

FTP

This lab will help guide you through the steps required to setup your system as a server for anonymous ftp and for the more restrictive guest-user ftp access. Perform the indicated steps, and write the answers to questions in the appropriate spaces on this hand-out. Turn-in one filled in handout per computer system by the beginning of class next week. Be sure to include your name (and your partner's name, if you have one).

Configure the System

Bring the system up single-user and perform the initial network configuration

- Step 1. Boot your system into single user mode. To accomplish this, reboot the system, and interrupt the Red Hat splash screen at the beginning of the boot process with Control-X. Then type **linux single** at the **boot:** prompt.
- Step 2. Configure your system to obtain its IP address using DHCP.
- Step 3. Bring up the **lo** and **eth0** interfaces. Check your connection by pinging a local host and a host on the internet. Your system's /etc/resolv.conf file will be automatically created, so DNS should work.

Install the FTP Server Packages

To run an ftp server, you need the *wu-ftp* package which includes the *in.ftpd* daemon.

- Step 4. The first step in setting up your system to provide FTP services is to obtain and install the **wu-ftp** software if it is not available on your system. Use **rpm** to determine if you have this package installed:

```
$ rpm -qa | grep ftpd
```

- Step 5. In addition, we will use another package that makes setup of anonymous ftp very easy. The package is named **anonftp**. The **wu-ftp** and **anonftp** packages can be found on a Red Hat installation CD, or at any of the Red Hat software mirrors. There are also copies available via ftp on the instructor's system (ask for the IP address). Use the **ftp** command to connect to the instructors system. Once connected, login in anonymously (user: **ftp** or **anonymous**). Get the rpm package files: **wu-ftp-2.6.1-20.i386.rpm** and **anonftp-4.0-9.i386.rpm**. Take note of the welcome message given to you by the ftp server – you will later setup your ftp server to include such a messages as well.
- Step 6. Once the packages are downloaded, exit ftp, and run the rpm command below to install (or update) the packages):

```
# rpm -Uhv wu-ftp* anonftp*
```

Enable ftpd via xinetd

The ftp server daemon is started via *xinetd*. Configure *xinetd* to start *in.ftpd* as needed.

- Step 7. The ftp daemon **in.ftpd** will be started via **xinetd**. The command below will enable this service for you:

```
# chkconfig wu-ftp on
```

- Q1. What file was changed by chkconfig? _____
- Q2. What line was changed within the file? _____
- Q3. What are the arguments passed to the ftpd server daemon when it starts? _____
- Q4. What do these arguments mean? _____
- Step 8. That's it! You should now be able to ftp to your own system using the anonymous account. Try it now.
- Q5. What transfer mode is currently selected? _____
- Q6. What is the indicated system type? _____

Examine the Anonymous User's chroot Environment

Step 9. Anonymous ftp users connect to the system in a restricted environment. This is accomplished by the ftp server via the **chroot** system call (**man 2 chroot**). It gives the system significant protection from the anonymous user's possibly malicious actions. The Red Hat chroot-ed environment is **/var/ftp**. Examine the contents of the directory and its subdirectories.

Q7. What directories exist? _____

Q8. What executable files are within the chroot area? _____

Customizing the FTP Server

Step 10. The default anonymous ftp setup provided by the Red Hat **anonftp** package is fairly simple. You have already seen the chroot area for anonymous ftp. We'll customize it now. A security leak is giving away information about our server. Connect anonymously to the ftp server (but don't log in). You should notice that the server immediately gives you back the ftp server name and version. There is no reason to give out this type of information. To disable it, we need to edit the wu-ftpd configuration file **/etc/ftppaccess**. Add the lines below to the configuration file:

```
greeting brief
signoff terse
```

Now connect again to test that the message is less revealing.

Q9. What is the new message given? _____

Step 11. Add our own welcome message. Again edit **/etc/ftppaccess** and look for the lines that start with the keyword **message**. The **message** keyword (see **man ftppaccess**) gives the name of a file that will be output to the screen upon login (or when the user changes directories). This message file must be within the chroot area, since after the chroot, the ftp server process can no longer access *anything* outside of this area. The line:

```
message /welcome.msg          login
```

indicates that the file **/welcome.msg** will be display upon ftp login. Note: This file is *not* in the root of the filesystem, but is in the root of the chroot area (**/var/ftp**). Create the file **/var/ftp/welcome.msg** and add a welcome message.

Step 12. Test that the message is displayed when you login as the anonymous user.

Q10. What was added to the beginning of each line of your message? _____

Q11. What do you suppose it means? _____

Step 13. The wu-ftpd server supports variables in the message files. Look at the man page for **ftppaccess**, and search for the **message** keyword. In the description are % variables, which will be substituted by the ftp server upon login. Customize your welcome message by adding the local time, the current user's login name, the remote host name, etc. Pick several, and see how they work. After you have added them, anonymously connect to ftp again to see the results.

Real Users

Logging into the ftp server as a real user allows access to the entire root filesystem.

Step 14. To see the added security of a chroot-ed ftp account, connect and login to your ftp server using your **student** account. After you are logged in, use ftp's **cd** and **ls** commands to examine **/**, **/dev**, **/etc**, in the file system. You should be able to see everything that user **student** can see as if **student** was actually logged into the system. This is a security hole and can be very dangerous. By default, all users (except ftp/anonymous) are real users in **wu-ftp** unless the system is configured otherwise. And such users have the same privileges as if the user was logged in.

Guest Users

Creating guest ftp accounts adds security by forcing access to a chroot area, thus eliminating access to the root file system.

Step 15. To create a guest user account, a chroot area must be created. Recall from earlier that a chroot area needs the following directories: bin, etc, and lib. The easiest way to create such a chroot-area is to copy an existing one. We'll create a new base directory for guest-user ftp access called `/home/ftpguests`. The commands below will copy the anonymous ftp user's chroot area to `/home/ftpguests`. This will act as a template for you:

```
# mkdir /home/ftpguests
# cd ftpguests
# ( cd /var/ftp ; tar -cvf - . ) | tar -xvf -
```

Step 16. In `/home/ftpguests` you should now have the directories bin, etc, lib, and pub, plus the login message file `welcome.msg` that you created earlier. File and directory permissions were retained by tar – these permissions are critical to security. Examine the file and directory permissions. In fact, the Red Hat **anonftp** package sets permissions on the executables in bin a bit too relaxed (they really should be 111 – execute only, so that users cannot read examine their contents). You can change them if you wish, but do not have to worry about this now.

Step 17. Remove the pub directory – it will not be used.

Step 18. Modify the welcome message, making guest-user welcomes different than those for anonymous users. This will help you verify your guest-user ftp login testing later.

Step 19. Now there must be a directory for our guest-user. We'll create an ftp guest-user account for our existing **student** user account. Create a private directory for **student** called `/home/ftpguests/student`. This is where **student** will upload/download files. Also change the owner and group of the new student directory and set permissions to 755.

Step 20. Edit the file `/home/ftpguests/etc/passwd` so that it contains only the lines below (lookup **student**'s UID and GID). Note: there is no reason to include more information than is necessary. The entries in this passwd file will only be used to determine the home directory for guest-users. Still, the file must be in the correct format.

```
root::0:0:::/:
student::students-uid:students-gid::/student/:
```

Step 21. Edit the file `/home/ftpguests/etc/group` so that it contains only the lines below (lookup **student**'s primary group name):

```
root::0:root
student-group-name::student-gid:student
```

Step 22. Add the lines below to the `/etc/ftpaccess` file. This sets all users to be guest-users, indicates that the anonymous ftp user is a real user (a workaround for pre wu-ftpd 2.6.2), and that the chroot area for guest-users is `/home/ftpguests`.

```
guestuser *
real-user ftp
guest-root /home/ftpguests
```

Step 23. Create a dummy test file in the directory `/home/ftpguests/student`, just to see that you are chroot-ed to the correct area when you login into ftp as **student**. Otherwise, when you login, and **ls** command will reveal an empty directory, indicating nothing about where you are (and since you are in a chroot-ed environment, there is no way to find out)!

Step 24. Test out the new **student** guest-user ftp login. Login to ftp as **student**. You should receive the new guest-user welcome message when you are logged in.

Step 25. Use **ls** to see if the dummy test file you created earlier is there. If so, you have successfully created your guest-user account.

Step 26. That's it for the basic setup. Again, file permissions and `/etc/ftpaccess` settings are very important to prevent security violations or DoS attacks. Make sure you understand the ramifications before setting up an ftp server on the internet.