System Administration for the Web:
Week 10 Lab

November 14, 2005

1 Staying Up to Date

In the first lab for your project, we instructed you to enter the following commands to install ssh on your servers:

```
apt-get update
apt-get install ssh
```

By now, the purpose of the second command should be clear to you. However, the purpose of the first command is still a mystery. In this section of the lab, you'll determine the purpose of `apt-get update` and learn how it'll help keep your server secure.

[1] Read the apt-get man page and see if you can figure out what `apt-get update` does.

Your server maintains a database of all the packages available to install. However, as new packages are created and older packages updated, your server's copy of the package database becomes outdated. `apt-get update` refreshes your server's copy of the package database and ensures that `apt-get` is aware of all new and updated packages.

[2] Perform an `apt-get update`. What happens?

[3] Once your server has obtained a new version of the package database, it is a good idea to check and see if there are any new versions of packages installed on your server. Read the `apt-get` man page and identify the command to upgrade your server's software.

[4] Upgrade the software on your server.

For those who use Microsoft Windows at home, this process should be somewhat familiar. Microsoft Windows Update works in a very similar manner. However, one should note that Microsoft Windows Update only updates the core of the Microsoft Windows operating system; `apt-get` updates all the software that has been installed on your server using `apt-get install`. In order words, you can upgrade all the software on your computer with one command (compare that to going to each individual program's website, downloading the latest version, and installing the upgrade manually)!

[5] Can you come up with reasons why you would want to insure that all the software on your server is up to date? Think about the majority of updates that are released through Microsoft Windows Update.

2 Access Control Lists

[1] At this point in the class, you should be familiar with UNIX filesystem permissions. What classes and types of permissions can you set?

Sometimes basic UNIX filesystem permissions are not flexible or specific enough. For example, how would you allow read and write access to a file for one specific person, while keeping you as the owner of the file? Such control is impossible to implement with basic UNIX filesystem permissions, but is easy with POSIX access control lists.

[2] Use `apt-cache` to search for the package that contains the access control list utilities and install the package.

Before you utilize access control lists, though, you must enable support for it. Open the `/etc/fstab` file with a text editor and change the first line in that file to match the following line:

```
/dev/sda1   /   reiserfs  defaults,acl     0    1
```

Reboot the server using the reboot command. You'll be disconnected from the server during the reboot. Please wait a minute and reconnect.

[3] The command to apply an access control list to a file is `setfacl`. Read the command's man page and figure out how to grant read and write access to a specific user. Hint: `setfacl`'s man page contains a nice list of examples at the end.

[4] In the first lab for your project, you were instructed to create a regular user account for yourself and your partner. Have each partner log into their respective account and create a text file. Using access control lists, grant your partner read and write access to the text file. Have your partner check to make sure they can read the write the file from their account.

[5] Now, create a file that can be read and modified by anyone. Using access control lists, remove read and write access for your partner only. Again, have your partner check to make sure the permissions are properly set.

As you can see, access control lists are extremely useful in multi-user environments. They allow you to grant specific access rights to individuals. An important principle of security is to grant the least amount of access necessary to any person. Access control lists allow system administrators to maintain a tight set of file permissions.

[6] See if you and your partner can come up with situations where access control lists would be useful.

3 Quota

Along with filesystem permissions, disk space quotas are probably one of the most important tools system administrators have for securing their systems and restricting abuse. Disk space quotas allow system administrators to specify exactly how much disk space a user can use. On any server with more than one user, it is imperative to set disk space quotas to prevent a single user from hogging up an entire server's hard drive.

[1] Use `apt-cache` to search for the two packages that contain the utilities to manage disk space quotas and install them.

As before, you must enable support for quotas before utilizing them. Open the `/etc/fstab` file with a text editor and change the first line in that file to match the following line:

```
/dev/sda1  /  reiserfs  defaults,acl,usrquota,grpquota  0  1
```

Reboot the server using the reboot command. You'll be disconnected from the server during the reboot. Please wait a minute and reconnect. Once reconnected, execute the following command to initialize your quota database:

```
quotacheck -mfug /
quotaon /
```

[2] Create a new user. Using `edquota`, give the user a quota of 1000 K.

[3] Login as the user you just created and try to download a file larger than 1000 K. What happens?

4 Insecure Directories

A common security hole associated with poor filesystem permissions management and lack of quotas is insecure directories. In general, an insecure directory is a directory that allows any user to write files to it. Ideally, users should only be able to write to their home directories; if users were allowed to write to all sorts of directories, it would be very easy for users to hide files from system administrators.

[1] Read the man page for the `find` command. Use the command to locate all world-writeable directories and files.