System Administration for Beginners

Week 5 Laboratory

March 16, 2009

1 Getting Started

You are getting full access to a virtual server, hosted courtesy of the Open Computing Facility (OCF). Because of the limited IP space and setup of the virtual servers, logging on to your server takes a bit more work than logging on to the `inst` or `OCF` servers. Read the following instructions carefully and thoroughly and ask the facilitators any questions if you need help.

1.1 Submission Instructions

At the top of each submission, please provide the assignment name, your full name, `inst` login (`cs198-XX`), and your email address. Answer the following questions below. If no specific information is being asked, include any output or answers that you think would help show us that you understand the material (text only). Turn in your *paper* submission at the start of class next week.

2 Login

You will be using SSH to connect to your private server via the login server. The server hostname is `decal.ocf.berkeley.edu`. Please login using the following command:

```
ssh username@decal.ocf.berkeley.edu
```

where `username` is the account name you created. Enter in the password you typed during account creation. This is the gateway server to get access to your virtual server. From here, log in to your virtual server using the following command:

```
ssh root@10.14.0.X
```

where you replace `X` with your server number. From here, you will have full control of your virtual server.

3 Create User Accounts

As mentioned during lecture, working in root is dangerous; a simple mistake can destroy a system completely. Consequently, system administrators usually create normal user accounts for themselves. Read the man page for the `adduser` command and figure out how to create a user account for each one of your group's members. Test these accounts by attempting to login to them using SSH.

If you ever need to perform something as root while working as a regular user, use the `su` command. `su` is short for switch user and is a UNIX command for temporarily switching to another user account on the system. `su` takes the name of the user you wish to switch to as its parameter, and if none is supplied, it defaults to root. For various technical reasons, you should also specify the `-` (dash) parameter to `su`. For example, to switch to another user account, you would use the following command:

```
su - account_name
```

And to switch to root, using the fact that `su` defaults to root:

```
su -
```

After entering in the command, you will be prompted for the password of the user to which you are switching. If you login successfully, you will have a new shell as that user; it will be just as if you logged in as that user in the first place. To end your `su` session, type `exit`.

One useful thing to note is that, as root, you will be able to `su` to any user without entering in their password. This is sometimes useful for repairing user accounts that have been damaged in such a way that the user cannot login normally.

4 Working with Apache

4.1 Download and Install Apache

Visit the Apache website (`http://httpd.apache.org`) and download the source code for the latest version of Apache 2.0 to your virtual server (the `wget` command may be useful). Extract the file and change into the directory that is created. Read through the `README` and `INSTALL` files to determine how you can compile and install Apache.

When using `./configure`, please specify a path within your home directory as the `PREFIX`. We're only going to do a test install of Apache this week, so you do not want to actually install Apache such that all users can access it. Thus, using the `--prefix` parameter to `./configure`, change the default installation directory of Apache to some place inside your home directory. Recall that you can use `pwd` to determine the full path to any directory.

TIP: You may use `$HOME` as a shortcut to your home directory; if you specify a prefix of `$HOME/test`, it will refer to the directory `test` under your home directory.

TIP: You will find that you are initially unable to configure apache because you do not have the tools to compile anything. Tools you will need are `make`, `gcc`, and other common compilation tools. These are all packaged in the Debian `build-essential` package. You can install this using apt-get: (`apt-get install build-essential`).

4.2 Configuring Apache

At this point, because everyone is on a virtual server on one specific machine, you would need to edit the configuration file of Apache so that it would listen on a different port. Usually the default configurations are good enough for basic service. One thing you will need to change is the port the server is looking on. Point it to 3XX80 where (`XX`) is your server number. Where in the configuration file would be the appropriate place to edit which port Apache listens on?

Open up the configuration file and try to see what kind of settings you can edit. Note that lines prefixed with the `#` symbol are comments and ignored by Apache.

4.3 Starting and Testing Apache

Execute the following command to start Apache:

```
$PREFIX/bin/apache2ctl start
```

where `$PREFIX` is the directory you used when you ran `./configure`. If you get a message stating that the server was started, you are ready to test Apache. If not, double check your configuration file. An easy way to test your configuration file is to run the following command:

```
$PREFIX/bin/apache2ctl configtest
```

To check if Apache is properly accepting requests, open Firefox or a browser and visit the following URL:

```
http://decal.ocf.berkeley.edu:3XX80
```

where `XX` is your server number. The latter part of the URL is necessary to tell the browser not to connect to the default HTTP port of 80. If everything works, you should get a page stating that Apache has been successfully configured.

5 Create a Sample Page

Apache can be extended beyond it's default capabilities with Apache modules. These are similar to plugins that are available for software. One useful module is the `UserDir` module. This module allows all users on a UNIX system to be granted their own webspace through their own home directories. By default, this module is enabled. To use the module, you must add directives to Apache's configuration file. Read through the configuration and online documentation to determine what you need to change.

Create a directory in your home folder which has read and execute permissions for everyone. Any files in this directory with permissions of at least read for other will be accessible to the Internet. Create an `index.html` file in this directory that lists the names of your group members and their `inst` account.

To access this directory via the web, type the following URL into your browser:

```
http://decal.ocf.berkeley.edu:3XX80/~user/index.html
```

where again, `XX` is your server number and `user` is the user you created using `adduser`.