Post Ubuntu Install Exercises

PacNOG 3 – June 18

Rarotonga, Cook Islands

1. Get used to using `sudo`
2. Create an "*inst*" account
3. Learn how to install software
4. Install `gcc` and `make`
5. Learn how to control services
6. Use the `ip` tool
7. See the state of your machine
8. Create the `locate` database
9. So, you wanna be root...
10. Install Gnome 2.18 and proper video driver
11. Configure your X server

Get used to using sudo

Ubuntu and Debian approach system administration a bit differently than other Linux distributions. Instead of logging in as the "*root*" user to do system tasks, or becoming *root* by using the `su` command you are encouraged to do your system administration using `sudo`. By default your user has privileges to do this. Let's practice this by running some privileged commands from your user account.

First, log in if you have not done so. Once you are logged in you'll see something like this:

```
user@pcn:~$
```

We'll represent this prompt with the abbreviation "$".

Now try to look at the system password file:

```
$ less /etc/passwd
```

The first time you attempt this it will fail. Instead do the following:

```
$ sudo less /etc/passwd
```

You will be prompted for a password. This is your user's password. Type it in and you should see the contents of the protected file /etc/passwd.

If you wish to issue a command that requires system privileges, use the `sudo` command. For instance, if you are interested in seeing what groups your account belongs to you can type:

```
$ sudo vigr
```

You are now in the vi editor (you have a handout to help you with this editor). Type:

```
/yourUserid
```

Then press the "n" key for "next" to see each group you belong to. Notice that you arein the "adm" group. To exit vi type:

```
:q!
```

Get used to using "sudo" to do your system administration work. The final exercise, number 9, will give you a couple of other options for using system privileged commands as well.

Create an *inst* account

If you are used to many Linux distributions, then you think of the `adduser` and the `useradd` commands as being equivalent. One is simply a link to the other. In Debian/Ubuntu this is not true. They are distinct commands with different capabilities. If you are interested in the differences type:

```
$ man adduser
$ man useradd
```

As you can see the `adduser` command has more options. This is what we will use to add a new user and to manipulate user accounts later on. Interestingly, it lacks one key ability, to create new user account *and* add it to multiple groups at the same time. We'll fix this issue at the end of this exercise.

At this point we would like you to create an account named *inst* with a password given in class. This allows your instructors, your fellow students or yourself a way in to your system if necessary. To do this type:

```
$ adduser --shell /bin/bash inst
```

You will be prompted for a password. Use 'XXXXXX' (password given in class). Please be sure to use this password. Your session will look like this:

```
user@pcn:~# adduser --shell /bin/bash inst
Adding user `inst' ...
Adding new group `inst' (1001) ...
Adding new user `inst' (1001) with group `inst' ...
Creating home directory `/home/inst' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:             <ENTER 'p4cn0g07'>
Retype new UNIX password:            <ENTER 'p4cn0g07>
passwd: password updated successfully
Changing the user information for inst
Enter the new value, or press ENTER for the default
```

```
        Full Name []:                      <Press ENTER for default>
        Room Number []:               <Press ENTER for default>
        Work Phone []:                     <Press ENTER for default>
        Home Phone []:                     <Press ENTER for default>
        Other []:                          <Press ENTER for default>
Is the information correct? [y/N] y        <Press ENTER for default>
user@pcn:~#
```

You are almost done. We want the user *inst* to belong to the *adm* group as well so that you can run privileged commands using *sudo* with this userid. If you use the *useradd* command it's possible to do this at account creation time (see *man useradd* for details). To do this now type the following command:

```
user@pcn:~# usermod -G adm inst
```

At this point you are done and the user *inst* now exists on your machine as we need it for the week.

Learn how to install software

This is a large topic. Your instructor should have discussed this with you previously. In general you can use `apt-get` to install software, clean up software installs, remove software and update your repositories. You can use `aptitude` as a meta-installer to control `apt`. The `dpkg` command extracts and installs individual Debian packages and is called by `apt`. Finally, `synaptic` is a graphical interface to `apt` that can be used in Gnome or KDE.

We are going to concentrate on the `apt-get` method of software installation. But you should most definitely spend some time reading about and learning about how `apt` (in general), `aptitude`, `dpkg` and `synaptic` work. To do this you might try doing:

```
$ man dpkg
$ man apt
$ man apt-get
$ man aptitude
```

Install gcc and make

Two items missing from a default Debian/Ubuntu installation are `gcc` and `make`. This can be quite disconcerting if you are used to compiling software under other versions of Linux. Luckily there is an easy way to install all the bits and pieces you need to use gcc and/or make. Simply do:

```
$ sudo apt-get install build-essential
```

In this case you are going to be asked to place the "Ubuntu-Server 7.04" CD in the cdrom drive. You don't want to do this. Press CTRL-C to get out of this dialogue.

This brings up the topic of software repositories. When using `apt`, `apt-get`, `aptitude` and/or `synaptic` there is a master file that tells Ubuntu where to look for software you wish to install. This file is /etc/apt/sources.list. You can update this file to point to different repositories (third party, local repositories, remove the cdrom reference, etc...). In our case we are now going to do this. We'll edit this file and we are going to edit out any reference to the Ubuntu 7.04 cdrom. In addition we are going to point our installation to use our local Ubuntu archive for software installs. This will save us a huge amount of time vs. attempting to download new software over our satellite link.

First to edit the file /etc/apt/sources.list do:

```
    $ sudo vi /etc/apt/sources.list
```

In this file we want to comment out any references to the Ubuntu cdrom. You'll see the following lines at the top of the file:

```
#
# deb cdrom:[Ubuntu-Server 7.04 _Feisty Fawn_ - Release i386 (20070415)]/ feisty main restricted
deb cdrom:[Ubuntu-Server 7.04 _Feisty Fawn_ - Release i386 (20070415)]/ feisty main restricted
```

Update this by simply commenting out the one line (see your vi reference sheet for help):

```
#
# deb cdrom:[Ubuntu-Server 7.04 _Feisty Fawn_ - Release i386 (20070415)]/ feisty main restricted
#deb cdrom:[Ubuntu-Server 7.04 _Feisty Fawn_ - Release i386 (20070415)]/ feisty main restricted
```

Once you've done this we want to remove references to the “ck.archive.ubuntu.com” archive. This is the default archive used for the Cook Islands – unfortunately this is in London. We have a local archive at “archive.conference.pacnog.org” that we should use instead. To do this enter the following in vi:

```
    :1,$s/ck.archive.ubuntu.com/archive.conference.pacnog.org/g
```

and press <ENTER>. Note the “:” to place you in command mode in vi.

This should do a global search and replace of “ck.archive.ubuntu.com” with “archive.conference.pacnog.org”.

Now that you have donethis you should save and exit from the file by doing:

```
    :wq
```

Now to tell `apt` that you have a new set of repositories to be used you do:

```
    $ sudo apt-get update
```

Now to actually install the `build-essential` meta-package type:

```
    $ sudo apt-get install build-essential
```

and respond with a “Y” when asked if you “...want to continue”. Once the installation process finishes you should have both `gcc` and `make` installed on your machine.

Learn how to control services

The first thing to remember is that if you install a new service, say a web server (Apache), then Ubuntu will automatically configure that service to run when you reboot your machine and it will start the service immediately! This is quite different from the world of Red Hat, Fedora, CentOS, etc. In order to configure and control services the core tool available to you is `update-rc.d`. This tool, however, may not be the easiest to use. Still, you should read and understand a bit about how this works by doing:

```
$ man update-rc.d
```

There are a couple of additional tools available to you that you can install. These are `sysvconfig` and `rcconf`. Both of these are console-based gui tools. To install them do:

```
$ sudo apt-get install sysvconfig rcconf
```

Did you notice that we specified two packages at the same time? This is a nice feature of `apt-get`. Try both these commands out. You'll notice that the `sysvconfig` command is considerably more powerful.

```
$ sudo sysvconfig
$ sudo rcconf
```

Finally, there is a nice Bash script that has been written which emulates the Red Hat `chkconfig` script. This is called `rc-config`. We have placed this script on our "noc" box. Let's download the script and install it for use on your machine:

```
$ cd
$ wget http://noc.conference.pacnog.org/workshop/scripts/rc-config
$ chmod 755 rc-config
$ sudo mv rc-config /usr/local/bin
```

At this point the script is installed. You should be able to just run the script by typing:

```
$ rc-config
```

Try viewing all scripts and their status for all run-levels:

```
$ rc-config -l
```

Now trying viewing the status of just one script:

```
$ rc-config -ls anacron
```

You can see how this script works, if you understand enough of bash scripts, by taking a look at it's code:

```
$ less /usr/local/bin/rc-config
```

Use the `ip` tool

The `ip` command is a powerful network debugging tool available to you in Ubuntu. As with any new command have a look at the help file by first doing:

```
$ man ip
```

As you can see this tool is designed to, "show/manipulate routing, devices, policy routing and tunnels."

For instance, if you are wondering what your default route is (or are) you can simply type:

```
$ ip route
```

This is actually short for "`ip route show`". Maybe you are wondering out which interface packets will go to a certain address? A quick way to find out is:

```
$ ip route get 128.223.32.35
```

Clearly you can substitute any IP address you wish above. This is useful for boxes that have multiple network interfaces defined.

Maybe you want to be able to sniff packets coming across an interface on your box. To do this you may wish to place your interface in promiscuous mode. Often this requires updating a kernel parameter. With the ip command you can do:

```
$ sudo ip link set eth0 promisc on
```

Note the use of "sudo" here as setting an interface requires admin privileges. Now you can snoop the packets on the eth0 interface by doing:

```
$ sudo tcpdump -i eth0
```

Be sure to read the man page for tcpdump if you want further information.

See the state of your machine

A critical piece of host-based security is to know what is running on your host at all times. To find out what network services are running and what connections are being made to your box you can use several commands, including LiSt of Open Files (`lsof`) and `netstat`. To see active network connections using `lsof` do:

```
$ sudo lsof -i
```

Read up on this command to better understand the output. Every service that is running and everything that is connected to that service should be expected by you. In addition, you should be aware of what is running and you should stay on top of security updates and warning for each of these.

Additionally you can view detailed information about processes and network status using the netstat command. For instance try doing:

```
$ sudo netstat -antlp
```

Read "`man netstat`" and try to figure out what all these options means.

To see every process currently running on your machine type:

```
$ ps -auxww | more
```

As usual, read "man ps" to understand what the switches mean. For the above, in short, "aux" is to see all processes in user-oriented format. The "ww" means include the entire process description, even if it wraps on multiple lines on the screen. Note that other versions of Linux require that you use "www"to get the full description.

More or less you should understand pretty much everything you see in this output.

A couple of more useful commands include:

```
$ w
```

And the `top` command. To break out of `top` press the "q" key. The top command can show you many variations of information dynamically by pressing various keys. Try pressing "l" and "m" after you type:

```
$ top
```

To find out how much physical disk space is in use (note that `top` includes how much RAM and SWAP is in use) use:

```
$ df -h
```

The "-h" is for "human readable" format. It is not as exact. To see more exact numbers remove the "-h" option.

There are many more commands for understanding what is going on with your system, but these are some of the most commonly used ones.

Create the `locate` database

One of the easiest ways to find files on your system is to use the `locate` command. For details, as usual, read the man pages:

```
$ man locate
```

Locate uses a hashed database off filenames and directory paths. the command searches the database instead of the file system to find files. While this is *much* is much more efficient it has two downsides:

1. If you create the `locate` database as root then users can see files using `locate` that they otherwise would not be able to see. This is considered a potential security hole.
2. The `locate` command is only as precise as the `locate` database. If the database has not been recently updated, then newer files will be missed. Many systems use an automated (cron) job to update the `locate` database on a daily basis.

To create an initial `locate` database, or update the current one do:

```
$ sudo updatedb
```

Once this process completes (it maytake a few minutes) try using the command:

```
$ locate ssh
```

Quite a few files go past on the screen.To find any file with “ssh” in it's name or it's path and which has the string “conf” you can do:

```
$ locate ssh | grep conf
```

Read about “`grep`” using “`man grep`” for more information. The `locate` command is very powerful and useful. For a more exacting command you can consider using “`find`”. This is harder to use and works by brute-force. As usual do “`man find`” for more information.

So, you wanna be root...

As you have noticed Ubuntu prefers that you do your system administration from a general user account making use of the `sudo` command.

If you must have a root shell to do something you can do this by typing:

```
$ sudo bash
```

This is useful if you have to look for files in directories that would otherwise be unavailable for you to see. Remember, be careful. As root you can move, rename or delete any file or files you want.

What if you really, really want to log in as root? OK, you can do this as well. First you would do:

```
$ sudo passwd root
```

Then you would enter in a root password – definitely picking something secure and safe, right?! For now there is now reason to do this, so please don't. :-) Once you've set a root password, then you can log in as root using that password if you so desire.

Install Gnome 2.18 and proper video driver

NOTE! Please do not do these last two exercises until just before the lunch break.

It is actually quite simple to install a graphical desktop on Ubuntu. By default Ubuntu uses the Gnome desktop. If you wish to use KDE with Ubuntu there is a separate version of the Ubuntu distribution called Kubuntu that you can find at www.ubuntu.com.

We have configured your workshop lab so that the files for Gnome are on a local machine. The installation requires over 400MB of files to download and over 1GB of total space. Downloading will not take long, but unpacking and installing will take some time.

In addition, with the same command we are going to tell Ubuntu to download an updated video driver for the particular machines in our classroom. Ubuntu version 7.04 has an issue with the Intel i810 chipset and the specific Intel i810 driver. There is a newer driver that works just fine called "intel". By default Ubuntu will first install the i810 driver when you install Gnome, so we'll specify to Ubuntu to install Gnome *and* the correct video driver using the following command:

```
$ sudo apt-get install ubuntu-desktop xserver-xorg-video-intel
```

This will now take quite some time. Feel free to go to lunch if it is time do to that. If you are around when this install prompts you to pick a default resolution for your Gnome desktop, then you should choose: **1280x1024**.

Except for the few workstations that have smaller Dell LCD panels. You should pick **1024x768** as your default resolution (which will be plenty to work with during the week).

Configure your X server

Ubuntu uses the Xorg XWindow system for the underlying graphics engine that drives the Gnome Desktop. Once the Gnome desktop is installed along with Xorg and the correct graphics driver you need to configure Xorg to work with your hardware, the installed driver and the resolution you have chosen. Luckily Xorg has made this quite easy to do. First do:

```
$ cd
$ sudo Xorg -configure
```

This should create the file xorg.conf.new. You can test this file is you wish, but we are pretty confident it should work. To finalize configuring your X Server do:

```
$ sudo cp xorg.conf.new /etc/X11/xorg.conf
```

Now type:

```
$ gdm
```

and your Gnome desktop environment should start. You can log in with your username and password.