Some Ubuntu Practice...

SANOG 10 – August 29

New Delhi, India

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...

1.) 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 with actual encrypted passwords:

```
$ less /etc/shadow
```

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

```
$ sudo less /etc/shadow
```

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/shadow (press "q" to exit the output on the screen).

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 are in 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.

2.) 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 is considerably more powerful. This is what we will use to add a new user and to manipulate user accounts later on.

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:

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

You may be be prompted for your user password to use the `sudo` command.

When prompted for the new user password use '10s4n0g' (zeroes, not the letter 'o'). 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 '104s4n0g'>
Retype new UNIX password:             <ENTER '104s4n0g'>
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:~#
```

At this point you are done and the user *inst* now exists on your machine.

In order to allow the new *inst* user to use the `sudo` command it must be a member of the *adm* group. To do this you can type:

```
$ sudo usermod -G adm inst
```

And, to verify that *inst* is now a member of the *adm* group:

```
$ groups inst
```

3.) 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`. In addition, `synaptic` is a graphical interface to `apt` that can be used in Gnome or KDE. Finally, `apt-cache` allows you to view information about already installed packages.

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`, `apt-cache`, and `synaptic` work. To do this you might try doing:

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

4.) Update /etc/apt/sources.list

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, which is left from the initial install. In addition we are going to point our installation to use our local Ubuntu archive for software installs. This will save us time vs. attempting to download all new software across our external 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
```

4.1) Change your sources list (NOTE: we might not do this in class)

Once you've done this we want to remove references to the "in.archive.ubuntu.com" archive. This is the default archive used for India– unfortunately this is in London (via Los Angeles). We have a local archive at ".169.223.5.254" that we should use instead. To do this enter the following in vi:

```
:1,%s/in.archive.ubuntu.com/169.223.5.254/g
```

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

This should do a global search and replace of "in.archive.ubuntu.com" with "169.223.5.254".

Now that you have done this 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
```

5.) Install libc, gcc, g++ and make

Two items missing from a default Debian/Ubuntu installation are `gcc` and `make` plus their associated bits and pieces. 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
```

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.

This is an example of installing software using a "meta-package." If you type in the command:

```
$ sudo apt-cache showpkg build-essential
```

You will see a descriptive list of all the various pieces of software that are installed for this package.

6.) 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://169.223.5.254/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
```

7.) Use `ping`, `traceroute` and `mtr`

During the week you will use these commands

`ping : traceroute : mtr`

ping

This command is used to measure latency between you and other hosts. Or, more formally, it sends an ICMP ECHO_REQUEST to a network host. You can see packet travel times as well as packet loss statistics using ping. Several tools during the week will take advantage of ping to measure network performance, such as the *Smokeping* package.

Give the command a try:

```
$ ping yahoo.com
$ ping -c 20 yahoo.com
```

You can send packets of specified size using the "-s" option. This can be useful to help troubleshoot suspected mtu issues on a network. It is, also, the source of some known network attacks for older TCP/IP stacks that did not correctly deal with odd-sized packets.

traceroute

Print the route packets take to a network host. By default send 3 queries per host, show the round trip time for each. Very useful to figure out the route that data takes to get from one host to another.

First we need to install the standard, IPv4 `traceroute` package. We can do this like so:

```
$ sudo apt-get install traceroute
```

Respond "Y" if asked whether you wish to install. Next, let's try doing a traceroute on the Ubuntu archive servers for India:

```
$ traceroute in.archive.ubuntu.com
```

What did you see? Can you draw any conclusions from the output? Ask your instructor or assistant if you don't understand what's being show, but first, as always, try:

```
$ man traceroute
```

mtr

My TRaceroute combines the functionality of the traceroute and ping programs in a single network diagnostic tool. Taken directly from the man page (`$ man mtr`):

> *As mtr starts, it investigates the network connection between the host mtr runs on and HOSTNAME. by sending packets with purposly low TTLs. It continues to send packets with low TTL, noting the response time of the intervening routers. This allows mtr to print the response percentage and response times of the internet route to HOSTNAME. A sudden increase in packetloss or response time is often an indication of a bad (or simply overloaded) link.*

Give `mtr` a try:

```
$ mtr -t sageduck.org
```

That is your instructor's home machine in Santiago, Chile. Read the mtr man page (`$ man mtr`) if you don't understand the output. We'll be discussing all three of these tools in more detail during the week.

8.) 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.

9.) 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.

10.) 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?! Once you've set a root password, then you can log in as root using that password if you so desire.That's a controversial thing to do in the world of Ubuntu and Debian Linux.