UNIT 6.
System Administration

Structure

6.0 Introduction

This chapter covers the most important things that you need to know about system administration under Linux in sufficient detail to start using the system comfortably. In order to keep the chapter manageable, it covers just the basics and omits many important details. It will help you understand bet-ter how things work and hang together. At least, skim through the SAG so that you know what it contains and what kind of help you can expect from it.

6.1 Objectives

At the end of this unit, You would be able to

- Undestand the concept of system administration
- Elaborate the views on user administration
- Know about password and protection
- Be Confident of managing the system for users
- Describe the analogy of storage space manipulation and importance
- Exhibit Login, Logoff, Shutdown and Restart.
- Describe the management of processes.

6.2 Root Account

The “root” account is the most privileged account on a Linux system. This account gives you the ability to carry out all facets of system administration, including adding accounts, changing user passwords, examining log files, installing software, etc.

When using this account it is crucial to be as careful as possible. The “root” account has no security restrictions imposed upon it. This means it is easy to perform

administrative duties without hassle. However, the system assumes you know what you are doing, and will do exactly what you request -- no questions asked. Therefore it is easy, with a mistyped command, to wipe out crucial system files.

When you are signed in as, or acting as “root”, the shell prompt displays '#' as the last character (if you are using bash). This is to serve as a warning to you of the absolute power of this account.

The rule of thumb is, never sign in as “root” unless absolutely necessary. While “root”, type commands carefully and double-check them before pressing return. Sign off from the “root” account as soon as you have accomplished the task you signed on for. Finally, (as with any account but especially important with this one), keep the password secure!

6.3 Creating User Accounts

This section assumes you are using the Shadow password suite on your Linux system. If you are not, you should consider doing so, as it helps to tighten up security somewhat. The Shadow suite is fairly easy to install and will automatically convert your non-shadow password file format over to the new shadow format.

There are two steps to creating a new user account.

i) actually create the account itself,
ii) provide an alias to their e-mail address (at my place of employment, we follow the convention of Firstname.Lastname@our.domain.name”.)

To create the account, decide on the username you are going to assign to the user. The username is at most 8 characters long, and wherever possible you should choose their last name, or last name and first initial if a user account already exists (the adduser script will detect and prevent you from adding duplicate account names).

You will then be prompted to enter other information: full name of user, user group (usually the default value), a user id # (automatically assigned), home directory (automatically assigned), a user shell, some password expiration values, and finally the desired password (which won't echo to the screen; you should have the user choose a password between 6 to 8 characters in length for security reasons).

Please note that *everything* should be entered in lowercase, except for the full name of the user which can be entered in a “pleasing format” (eg. Joe Smith) and the password. Case is sensitive, so inform your user(s) they must use identical case when entering their username and password.

Here is a sample session where we will add a user named Joe Smith:

```
mail:~# /sbin/adduser
User to add (^C to quit): smith
That name is in use, choose another.
User to add (^C to quit): smithj
Editing information for new user [smithj]
Full Name: Joe Smith
GID [100]:
```

```
Checking for an available UID after 500
First unused uid is 859
UID [859]:
Home Directory [/home/smithj]:
Shell [/bin/bash]:
Min. Password Change Days [0]:
Max. Password Change Days [30]: 90
Password Warning Days [15]:
Days after Password Expiry for Account Locking [10]: 0
Password [smithj]:</ FL1539
Retype Password:</ Fl1539
Sorry, they do not match.
Password:</> FL1539
Retype Password:</ FL1539

Information for new user [smithj]:
Name: Joe Smith
Home directory: /home/smithj
Shell: /bin/bash
Password: <hidden>
Uid: 859        Gid: 100
Min pass: 0     maX pass: 99999
Warn pass: 7    Lock account: 0
public home Directory: no
Type 'y' if this is correct, 'q' to cancel and quit the program,
or the letter of the item you wish to change: Y
```

The next step is to create the alias for the person's e-mail account. This gives people the choice of using their account name for their e-mail address, or their full name (First.Last combination) to make it “easier” for the outside world to guess their e-mail address when trying to contact them for the first time.

To add the e-mail alias, edit the “`/etc/aliases`" file as follows:

```
mail# pico -w /etc/aliases
```

Add the new alias at the bottom of the file. The format for an alias is:

```
First.Lastname:username
```

You should ask the user what preference they have for this (eg. Joseph.Smith or Joe.Smith). For our new Joe Smith user, the entry would be as follows:

```
Joe.Smith:smith
```

When finished adding the alias, press <**Ctrl**>-<**X**> and save the file. Then, type “`newaliases`" to update the aliases database.

At this point the user account has been created and is ready for use. It is a good idea to remind the user that his username and password must be entered in lowercase characters.

6.4 Changing User Passwords

To change a password on behalf of a user, first sign on or “su” to the “root” account. Then type, “`passwd user`" (where user is the username for the password you are changing). The system will prompt you to enter a password. Passwords do not echo to the screen when you enter them.

You can also change your own password, by typing “`passwd`" (without specifying a username). You will be prompted to enter your old password for verification, and then a new password.

6.5 Disabling User Accounts

To disable a user account, edit, as root, the “`/etc/shadow`" file (assuming you're using shadow passwords; if not, edit the “`/etc/passwd`" file instead), and replace the password (which is stored in its encrypted form) with a “*" asterisk character. All Unix passwords, regardless of length (up to a maximum of 8 characters), are stored in the password file as encrypted strings of 13 characters. Therefore, by replacing the password with a single “*" character, it is impossible for the user to sign in.

Note: This method will require you to assign a new password to the user if you re-enable the account, since the encrypted password field will have been replaced. One solution to this which seems to be popular among system administrators is to simply prefix the ``*" asterisk character in front of the encrypted password to disable the account, and simply removing the asterisk to enable it.

6.6 Removing User Accounts

On occasion, you may wish to remove a user's access from your server altogether.

If you are a Red Hat user, the easiest way to remove an unneeded user account is with the “`userdel`" command, which must be typed as “root". An example follows:

```
/usr/sbin/userdel baduser
```

The above command will remove the entry matching the username “*baduser* from the “`/etc/passwd`", file, and, if you're using the Shadow password format, the “`/etc/shadow`".

Note: The “`/etc/group`" is *not* modified, to avoid removing a group that other user(s) may also belong to. This isn't much of a big deal, but if this bothers use, you can edit the group file and remove the entry manually.

Should you wish to remove the user's home directory as well, add the “`-r`" option to the “userdel" command. For example:

```
/usr/sbin/userdel -r baduser
```

I recommend not removing an account right away, but first simply *disable* it, especially if you are working with a corporate server with lots of users. After all, the former user may one day require the use of his or her account again, or may request a file or two which was stored in their home directory. Or perhaps a *new* user (such as an employee replacement) may require access to the former user's files. In any event, make sure you have backups of the former user's home directory, “just-in-case”.

6.7 Linux Password & Shadow File Formats

Traditional Unix systems keep user account information, including one-way encrypted passwords, in a text file called “`/etc/passwd`". As this file is used by many tools (such as “ls") to display file ownerships, etc. by matching user id #'s with the user's names, the file needs to be world-readable. Consequentally, this can be somewhat of a security risk.

Another method of storing account information, one that we always use, is with the shadow password format. As with the traditional method, this method stores account information in the /etc/passwd file in a compatible format. However, the password is stored as a single “x” character (ie. not actually stored in this file). A second file, called “`/etc/shadow`", contains encrypted password as well as other information such as account or password expiration values, etc. The /etc/shadow file is readable only by the root account and is therefore less of a security risk.

While some other Linux distributions forces you to install the Shadow Password Suite in order to use the shadow format, Red Hat makes it simple. To switch between the two formats, type (as root):

```
  /usr/sbin/pwconv     To convert to the shadow format
  /usr/sbin/pwunconv   To convert back to the traditional format
```

With shadow passwords, the “`/etc/passwd`" file contains account information, and looks like this:

```
smithj:x:561:561:Joe Smith:/home/smithj:/bin/bash
```

Each field in a passwd entry is separated with “:” colon characters, and are as follows:

- Username, up to 8 characters. Case-sensitive, usually all lowercase
- An “x” in the password field. Passwords are stored in the “`/etc/shadow`" file.
- Numeric user id. This is assigned by the “`adduser`" script. Unix uses this field, plus the following group field, to identify which files belong to the user.
- Numeric group id. Red Hat uses group id's in a fairly unique manner for enhanced file security. Usually the group id will match the user id.
- Full name of user. I'm not sure what the maximum length for this field is, but try to keep it reasonable (under 30 characters).
- User's home directory. Usually /home/username (eg. /home/smithj). All user's personal files, web pages, mail forwarding, etc. will be stored here.
- User's “shell account”. Often set to “`/bin/bash`" to provide access to the bash shell (my personal favorite shell).

Perhaps you do not wish to provide shell accounts for your users. You could create a script file called "`/bin/sorrysh`", for example, that would display some kind of error message and log the user off, and then set this script as their default shell. The "`/etc/shadow`" file contains password and account expiration information for users, and looks like this:

```
smithj:Ep6mckrOLChF.:10063:0:99999:7:::
```

As with the passwd file, each field in the shadow file is also separated with ":" colon characters, and are as follows:

- Username, up to 8 characters. Case-sensitive, usually all lowercase. A direct match to the username in the /etc/passwd file.
- Password, 13 character encrypted. A blank entry (eg. ::) indicates a password is not required to log in (usually a bad idea), and a "*" entry (eg. :*:) indicates the account has been disabled.
- The number of days (since January 1, 1970) since the password was last changed.
- The number of days before password may be changed (0 indicates it may be changed at any time)
- The number of days after which password *must* be changed (99999 indicates user can keep his or her password unchanged for many, many years)
- The number of days to warn user of an expiring password (7 for a full week)
- The number of days after password expires that account is disabled
- The number of days since January 1, 1970 that an account has been disabled
- A reserved field for possible future use

6.8 System Shutdown & Restart

To shut down the system from a terminal session, sign in or "su" to the "root" account. Then type "`/sbin/shutdown -r now`". It may take several moments for all processes to be terminated, and then Linux will shut down. The computer will reboot itself. If you are in front of the console, a faster alternative to this is to press **<Ctrl>-<Alt>-<Del>** to shut down. Please be patient as it may take a couple of minutes for Linux to terminate.

You can also shut down the system to a halt (ie. it will shut down and not reboot the system). The system will be unavailable until power-cycled or rebooted with **<Ctrl>-<Alt>-<Del>**. This can be useful if you need to power down the system and move it to a different location, for example. To do this, type ``/sbin/shutdown -h now`" when signed into or "su"ed to the "root" account. Linux will shut itself down then display a message, "System halted". At this point you can power down the computer.

It is probably a good idea to only shut down the system when you are at the console. Although you can shut it down remotely via a shell session, if anything goes wrong and the system does not restart properly, the system will be unavailable until action is taken at the system unit. (I haven't experienced any problems doing this myself, however).

Upon system bootup, Linux will start automatically, and load all necessary services including networking support, and Internet services.

Tip: If you wish to provide some kind of warning to any online users (online meaning logged in to shell accounts), you can substitute a time value instead of the "now" keyword. You can also customize the shutdown warning message. For example, "`/sbin/shutdown -r +5 Hardware upgrade`" would inform users that the system was about to shutdown for the given reason. They are then given periodic warnings that they should close files and log off before the big moment arrives.

6.9 Checking Storage Space

It is important to check from time to time that adequate free space remains on the storage devices. Use the "`df`" command to get a report of available space. It will look as follows (information shown is from the Internet server at my place of employment):

```
Filesystem         1024-blocks  Used Available Capacity Mounted on
/dev/sda1            1888052  135908  1654551      8%   /
/dev/sdd1            4299828  100084  3977246      2%   /archive
/dev/hda2            3048303  897858  1992794     31%   /archive2
/dev/hda1              11677    1380     9694     12%   /boot
/dev/sdc1            4299828  350310  3727020      9%   /home
/dev/sdb1            4299828  598504  3478826     15%   /usr
/dev/sda2            1888083  700414  1090075     39%   /var
/dev/scd0             593958  593958        0    100%   /cdrom
```

These file-systems are pretty stable in that they have a fairly slow growth pattern.

The "*/*" (aka root) file-system, mounted on /dev/hda1, contains the Linux kernel, device drivers, and other directories. It also is where user mail messages are stored (*/var/spool/mail/*) as well as log files (*/var/adm/*) but as mail messages are received and log files are recycled, the available capacity stays fairly stable (an estimated growth of about 1% per month). Log files are rotated and purged automatically on a weekly basis, so you'll always have about a month's worth of log information available to you.

Tip: If this file-system is growing rapidly, concentrate your efforts in the /var/spool/mail directory -- look for huge mailboxes (something like "`find /var/spool/mail -size +1000k`" would display a list of mailboxes larger than 1Mb in size). If you find a file much larger than 1,000,000 bytes in size, the user probably isn't retrieving their mail, is on a high-volume mailing list, or their e-mail package isn't configured to remove the mail from the server. Contact the user and/or clear the mail file, using "`> mailbox`", (eg. "`>smithj`" to clear Joe Smith's mail box). Also check the ``/tmp/`" directory, which may need to be cleaned out on an occasional basis (usually old tin* files left over from aborted newsreader sessions, old print files, etc).

The "/usr/" (aka user) file-system, mounted on /dev/hda2, contains user-installable (user meaning user-installed by system administrator) software, things like your web site pages, etc. This is the largest file-system, and is also fairly slow-growth. The log files for the web pages may also be stored here, and grow in size; check and trim them periodically as needed. On my machines, at the beginning of each month the current

web log files are moved to month summary logs (eg. access_log.11 for November's log entries). At the end of the year these logs are all deleted and the cycle starts again (which means each January 1st should see a fair improvement in available space).


Tip: If this file-system is growing rapidly, check the "`/usr/local/etc/httpd/logs`" and the "`/usr/local/squid/logs/`" directories (if you have them). There may be log file(s) that are getting too large (if, perhaps, the web site received a high number of visits). If, however, the logs are purged automatically on a regular basis as I have them, you shouldn't run into any problems with space here (indeed, as the logs are used for statistical analysis of my site's traffic I'd rather not have to delete them if possible). Another place to check for potentially deletable files is in "`/usr/tmp/`".

The "/home/" (aka user's personal home) file-system, mounted on /dev/hda3, contains all the user directories and personal files. Unless you are giving out shell accounts, most of these will be useless and inaccessible to the user (these directories are created when each users' accounts are created, and can later be used to forward the user's mail, etc.). However shell account users, as well as any non-shell accounts which have web pages (eg. personal web pages) will probably have them stored here. In addition, main server pages are stored here in the /home/httpd directory under Red Hat, while other distributions usually place them in the /usr file system.

This file-system is probably the slowest growth unless you are offering a lot of shell accounts.


Tip: If this file-system suddenly grows in size, it is probably because one of your users is adding web pages or binary files in his/her personal space. Check the "`/var/adm/xferlog.*`" log files for recent activity, which should show you which user has added to their web pages.

We also have an "/archive/" (aka archive files) file-system, mounted on /dev/hdb1, which is a spare 1.02 Gb hard drive that can be used for any purpose (eg. data files, software kits, etc.).

We also have a CD-ROM drive, mounted as "/mnt/cdrom/" on /dev/scd0, which is a 24X-speed SCSI CD-ROM device that can read any ISO9660 formatted CD. It is used primarily for software installation, but DOS/Windows CD's can be mounted and then accessed from Windows 3.x/95/NT network shares as needed via a Samba service.

The "`rm`" command will delete a file. Usage is "`rm filename`". If you want confirmation of deletion, use the "`-i`" option (eg. "`rm -i *`"). You would then be asked to confirm each file before it is deleted.


Note: This is the default for normal shell users, but beware -- the root account will not confirm before deleting files unless you specify the "-i" option!)

6.10 Managing Processes

From time to time you may wish to view processes that are running on Linux. To obtain a list of these processes, type “`ps -aux`", which will look similar to the following:

```
USER       PID %CPU %MEM  SIZE   RSS TTY STAT START   TIME COMMAND
bin            69   0.0   1.0    788    320   ?   S     Nov  30    0:00
/usr/sbin/rpc.portmap
frampton 10273  0.0  2.1  1136   664  p0 S    14:12   0:00 -bash
frampton 10744  0.0  1.1   820   360  p0 R    17:25   0:00 ps -aux
frampton 10745  0.0  0.8   788   264  p0 S    17:25   0:00 more
nobody   10132  0.0  1.8  1016   588  ?  S    13:36   0:00 httpd
nobody   10133  0.0  1.8   988   568  ?  S    13:36   0:00 httpd
nobody   10413  0.0  1.8  1012   580  ?  S    14:56   0:00 httpd
nobody   10416  0.0  1.8  1012   580  ?  S    14:56   0:00 httpd
nobody   10418  0.0  1.8  1012   588  ?  S    14:57   0:00 httpd
nobody   10488  0.0  1.7   976   556  ?  S    15:34   0:00 httpd
nobody   10564  0.0  1.8   988   564  ?  S    16:06   0:00 httpd
nobody   10600  0.0  1.8   988   564  ?  S    16:15   0:00 httpd
nobody   10670  0.0  1.8   988   568  ?  S    16:45   0:00 httpd
nobody   10704  0.0  1.7   976   552  ?  S    17:03   0:00 httpd
root         1  0.0  1.0   776   312  ?  S   Nov 30   1:13 init [3]
root         2  0.0  0.0     0     0  ?  SW  Nov 30   0:00 (kflushd)
root         3  0.0  0.0     0     0  ?  SW  Nov 30   0:00 (kswapd)
```

The list shows you the owner of the process (“nobody” for special services such as web servers), the process identification number, the % of CPU time the process is currently using, the % of memory the process is consuming, and other related information, as well as a description of the task itself.

To get more information on a given process, type “`ps pid`" (where “`pid`” is the process identification number). Looking at our example above, “`ps 10704`” would display:

```
10704  ?  S     0:00 /usr/local/etc/httpd/httpd
```

This would tell you that this particular process is a web server.

If you happen to notice a service is not operating, you can use the “`kill -HUP pid`” (where “`pid`” is the process identification number as shown in the process list produced with “ps”). For example, if Internet services (a process called inetd, process #123 in our example) are not working as they should, a “`kill -HUP 123`" (or even safer, use the “`killall`" command and specify the process name: “`killall -HUP inetd`") should restart the process. The -HUP option to the kill command means “hang up”; the process knows that it is supposed to reload itself.

At times, you may find it necessary to temporarily suspend a process, and then resume its execution at a later time. For example, you may be running a CPU-intensive job and wish to burn an IDE-based CDRecordable. Since IDE-based devices rely on the CPU for much of the work behind input/output, they are prone to buffer starvation if your CPU is too busy, and you end up with a useless coaster instead of a properly prepared CD! The following two commands will suspend a process, and the resume it, respectively:

```
kill -STOP 945
kill -CONT 945
```

6.11 Starting and Stopping Processes

The Red Hat distribution of Linux provides a slightly more organized way of managing processes. Instead of hunting and killing them by finding their process id in the process table, Red Hat provides a collection of scripts in the "`/etc/rc.d/init.d`" directory which will allow you to start and stop processes as desired.

For example, to shut down the "`httpd`" (Apache web server) service, simply run the httpd script, as follows:

```
/etc/rc.d/init.d/httpd stop
```

In much the same manner, you can use the "`start`" option to start a service. Or, if you have made changes to a configuration file and wish to restart a service so those changes are recognized, you can use the "`restart`" option.

6.12 Summary

The system keeps track of the following information about each user:
user name This identifier is unique for every user. Example user names are larry, karl,and mdw. Letters and digits may be used, as well as " "and "." (period). User names are usually limited to 8 characters in length.
user ID This number, abbreviated UID, is unique for every user. The system generally keeps track of users by UID, not user name.
group ID This number, abbreviated GID, is the user's default group. Each user belongs to one or more groups as defined by the system administrator.
password This is the user's encrypted password. The passwd command is used to set and change user passwords.
full name The user's "real name," or "full name," is stored along with the user name. For example, the user schmoj may be "Joe Schmo" in real life.

6.13 Check Your Progress

I. Choose the correct answer

1. For creating a new user account we require _____.
    a. create the account
    b. provide an alias to their e-mail address
    c. both a & b
    d. none of the above
2. All Unix passwords, regardless of length (up to a maximum of 8 characters), are stored in the password file as encrypted strings of _____.
    a. 13 characters

b. 16 characters
c. 8 characters

3. user account is removed using _____ command.
a. rm
b. userdelete
c. userdel

4. Unix systems keeps user account information, including one-way encrypted passwords, in a text file called _____.
a. /etc/password
b. /etc/passwd
c. /etc/pwd

5. The /etc/shadow file contents are read _____.
a. only by the current account
b. only by the root account
c. both a & b

6. df command gives the information about _____ space.
a. free space
b. used space
c. none of the above

7. The "/" (aka root) file-system, mounted on /dev/hda1, contains the_____.
a. Linux kernel
b. device drivers
c. Other directories
d. All of the above.

II. Say True or False

1. Passwd command is used for changing the password. True/False
2. The "root" account has security restrictions. True/False
3. The username is at atleast 8 characters long. True/False
4. Change the password using "su" command. True/False

III. Essay type questions

1. Explain the different steps of creating user account.
2. Write a note on the following.
   i. disabling user accounts
   ii. removing user accounts
   iii. changing user passwords
   iv. starting & stopping processes
3. Elaborate the command used for checking storage space.
4. Explain the technique of managing processes**.**

IV. Further readings and other activities

1. Get more information using man or info or any help command for userconf, adduser, chgrp, chown, chmod, uname, umask, id, halt, reboot

EX: 1. $ man passwd 2. $ apropos shutdown

2. Work on all the commands described in the unit and also login as root in your system and carry out the jobs of an System Administrator.
3. For further readings you can refer the following books

1. Title: Linux System Administration Handbook - for detail information on problems of linux installation and administration.

2. Title: Red Hat Linux Starter kit in 24 hours
   Author: Judith Samson etal
   Publication: Sams Techmedia

3. Title: Linux Kernel Internals
   Author: M. Beck etal
   Publication: Addison-Wesley, Second Edition

Reference e-mails: raomvp@yahoo.com roopasindhe@lycos.com
URL / Web Site: http://www.raomvp.bravepages.com

Good-Luck