**Linux Command Synopsis & Bash Scripts**

**Commonly Used Linux Commands**

**pwd** – print working directory

**whoami** – displays who user is logged in as

**ls (-la)** – lists files, regular (l) and hidden (a)

**mkdir <directory name>** – make directory

**rmdir <directory name>** - removes directory

**cd <directory name>** - change directory

**cd ~** - change directory to $HOME

**cd /** - change directory to root directory

**cd ..** – go up a directory

**touch <file name>** – create file

**cp <file name/directory>** - copies file or directory

**mv <file name/directory>** - moves file or directory

**rm <file name>** - remove file

**rm <directory>** - remove directory

**cat <file name>** - concatenates and prints contents of the file

**cat > <file name>** - standard input into a file

**cmd > <file name>** - standard output of command into a file

**more <file name>** - view content of file one page at a time

**less <file name>** - scrollable output of a file

**head** <file name> - displays first ten lines of the file

**tail** <file name> - displays last ten lines of the file

**wc <file name>** - displays the word count for the file

**grep <search string>** - print lines matching pattern

**sed** – stream editor

**awk** – pattern scanning

**cron – Command Run On; crontab – cron table**

\* \* \* \* \* Location of Action

Value Range Value Range Value Range Value Range Value Range  
0-59 0-23 1-31 1-12 0-7  
Minute Hour Day of Month Month Day of Week

January = 1 Sunday = 0

February = 2 Monday = 1

March = 3 Tuesday = 2

April = 4 Wednesday = 3

May = 5 Thursday = 4

June = 6 Friday = 5

July = 7 Saturday = 6

August = 8 Sunday = 7

September = 9

October = 10

November = 11

December = 12

**Example:**

0 23 \* \* 6 rm~/Downloads/\*

This cron job takes place every Saturday (6) at 11pm (2300 hours) for all months (\*) and day of the month where the download directory is removed.

**Example:**

0 6 \* \* 3 /var/spool/cron/crontabs/crontab1.sh

0 = minute; 6 = hour; \* = match any; \* = match any; 3 = Wednesday; /var/spool/cron/crontabs/crontab1.sh executes the crontab1 script located in the /var/spool/cron/crontabs/ directory

**Example:**crontab –e  
59 23 \* \* \* /var/spool/cron/crontabs/priority\_filter.sh – This crontab is set to run at the 59th minute on the 23rd hour on every day – a.k.a. – 11:59 PM every day.

**Bash Scripts**

Nano system.sh  
#!/bin/bash  
echo “Let’s See If This Works!”  
mkdir -p {~/home/backups/diskuse}, {~/home/backups/openlist}, {~/home/backups/freedisk}

Nano system.sh opens the nano editor for a file named system.sh  
mkdir -p {~/home/backups/diskuse}, {~/home/backups/openlist}, {~/home/backups/freedisk} command to make the directories (mkdir) if they do not already exist (-p). The brackets denote specific directories and each bracket separated by the comma (,) and the different directories are strung together using commas in-between.

It can also be written as:

Mkdir -p {~/home/backups/freemem, ~/home/backups/diskuse, ~/home/backup/openlist, ~/home/backup/freedisk

**Rational:**  
This command makes the directories (mkdir) if they don’t already exist (-p) and each of the directories is listed inside a single set of brackets ({}), separated by comma (,).

Nano system1.sh  
#!/bin/bash  
echo “Let’s See If This Works!”  
Free free | mkdir -p >> /~home/backups/diskuse/disk\_usage.txt  
free used | mkdir -p >> /~home/backups/diskuse/disk\_usage.txt  
Isof | mkdir -p >> /~home/backups/openlist/openlist.txt

**Rational:**Nano system1.sh opens the nano editor and file named system1.sh  
#!/bin/bash – bash script  
echo “Let’s See If This Works!” Linux Neophyte wishful thinking  
Free free | mkdir -p >> /~home/backups/diskuse/disk\_usage.txt – the free command (free) orders the amount of free memory (free) to be piped (|) into a new directory and file if it doesn’t exist (-p) ~/home/backups/freemem/free\_mem.txt  
free used | mkdir -p >> /~home/backups/diskuse/disk\_usage.txt – The free command (free) orders the amount of used memory (used) to be piped (|) and append to (>>) the file ~/home/backups/diskuse/disk\_usage.txt if the file does not already exist (mkdir -p).  
Isof | mkdir -p >> /~home/backups/openlist/openlist.txt – The Isof command data is piped (|) and appended (>>) into the directory file /~home/backups/openlist/openlist.txt if the directory file

**Journalctl**

Nano priority\_filter.sh  
mkdir /home/student/directory  
cd/home/student/directory  
Journalctl –l –p “emerg”..”crit” | >>Priority\_High.txt  
Cntl-X   
Chmod +x priority\_filter.sh  
Bash priority\_filter.sh

**Rational:**Nano priority\_filter.sh – Nano editor opens file priority\_filter.sh  
mkdir /home/student/directory – make directory /home/student/directory  
cd/home/student/directory – change directory to /home/student/directory  
Journalctl –l –p “emerg”..”crit” | >>Priority\_High.txt – Journalctl command listing (-l) and filtering (-p) “emerg (0)”..”crit (2)” to the file (>>) Priority\_High.txt  
Cntl-X – Save and close  
Chmod +x priority\_filter.sh – Sets file to execute  
Bash priority\_filter.sh – executes file

**Auditctl**

**auditctl –w /etc/shadow/ - p wra – k hashpass\_audit**

**Rational:**For /etc/shadow/, set wra for the permissions to monitor and set the keyname for this rule to hashpass\_audit

**Auditctl –w /etc/passwd/ -p wra –k userpass\_audit  
  
Rational:**For /etc/passwd/, set wra for the permissions to monitor and set the keyname for this rule to userpass\_audit.

**auditctl –w /var/log/auth.log –p wra –k authlog\_audit**

**Rational:**For /var/log/auth.log, set wra for the permissions to monitor and set the keyname for this rule to authlog\_audit.

**Chmod – Change Mode**

0 = --- 1 = --x 2 = -w- 3 = -wx 4 = r- 5 = r-x 6 = rw-

7 = rwx

Chmod 600 /etc/shadow/ - 6=rw- owner; 0 = --- user; 0=--- others = 600 – sets permissions on /etc/shadow to allow only root read and write access.

Chmod 600 /etc/gshadow/ - 6 = rw- owner; 0 = --- user; 0 = --- others = 600 – sets permission on /etc/gshadow/ to allow only root read and write access.

Chmod 644 /etc/group/ - 6 = rq- owner; 4 = r- user; 4 = r- others = 644 – sets permission on /etc/groups/ to allow root read and write access, and allow everyone else read access only.

​

Chmod 644 /etc/passwd/ - 6 = rq- owner; 4 = r- user; 4 = r- others = 644 – sets permission on /etc/groups/ to allow root read and write access, and allow everyone else read access only.

**User Accounts**

Sudo useradd –m <username> - Add user

/etc/security/ chage –lastday 90 <username> - 90 day password expiration

Sudo usermod –a –G Engineers <username> - command sets all (-a) to the group (-G) Engineers <username>