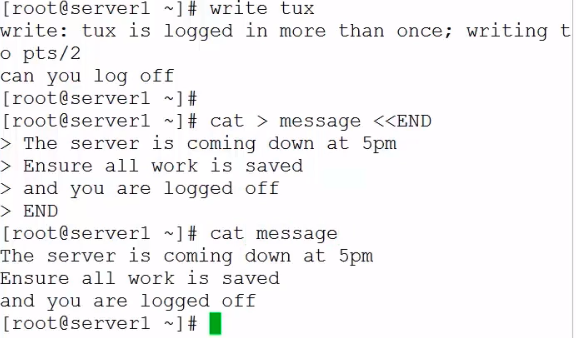
# cat /etc/system-release = to check version of your release

# uname –r OR cat /proc/version = Kernel version

# lsblk = This list block devices eg sda1

SENDING MESSAGES

# writes username 🡪 type your messages 🡪 Cntrl+D



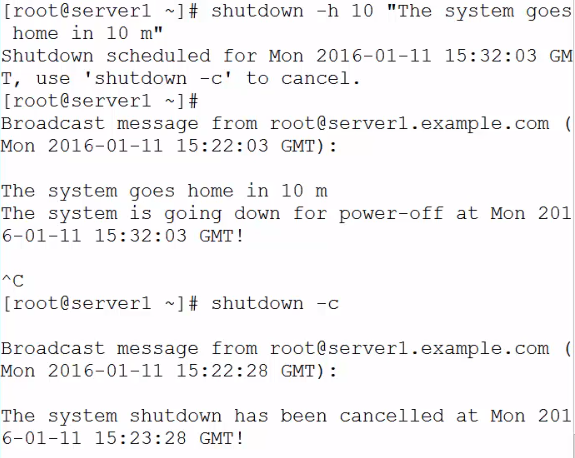
To send to all users 🡪 follow the instruction above

# wall < messages

SHUTDOWN

Rebot | halt | poweroff | init –help | telinit --help

# shutdown –h 10 “The system goes home in 10 m”



Where ^c will terminate shutdown

Shutdown –c = to force termination of shutdown

\*\* when a shutdown message is sent out, 5 mins to shotdown, the /run/nologin file is automatically created

**CHANGING RUNLEVEL & SETTING DEFAULT**

# who –r OR # runlevel = shows you your run level

# systemctl get-default

To link the default target to the names user

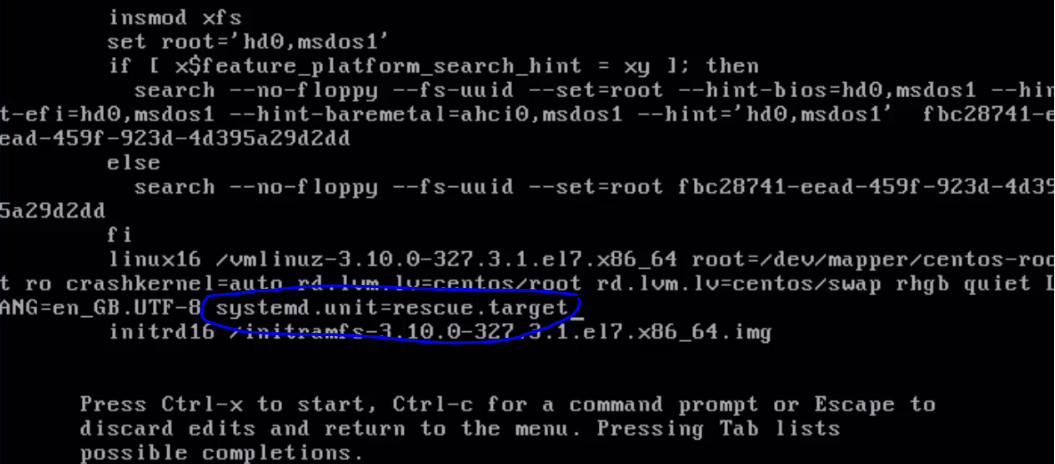
# systemctl set-default multi-user.target

**BOOTING TO RESCUE.TARGET**

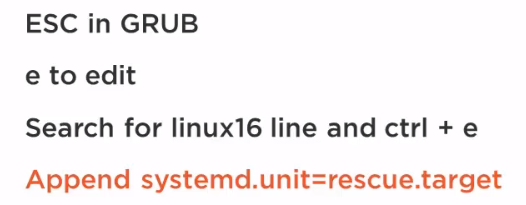
This shows how to interact

with our system by moving into a single user mode and selecting the run level at the command land.

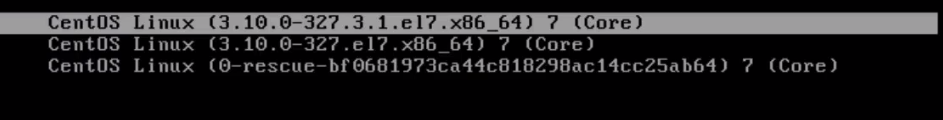
Power on system & click on ESC 🡪 the boot process will proceed into GRUB [the grub 2 menu shows you the kernel you can boot to ]🡪 at the moment the default entry is selected, click E to edit 🡪 move down to the Linux 16 Line to book from the kernel 🡪 type system.unit=rescue.target 🡪 F10



Log in as root user 🡪 runlevl [to know what level]



ENABLE RECOVERY MODE



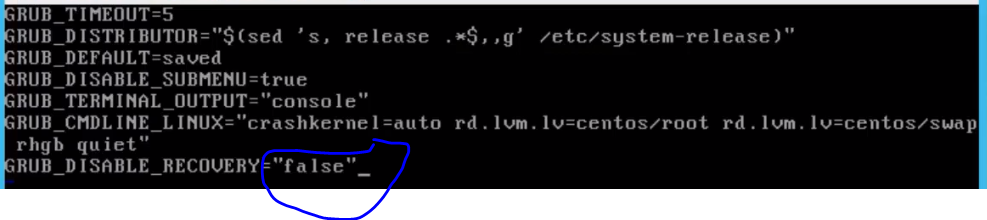
The middle kernel is the kernel at boot up while the top kernel is the updated kernel after running update.

The bottom kernel is the rescue; it is a fall-back kernel if there is no other system. This will just be a rescue image we can boot to.

We will continue the book process with the default run level 🡪 log in to account

Vim /etc/default/grub

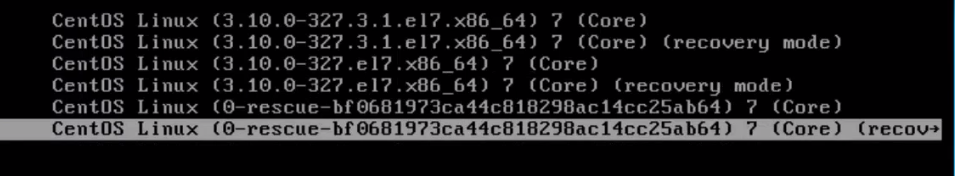
Change GRUB\_DISABLE\_RECOVER=”true” to false 🡪 save and exit



Update the grub configuration

# grub2-mkconfig –o /boot/grub2/grub.cfg

Now reboot the system. You will notice that for each kernel entry, it would have created a recovery entry to be the rescue target

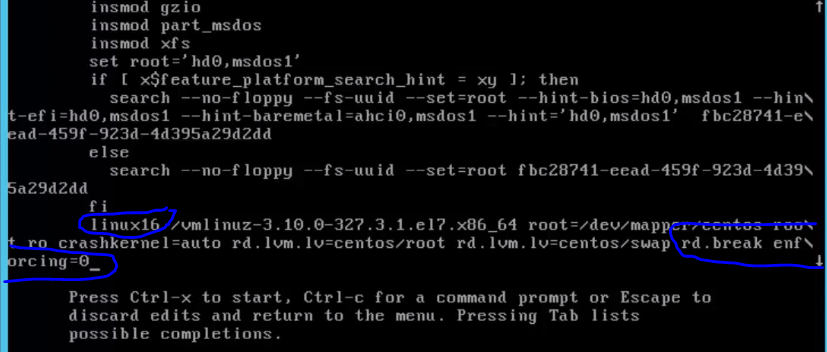


**RECOVER LOST PASTWORDS**

To access account with no password

🡪 power on the system and then click escape

🡪 click on E to edit 🡪 go down to the Linx16 entry



🡪 go to the end of the line, take out **quiet and rhgb**[this is taken off to see the boot process] then replace the end of the line with **rd.break** and **enforcing=0** [this ensures we stop after we have completed the RAM disk space] and because we are using SElinux , we have to change to permissive mode becos it will see errors if we make changes to the password file outside of the normal environment.

Then control+X to continue with the boot process

Then it will change to the switch root file system so we move over,

🡪 control+L

🡪 we have to remount the file system, currently it is mounted read only

🡪 # mount –o remount,rw /sysroot [with this you point a false root that point to the sys root]

🡪 # chroot /sysroot [now we are working on the real file system that was previously mounted on sysroot]

🡪 now we can use the passwd command to change the passwd

🡪 --exit of the chroot

🡪 remount the file system as read only

# mount –o remount, ro /sysroot

🡪 exit

Log in as root user

🡪 to restore the security content for the password file to reset sec content. It was edited outside security linux and lost some settings

# restorecon /etc/shadow

🡪 setenforce 1 = this takes the system back to the normal enforcing mode

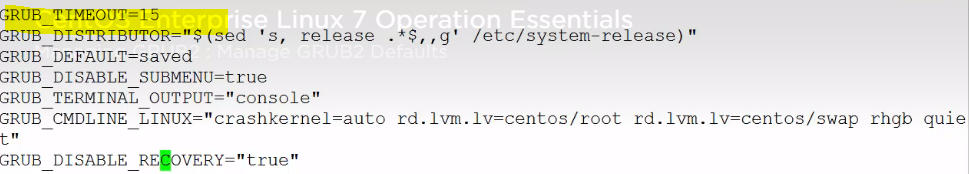
RE-INSTALL GRUB2

# grub2-install /dev/sda

MANAGE GBUB2 DEFAULTS

To modify the time out when making changes to kernels

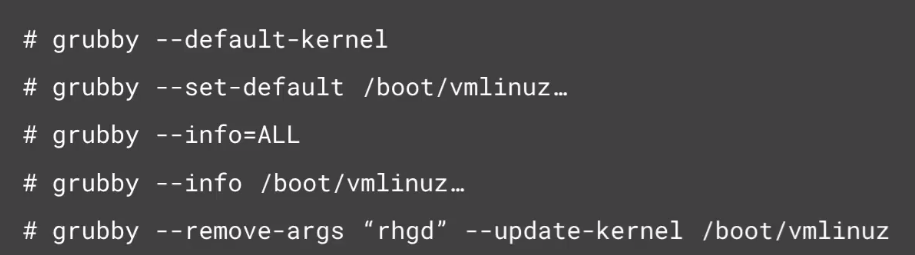
# vi /etc/default/grub2



To reboot grub2 to effect modification on the grub2 file

# grub2-mkconfig –o /boot/gru2/grub.cfg = This is to regenerate the grub file after changes

**MANAGING SETTINGS WITH GRUBBY**



To see kernel settings

# grubby – info=ALL

To view particular kernel

Kernel=/boot/vmlinux-3.10.0.-327.e17.46-64

To make ahcnages

# grubby –remove-arg=”rhgb quiet” –update-kernel !$

Where !$ = the last argument

**PASSWORD PROTECT THE GRUB2 MENU**

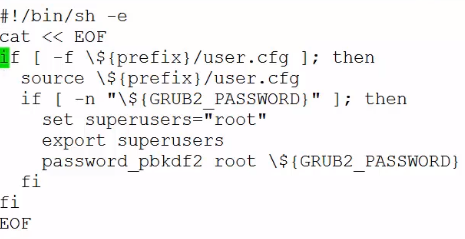
First we will make a copy of the file we are making changes to the users file

# cp /etc/grub.d/01\_users .

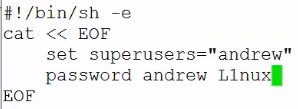
# cd /etc/grub.d/01\_users

# cd /etc/grub.d/

# vi 01\_users



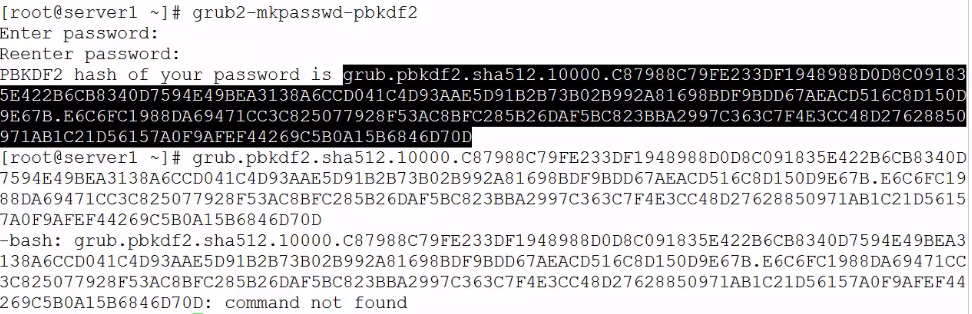
Makes changes to the below



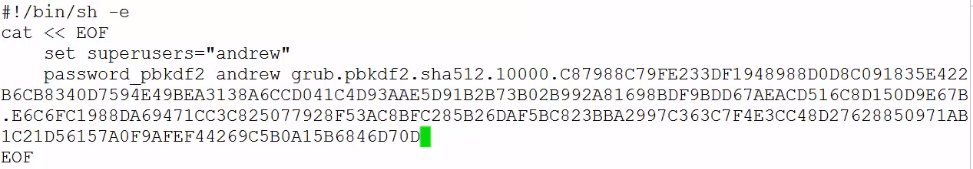
Fyi = The account in the grub file is not related to any linux account, they are restricted to the grub files

Grub2-mkconfig –o /boot/grub2/grub.cfg 🡪 enter

Robot 🡪 choice kernel 🡪 enter username and password 🡪 ctrl+X to allow system boot normally



# vi /etc/grub.d/01\_users





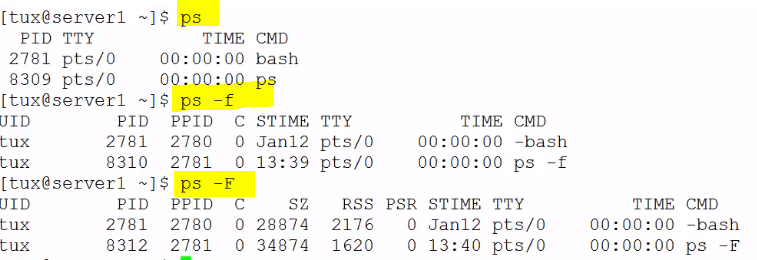
USING PS

Ps –e 🡪 this shows you all the processes on the system

Ps –aux or ps -elf 🡪 to see user processes and processes not assigned to terminal

Ps –e –forest 🡪 shows you a process tree [shows process in hirache]

Pstree



Where –F shows you much more like full size

# ps –l 🡪 this is a long listing, shows you user ID

# ps –ly 🡪 replaces address column with size of memory been use

PROC DIRECTORY

Proc 🡪 ls = shows you directory of all directory

Ps – p1 –f = gives you information about process 1

Ps –p $$ -F

Cd proc 🡪 cat loadave = this shows you load average for 1, 5 and 15mins

KILL

Kill –l = a list of all the signals that can be sent

Kill –p pid | kill –sigkill pid | kill –kill pid = This is to force kill any process

Pgrep sshd = give you all sshd processes

Sleep 100& | pgreep sleep = processes running in the background

Pkill sleep = kill all the processes running in the background

MANAGING PROCESS PRIORITY

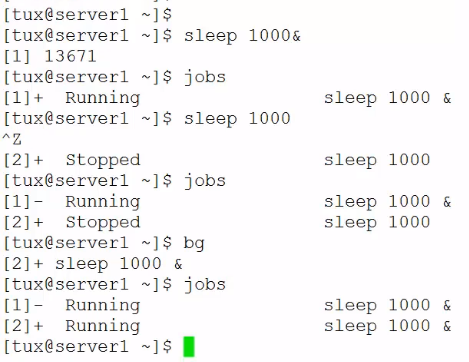
Jobs = to see jobs in the background

Fg = to bring a job foregroud

Sleep 1000 = makes a process to sleep

Sleep 1000& = makes the process sleep in the background

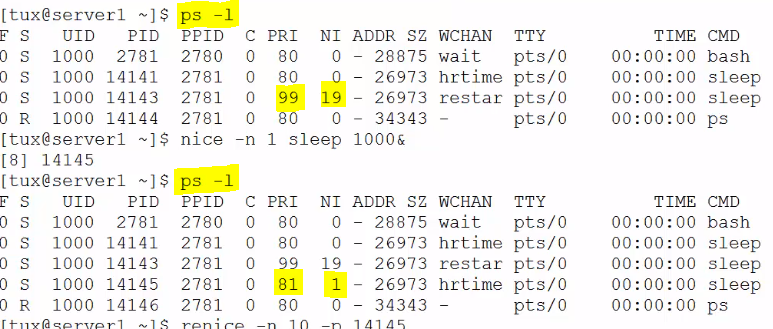
Jobs = to show you running jobs

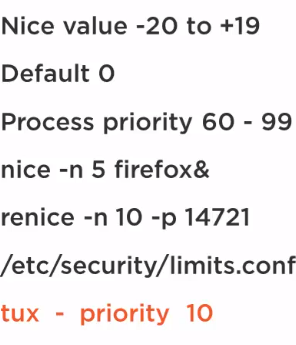


Where bg makes the job run in the background

Fg pid = makes job run in the foreground

NICE





MONITOR LINUX PERFORMANCE

Pgrep sshd = this is to view the pid of running processes

Pmap $$ = shows you the current running process

Pmap 14215 and this tell you Memory address, size of memory in useand what it is been used for

Pwdx $$

Pwdx $(pgrep sshd)

WORKING WITH UPTIME AND LOAD AVERAGE

* LOAD AVERAGE is dependent on how many cpu you have, if you have one CPU, you do not want a value bigger than 1.0. If you get a larger value all it means it that you are queuing CPU request.
* If your idle time is greater than your worktime it means that you have more than 1 CPU

Who | w = you can use this to check loadaverage

Lscpu = that will check how many cpu that you have

Cat /proc/uptime

# watch –n 4 update

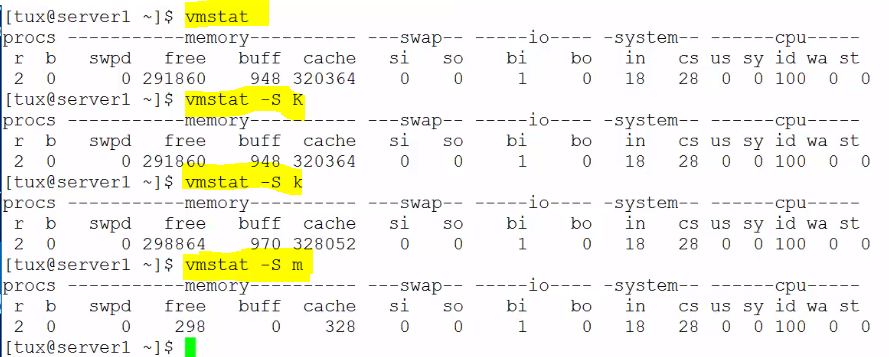
# tload

TOP & VMSTAT

Top –b –n1

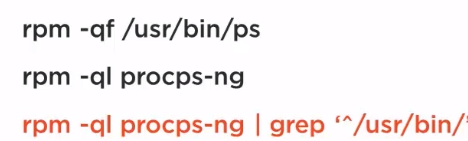
Ni – number of iterations

Vmstat - this is used to collect information



To collect information over a bit of time

# vmstat 5 3 = means we collect 3 iteration. [to collect information 3 times with 5 seconds in between]

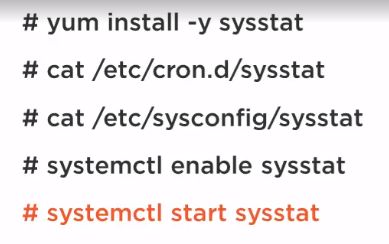


Where rpm – Quarry package management system

SYSSTAT TO MONITOR SYSTMS

To produce and deliver reports on systsm use [processor, memory, disk, and network], outages, and user requests

INSTALLING



Using root 🡪 yum install –y sysstat

# cat /etc/sysconfig/sysstat

Now that sysstat is up and running, we can look at things like;

# iostat | iostat –m | iostat 5 3 = this is looking at our disk activity displayed in megabite, with a delay of 5 secs and reading information 3 times.

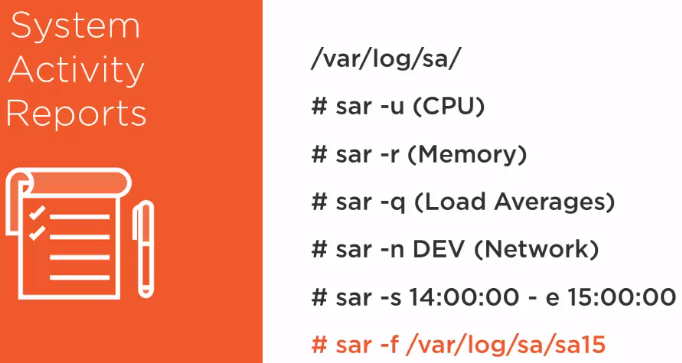
#pidstat –p $$ 5 3

# mpstat –P all 2 3 = This has a 2 second pag and it running you 3 times

# sar –u – reportingon memory utilization

# sar –b disk I/O

# SAR –q



The idea of having information on cron file is so that we can have information captured every 10mins so we are able to build up a picture of our system performance over a period of time.

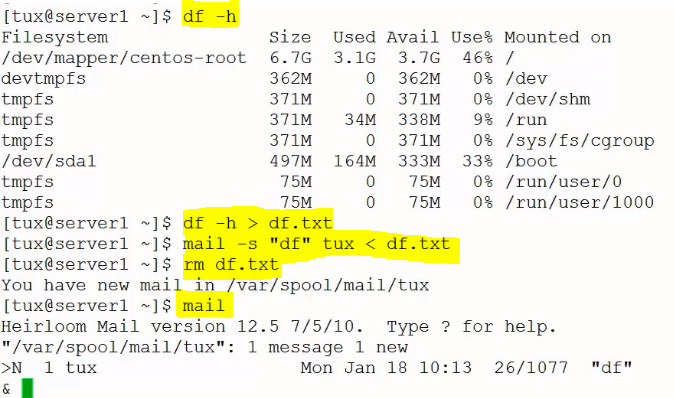
Then we can use our # sar tool to report on that data

SHARED LIBRARIES

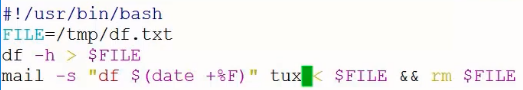
A shared library is a file containing object code that several a.out files may use simultaneously while executing. When a program is link edited with a shared library, the library code that defines the program's external references is not copied into the program's object file.

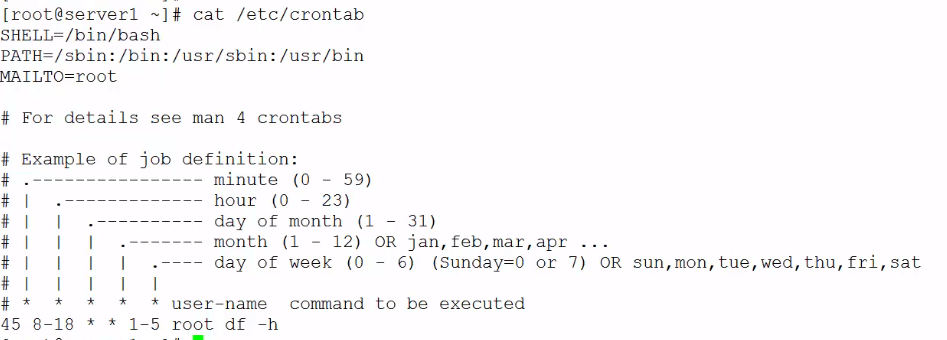
\*\* To create your own directory structure, we can use **/etc/ld.so.confog.d/** , rather than just hae a singl conf fit, it is generally most linux distribution linux including centos 7 to use an extention difrectory, so we can then go and add in our new location within the configuration file, .conf file within the /ld.so.conf.d/

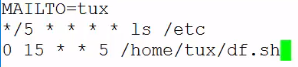




Vi df.sh







# crontab –l > shows you cron tabs

# crontab –e > prompts you to start a new cron

# crontab –r > this removes the cron user

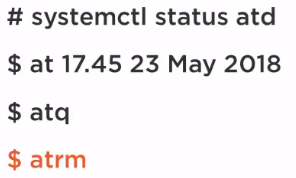
ANACRONTAB

# cat /etc/anacrontab

#v i/etc/anacrontab to insect details on your laste argument…

Period delay job-id command

SCHEDULING JOBS WITH AT

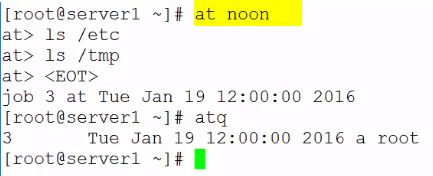


# systemctl status atd > check if “at demon is running”

To exit the file below is ctrl+D to end

Atq will list jobs

Atrm will remove jobs



To schedule a job at in the future

# at 13:23 jun 23

Ls /etc

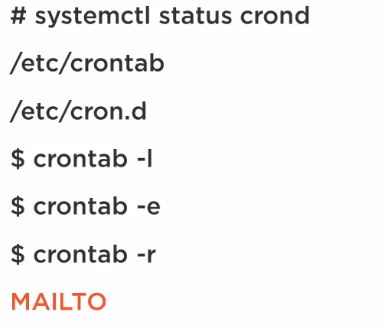
Ctrl + D

# atq

To remove job atrm4, where 4 stands for job num

MALTO

If we populate the mailto variable, either in the system crontab or the user crontab, then we can ensure that if a command has output, we can ensure that the output is mailed to our user



LOG FILES

AUDITING LOGIN EVENTS

# lastlog | grep –v “Never” - This pulls lines from a file not containing the word never

Last user | last –n 10 user | last reboot | last wtemp | subo lastb

Wtemp = shows good login events

Btemp = bad login events

Auditing Root Access

This shows you SU and SUDO effect in the /var/log/ files

# grep sudo secure\*

AWK TO ANALYSE LOGS

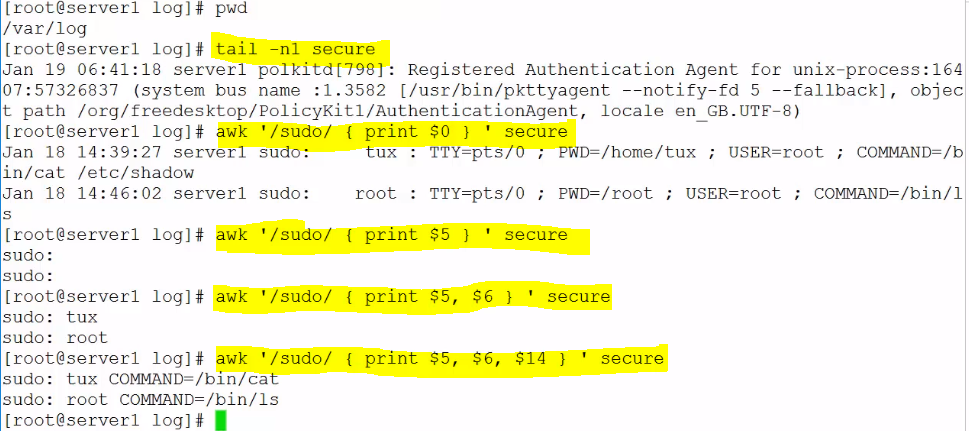
Example: # tail –n1 secure

# awk ‘ /sudo/ { print $0 } ’ secure

Using a default separator of wide space, show the rant of line to look for which is /sudo/

Then the main block of code to execute print , $0 means the whole line

# awk ‘ /sudo/ { print $5, $6 } ’ secure - This chooses field which you want



Writing a script to perform the above function

Cd ~

Vi secure.sh

Where

Awk /1/ is the range to the system, it is an input to the sript

{ print $5, $6, $14} block of codes

$2 is the file to look at

Using single quotes, we do not allow our variables to expand





RSYSLOGD

# rsyslog\*

The rule content lets us know how contents get into the file. In this structure, /var/log/messages,here we are sending anything .info or higher. But we do not include

mail, authpriv and crone



**Adding our own rule**

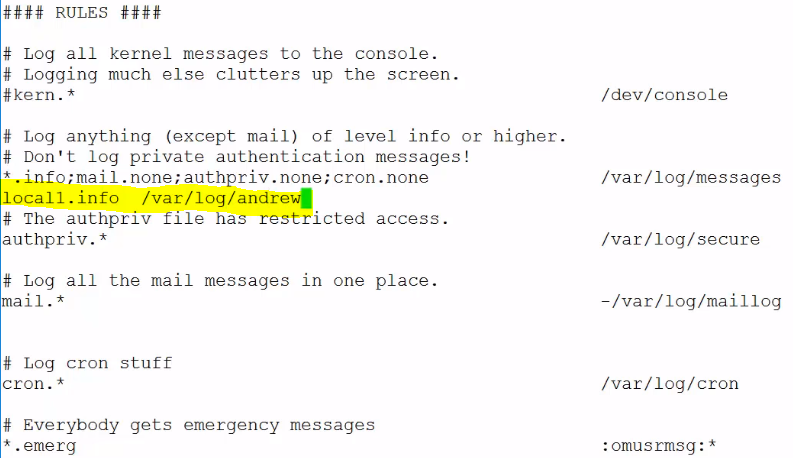
# vi ryslog.conf

Local.info /var/log/andrew

After adding rules, restart rsyslod

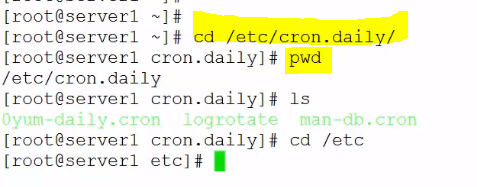
# systemctl restart rsyslog

# logger –p local1.warn “Test message”



ROTATING LOG FILES

This helps us to reduce the amount of files we save in the /var/log/ structure. By default log rotate runes once a day via cron

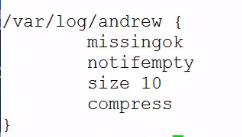


# less logrotate.conf

In here we can see that the default says weekly files are rotated weekly keeping 4 copies. In here we can create empty logrotate and specify permissions with them.

# Vi logrotae.conf

/var/log/Andrew



Missingok - don’t worry, don’t create errors if it is not there

Noifempty – don’t rotate it if there is no content

Size 10 - if it gets to 10 bytes or more

Ls /var/log/Andrew\*

Logrotate /etc/logrotate.conf

Run listing again: ;s /var/log/Andrew\*

JOURNALCTL

# journalctl | journalctl -n | journalctl n 15 | journalctl -f | journalctl –u sshd.service |

Journalctl --since “2016-01-21 12:00:00” | journalctl –list-bots | journalctl –b -1 (everything from previous boot) while journalctl –b shows you the current boot

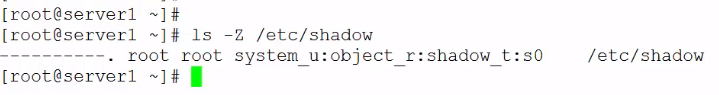
# journalctl –b = Information since the last boot

To modity the journalct, there are different options which are;

1 – mkdir /var/log/journal

2 – vi /etc/system/journal.conf

SELINUX



U= user

R = object

T=type

Gentenforce | sestatus |

With enforcing, we both read the rules and apply the rules

In Permissive, it allows things to happen but would log everything happening

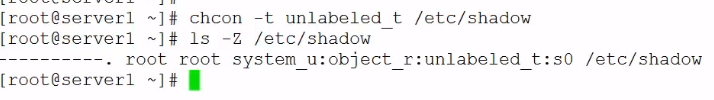
Sentenforce 0 is permissive while setenforce 1 is enforcing

VIEW SELINUX LOGS

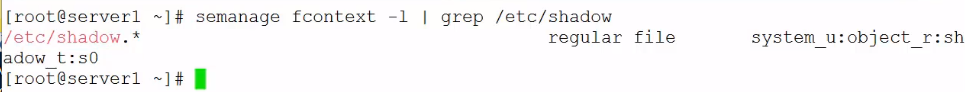
# ls –z /etc/shadow

# ausearch –m avc (access vector control)

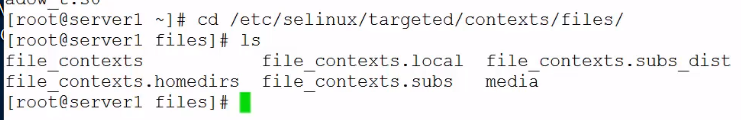
To change contest

  
# ausearch –m avc –ts recent

# restorecon /etc/shodow = to get this file labelled correctly



The default file where you can find where all label’s are listed for files and directory



BOOLEANS AND PORTS

Sebooleans allows policies to be changed with on and off settings.

# getsebool -a | wc –l

# semanage bolean –l = description of the current and default state



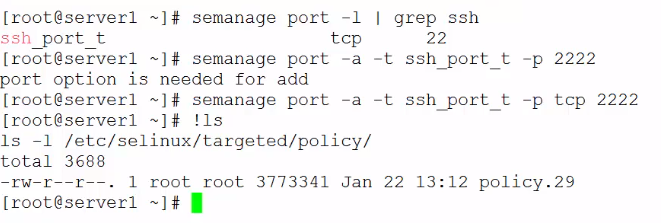
To turn this on;

Setsebool getsebool httpd\_read\_user\_content on



The –p makes it permanent but setting it without the p makes it memory.

Making changes to policy so you do not have to disable selinux

g



Notes: The audit logs helps you check everything not working properly

Ausearch is to search for av and the recent show you more recent error.

You can restore a file to restore the password file to it original context

To manually change, you can use chcon



If we want to look for ways to modifiy policies just to tune them abit to work in an eviroment then we use Booleans.

Semanage shows you current and default state as well as descriptions.

If you wan to change a Boolean value it is setsebool

Using the –P would make it permanent

Semanage ports will list all the defined ports and the value they can manage

To add in a port number, we add the port to the end.