bare metal approach

OS - kernel + appln s/w

2 parts

kernel

appln s/w

kernel-a platform for h/w

collection of drivers

drivers - connection to peripherals(piece of code)

types

-windows

1. best graphics

2. heavy

3. cost ineffective

4. less secure-kernel is not accessible or not open source

5. not customizable

=Linux 1991, Linus Torvalds

1. best command line interface

2. light weighted

3. free

4. customizable

5. secure (open source)

-unix

hypervisor

A hypervisor is a software that allows multiple virtual machines (VMs) to run on a single physical machine

types

Type 1

Also known as "bare metal", this type of hypervisor runs directly on the host's hardware. It's lightweight and acts like an operating system.

Type 2

Also known as "hosted", this type of hypervisor runs as a software layer on an operating system.

types of user

full privileges

appln associated user

least privilege account

Shell

hostname

path

2types

1. absolute

full path of file including / root dir

2. relative

starts with . .. ~

. means pwd

.. means pwd's parent dir

eg touch gip -> system assume it as touch ./gip

this path represents in absolute path as /root/gip

touch ../gip

file created in / as it is root parent dir

commands

.pwd

.mkdir /techno

.mkdir /copy

.pwd

.ls /techno/

.touch .techno/file{1..5}.txt

.ls /techno/

.cp /techno/file1.txt /copy/

.ls /copy/

.cp /techno/\* /copy/

.rm - remove

.rm -f dest (-f to remove all files forcelyw/o permission)

.cp - copy

copy specific file of similar names

cp /techno/\*.txt /copy/

.mv - move

mv /copy/dire2/ /move/jk --here jk is new name of dire2

mv /move/jk /move/pk -- rename dir

.mkdir - create directory

copy directory with -r copy all files in dir recursively (cp -r)

.cp -rv - print the info that what we have done

.rmdir - remove directory

removing dir contains data then remove it by "rm" cmd with -r to remove content also and also use -f to remove forcely so it dont ask every time for permission to remove files

to "remove only" files in dir then dont use -r with rm cmd

.ls - list

hidden files are starts with .

to see hidden files -> ls -a

.!(line no) to run cmds from history

.!!: Repeats the last command.

.!string: Executes the most recent command that starts with string.

eg. !ls

.

.export HISTTIMEFORMATE = '%d/%m/%y %T' changes format of history

editors

vim

3modes

insert esc+i

execute(default) esc + nothing

yy - copy

3yy - 3 lines copy

b - paste

2b - 2 times paste

dd - delete

shift+g - bottom of page

u - undo

command esc + :

.:q! forcefully exit

grep

used to search pattern in files

grep "pattern" "files name"

-i ignore case sensitivity

-n no of line

-c count

-v print line in which pattern not matched

-w

use on output

ps -aux | grep system

ps -aux | grep system >/redirection

redirection

redirect input output or errors to a file

> - overwrite

>> - append

types

-inout

-output

-error

-output and error redirection

> file Redirect stdout to overwrite a file

>> file Redirect stdout to append to a file.

2> file Redirect stderr to overwrite a file.

2> /dev/null Discard stderr error messages by redirecting them to /dev/null

> file 2>&1 || &> file Redirect stdout and stderr to overwrite the same file

>> file 2>&1 || &>> file Redirect stdout and stderr to append to the same file.

store error and output in diff. files - grep -inw jenkins ansible new devops\_tools > output.txt 2> error.txt

pipeline("|")

cmd 1 (output)|(input) cmd 2 (output)|(input) cmd 3

eg head -15 file | tail -4

man chattar

chattar +a /redirection

lsattr /redirection

chattr -a /redirection

stat - see metadata

user mgmt.

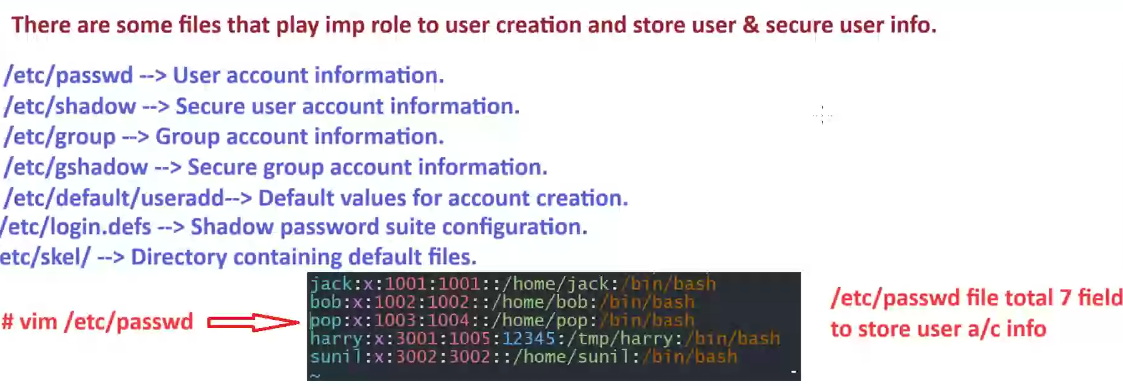
useradd "name"

id "name"

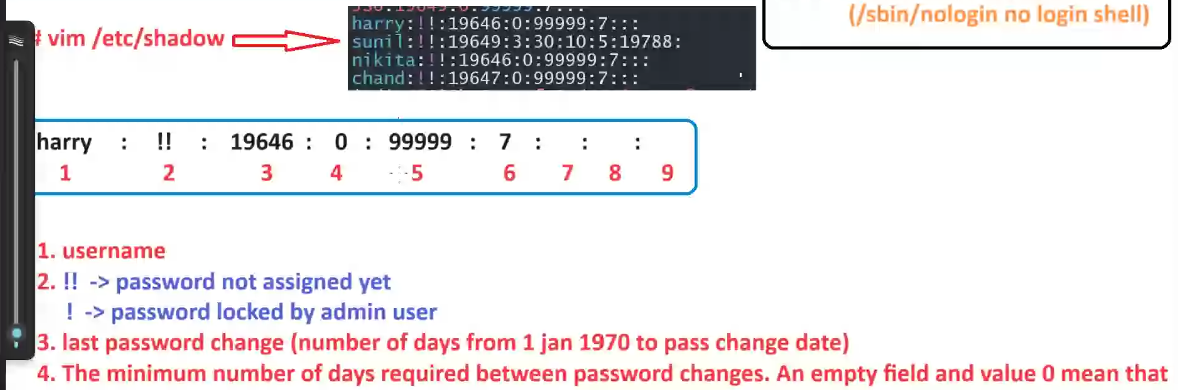
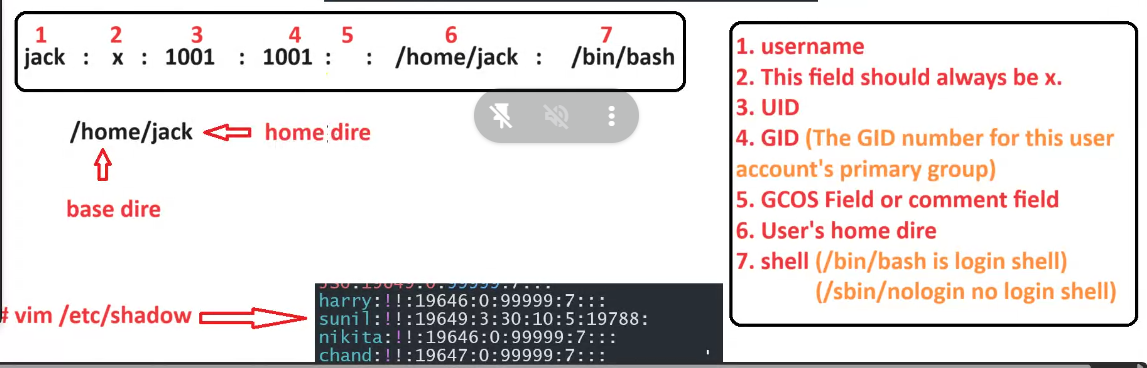
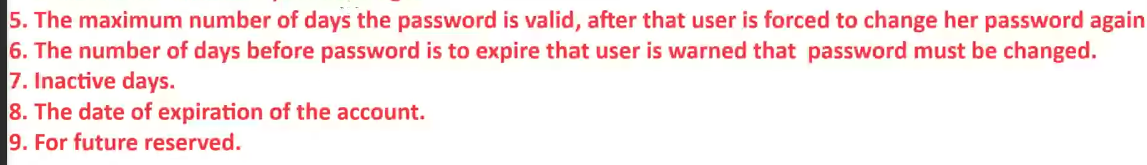
gid - primary group id

groups - tells all groups including secondary group

group requirement - to give permission to multiple users

 man useradd

/etc/passwd – fields 7



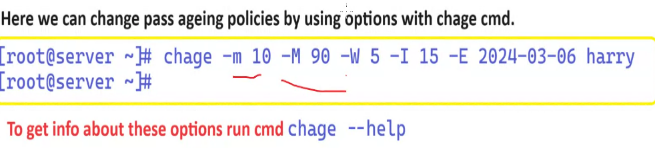
su – “username” -> user change

exit - > back to previous user

passwd “user” -> change pass

chage -l -> list pass aging policy

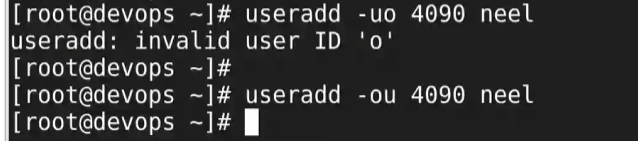
chage –help

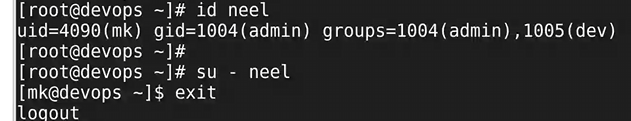
chage -d 0 harry -> user has to change password while login first time

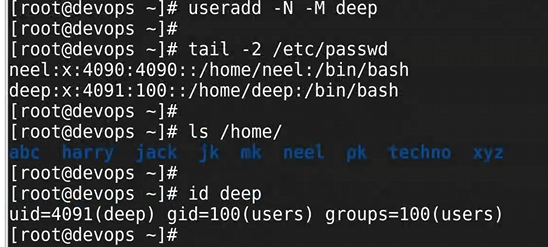
to see groups – vim /etc/group

secondary group user will display in group file

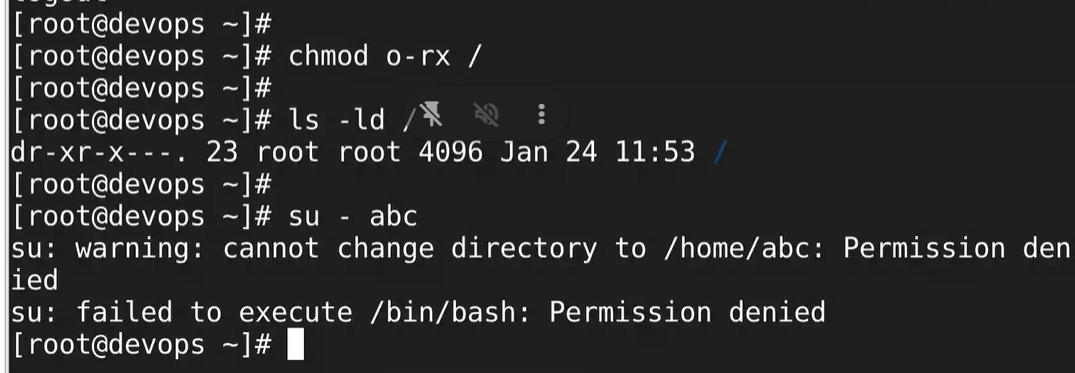
in -u value is with itself







Chown



file permissions

root can read write file if all permissions are 0

read -> cat, head, tail, grep(read only mode)

write -> vim, nano, redirection(>,>>)

execute -> run a program, run a script(sh)

.if any file has no read permission then even root cannot execute that file.o

by default there is no execute permission so we manually give execute permission.

for directory

read -> ls, ls, -l, ll

write -> mkdir, touch, mv, cp, rm, rmdir

execute -> to switch into the dir

execute permission give process to enter resources/directory.

Sudo – super user ddo

User mgmt. – useradd userdel usermod

Package mgmt. – rpm , yum

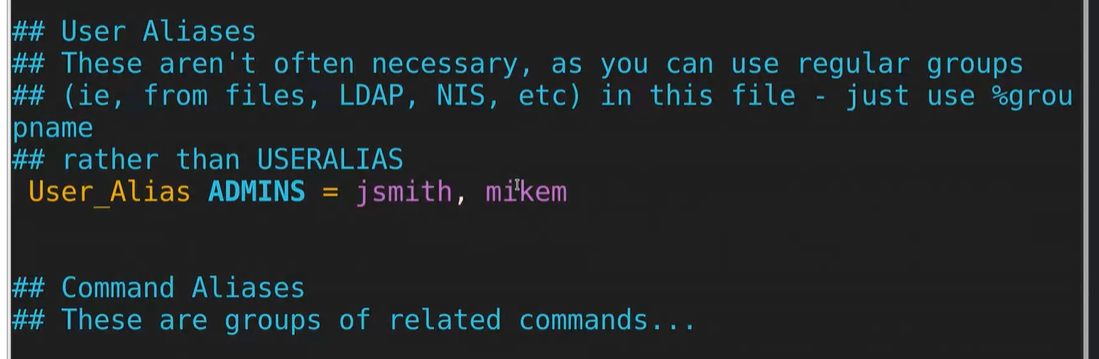
Networking –

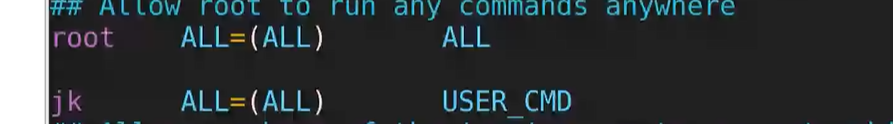
/etc/sudoers – modify this file (vim /etc/sudoers don’t use)

Use visudo because it tells the syntax error while vim not gives syntax error

visudo

* <user> ALL=(root) if root->All then it gives all user permissions with root



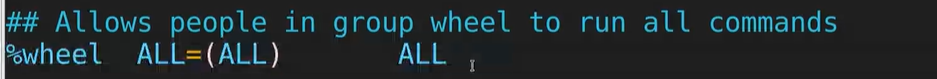


To check access of user -> sudo -m

To give a user permission of a alias excluding some permission



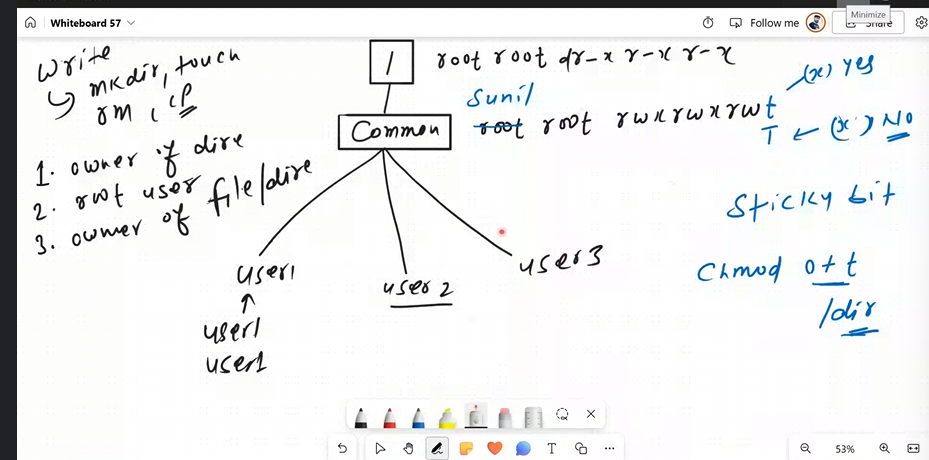
Give access to group



**Special permission**

Sticky bit, SUID, SGID

Sticky bit – In common file of users any user can delete a file of other user



t-execute

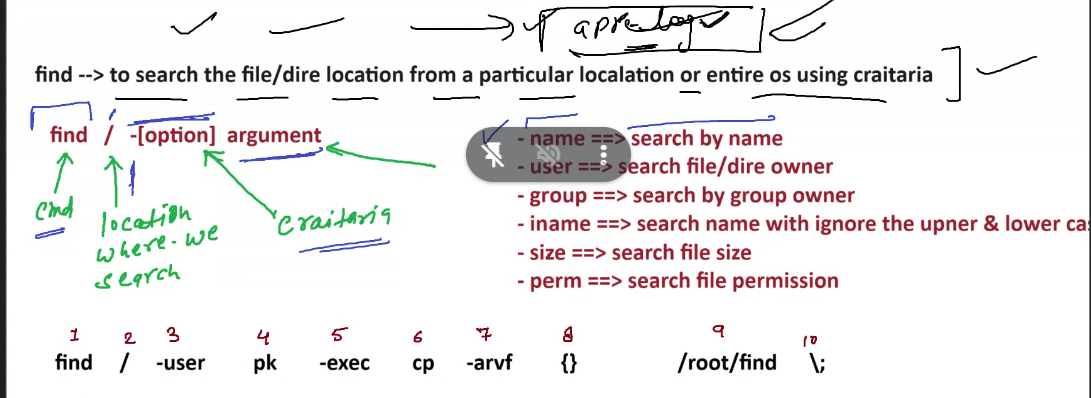
T-no execute

Suid – on an executable file if we implement suid then

Package management

Find cmd : to search a particular location of file

find <location> –[option] <argument>



type f\d

du -h <location>– disk usage

-sh – individual diro

NTP – network time protocol

**Key based authentication-**

Ifconfig – ip address