USERS AND GROUPS

Jevitha K.P

Secure Coding Lab 5

Adapted from "Computer Security: A Hands-on Approach" by Wenliang Du

USER

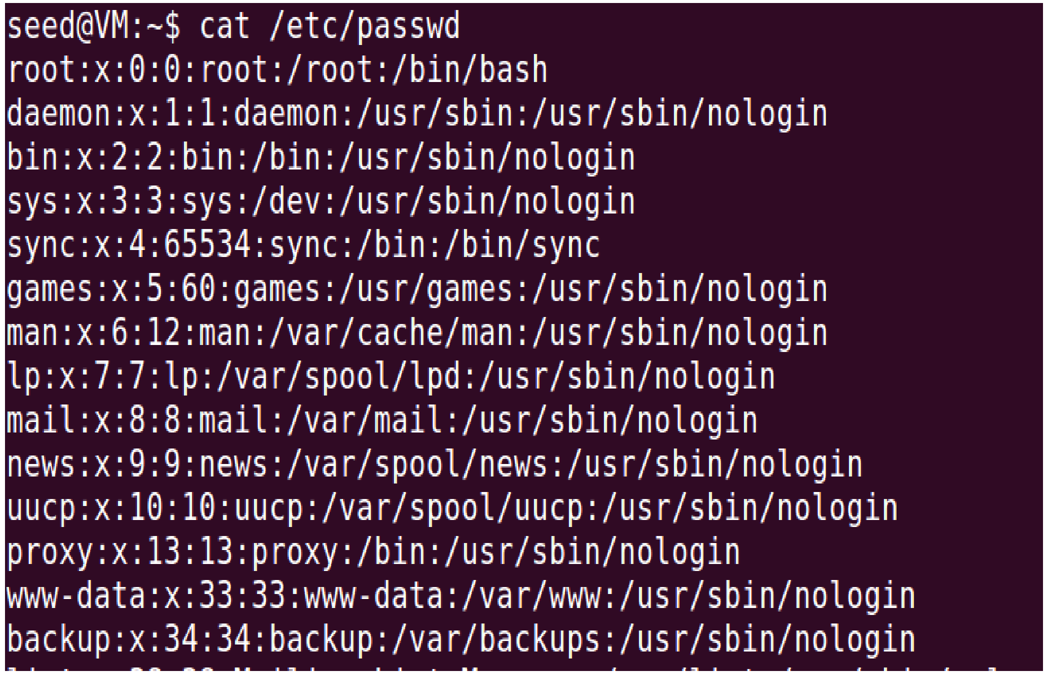
- OS identifies users who login, using userid
- In Linux, userid is just a number and every user is assigned a unique number (userid)
- Request for accessing a resource is verified using userid by the OS using the access control list
- Special user in linux – Root – which has user id 0 – Privileged account
- Any account can be a root user by having userid as 0

USER INFORMATION

- When creating a user account, where are the user information stored ?
- Where can we find the list of users on the system ?

USER INFORMATION

- User information is available either in database or files, depending on the OS.
- In linux, user information is available in a file - /etc/passwd
- Every user is listed in a line
- Contains – userid, groupid, home directory, shell pgm to be used, etc



```
seed@VM:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
```

USER INFORMATION

- Shell – first command that will be executed after user login
- Notice the root user, seed user, etc.
- Others are not real users, the ones without a shell – account created for special use

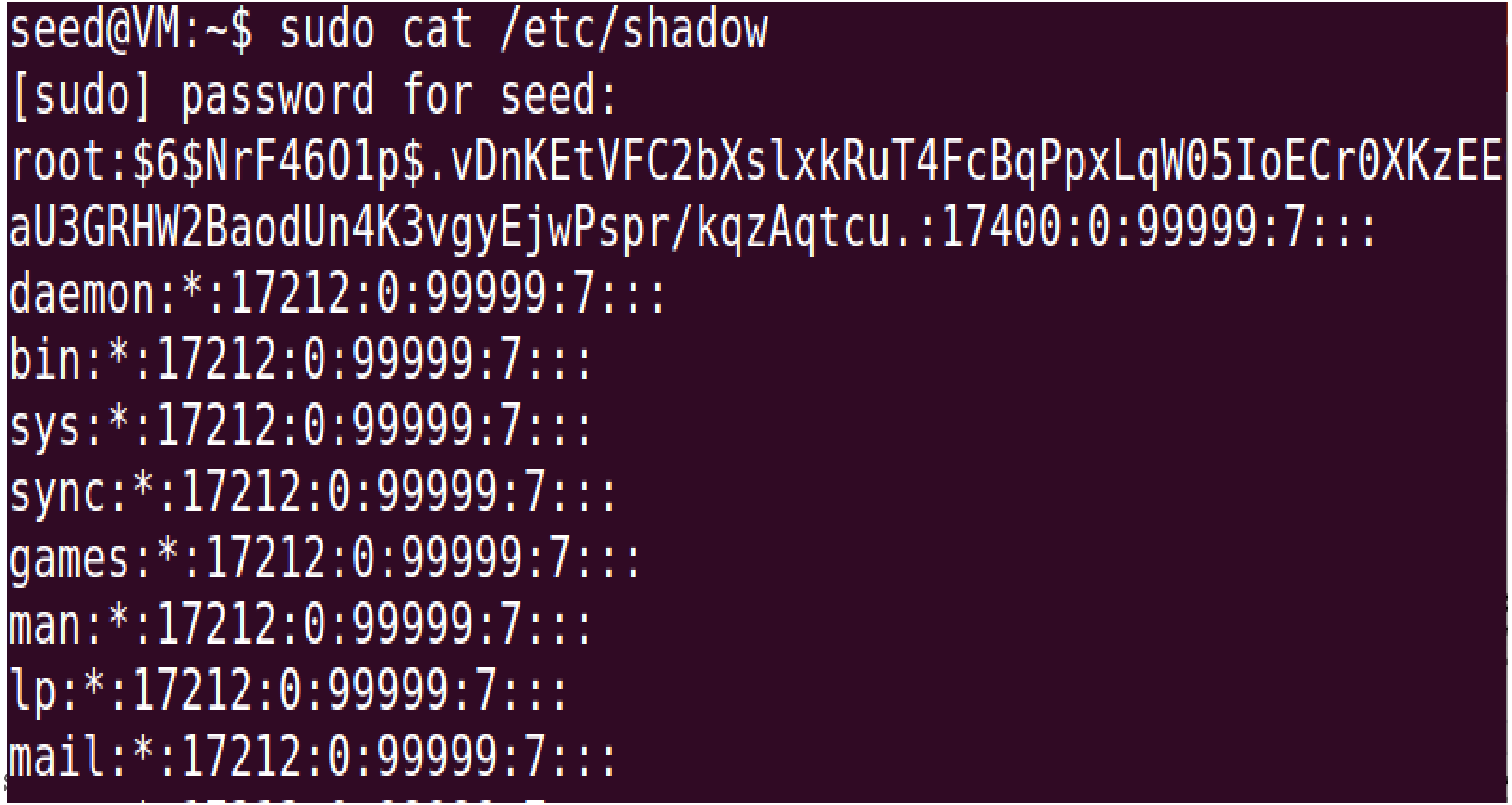
```
seed@VM:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
seed:x:1000:1000:seed,,,:/home/seed:/bin/bash
alice:x:1001:1001:alice,,,:/home/alice:/bin/bash
...
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
telnetd:x:121:129::/nonexistent:/bin/false
sshd:x:122:65534::/var/run/sshd:/usr/sbin/nologin
ftp:x:123:130:ftp daemon,,,:/srv/ftp:/bin/false
bind:x:124:131::/var/cache/bind:/bin/false
```

Where is the password field??

PASSWORD INFORMATION

- Second field – x - password field
- X – look for password in a separate file
- /etc/shadow file contains the password.
- Why two files?
- Password was stored in passwd file, but it is world readable and also contains other useful information (home dir, shell, etc) which is required by other programs
- Also, users use weak passwords, so even though encrypted, can easily expose them
- Hence, linux moved passwords to a different file – /etc/shadow file, readable only by root.

PASSWORD INFORMATION



```
seed@VM:~$ sudo cat /etc/shadow
[sudo] password for seed:
root:$6$NrF46O1p$.vDnKEtVFC2bXslxkRuT4FcBqPpxLqW05IoECr0XKzEE
aU3GRHW2BaodUn4K3vgyEjwPspr/kqzAqtcu.:17400:0:99999:7:::
daemon:*:17212:0:99999:7:::
bin:*:17212:0:99999:7:::
sys:*:17212:0:99999:7:::
sync:*:17212:0:99999:7:::
games:*:17212:0:99999:7:::
man:*:17212:0:99999:7:::
lp:*:17212:0:99999:7:::
mail:*:17212:0:99999:7:::
```

RELATED COMMANDS

- $ ls –l /etc/passwd /etc/shadow
- $ cat /etc/passwd
- $ sudo cat /etc/shadow
- Print id and group information
- $ id <-- user id, group id and group information

```
seed@VM:~$ ls -l /etc/passwd /etc/shadow
-rw-r--r-- 1 root root   2571 Oct 20 05:05 /etc/passwd
-rw-r----- 1 root shadow 1621 Oct 20 05:08 /etc/shadow
seed@VM:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
…
seed@VM:~$ cat /etc/shadow
cat: /etc/shadow: Permission denied
seed@VM:~$ id
uid=1000(seed) gid=1000(seed) groups=1000(seed),
4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpa
dmin),128(sambashare)
seed@VM:~$
```

RELATED COMMANDS

- Add a new user
- $ sudo adduser bob

- Alternate manual way :
- Add entry to passwd and shadow file

```
seed@VM:~$ sudo adduser alice
Adding user `alice' ...
Adding new group `alice' (1001) ...
Adding new user `alice' (1001) with group `alice' ...
Creating home directory `/home/alice' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for alice
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n]
seed@VM:~$
```

RELATED COMMANDS

- Switch user
- Su alice
- $ id
- Change password
- $ passwd
- ls –l /etc/passwd /etc/shadow
- Notice that the time stamp of the shadow file is updated

```
seed@VM:~$ su alice
Password:
alice@VM:/home/seed$ id
uid=1001(alice) gid=1001(alice) groups=1001(alice)
alice@VM:/home/seed$ ls -l /etc/passwd /etc/shadow
-rw-r--r-- 1 root root   2566 Oct 20 06:19 /etc/passwd
-rw-r----- 1 root shadow 1621 Oct 20 06:19 /etc/shadow
alice@VM:/home/seed$ passwd
Changing password for alice.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
alice@VM:/home/seed$ ls -l /etc/passwd /etc/shadow
-rw-r--r-- 1 root root   2566 Oct 20 06:19 /etc/passwd
-rw-r----- 1 root shadow 1621 Oct 20 06:22 /etc/shadow
alice@VM:/home/seed$ sudo cat /etc/shadow
[sudo] password for alice:
alice is not in the sudoers file.  This incident will be
reported.
alice@VM:/home/seed$
```

GROUP

- Users can be added to one or more Groups
- Groups are created by assigning the required users to a specific group
- Manage permissions on the group rather than the individual users separately
- Group details are available in /etc/group file
- Just need to add to that line to become a member of the group, which provides the user the permissions assigned to that group
- $ cat /etc/group

```
alice@VM:/home/seed$ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,seed
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:seed
floppy:x:25:
tape:x:26:
sudo:x:27:seed
alice:x:1001:
```

ACCESS CONTROL

- When a process accesses a resource like file, OS needs to know whether the process is allowed to access it or not – Access Control
- Different models, different mechanisms
- Most used is ACL – Access Control List
- Example:

```
seed@VM:~$ ls -l
...
drwxr-xr-x 2 seed seed 4096 May  9  2018 Downloads
-rw-rw-r--  1 seed seed    0 Oct 20 05:25 file1
```

- Permissions - Owner , group and others (r- read, w-write, x-execute)
- Ownername
- Groupname

CHANGE PERMISSIONS

- Owner | Group | Others : rw- | r--| ---
- Binary to decimal – 110 | 100| 000 = 6 | 4 | 0
- $ chmod 640 myfile
- Alternate ways : check man page
- Homework – remove x from all groups

```
seed@VM:~$ ls -l file1
-rw-rw-r-- 1 seed seed 0 Oct 20 05:25 file1
seed@VM:~$ chmod +x file1
seed@VM:~$ ls -l file1
-rwxrwxr-x 1 seed seed 0 Oct 20 05:25 file1
```

PERMISSIONS ON DIRECTORIES

- $ ls –l dirname

drwxrwxr-x

Three groups – owner | group | others

Permissions :

R - list contents of a folder

W - Create files / sub folders in a folder

X - Enter a folder (Cannot execute a folder)

```
seed@VM:~$ ls -l
total 64
drwxrwxr-x 4 seed seed 4096 May  1  2018 android
drwxrwxr-x 2 seed seed 4096 Jan 14  2018 bin
-rw-rw-r-- 1 seed seed    0 Oct 20 05:25 file2
```

DEFAULT PERMISSIONS

- When a new file is created, OS assigns a default set of permissions
- Default permission when a file is created – 110 | 110| 110
- Default permissions are decided by 'umask' - umask of the current process
- $ umask

0002

$ touch file1 && ls –l file1

-rw-rw-r--1 seed seed date file1

$ umask 0077

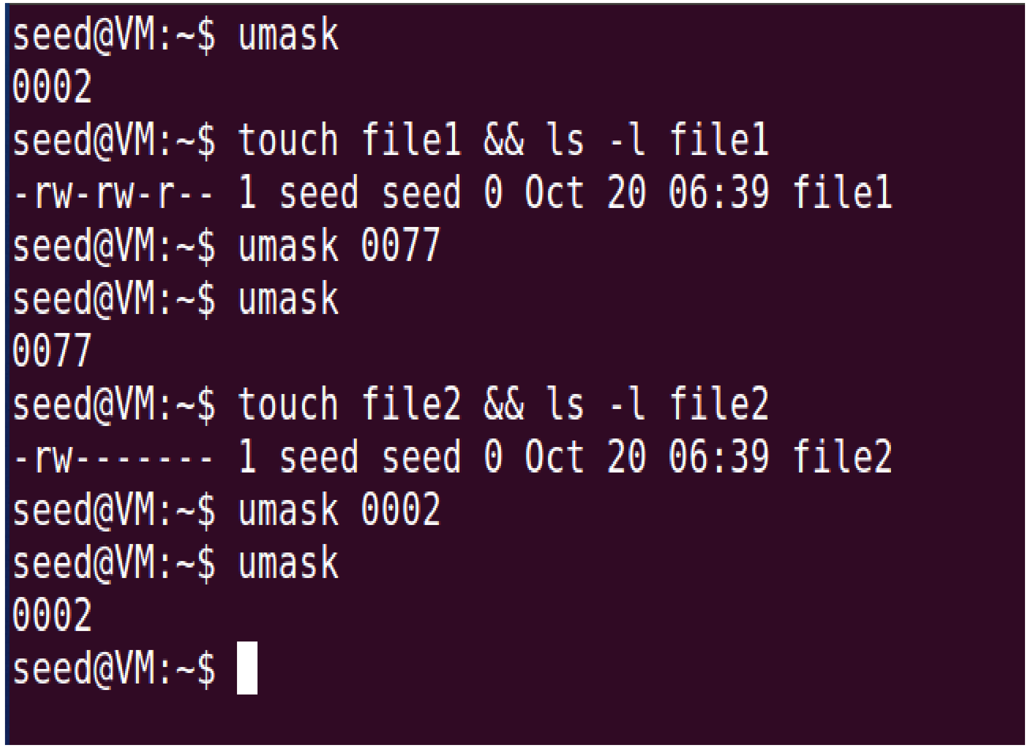
$touch file2 && ls –ld file2

-rw------ 1 seed seed date file1

File1: 1 1 0 1 1 0 1 1 0 mask with 0 0 0 0 0 0 0 1 0 = 1 1 0 1 1 0 1 0 0

File2: 1 1 0 1 1 0 1 1 0 mask with 0 0 0 1 1 1 1 1 1 = 1 1 0 0 0 0 0 0 0

$ umask 0002



```
seed@VM:~$ umask
0002
seed@VM:~$ touch file1 && ls -l file1
-rw-rw-r-- 1 seed seed 0 Oct 20 06:39 file1
seed@VM:~$ umask 0077
seed@VM:~$ umask
0077
seed@VM:~$ touch file2 && ls -l file2
-rw------- 1 seed seed 0 Oct 20 06:39 file2
seed@VM:~$ umask 0002
seed@VM:~$ umask
0002
seed@VM:~$ 
```

CHANGE OWNERSHIP

- The user who creates file is the owner of the file
- $ sudo chown root file1 <-- change ownership to root

```
seed@VM:~$ ls -l file2
-rw------- 1 seed seed 0 Oct 20 06:39 file2
seed@VM:~$ chown alice file2
chown: changing ownership of 'file2': Operation
not permitted
seed@VM:~$ sudo chown alice file2
seed@VM:~$ ls -l file2
-rw------- 1 alice seed 0 Oct 20 06:39 file2
seed@VM:~$ sudo chown seed file2
seed@VM:~$ ls -l file2
-rw------- 1 seed seed 0 Oct 20 06:39 file2
seed@VM:~$
```

FULL ACCESS CONTROL LIST

- getfacl displays the file name, owner, the group, and the ACL (Access Control List).
- $ getfacl file2
- setfacl utility sets ACLs (Access Control Lists) of files and directories.
- $ setfacl -m user:alice:r file2
- $ getfacl file2

```
seed@VM:~$ ls -l file2
-rw------- 1 seed seed 0 Oct 20 06:39 file2
seed@VM:~$ getfacl file2
# file: file2
# owner: seed
# group: seed
user::rw-
group::---
other::---
seed@VM:~$ setfacl -m user:alice:r file2
seed@VM:~$ getfacl file2
# file: file2
# owner: seed
# group: seed
user::rw-
user:alice:r--
group::---
mask::r--
other::---
```

SUDO - RUN COMMAND AS ANOTHER USER

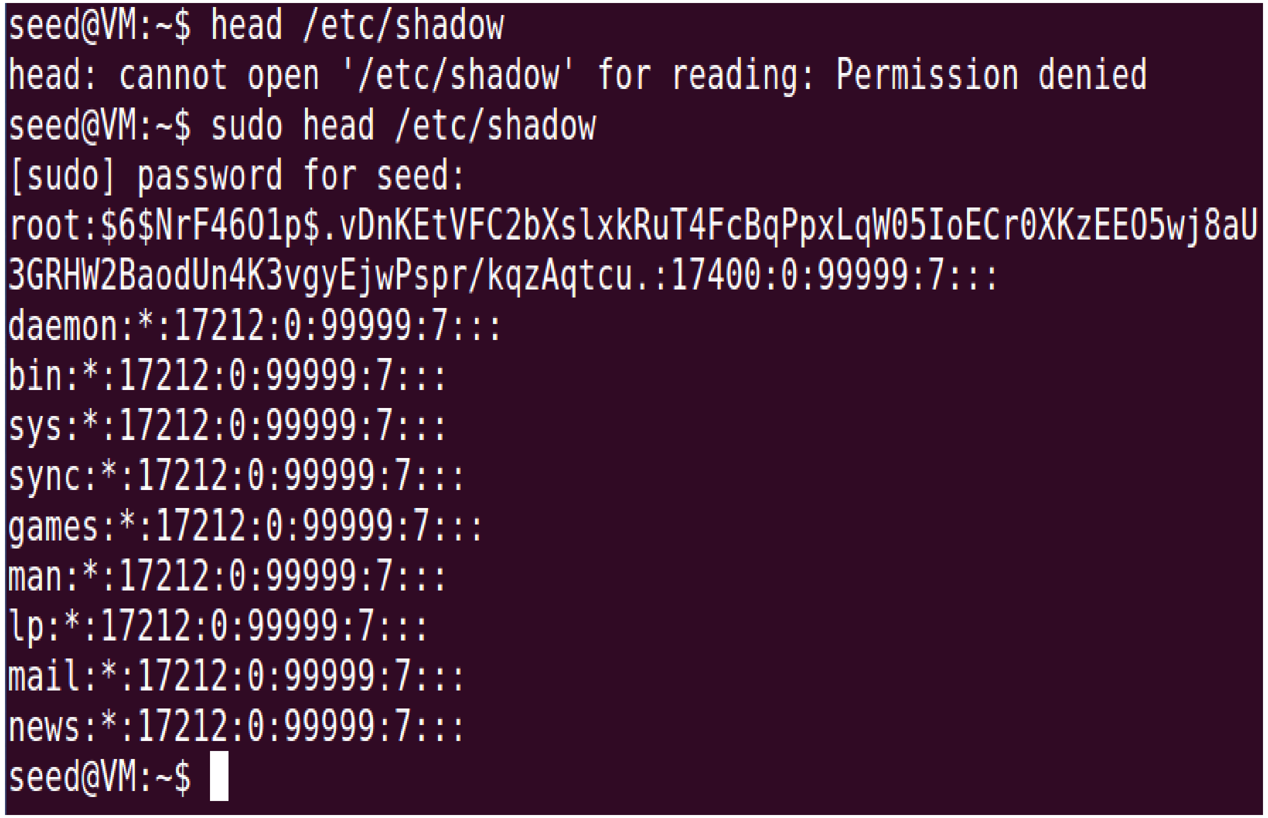
- $whoami
- $ sudo –u alice whoami

Sudo - Mostly used to run the command as superuser

```
seed@VM:~$ whoami
seed
seed@VM:~$ sudo -u alice whoami
Password for alice :
alice
seed@VM:~$
```

NEED FOR USER SUPERUSER PRIVILEGES

- **Sudo - Super user do**
- $ head /etc/shadow
- Permission denied
- $ **sudo** head /etc/shadow
- Password for the user:
- File contents
- Command is run with user id 0
- When the system looks at process user id, it will be 0 and hence allowed
- Is that not a security problem ?
- How the seed user is allowed to run a command as superuser?



```
seed@VM:~$ head /etc/shadow
head: cannot open '/etc/shadow' for reading: Permission denied
seed@VM:~$ sudo head /etc/shadow
[sudo] password for seed:
root:$6$NrF4601p$.vDnKEtVFC2bXslxkRuT4FcBqPpxLqW05IoECr0XKzEE05wj8aU
3GRHW2BaodUn4K3vgyEjwPspr/kqzAqtcu.:17400:0:99999:7:::
daemon:*:17212:0:99999:7:::
bin:*:17212:0:99999:7:::
sys:*:17212:0:99999:7:::
sync:*:17212:0:99999:7:::
games:*:17212:0:99999:7:::
man:*:17212:0:99999:7:::
lp:*:17212:0:99999:7:::
mail:*:17212:0:99999:7:::
news:*:17212:0:99999:7:::
seed@VM:~$
```

SUDO CONFIGURATION FILE

- $cat /etc/sudoer file
- %sudo ALL=(ALL:ALL) ALL
- Sudo group – is allowed to run any command as super user

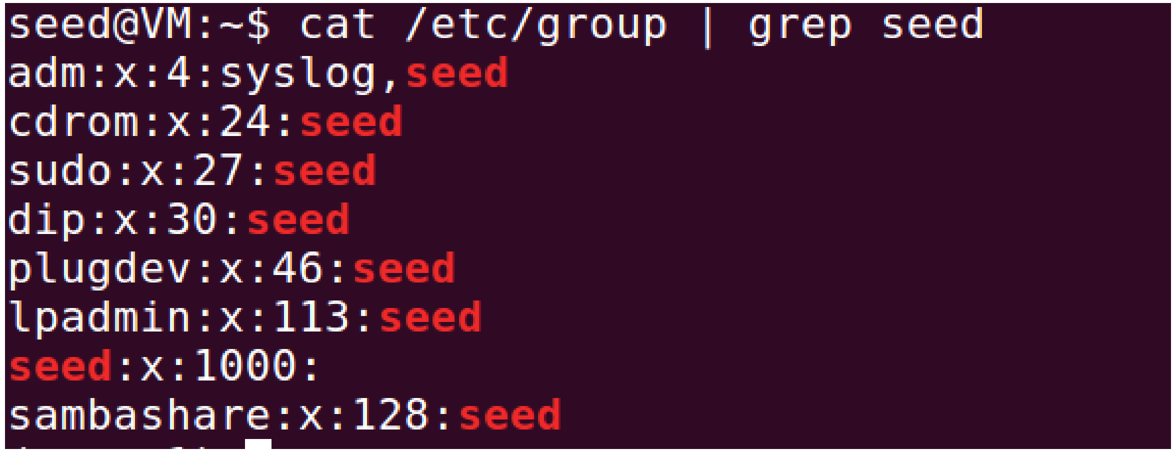
```
seed@VM:~$ cat /etc/sudoers
cat: /etc/sudoers: Permission denied
seed@VM:~$ sudo cat /etc/sudoers
...
# User privilege specification
root ALL=(ALL:ALL) ALL
# Members of the admin group may gain root
privileges
%admin ALL=(ALL) ALL
# Allow members of group sudo to execute any
command
%sudo ALL=(ALL:ALL) ALL
# See sudoers(5) for more information on "#include"
directives:
#includedir /etc/sudoers.d
seed@VM:~$
```

SUDO CONFIGURATION FILE

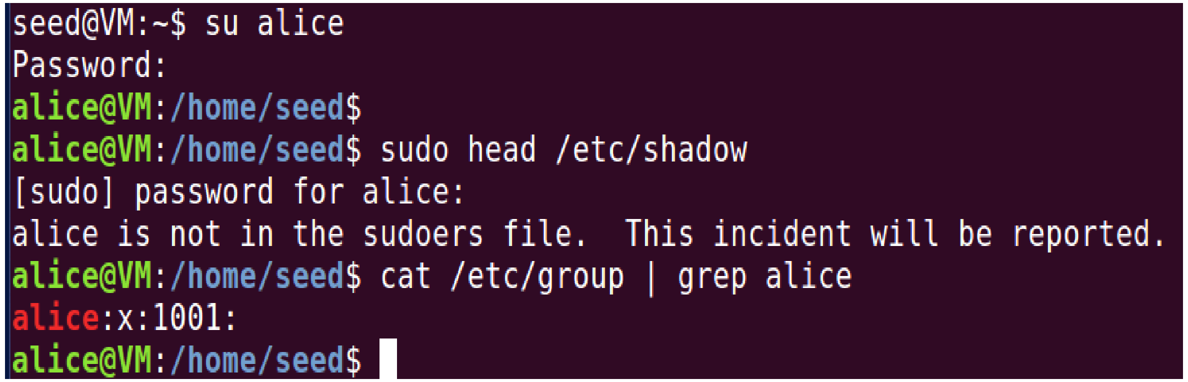
- $ cat /etc/group | grep seed

sudo : x: 27:seed

- If the above entry is not there, then seed user will not be able to run super user command
- Seed – normal user account
- By adding seed to sudo group, we can do sudo in seed account rather than switching to su account



```
seed@VM:~$ cat /etc/group | grep seed
adm:x:4:syslog,seed
cdrom:x:24:seed
sudo:x:27:seed
dip:x:30:seed
plugdev:x:46:seed
lpadmin:x:113:seed
seed:x:1000:
sambashare:x:128:seed
```



```
seed@VM:~$ su alice
Password:
alice@VM:/home/seed$
alice@VM:/home/seed$ sudo head /etc/shadow
[sudo] password for alice:
alice is not in the sudoers file.  This incident will be reported.
alice@VM:/home/seed$ cat /etc/group | grep alice
alice:x:1001:
alice@VM:/home/seed$
```

ASSIGNMENT: TRY OUT ALL THESE COMMANDS ON SEED VM AND DOCUMENT THEM.