Operating System Security #2 Securing User Env.

Herman Kabetta, M.T.

Understand /etc/passwd File

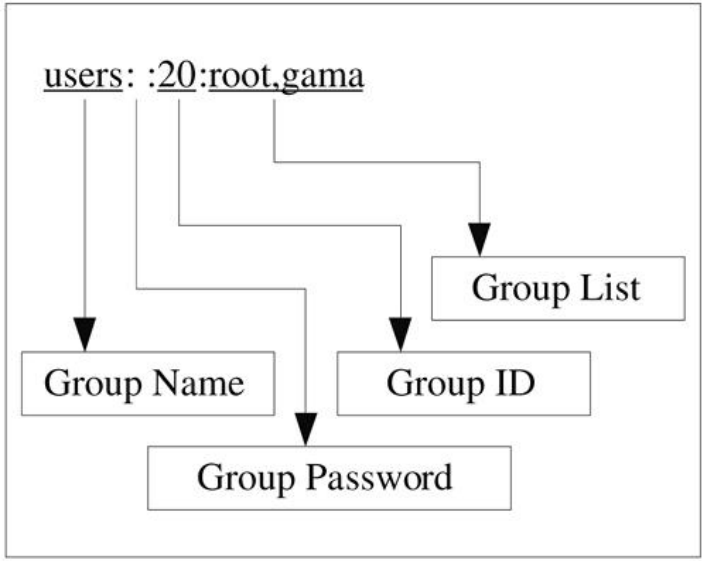
- What is **/etc/passwd** file?
  - /etc/passwd is a user configuration file
- What is Configuration File?
  - All configuration files resides in **/etc** directory which dictates how your system or application should work
- The meaning of **/etc** directory
  - Stands for et cetera where eveything else would go
  - All configuration files in **/etc** are editable and customizable

```
root:x:0:0:root:/root:/bin/bash
 1   2 3 4  5     6      7

 1.root: username
 2.x: password (saved in /etc/shadow in encrypted form)
 3.0: UID (0 is for root)
 4.0: GID (0 is for root)
 5.root: comments
 6./root: Home directory
 7./bin/bash: Login Shell
```

Understand /etc/group File

- What is **/etc/group** file?
  - File that has information about user group
- Some Basic Rules
  - User account cannot be created without a group
  - Each user belongs to a group or multiple user can belong to one single group
  - Each user can belong to multiple groups as well



Understand /etc/shadow File

- What is **/etc/shadow** file?
  - Stores actual password in encrypted format for user's account with additional properties related to user password
- Example

  john:6iTEFbMTM$CXmxPwErbEef9RUBvf1zv8EgXQdaZg2eOd5uXyvt4sFzi6G4lIqavLil TQgniAHm3Czw/LoaGzoFzaMm.YwOl/:17707:0:90:14:::

Understand /etc/shadow File

john:6iTEFbMTM$CXmxPwErbEef9RUBvf1zv8EgXQdaZg2eOd5uXyvt4sFzi6G4lIqavLilTQgniAHm3Czw/LoaGzoFzaMm.YwOl/:17707:0:90:14:::

1. **Username** : It is your login name.
2. **Password** : It is your encrypted password. The password should be minimum 8-12 characters long including special characters, digits, lower case alphabetic and more. Usually password format is set to idsalt$hashed, The $id is the algorithm used On GNU/Linux as follows:

   1 is MD5 5 is SHA-256
   $2a$ is Blowfish 6 is SHA-512
   $2y$ is Blowfish
3. **Last password change** (**lastchanged**) : Days since Jan 1, 1970 that password was last changed
4. **Minimum** : The minimum number of days required between password changes i.e. the number of days left before the user is allowed to change his/her password
5. **Maximum** : The maximum number of days the password is valid (after that user is forced to change his/her password)
6. Warn : The number of days before password is to expire that user is warned that his/her password must be changed
7. **Inactive** : The number of days after password expires that account is disabled
8. **Expire** : days since Jan 1, 1970 that account is disabled i.e. an absolute date specifying when the login may no longer be used.

Understand /etc/login.defs File

- The chage command - per user
  - Example

```
chage [-m mindays] [-M maxdays] [-d lastday] [-I inactive]
[-E expiredate] [-W warndays] user
```

- File **/etc/login.defs**
  - PASS_MAX_DAYS 99999
  - PASS_MIN_DAYS 0
  - PASS_MIN_LEN 5
  - PASS_WARN_AGE 7

Create User Account and Set Password

- Command we will learn
  - **useradd**
  - passwd
  - userdel
  - groupadd
  - groupdel
  - usermod

```
[root@rkss ~]# useradd centuser
[root@rkss ~]# id centuser
uid=1001(centuser) gid=1001(centuser) groups=1001(centuser)
[root@rkss ~]# grep centuser /etc/passwd
centuser:x:1001:1001::/home/centuser:/bin/bash
[root@rkss ~]# grep centuser /etc/group
centuser:x:1001:
[root@rkss ~]# grep centuser /etc/shadow
centuser:!!:18169:0:99999:7:::
```

Create User Account and Set Password

- Command we will learn
  - useradd
  - **passwd**
  - userdel
  - groupadd
  - groupdel
  - usermod

```
[root@rkss ~]# passwd centuser
Changing password for user centuser.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@rkss ~]# grep centuser /etc/shadow
centuser:$6$cVAdrtJj$u3Hyn7ajotT/.EV/K5afw3qzHzGdtPHOXICm5W.KOB36FdqyaqGWfMJYY/q
UHRzv8xbXk/9UFalblfXGMPqkgl:18169:0:99999:7:::
```

Create User Account and Set Password

- Command we will learn
  - useradd
  - passwd
  - **userdel**
  - groupadd
  - groupdel
  - usermod

```
USERDEL(8)                System Management Commands                USERDEL(8)

NAME
       userdel - delete a user account and related files

SYNOPSIS
       userdel [options] LOGIN

DESCRIPTION
       The userdel command modifies the system account files, deleting all
       entries that refer to the user name LOGIN. The named user must exist.

OPTIONS
       The options which apply to the userdel command are:

       -f, --force
           This option forces the removal of the user account, even if the
           user is still logged in. It also forces userdel to remove the
           user's home directory and mail spool, even if another user uses the
           same home directory or if the mail spool is not owned by the
           specified user. If USERGROUPS_ENAB is defined to yes in
           /etc/login.defs and if a group exists with the same name as the
           deleted user, then this group will be removed, even if it is still
           the primary group of another user.

           Note: This option is dangerous and may leave your system in an
           inconsistent state.
```

```
userdel -r centuser
```

Setting Password Policy

- **/etc/login.defs**
- **/etc/pam.d/system-auth**
- **/etc/security/pwquality.conf**

Lock or Disable User Accounts

- Disabling inactive accounts ensure that account which may not have been responsibly removed are not available to attacker who may have compromised their credentials
- A file **/etc/default/useradd** is used to set default configuration of a user whenever useradd command is ran
- To specify the number of days after password expires (which signifies inactivity) until an account is permanently disabled, add or correct the following lines in **/etc/default/useradd**, substituting “[NUM_DAYS]” appropriately
- A value of 35 is recommended. If a password is currently on the verge of expiration, then 35 days remain until the account is automatically disabled. However, if the password will not expire for another 60 days, then 95 days could elapse until the account would be automatically disabled.

Lock or Disable User Accounts

/etc/defaults/useradd

GROUP : Maximum number of groups for which a user can be a member of

HOME : Directory where the user's home directory will be created

INACTIVE : Number od=f days the account should be inactive after creation. Note “-1” means never inactive.

EXPIRE : Date on which the account should expire. It is given the form YYYY-MM-DD

SHELL : Default login shell for the user

SKEL : Directory from where the default user profile files will bcopied to the user's home directory

CREATE_MAIL_SPOOL : This option ensures that a new user will have a directory of its username in /var/main where the mail process can store mail messages

Lock or Disable User Accounts Manually

- Another comman to check users status

  `lastlog -b 90`

  which directly lists users who have not logged in in the past 90 days

  `lastlog -b 90 | tail -n+2`

  You could use grep filter out system users to be specific

  `lastlog -b 90 | tail -n+2 | grep -v  Never`

- To disable user account

  `usermod -L username`

  `passwd -l username`

  `Change user shell to /bin/nologin n /etc/passwd`

- To re-enable user account

  `usermod -U username`

  `passwd -u username`

Lock User Account After 3 Failed Attempts

- Being root first copy the following files

  `cp /etc/pam.d/password-auth /tmp`

  `cp /etc/pam.d/system-auth /tmp`

- Edit these files one by one

  `vi /etc/pam.d/system-auth`

  `vi /etc/pam.d/password-auth`

- Add the First Line

  `auth required pam_tally2.so deny=3 other=fail unlock_time=600`

- Next, scroll down to the account section and add the second line at the bottom of that section

  `account required pam_tally2.so`

- Try to Login as any user and then enter wrong password 3 times

- You can view how many failed attempts a user had, by issuing the command

  `pam_tally2 --user=USERNAME`

- Users will be automatically unlocked when using a lock time. If you want to enforce this, or unlock a permanently locked user, use the -r option together with the -u option

  `pam_tally2 -r -u username`

Restrict Direct “root” Login

- “root” a very powerful account in Linux and its access should be closely monitored
- Therefore one security measure is to restrict direct root login access
- IMPORTANT! Before you restrict root login, make sure you have a regular user account

- Become root
- **`cp /etc/ssh/sshd_config`**
- **`vi sshd_config file`**

- Change
  - **`PermitRootLogin  No`**

- Restart ssh service
  - **`systemctl restart sshd`**

Disable SSH Access for a Specific User

- Don't forget to make backup file
- **`vi sshd_config file`**

- Edit the following line in sshd_config file
- If the line does not exists then add in the last line
  - **`DenyUsers USERNAME`**

- Restart ssh service
  - **`systemctl restart sshd`**

Implement UID/GID Policy

- One of the security practice in Linux is to separate sers UID/GID from application accounts or systems account
- Usually the systems accounts UID/GID range from 0-99
- User accounts starts above 100+
- UID/GID are assigned by default when running useradd command
- The best practice is to create the first user with manual ID assignment so the OS will assign the UID by incrementing the last UID assigned
  - e.g. **`useradd -u 10000 USERNAME`**
- The security documentation should also include the policy of separating UID for different groups as well
  - e.g. QA, DevOps, System, etc.

sudo Access

- sudo stands for either “substitute user do” or “super user do” (depending upon how you want to look at it)
- There are two ways to run administrative application in Linux. You can either switch to the super user (root) with the su command, or you can take advantage of sudo.
- sudo allows an ordinary user to run a program as a root user
- visudo = edit the /etc/sudoer fil
- Parameter example :
  - **`root ALL=(ALL) ALL`**

so, basically it says that any user caxn run any command on any host as any user and yes, the user just has to authenticate, but with the password of the other user, in order to run anything.

- The first ALL is the users allowed
- The second one is the hosts
- The third one is the user as you are running the command
- The last one is the commands allowed

sudo Access

- Example of root commands:
  - shutdown, reboot, init
  - useradd, userdel
  - dmidecode, fdisk -l
- To allow a user to run specific command
  - **`akeo ALL=(ALL)  NOPASSWD: /sbin/fdisk`**
- To allow a user to run any root privileged command then add to wheel group
  - **`usermod -aG wheel akeo`**
- To remove a user from a group
  - **`gpasswd -d user group`**

Monitor User Activity

To know when a particular user logged in to the Linux System or from which IP Address your system was accessed.

Security point of view

- Logging = Log or record the activity
- Auditing = Validate the activity
- Default logging by the OS
  - **/var/log/messages** = All system related messages
  - **/var/log/secure** = user login activity including failures
  - **last** = view history of all logged user
  - **last <username>** = view a login history of a certain user
  - **lastb** = view all bad login attempts
  - **who or w** = who is currently logged in (tty and pts)
    - tty = console
    - pts = xterm, putty, other terminals
  - **/home/user/.bash_history** = user command history
  - **tcpdump** = server incoming and outgoing traffic
  - **/etc/rsyslog.conf** = logging configuration
  - **ps -ef | grep <username>** = monitor user running process