RHSA1

Red Hat System Administration I

Information Technology Institute

COURSE MATERIALS

You can access the course materials via this link

https://goo.gl/ezCT7j

DAY 2 CONTENTS

- User and group administration
- Permissions
- Switching to other accounts
- Shutting down the system

USERS AND GROUPS

- The `/etc/passwd` file

```
loginame:x:uid:gid:comment:home-directory:login-shell
```

- Included fields are:
  - Login name
  - User Id (uid)
  - Group Id (gid)
  - Comment about the user
  - Home Directory
  - Login shell

USERS AND GROUPS

- The `/etc/shadow` file

```
username:encrypted passwd:last
changed:min:max:warn:inactive:expire:future-use
```

- Included fields are:
  - Login name
  - Encrypted password
  - Days since Jan 1, 1970 that password was last changed
  - Days before password may not be changed
  - Days after which password must be changed
  - Days before password is to expire that user is warned
  - Days after password expires that account is disabled
  - Days since Jan 1, 1970 that account is disabled

USERS AND GROUPS

- The `/etc/group` file

```
groupname:x:gid:comma-separated list of group members
```

- The `/etc/gshadow` file ???

ADDING NEW USER

```
# useradd username
```

- The `useradd` command populates user home directories from the `/etc/skel` directory.

- To view and modify default setting

```
useradd -D
```

```
# passwd username
```

- Adding multiple user accounts

```
# newusers filename
```


MODIFYING USER ACCOUNTS

- To change a user's account information, you can:
  - Edit the `/etc/passwd` or `/etc/shadow` files manually
  - Use the `chage` or `usermod` commands which are discussed later

MODIFYING USER ACCOUNTS

- To change a user's account information, you can:
  - Use the `usermod` command:
  - `usermod [options] username`
    - Useful options
      - To changes the login name use -l <login name>
      - To lock the password use -L
      - To unlock the password use -U

DELETING A USER ACCOUNT

- To delete a user account you can
  - Manually remove the user from
    - `/etc/passwd` file
    - `/etc/shadow` file
    - `/etc/group` file
    - remove the user's home directory (`/home/username`)
    - and mail spool file (`/var/spool/mail/username`)
  - Use the `userdel` command.

```
# userdel [-r] username
```


PASSWORD AGING POLICIES

- The `chage` command sets up password aging

```
# chage [options] username
```

- Options
  - `-m`: to change the min number of days between password changes
  - `-M`: to change the max number of days between password changes
  - `-E` date: change the expiration date for the account
  - `-W`: change the number of days to start warning before a password change will be required

PRIVATE GROUP SCHEME

- A traditional problem found in many UNIX/Linux environments is when administrators place all users in the same primary group. When users on such systems use a umask value of 002.

- Ubuntu solves this problem by assigning user a primary group for which they are the sole members.

- This "private" primary group has the same name as the user's username

MANAGING GROUPS

- Creating New Group

```
# groupadd groupname
```

- Modifying an Existing Group

```
# groupmod [options] groupname
```

- Deleting a Certain Group

```
# groupdel groupname
```

- List all file which are owned by groups not defined in `/etc/group` file

```
# find / -nogroup
```

MANAGING GROUPS

- You can use the `gpasswd` command to define
  - Group members
  - Group administrators
  - And to create or change group passwords

- Use the -r option to the `groupadd` command avoids using a GID within the range typically assigned to users and their private groups.

CHANGING ACTIVE GROUP

- To switch between groups you are member in, use `newgrp` command.

```
newgrp group
```

- To display the groups you are member in use `groups` command

```
groups
```

```
other root bin sys adm uucp mail tty lp
```

SWITCHING ACCOUNTS

```
# su [-] [username]

# su [-] [username] -c command
```

THE `whoami` COMMAND

- After switching into several users, it is a sever issue to know your current (effective) user

```
whoami

root
```

THE `id` COMMAND

- Displays
  - Effective user id
  - Effective user name
  - Effective group id
  - Effective group name
- Examples

```
id
```

uid=101(user1) gid=100(user1)groups=101(user1)

```
id user2
```

uid=500(user2)gid=500(user2) groups=500 (user2)

THE `who` COMMAND

- `who` is on the system
- Displays
  - User Login name
  - Login device (tty)
  - Login date and time
- Example

```
who
```

```
Islam pts/2 2010-09-28  8:35 (:0)
```

THE w COMMAND

- The w command display a summary of the current activity on the system, including what each user is doing.

```
w [user]
```

- Example

```
w
```

1:23pm up 4 days(s), 20:36, 1 user, load average: 0.04, 0.04, 0.03

USER	TTY	FROM	LOGIN@	IDLE	JCPU	PCPU	what
Islam	pts/2	:0	11:26am		0.00s	0.05s	w

THE `finger` COMMAND

- finger is a useful command that reveals details of users.

```
finger
```

```
Login    Name                TTY     Idle Login Time    Office      Office
Phone
iaskar Islam Askar  pts/1        Nov 27 12:34  (:0.0)
```

```
finger @163.121.12.30
```

```
Login      Name          TTY      Idle     When          Where
iaskar Islam Askar  pts/2          Mon 12:34   iti-198.iti.gov.eg
```

THE `finger` COMMAND

```
finger iaskar
```

Login name: iaskar Name: Islam Askar

Directory:/home/iaskar Shell:/usr/bin/bash

On since Jun 6 12:34:09 on pts/2 from :0

20 minutes Idle Time

No mail

No plan

USING `sudo` COMMAND

- sudo is more secure
- sudo access is controlled by the `/etc/sudoers`.
  - This file is edited by visudo, an editor and syntax checker.
  - To give a specific group of users limited root privileges
    - User_Alias LIMITEDTRUST=st1,st2
    - Cmnd_Alias MINUMUM=/etc/init.d/httpd
    - Cmnd_Alias SHELLS=/bin/sh,/bin/bash
    - LIMITEDTRUST ALL=MININUM
    - user5 ALL=ALL,!SHELLS
    - %development station1=ALL, !SHELL

OWNERSHIP AND PERMISSIONS

- Every file and directory has both **user** and **group** ownership. A newly-created file will be owned by:
  - The user who creates it
  - That user's primary group (unless the file is created in a set group ID (SGID) directory; more on this file in the next lesson)

OWNERSHIP AND PERMISSIONS

- File ownership can be changed using `chown` command.
- Example

```
# chown user1 file1

# chown user1:group1 file1

# chown :group1 file1
```

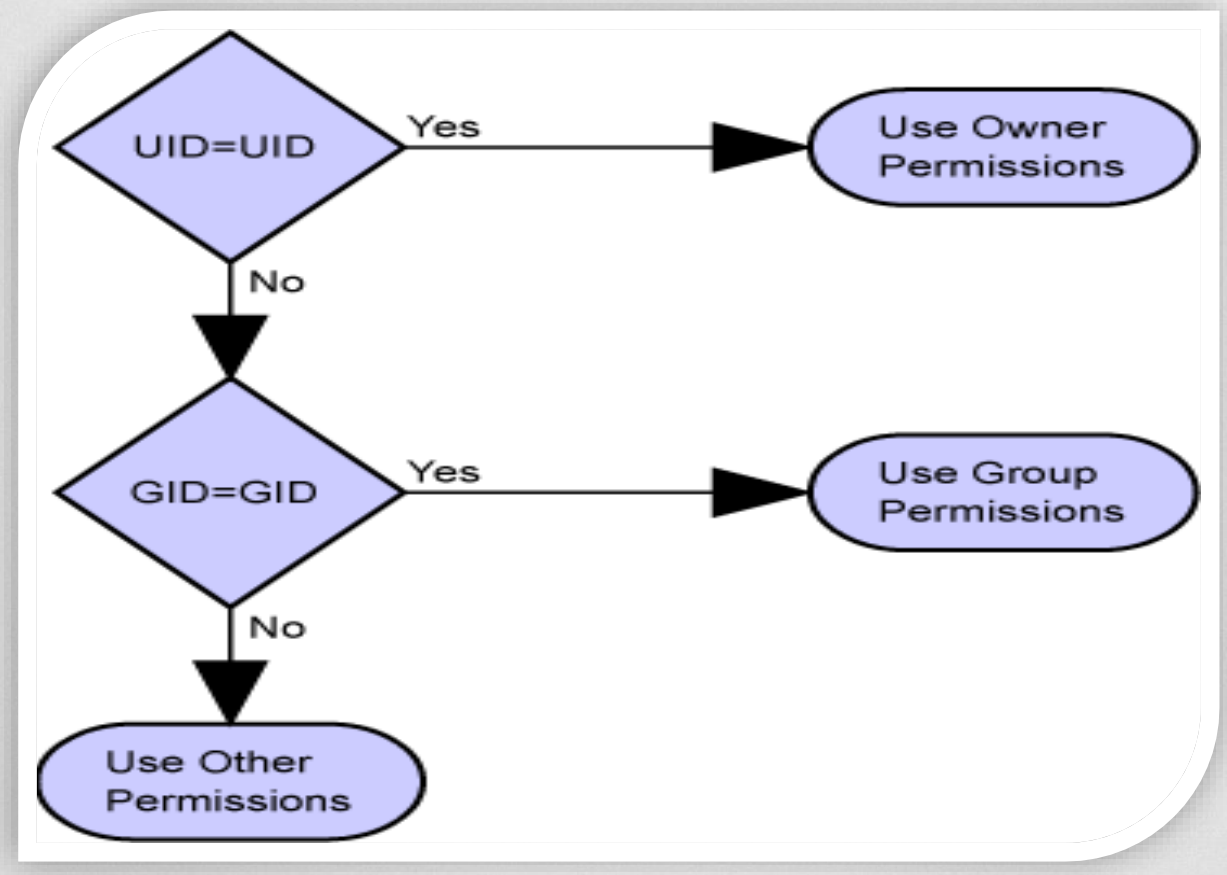
SECURITY SCHEME

- Each file has an owner and assigned to a group.
- Linux allows users to set permissions on files and directories to protect them.
- Permissions are assigned to
  - File owner
  - Members of the group the file assigned to
  - All other users
- The most specific permissions apply
- Permissions can only be changed by the owner and root

PERMISSION NOTATIONS

Permission	Access for a File	Access for a Directory
Read	You can display file contents and copy the file.	You can list the directory contents with the ls command
Write	You can modify the file contents.	If you also have execute access, you can add and delete files in the directory.
Execute	You can execute the file if it is an executable. You can execute a shell script if you also have read and execute permissions.	You can use the cd command to access the directory. If you also have read access, you can run the ls -l command on the directory to list contents.

DETERMINING PERMISSIONS



CHANGING PERMISSIONS

`chmod permission filename`

- Permissions are specified in either
  - Symbolic mode
    - Who
      - u: Owner permissions
      - g: Group permissions
      - o: Other permissions
      - a: all permissions
    - Operator
      - + Add permissions
      - - Remove permissions
      - = Assign permissions absolutely
    - Permissions
      - r: read
      - w: write
      - x: execute

CHANGING PERMISSIONS

- Permissions are specified in either
  - Octal mode
    - 4 read
    - 2 write
    - 1 execute

EXAMPLES

ls -l file1

```
-rw-r--r-- 1 user1 staff 1319 Mar 22 14:51 file1
```

chmod o-r file1

ls -l file1

```
-rw-r----- 1 user1 staff 1319 Mar 22 14:51 file1
```

chmod g-r file1

ls -l file1

```
-rw------- 1 user1 staff 1319 Mar 22 14:51 file1
```

EXAMPLES

chmod u+x,go+r file1

ls -l file1

```
   -rwxr--r-- 1 user1 staff 1319 Mar 22 14:51
file1
```

chmod a=rw file1

ls -l file1

```
   -rw-rw-rw- 1 user1 staff 1319 Mar 22 14:51
file1
```

chmod 555 file1

ls -l file1

```
   -r-xr-xr-x 1 user1 staff 1319 Mar 22 14:51
file1
```

EXAMPLES

chmod 775 file1

ls -l file1

```
-rwxrwxr-x 1 user1 staff 1319 Mar 22 14:51 file1
```

chmod 755 file1

ls -l file1

```
-rwxr-xr-x 1 user1 staff 1319 Mar 22 14:51 file1
```

DEFAULT PERMISSIONS

- The `umask` command sets the default permissions for files and directories

- Example
  - `# umask 002`
  - `# umask`
  - 022

VIRTUAL CONSOLES

- Accessed with Ctrl-Alt-F_key
- Consoles 1-6 accept logins
- X server starts on the console 7

SYSTEM SHUTDOWN

- It only requires reboot or shutdown when you need to
  - Add or remove hardware
  - Upgrade to a new version of Ubuntu
  - Or upgrade your kernel
  - `shutdown -k now`
    - # doesn't really shutdown only send the warning messages and disable logins.
  - `shutdown -h  time` # Halt after shutdown
  - `poweroff`
  - `init 0`