Linux

Users and Groups

CONFIDENTIAL | © 2021 EPAM Systems, Inc.

Users and groups in Linux

- Every process (running program) on the system runs as a particular user.
- Every file is owned by a particular user and group.
- Access to files and directories are restricted by user and group.

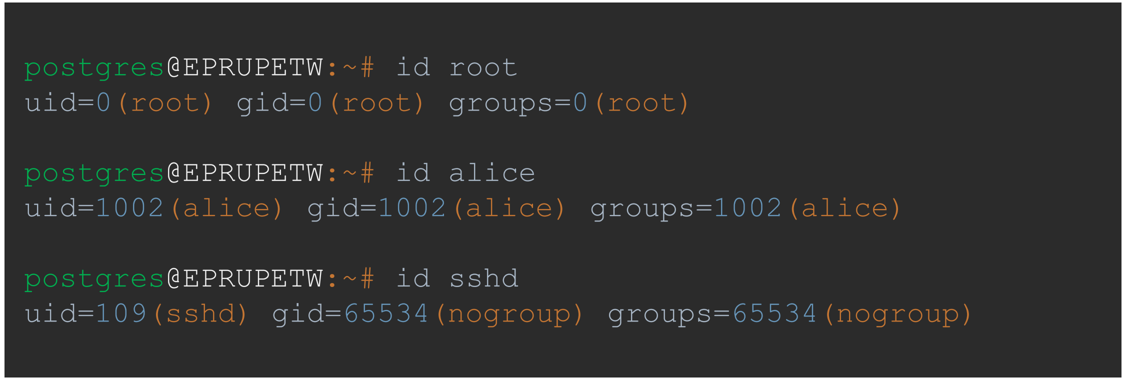
```
postgres@EPRUPETW:~$ id
uid=113(postgres) gid=119(postgres) groups=119(postgres),115(ssl-cert)
```

Uid (user id): represents a user
Gid (group id): represents a group

Uid and Gid numbers are unique in the system but not unique across systems without the proper technologies (NIS+, LDAP)

Types of users

Root
Non-root (alice, bob, ntp, hdfs, sshd, postgres)



```
postgres@EPRUPETW:~# id root
uid=0(root) gid=0(root) groups=0(root)

postgres@EPRUPETW:~# id alice
uid=1002(alice) gid=1002(alice) groups=1002(alice)

postgres@EPRUPETW:~# id sshd
uid=109(sshd) gid=65534(nogroup) groups=65534(nogroup)
```

Types of users

Use command **id** to print user and group information

```
postgres@EPRUPETW:~# id --help
Usage: id [OPTION]... [USER]
Print user and group information for the specified USER, or (when USER omitted) for the current user.
```

last – shows recent logins

```
postgres@EPRUPETW:~# last

wtmp begins Thu Aug  5 06:46:54 2021
postgres@EPRUPETW:~#
```

Superuser Access

The root user is an account with administrative permissions:

1. No access restrictions
2. Usually have inactive password (can't login using **su**)

Switching users with su

The su command allows a user to switch to a different user account. If a username is not specified, the root account is implied.

su -> root password -> bash from root

```
postgres@EPRUPETW:~$ su root
Password:
root@EPRUPETW:/var/lib/postgresql#
```

Running commands as root with sudo (Super user do)

```
postgres@EPRUPETW:~# sudo id
uid=0(root) gid=0(root) groups=0(root)

postgres@EPRUPETW:~# sudo -l
Matching Defaults entries for postgres on EPRUPETW:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User postgres may run the following commands on EPRUPETW:
    (ALL : ALL) ALL
    (ALL : ALL) NOPASSWD: ALL

postgres@EPRUPETW:~# sudo –u alice  id  # only if the target user has access to the shell
uid=1002(alice) gid=1002(alice) groups=1002(alice)

postgres@EPRUPETW:~# sudo –i  # Login as root
root@EPRUPETW:~#

root@EPRUPETW:~# visudo  # edit file /etc/sudoers with syntax check
```

Superuser vs wheel

Root

- Separate user with the password
- Full privileges without restrictions
- One password for distribution
- Always there for you
- Need to be very careful in use

Wheel/Sudo

- Can be applied to any user or group (or user collection)
- Allows more narrow control over the command scope
- Can be absent on the machine

Managing Local User Accounts (useradd and usermod)

useradd Create a user with set of defined parameters

```
root@EPRUPETW:~# useradd user
root@EPRUPETW:~# useradd -m -d /var/www/userhome -s /bin/zsh -c "generic description" -u 1000 -g
1000 username
```

usermod Modify a user in quite a number of ways (name change, moving home dir, group attachment and etc)

```
root@EPRUPETW:~# usermod -u 1111 -g 2222 username
```

Managing Local User Accounts (passwd and userdel)

passwd Sets passwords

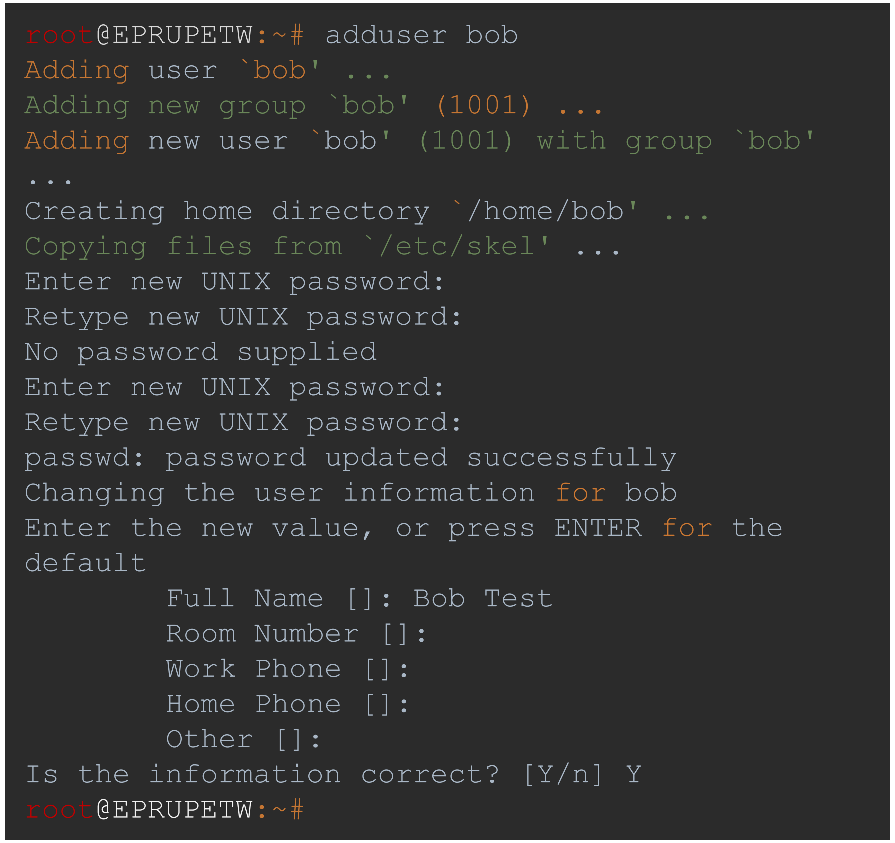
```
root@EPRUPETW:~# passwd user
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

userdel Delete a user

```
root@EPRUPETW:~# userdel user
```

Managing Local User Accounts (adduser)

adduser Interactive interface for **useradd**



```
root@EPRUPETW:~# adduser bob
Adding user `bob' ...
Adding new group `bob' (1001) ...
Adding new user `bob' (1001) with group `bob'
...
Creating home directory `/home/bob' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
No password supplied
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for bob
Enter the new value, or press ENTER for the
default
        Full Name []: Bob Test
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
root@EPRUPETW:~#
```

Managing Local Groups (groupadd and groupmod)

groupadd Create a user or a group with set of defined parameters

```
root@EPRUPETW:~# groupadd testgr
root@EPRUPETW:~# groupadd -g 1234 -o testgr1
root@EPRUPETW:~# groupadd -g 1234 -o testgr2
root@EPRUPETW:~# groupadd -g 1234 testgr2
groupadd: GID '1234' already exists
```

groupmod Modify a group (basically renaming or changing ID)

```
root@EPRUPETW:~# groupmod -g 1235 testgr
```

Managing Local Groups (addgroup, groupmems, groupdel)

addgroup Interactive interface for groupadd

```
root@EPRUPETW:~# addgroup testgr33
addgroup testgrp33
Adding group `testgrp33' (GID 1004) ...
Done.
```

groupmems Controls the users in the group (add, delete, list)

```
root@EPRUPETW:~# groupmems -g testgr -a bob
Password:
```

groupdel Delete a group

```
root@EPRUPETW:~# groupdel testgr
```

Databases

Local database:

User store: Passwd file (**/etc/passwd**)

Group store: Group file (**/etc/group**)

Shared database:

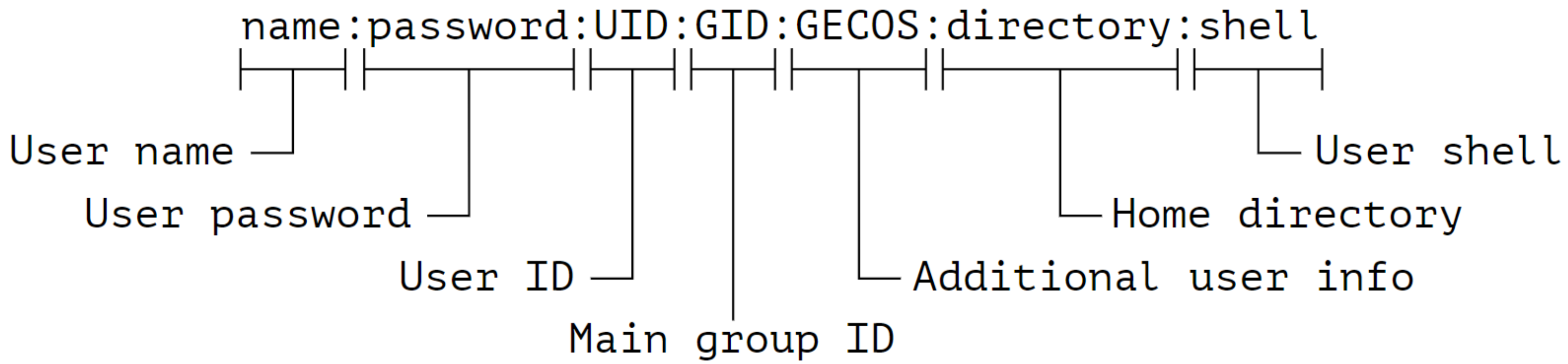
NIS, NIS+ (shared passwd and group files)

Remote database:

LDAP, Kerberos (also thru SSSD)

Password store: **/etc/shadow**, need this since /etc/passwd should world readable. Shadow file only readable for root "database" formats User and Group "database" formats

Passwd structure



Important note: User shell and home directory are basically set for login purposes.

Contents of /etc/shadow

name:password:last_change_date:min_age:max_age:warning:inactive:expire:reserved

Username: User name from /etc/passwd

Password: Encrypted password

last_change_date: When password was changed

min_age: Minimum pause before change

max_age: How long to accept old password

Warning: Warning period before password is expired

Inactive: When to change after password expires

Expire: When the account (not password!) is expired

Reserved: Unused for now

/etc/shadow- is a backup file for /etc/shadow.

Contents of /etc/sudoers

```
root@EPRUPETW:~#  cat /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#

Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/b
in"

watson ALL=(ALL:ALL) NOPASSWD: /bin/cat  /root/secret
%wheel ALL=(ALL:ALL) NOPASSWD: /bin/cat  /root/secret
```

watson %wheel | ALL= | (user1, user2 : group1, group2) | NOPASSWD: | /bin/cat | /root/secret

User or group ID | Host | By which users : by which groups | Tag | Command | Args

Useful links

- https://man7.org/linux/man-pages/man5/passwd.5.html Passwd description
- https://man7.org/linux/man-pages/man5/shadow.5.html Shadow description
- https://man7.org/linux/man-pages/man5/login.defs.5.html /etc/login.defs description
- https://www.redhat.com/sysadmin/linux-gecos-demystified Passwd GECOS description
- https://www.digitalocean.com/community/tutorials/how-to-edit-the-sudoers-file Sudoers for a beginners

CONFIDENTIAL | © 2021 EPAM Systems, Inc.

Task

1. Try to login as root with **su**
2. Set password to root with **sudo**
3. Try to login as root with **sudo**
4. Add a file which will be placed into a user's home directory automatically after user's creation
5. **Create** a new user with home directory
6. **Add** sudo to the new user

CONFIDENTIAL | © 2021 EPAM Systems, Inc.

THANK YOU