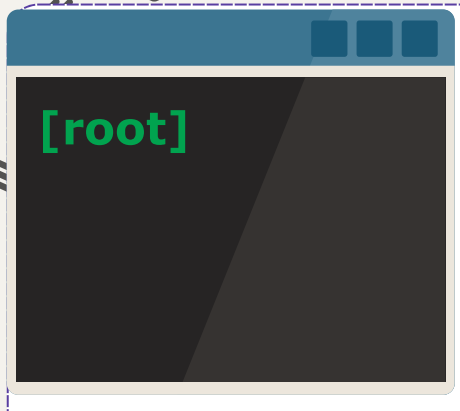
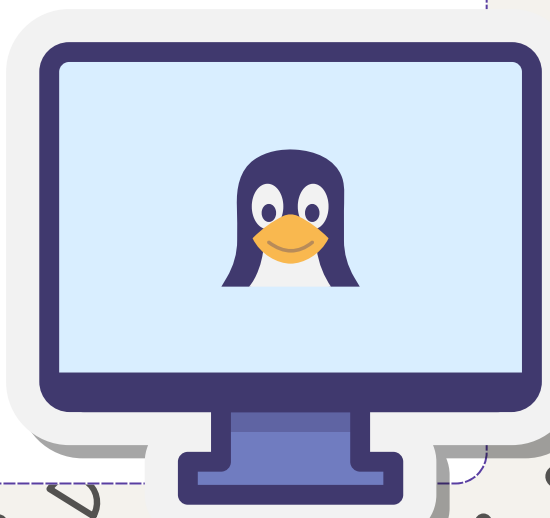


USER MANAGEMENT IN LINUX

BY - Gauri Yadav



LIST OF TOPICS

LIST OF TOPICS

LIST OF TOPICS

How to list logged in user?

- A list of the users who have successfully signed on to the system with valid credentials can be printed using one of the two basic Linux tools: who and w. These commands show various pieces of information separated in multiple columns.
- The who command references the /run/utmp file and displays the information. Here is a sample from server1:

Command 1: *who*

```
[root@localhost ~]# who
root     seat0         2024-04-02 14:31 (login screen)
root     tty2          2024-04-02 14:31 (tty2)
```

Column 1 displays the login name of the user. Column 2 shows the terminal session device filename (pts stands for pseudo terminal session, and tty identifies a terminal window on the console).

Command 2: *whoami*

```
[root@localhost ~]# whoami
root
[root@localhost ~]#
```

The who command can only print information about the user who executes it with the arguments “am i”

Command 3: *w*

```
root@localhost ~]# w
 14:35:17 up 5 min,  2 users,  load average: 0.09, 0.56, 0.33
SER     TTY        LOGIN@   IDLE   JCPU   PCPU WHAT
oot     seat0     14:31    0.00s  0.00s  0.02s /usr/libexec/gdm-
oot     tty2      14:31    6:15   0.15s  0.13s /usr/libexec/gnom
root@localhost ~]#
```

The w (what) command displays information in a similar format as the who command, but it also tells the length of time the user has been idle for (IDLE), along with the CPU time used by all processes including any existing background jobs attached to this terminal (JCPU), the CPU time used by the current process (PCPU), and current activity (WHAT).

Inspect history of successful login attempts

- The last command reports the history of successful user login attempts and system reboots by consulting the wtmp file located in the /var/log directory. This file keeps a record of all login and logout activities, including the login time, duration a user stayed logged in, and tty (where the user session took place). Consider the following two examples.

Command 1: *last*

```
[root@localhost ~]# last
root     tty2         tty2             Tue Apr  2 14:31   still logged in
root     seat0        login screen     Tue Apr  2 14:31   still logged in
reboot   system boot  5.14.0-284.30.1. Tue Apr  2 14:28   still running
root     tty2         tty2             Mon Apr  1 14:48 - crash  (23:40)
```

Column 1: Login name of the user
Column 2: Terminal name assigned upon logging in
Column 3: Terminal name or IP address from where the connection was established
Column 4 to 7: Day, month, date, and time when the connection was established
Column 8: Log out time. If the user is still logged on, it will say "still logged in"
Column 9: Duration of the login session

Command 2: *last reboot*

```
[root@localhost ~]# last reboot
reboot   system boot  5.14.0-284.30.1. Tue Apr  2 14:28   still running
reboot   system boot  5.14.0-284.30.1. Mon Apr  1 14:47   still running
reboot   system boot  5.14.0-284.30.1. Mon Apr  1 10:54   still running
```

Column 1: Action name (reboot)
Column 2: Activity name (system boot)
Column 3: Linux kernel version
Column 4 to 7: Day, month, date, and time when the reboot command was issued
Column 8: System restart time
Column 9: Duration the system remained down. If the system is running, it will say "still running".

View history of failed user login attempts

- The lastb command reports the history of unsuccessful user login attempts by reading the btmp file located in the /var/log directory. This file keeps a record of all unsuccessful login attempts, including the login name, time, and tty (where the attempt was made). Consider the following example. To list all unsuccessful login attempts, type the lastb command without any arguments. You must be root in order to run this command.

Command 1: *lastb*

```
[root@localhost ~]# lastb

btmp begins Tue Apr  2 14:29:04 2024
```

Column 1: Name of the user who made the login attempt
Column 2: Name of the protocol used. No tty was assigned as the attempt failed
Column 3: Terminal name or IP address from where the connection attempt was launched
Column 4 to 7: Day, month, date, and time of the attempt
Column 8: Duration the login attempt was tried
Column 9: Duration the login attempt lasted for

Examine user and group info

Command 1: *id*

```
[root@localhost ~]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconf
ned_t:s0-s0:c0.c1023
```

The id (identifier) command displays the calling user's UID (User IDentifier), username, GID (Group IDentifier), group name, all secondary groups the user is a member of, and SELinux security context. Here is a sample output for the root user when this command is executed without an option or argument.

Command 2: *id user1*

```
[root@localhost ~]# id user1
uid=1006(user1) gid=1007(user1) groups=1007(user1)
```

The id (identifier) command displays the calling user's UID (User IDentifier), username, GID (Group IDentifier), group name, all secondary groups the user is a member of, and SELinux security context. Here is a sample output for the root user when this command is executed without an option or argument.

Command 3: *whoami*

```
[root@localhost ~]# groups
root
```

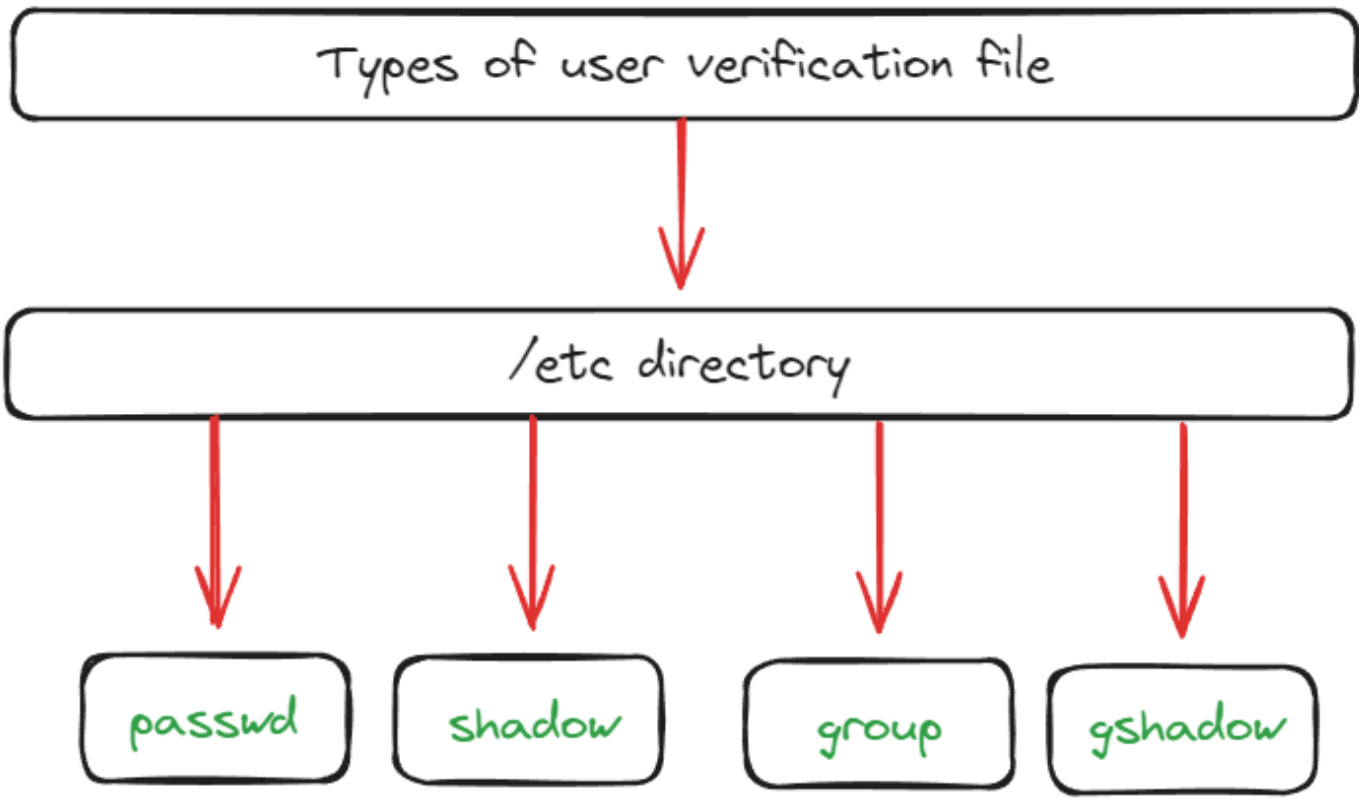
The first group listed is the primary group for the user who executed this command; all other groups are secondary (or supplementary). The groups command can also be used to view group membership information for a different user

What is local user Authentication

- RHEL supports three fundamental user account types: root, normal, and service. The root user (a.k.a. the superuser or the administrator), has full access to all services and administrative functions on the system. This user is created by default during installation. Normal users have user-level privileges; they cannot perform any administrative functions but can run applications and programs that have been authorized. Service accounts take care of their respective services, which include apache, ftp, mail, and chrony.

Types of user verification file?

- User account information for local users is stored in four files that are located in the /etc directory. These files—passwd, shadow, group, and gshadow—are updated when a user or group account is created, modified, or deleted. The same files are referenced to check and validate the credentials for a user at the time of their login attempt, and hence the files are referred to as user authentication files. These files are so critical to the operation of the system that the system creates their automatic backups by default as passwd-, shadow-, group-, and gshadow- in the /etc directory.



Command 1: *ls -l /etc/passwd* /etc/shadow* /etc/group* /etc/gshadow**

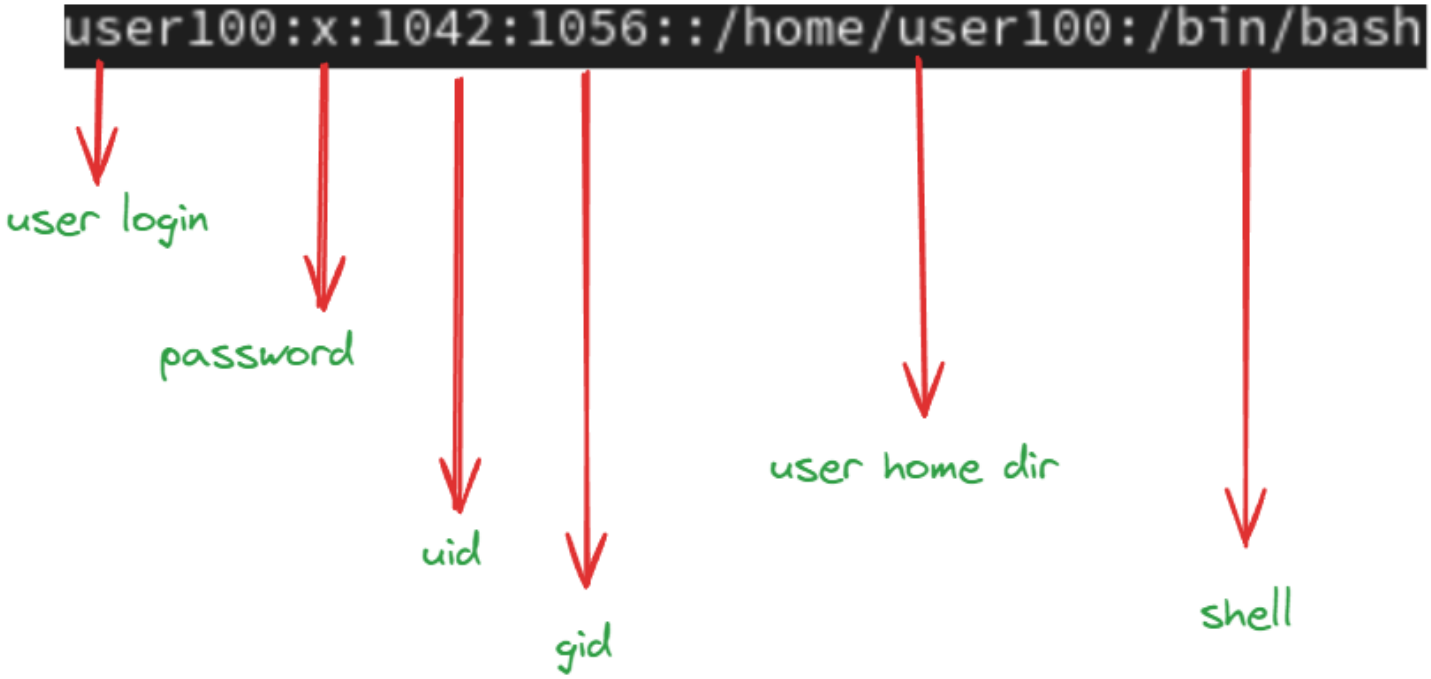
```
[root@localhost ~]# ls -l /etc/passwd* /etc/shadow* /etc/group* /etc/gshadow*
-rw-r--r--. 1 root root 1751 Apr  1 12:40 /etc/group
-rw-r--r--. 1 root root 1735 Apr  1 12:18 /etc/group-
----------. 1 root root 1376 Apr  1 12:40 /etc/gshadow
----------. 1 root root 1364 Apr  1 12:18 /etc/gshadow-
-rw-r--r--. 1 root root 4045 Apr  1 12:40 /etc/passwd
-rw-r--r--. 1 root root 4000 Apr  1 12:18 /etc/passwd-
----------. 1 root root 2490 Apr  1 12:40 /etc/shadow
----------. 1 root root 2460 Apr  1 12:18 /etc/shadow-
```

- ***These files—passwd, shadow, group, and gshadow—are updated when a user or group account is created, modified, or deleted. The same files are referenced to check and validate the credentials for a user at the time of their login attempt, and hence the files are referred to as user authentication files. These files are so critical to the operation of the system that the system creates their automatic backups by default as passwd-, shadow-, group-, and gshadow- in the /etc directory.***
- ***All files are short in size, but they grow bigger as new users are added. Two of the files—gshadow and shadow—along with their backups have no access permissions for any user, not even for root. Let's analyze these files and see what information they store and how.***

Examine passwd file?

- The passwd file is a simple plaintext file but it contains vital user login data. Each row in the file holds information for one user account.

Command 1: *cat /etc/passwd*

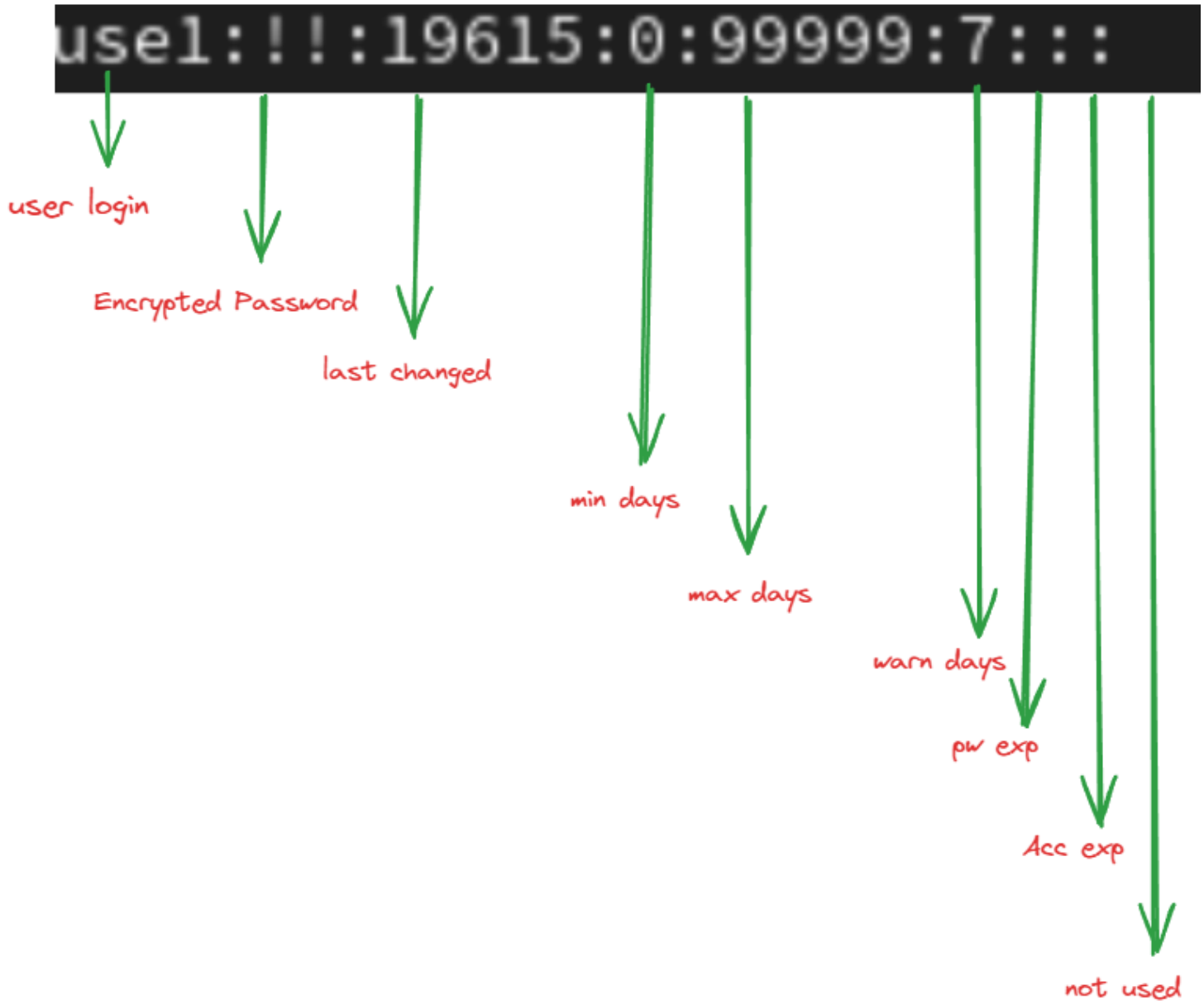


- *Field 1 (Login Name): Contains the login name for signing in. Login names up to 255 characters, including the underscore (_) and hyphen (-) characters, are supported. It is not recommended to include special characters and uppercase letters in login names.*
- *Field 2 (Password): Can contain an "x" (points to the /etc/shadow file for the actual password), an asterisk * to identify a disabled account, or a hashed password.*
- *Field 3 (UID): Comprises a numeric UID between 0 and approximately 4.2 billion. UID 0 is reserved for the root account, UIDs between 1 and 200 are used by Red Hat to statically assign them to core service accounts, UIDs between 201 and 999 are reserved for non-core service accounts, and UIDs 1000 and beyond are employed for normal user accounts. By default, RHEL begins assigning UIDs to new users at 1000.*
- *Field 4 (GID): Holds a GID that corresponds with a group entry in the /etc/group file. By default, RHEL creates a group for every new user matching their login name and the same GID as their UID. The GID defined in this field represents the user's primary group.*
- *Field 5 (Comments): Called GECOS (General Electric Comprehensive Operating System and later changed to GCOS), optionally stores general comments about the user that may include the user's name, phone number, location, or other useful information to help identify the person for whom the account is set up.*
- *Field 6 (Home Directory): Defines the absolute path to the user home directory. A home directory is the location where a user is placed after signing in and it is used for personal storage. The default location for user home directories is /home.*
- *Field 7 (Shell): Consists of the absolute path of the shell file that the user will be using as their primary shell after logging in. The default shell used in RHEL is the Bash shell (/bin/bash).*

Examine shadow file?

- RHEL has a secure password control mechanism in place that provides an advanced level of password security for local users. This control is referred to as the shadow password. With this control mechanism in place, user passwords are hashed and stored in a more secure file /etc/shadow, but there are certain limits on user passwords in terms of expiration, warning period, etc., that can also be applied on a per-user basis. These limits and other settings are defined in the /etc/login.defsfile, which the shadow password mechanism enforces on user accounts. This is called password aging. Unlike the passwd file, which is world-readable and owner-writable, the shadow file has no access permissions at all. This is done to safeguard the file's content
- With the shadow password mechanism active, a user is initially checked in the passwd file for existence and then in the shadow file for authenticity
- The shadow file contains user authentication and password aging information. Each row in the file corresponds to one entry in the passwd file.

Command 1: *cat /etc/shadow*

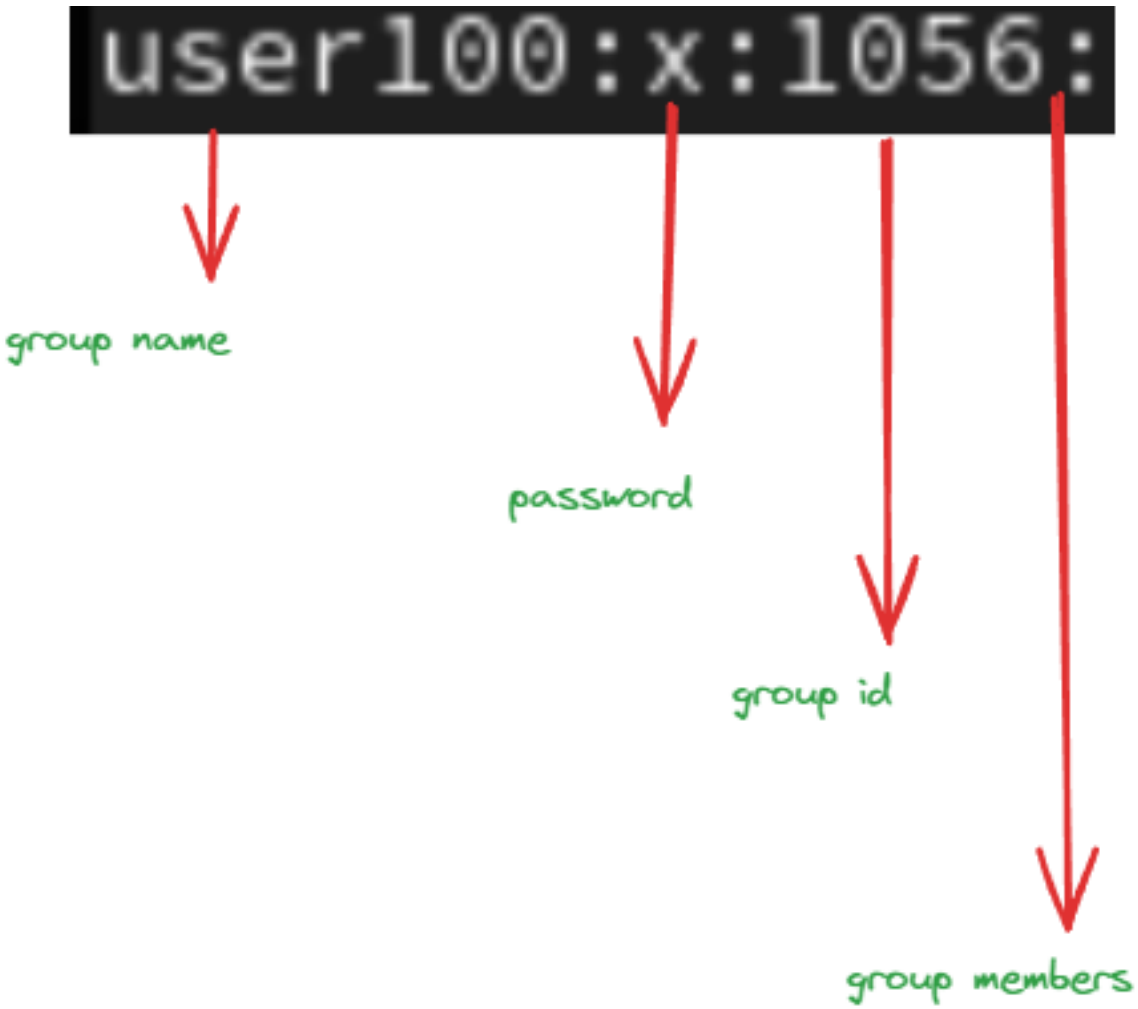


- *Field 1 (Login Name): Contains the login name as appears in the passwd file.*
- *Field 2 (Encrypted Password): Consists of a hashed password. A single exclamation mark (!) at the beginning of this field implies that the user account is locked. If this field is empty, the user will have password-less entry into the system.*
- *Field 3 (Last Change): Sets the number of days (lastchg) since the UNIX epoch, a.k.a. UNIX time (January 01, 1970 00:00:00 UTC) when the password was last modified. An empty field represents the passiveness of password aging features, and a 0 forces the user to change their password upon next login.*
- *Field 4 (Minimum): Expresses the minimum number of days (mindays) that must elapse before the user is allowed to change their password. This field can be altered using the chage command with the -m option or the passwd command with the -n option. A 0 or null in this field disables this feature. Field 5 (Maximum): Defines the maximum number of days (maxdays) of password validity before the user password expires and it must be changed. This field may be altered using the chage command with the -M option or the passwd command with the -x option. A null value here disables this feature along with other features such as the maximum password age, warning alerts, and the user inactivity period.*
- *Field 6 (Warning): Denotes the number of days (warndays) for which the user gets warnings for changing their password before it actually expires. This field may be altered using the chage command with the -W option or the passwd command with the -w option. A 0 or null in this field disables this feature.*
- *Field 7 (Password Expiry): Contains the maximum allowable number of days for the user to be able to log in with the expired password. This period is referred to as the inactivity period. This field may be altered using the chage command with the -I option or the passwd command with the -i option. An empty field disables this feature.*
- *Field 8 (Account Expiry): Determines the number of days since the UNIX time when the user account will expire and no longer be available. This field may be altered using the chage command with the -E option. An empty field disables this feature.*
- *Field 9 (Reserved): Reserved for future use.*

Examine group file?

- The group file is a simple plaintext file and contains critical group information. Each row in the file stores information for one group entry. Every user on the system must be a member of at least one group, which is referred to as the User Private Group (UPG). By default, a group name matches the username it is associated with. Additional groups may be set up, and users with common file access requirements can be added to them. There are four colon-separated fields per line entry.

Command 1: *cat /etc/group*

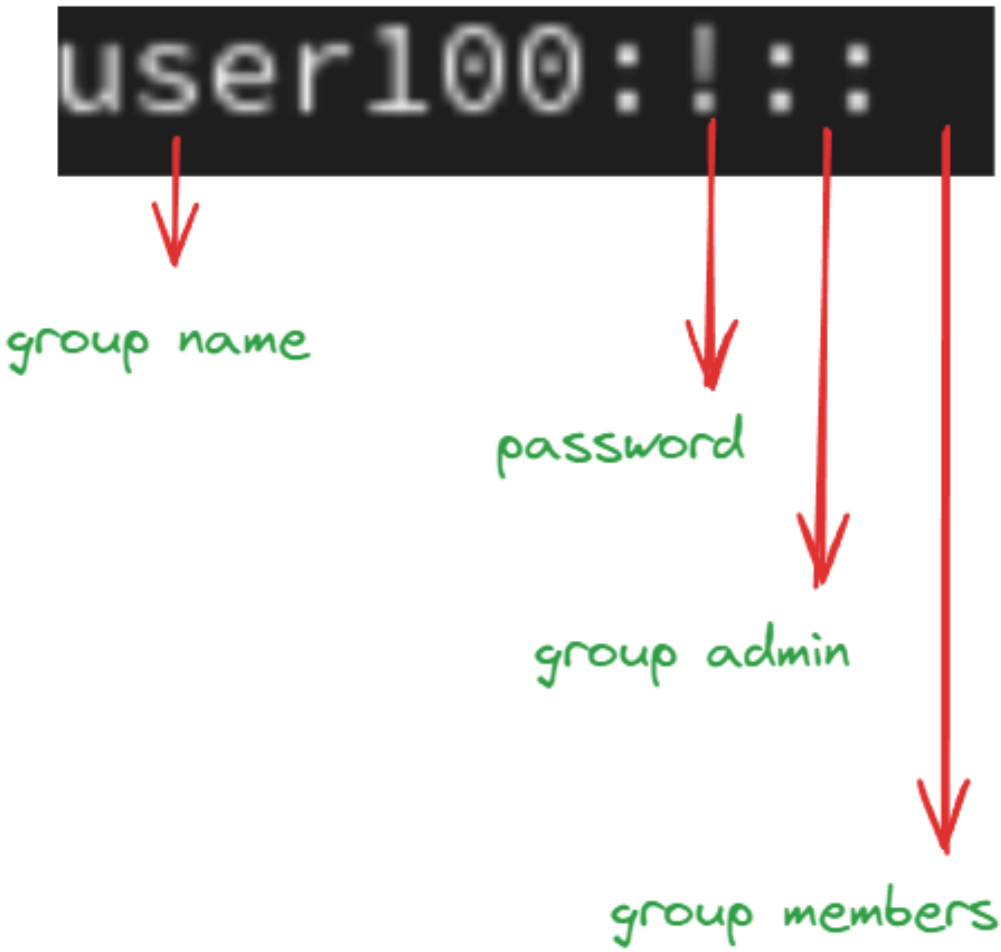


- ***Field 1 (Group Name): Holds a group name that must begin with a letter. Group names up to 255 characters, including the underscore (_) and hyphen (-) characters, are supported. It is not recommended to include special characters and uppercase letters in group names.***
- ***Field 2 (Encrypted Password): Can be empty or contain an "x" (points to the /etc/gshadow file for the actual password), or a hashed group-level password. You can set a password on a group if you want non-members to be able to change their group identity temporarily using the newgrp command. The non-members must enter the correct password in order to do so.***
- ***Field 3 (GID): Holds a GID, which is also placed in the GID field of the passwd file. By default, groups are created with GIDs starting at 1000 and with the same name as the username. The system allows several users to belong to a single group; it also allows a single user to be a member of multiple groups at the same time.***
- ***Field 4 (Group Members): Lists the membership for the group. Note that a user's primary group is always defined in the GID field of the passwd file.***

Examine gshadow file?

- The group file is a simple plaintext file and contains critical group information. Each row in the file stores information for one group entry. Every user on the system must be a member of at least one group, which is referred to as the User Private Group (UPG). By default, a group name matches the username it is associated with. Additional groups may be set up, and users with common file access requirements can be added to them. There are four colon-separated fields per line entry.

Command 1: *cat /etc/gshadow*



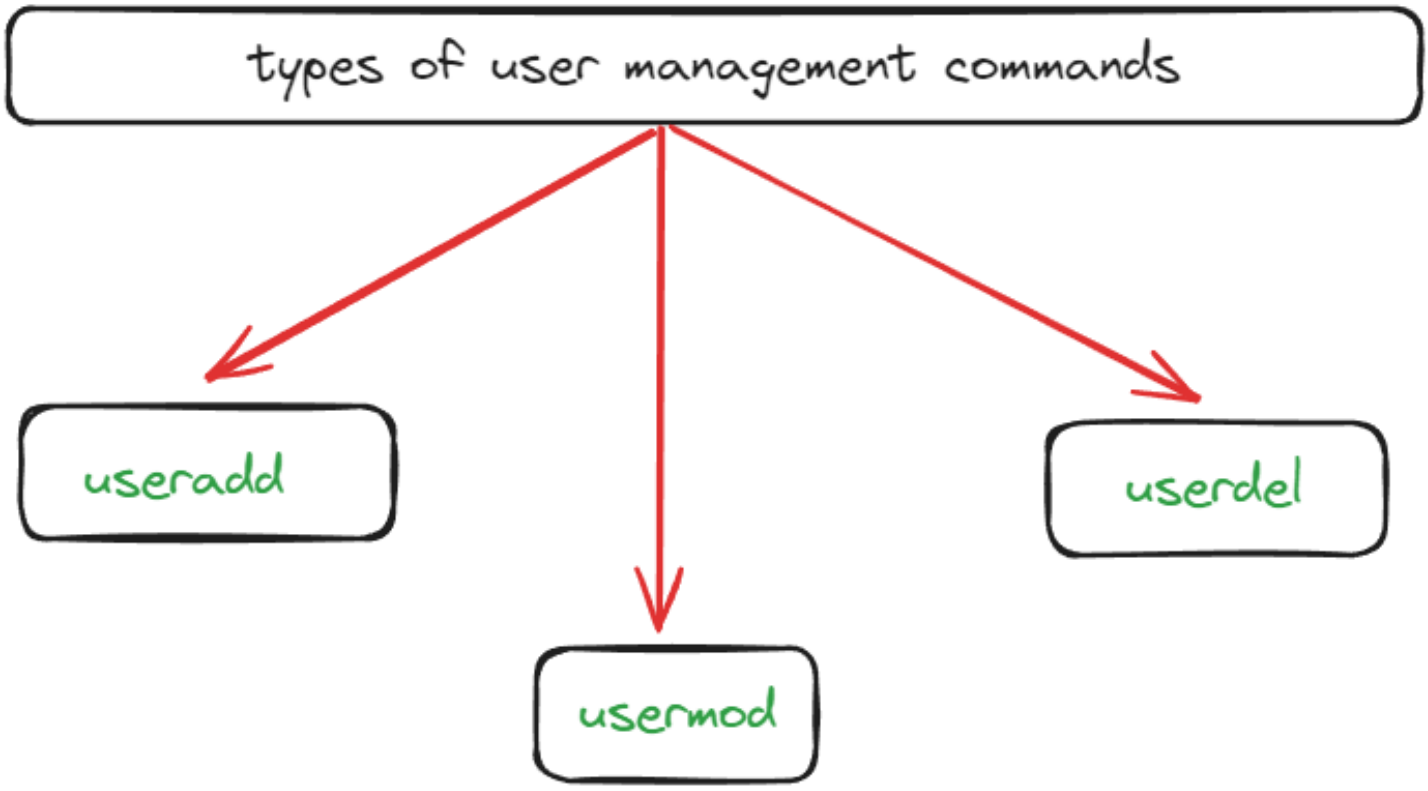
- ***Field 1 (Group Name): Consists of a group name as appeared in the group file.***
- ***Field 2 (Encrypted Password): Can contain a hashed password, which may be set with the gpasswd command for non-group members to access the group temporarily using the newgrp command. A single exclamation mark (!) or a null value in this field allows group members password-less access and restricts non-members from switching into this group.***
- ***Field 3 (Group Administrators): Lists usernames of group administrators that are authorized to add or remove members with the gpasswd command. Field 4 (Members): Holds a comma-separated list of members.***

What is user account managemt?

- Managing user accounts involves creating, assigning passwords to, modifying, and deleting them. RHEL provides a set of tools for performing these operations. These tools are useradd to add a new user to the system, usermod to modify the attributes of an existing user, and userdel to remove a user from the system. In addition, the passwd command is available to set or modify a user's password.

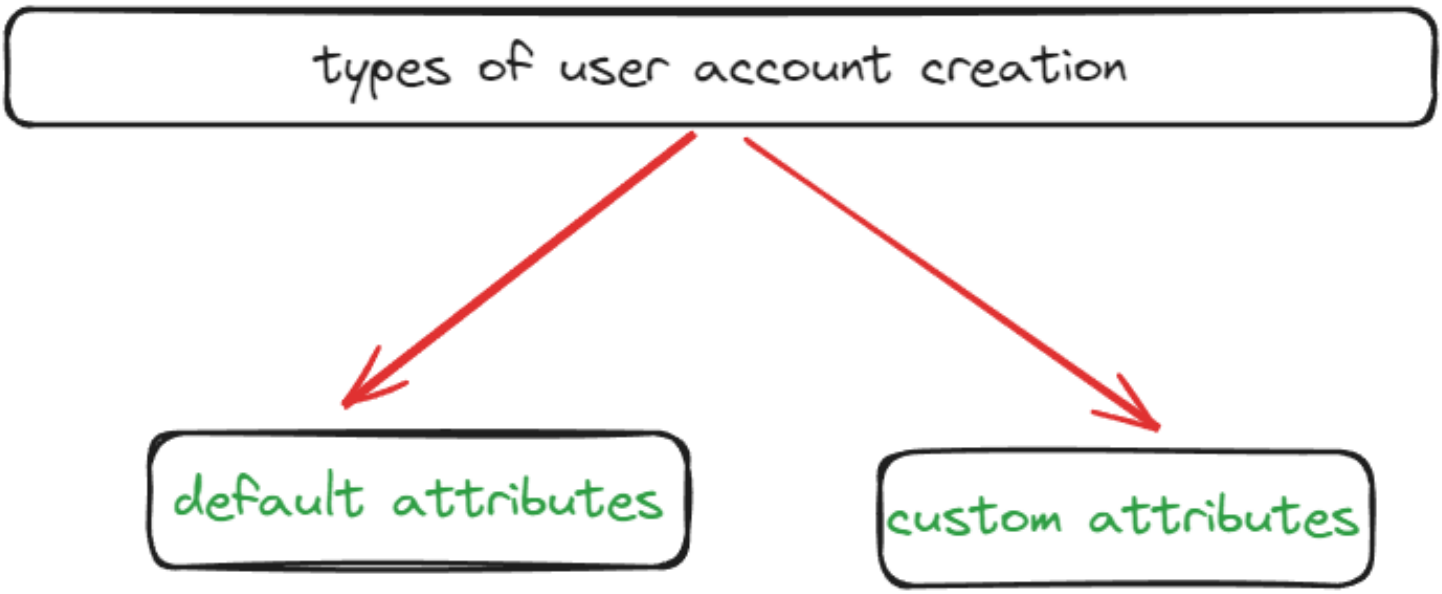
Types of user management commands

- This set of commands is used to add, modify, and delete a user account from the system. The useradd command adds entries to the four user authentication files for each account added to the system. It creates a home directory for the user and copies the default user startup files from the skeleton directory /etc/skel into the user's home directory. It can also be used to update the default settings that are used at the time of new user creation for unspecified settings.



Types of user account creation

- This set of commands is used to add, modify, and delete a user account from the system. The useradd command adds entries to the four user authentication files for each account added to the system. It creates a home directory for the user and copies the default user startup files from the skeleton directory /etc/skel into the user's home directory. It can also be used to update the default settings that are used at the time of new user creation for unspecified settings.



Q1 Create a User Account with **Default Attributes**?

Step 1 : *Create user300 with all the default values:*

```
[root@localhost ~]# useradd user300
[root@localhost ~]#
```

Step 2 : *Assign this user a password and enter it twice when prompted:*

```
[root@localhost ~]# passwd user300
Changing password for user user300.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost ~]#
```

Q1 Create a User Account with **Custom Attributes**?

Step 1 : *Create user400 with UID 2020 (-u), home directory /usr/user400a (-d), and shell /bin/sh (-s):*

```
[root@localhost ~]# useradd -u 2020 -d /usr/user400a -s /bin/sh user400
```

Step 2 : *Assign user1234 as password (passwords assigned in the following way is not recommended; however, it is okay in a lab environment):*

```
[root@localhost ~]# echo user1234 | passwd --stdin user400
Changing password for user user400.
passwd: all authentication tokens updated successfully.
[root@localhost ~]#
```

Step 3 : *grep for user400: on the four authentication files to see what was added for this user:*

```
[root@localhost ~]# cd /etc ; grep user400: passwd shadow group gshadow
passwd:user400:x:2020:2020::/usr/user400a:/bin/sh
shadow:user400:$6$lnegimXoRNOhMO27$4uEkx44YhIpLePsFWYPpP/8bvDVOou8C8S9IzLl
JxQ.PljsZSb8y1DJ9JMV9n/:19815:0:99999:7:::
group:user400:x:2020:
gshadow:user400:!::
```

Modify and Delete a user account

Step 1 : *Modify the login name for user300 to user300new (-l), UID to 2000 (-u), home directory to /home/user300new (-m and -d), and login shell to /sbin/nologin (-s). Notice that the options are specified in a different sequence.*

```
[root@localhost etc]# usermod -l user300new -m -d /home/user300new -s /sbin/nologin -u 2000 user30
0
[root@localhost etc]# #
```

Step 2 : *Obtain the information for user300new from the passwd file for confirmation:*

```
[root@localhost etc]# grep user300new /etc/passwd
user300new:x:2000:1059::/home/user300new:/sbin/nologin
[root@localhost etc]#
```

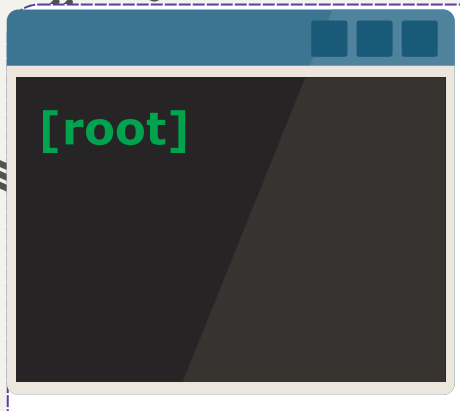
Step 3 : *Remove user300new along with their home and mail spool directories (-r):*

```
[root@localhost etc]# userdel -r user300new
[root@localhost etc]#
[root@localhost etc]#
```

Step 4 : *Confirm the user deletion:*

```
[root@localhost etc]# grep user300new /etc/passwd
[root@localhost etc]#
[root@localhost etc]#
[root@localhost etc]#
```

- ***Nothing is returned in the output, which means this user does not exist in the passwd file any longer. This confirms the removal.***



THANK YOU