Managing Local Users

Andrew Mallett
Linux Author and Trainer

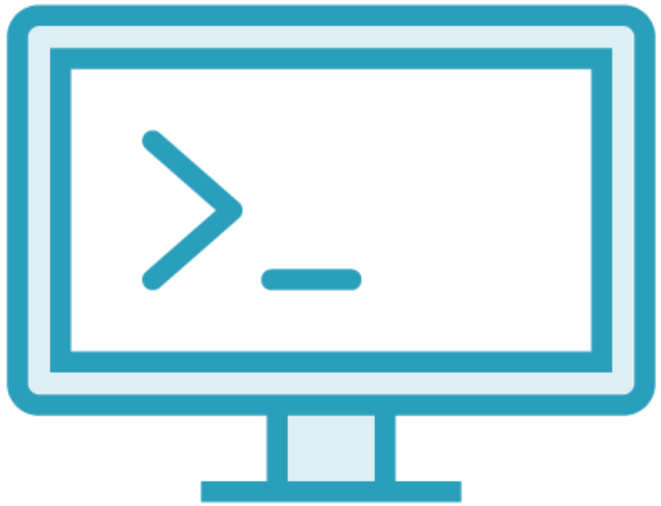
@theurbanpenguin www.theurbanpenguin.com

Overview

Local User Management

- Managing Users
  - Local Users
  - Identity Management
- Managing Groups
- Managing Passwords

Lab Systems



Alma Linux 8.5:

- VirtualBox / Vagrant

```
$ cat /etc/passwd

$ grep vagrant /etc/passwd

$ getent passwd vagrant

$ man 5 passwd
```

Listing Users

The local user account database is the file /etc/passwd. This is a text file accessible by all users. It is not usual to store passwords in this file anymore. The command getent can be used to display or search the database

The file /etc/nsswitch.conf tells getent where to look for entries. By default, locally but could also be an Identity Vault such as OpenLDAP or AD

```
$ cut -f1,3 -d: /etc/passwd | grep vagrant
$ awk -F: '/vagrant/{ print $1 " " $3}' /etc/passwd
```

Printing Selected Information

Typically, the command cut is is used to filter the fields; however, we may also need to use grep to search for a specific user. Using awk we can cover both with the one command

Demo

Listing Users

- The /etc/passwd file
- The command getent and /etc/nsswitch.conf
- Filtering user information

```
$ sudo -i
# useradd u1
# id u1
# getent passwd u1
# useradd -D
# grep -E '^(CREATE_HOME|USERGROUPS_ENAB)' /etc/login.defs
```

Default Settings

When creating a user we can specify just the login name, the rest will come from the user defaults.

User Groups

Primary Group

Defined in the /etc/passwd file. Affects the group ownership of new files created by the user

Complimentary Groups

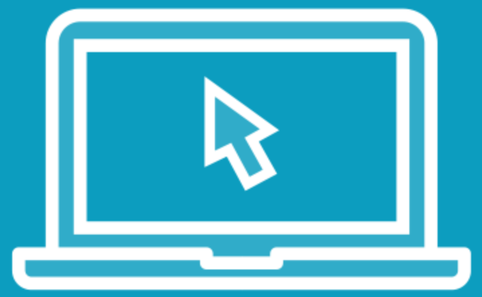
Defined in the /etc/group file along with membership. Affects the rights that a user has to resources and includes the users primary group

```
# useradd -N -G wheel -c 'user two' u2
```

Creating Users with Non-Default Settings

The option -N specifies not to create a user group, the primary group will now be from the defaults. The option -G allows us to specify complimentary groups, here we add the user to the wheel group. The option -c allows the setting of the full name or user comment.

Demo



We now create users:

- Using defaults
- Modify defaults
- And using non-default values

```
# usermod -c 'user one' -aG wheel u1
```

Modify User Accounts

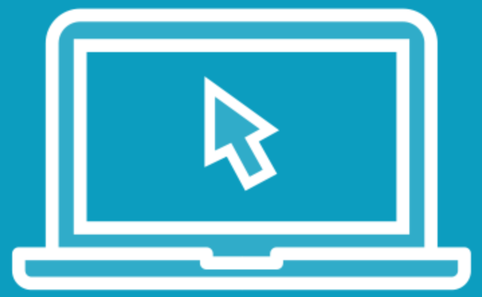
The same options to useradd or available to usermod which is used to modify user accounts

```
# userdel u1
# find /home /var -nouser
# userdel -r u2
# find /home /var -nouser
```

Deleting Users

Deleting users is affected using userdel. The option -r deletes the home directory, mail spool files and user cron jobs

Demo



We will will look at modifying and deleting users:

- Using the usermod command
- Using the userdel command

Linux Passwords

Although the user password could be in the file /etc/passwd, there is only one field to use; not allowing shadow (aging) data. Passwords are usually stored in /etc/shadow with the aging data and accessible only to root

/etc/shadow

Field	Purpose
1	Login name

/etc/shadow

Field	Purpose
1	Login name
2	Encrypted password, if it starts with ! the account is locked

/etc/shadow

Field	Purpose
1	Login name
2	Encrypted password, if it starts with ! the account is locked
3	Date of last password change, expressed as the number of days after 1, Jan 1970

/etc/shadow

Field	Purpose
1	Login name
2	Encrypted password, if it starts with ! the account is locked
3	Date of last password change, expressed as the number of days after 1, Jan 1970
4	Minimum password age, the minimum days a password has to be set for before it can be changed

/etc/shadow

Field	Purpose
1	Login name
2	Encrypted password, if it starts with ! the account is locked
3	Date of last password change, expressed as the number of days after 1, Jan 1970
4	Minimum password age, the minimum days a password has to be set for before it can be changed
5	Maximum password age, how often the password needs to be changed

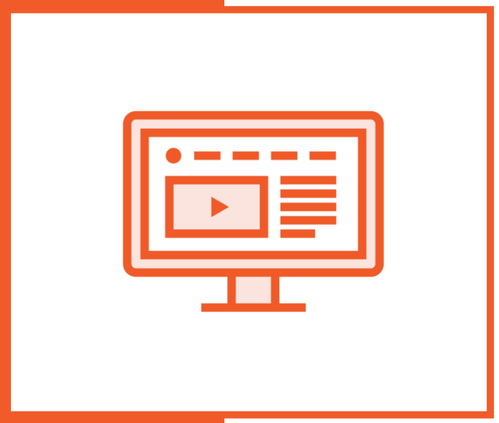
/etc/shadow

Field	Purpose
1	Login name
2	Encrypted password, if it starts with ! the account is locked
3	Date of last password change, expressed as the number of days after 1, Jan 1970
4	Minimum password age, the minimum days a password has to be set for before it can be changed
5	Maximum password age, how often the password needs to be changed
6	Password warning, the days before a password expires in which the user will be warned to change password

/etc/shadow

Field	Purpose
1	Login name
2	Encrypted password, if it starts with ! the account is locked
3	Date of last password change, expressed as the number of days after 1, Jan 1970
4	Minimum password age, the minimum days a password has to be set for before it can be changed
5	Maximum password age, how often the password needs to be changed
6	Password warning, the days before a password expires in which the user will be warned to change password
7	Password inactivity, the number of days after the password has expired that the user can still login using the old password.

/etc/shadow



Field	Purpose
1	Login name
2	Encrypted password, if it starts with ! the account is locked
3	Date of last password change, expressed as the number of days after 1, Jan 1970
4	Minimum password age, the minimum days a password has to be set for before it can be changed
5	Maximum password age, how often the password needs to be changed
6	Password warning, the days before a password expires in which the user will be warned to change password
7	Password inactivity, the number of days after the password has expired that the user can still login using the old password.
8	Account expiry date, the days after 1, Jan 1970 when the account will expire. If the password has expired users can still login using SSH keys for example. When the account expires, no login is possible

```
$ chage -l $USER
```

chage

Using the command chage (change age) a user can see their own shadow data, root can see and change all shadow data

Demo

Let's begin by examining the shadow data:

- /etc/shadow
- man 5 shadow
- chage -l

Default Password Aging Controls

The file /etc/login.defs allows for configuration of default aging settings

Demo

Working with /etc/login.defs

```
$ sudo getent shadow vagrant
$ sudo getent shadow vagrant|awk -F$ '{ print "alg: " $2 "\nsalt: " $3 "\npwd: " $4 }'
alg: 6
salt: AMgw75RpN3vBo1qO
pwd:
cs.DaVbaZ8R01mlA0LFtPjDffvFND6rGkQ/AAPzmtpm2maVHY/kEL3cNy3iyljZgubpxC.ObEz/L5dSWazMPL0
```

User Passwords

Passwords are stored within the second field of the shadow file. The entry itself is broken down 3 separate entities: the algorithm, the SALT and the password hash.

Passwords are one way password hashes. They can't be decrypted. Authentication occurs by comparing hash values. If the correct password is supplied the same hash will be produced with when combined with the same SALT

```
$ echo 'Password1' | sudo passwd u1 --stdin

$ sudo getent shadow u1 | awk -F$ '{ print "alg: " $2 "\nsalt: " $3 "\npwd: " $4 }'
alg: 6
salt: xda6csfZi9xqDYkX
pwd:
9r6fhE1qZFGEIBJrpF3ZoJaojw5kCEZIzFItIlAKxIpZXICNQ27meb2E4Kujmmt8GM0Cz3WzRlFSKOn/Z3q7d0

$ openssl passwd -6 -salt xda6csfZi9xqDYkX Password1
$6$xda6csfZi9xqDYkX$9r6fhE1qZFGEIBJrpF3ZoJaojw5kCEZIzFItIlAKxIpZXICNQ27meb2E4Kujmmt8GM0
Cz3WzRlFSKOn/Z3q7d0
```

Authenticating Users

We can quickly create a user account setting its password. RedHat based systems have the option --stdin but Debian based systems do not. We can show the authentication process by using the openssl command

Demo

We now look at user authentication

- Create user with password
- Read the password hash and compare using the openssl command

```
$ echo Password1 | sudo passwd u2 --stdin

$ vim users
u1:Password123
u2:Password123

$ cat users | sudo chpasswd

$ sudo passwd -l | -S | -u u1
```

Changing and Setting Passwords

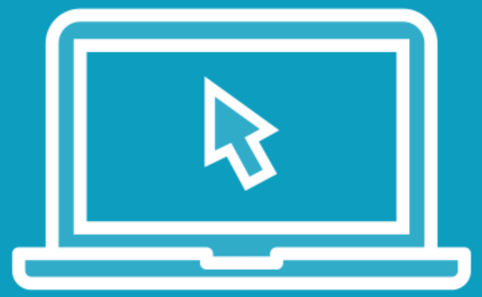
Commonly, the command passwd is used to set the password. If you use both Debian and RedHat based systems, the chpasswd command is useful for non-interactive password setting. The passwd command is also useful for locking a users account, perhaps while they are on vacation. Use -l to lock, -S to check status and -u to unlock

```
$ sudo useradd -r u3 ; echo Password1 | sudo passwd u3 --stdin

$ sudo chage -l u3

$ sudo chage -M 99999 -m 0 -E -1 -I -1 u3
```

System Accounts

Having modified the login.defs, this will work for standard interactive users. For systems accounts we don't want the password to expire using the useradd option -r we by-pass the restrictions in login.defs. We can use chage to set explicit information for any account.

Demo



Managing User Passwords:

- passwd
- chpasswd
- Allowing for system accounts

```
$ grep vagrant /etc/group
$ gid=$(awk -F: '/^vagrant/ { print $3 }' /etc/group)
$ echo $gid
$ awk -F: -v gid="$gid" 'gid == $4 { print $0 }' /etc/passwd
vagrant:x:1000:1000::/home/vagrant:/bin/bash
```

/etc/group

Groups are created and stored in the file /etc/group. Membership of secondary groups is maintained here also, but if the group is a user's primary group, the membership is maintained by the /etc/passwd file. Getting clever with awk we can match the GID in the groups file with the associated user in the /etc/passwd file

Demo

Let's begin by examining the group file:

- /etc/group
- becoming familiar with awk
- match GID in /etc/group to /etc/passwd

```
$ sudo groupadd mkt
$ sudo usermod -aG wheel,mkt vagrant
```

Adding Users to Groups

Within the useradd or usermod command we have -g for the user's primary group. This can be a single value only; however, the -G for secondary groups allow for many groups. We can specify a new list or use -a the append the existing groups

Demo

Working with groups:

- Creating groups
- Adding users to groups

Group Passwords

Adding a group passwords will make your group less secure NOT more secure. That said the gpasswd command has more uses!

```
$ sudo groupadd sales

$ sudo gpasswd -a vagrant sales

$ sudo gpasswd -A vagrant sales

$ gpasswd -a u1 sales
```

gpasswd

The gpasswd command does more than set the group password. It can be used to set group administrators with the option -A. The administrator is stored in the /etc/gshadow file

Demo

We now look at user gpasswd

- self-service groups
- group administrators

Summary

User Management

- Managing Users
  - Local Users
  - Identity Vaults
- Managing Passwords
  - SALT
  - HASH
- Managing Groups

Implementing HTTPS with the Apache Web Server