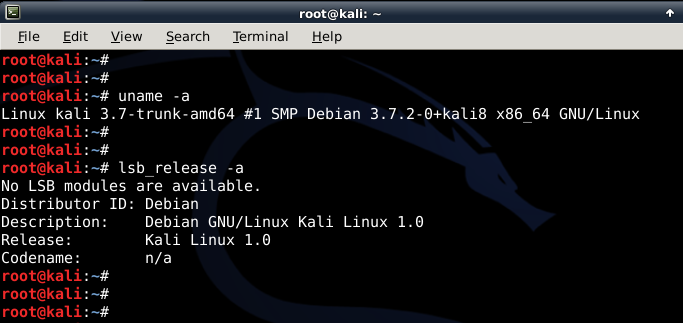
Create a user named bob in Kali.

First of all let’s confirm which version of Linux and Kernel I’m running.In command prompt type in

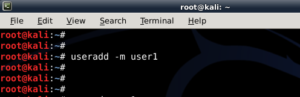
Uname -a

Lsb\_release -a



Now let’s add user. Open terminal and type following to create new user **(replace** user1 with your desired user name)

Useradd -m user1

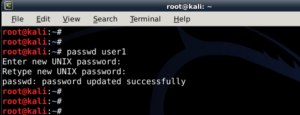


Note: **-m** means create home **directory** which is usually **/home/username)**

Now set password for this user

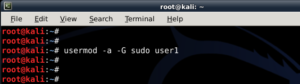
Passwd user1

Enter desired password twice



Add user to **sudo** group (to allow user to install software, allow printing, use privileged mode etc.

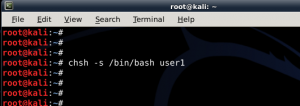
Usermod -a -G sudo user1



(Note:**-a** means **append** or add and **–G** mean to specified **group/groups)**

Change default shell of previously created user to bash

Chsh -s /bin/bash user1



(Note: **chsh** mean change login shell, **-s** is the name of the specified shell you want for the user, in this case /bin/bash)

all worked out as expected.

-------------------------------------------------------------------------------------------

Enable ssh on Kali

The openssh-server package should already be installed, to verify this you can use the following command:

# apt list openssh-server

If it’s not installed, you can use this command to install it:

# apt install openssh-server

When enabling the service, be sure to fully secure SSH first.  I will cover some of the basics briefly, but this is not meant to be a guide on securely running an SSH server.  Since Kali comes with pre-generated SSH keys, to make it more secure, the first thing we will do is generate new ones.

To backup the original keys first as a precaution use:

# mkdir /etc/ssh/default\_keys

# mv /etc/ssh/ssh\_host\_\* /etc/ssh/default\_keys/

Then to regenerate the keys:

# dpkg-reconfigure openssh-server

The next step is to edit the SSH server configuration file with the settings you need:

# nano /etc/ssh/sshd\_config

If you are only planning on using SSH briefly the defaults are probably fine.  If you think you will use it for a length of time I would recommend at minimum enabling public key authentication:

PubkeyAuthentication yes

Then disabling password authentication:

PasswordAuthentication no

You could also allow the root user login here, but instead consider creating a non-privileged user account instead.

It’s useful to know that Systemd has different units, a unit configuration file encodes information.  The units relevant to SSH are ssh.service and ssh.socket. At a basic level a service unit controls a process and a socket unit controls a filesystem or network socket.

If you only need to temporarily start up the SSH service it’s recommended to use ssh.socket:

# systemctl start ssh.socket

When finished:

# systemctl stop ssh.socket

To instead permanently enable the SSH service to start whenever the system is booted use:

# systemctl enable ssh.service

Then to use SSH immediately without having to reboot use:

# systemctl start ssh.service

To check the status of the service you can use:

# systemctl status ssh.service

To stop the SSH service use:

# systemctl stop ssh.service

And to disable the SSH service so it no longer starts at boot:

# systemctl disable ssh.service