# OpenSSH server and RSA authentication

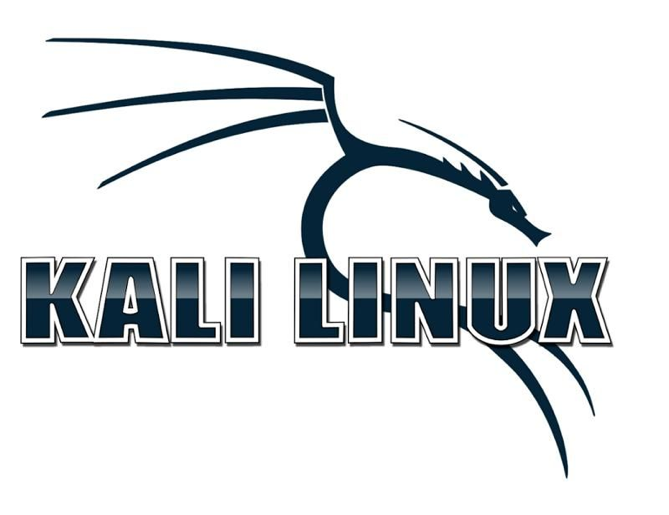
Goal of this Lab:

Installation of OpenSSH server on Debain Linux and demonstrating use of RSA keys.

Pre-requisites for this lab:

1. Debian Linux VM
2. Kali Linux VM
3. Both VMs needs to be connected through NAT Network.





A logo of a company

Description automatically generated

Table of Contents

[OpenSSH server and RSA authentication 1](#_Toc167804438)

[OpenSSH 3](#_Toc167804439)

[Task 1: Indenifying status of SSH server 4](#_Toc167804440)

[Task 2: Installing SSH server if it is required 5](#_Toc167804441)

[Task 3: Generate key-pair on Debian 6](#_Toc167804442)

[Task 4: Generate key-pair on Kali 7](#_Toc167804443)

[Task 5: Verifying the key has been created 7](#_Toc167804444)

[Task 6: Connecting to SSH server using username and password 8](#_Toc167804445)

[Task 6: Using RSA authentication to connect to SSH server. 9](#_Toc167804446)

[Step 1: Make sure you have closed pervious SSH connection. 9](#_Toc167804447)

[Step 2: Copying public key to SSH server using ssh-copy-id 10](#_Toc167804448)

[Step 3: Login to SSH server 11](#_Toc167804449)

[Task 7: Disabling username - password login. 12](#_Toc167804450)

[Task 8: Creating new user on Kali and adding it to sudo group 14](#_Toc167804451)

[Step 1: Create a new user. 14](#_Toc167804452)

[Step 2: Add user new\_user to sudo group 15](#_Toc167804453)

[Step 3: Change to new\_user 15](#_Toc167804454)

[Task 9: Login to the SSH server from the new user account 16](#_Toc167804455)

[Task 10: Generate RSA key for new\_user. 16](#_Toc167804456)

[Task 11: Copy RSA key of new\_user from Kali to authorized\_keys files located on SSH server(Debian). 16](#_Toc167804457)

[Step 1: First we need to read the key, then copy it. 17](#_Toc167804458)

[Step 2: Select the key and copy it 17](#_Toc167804459)

[Step 3: We need to add the key we copied from Kali to authorized\_keys file in SSH server(Debian). 17](#_Toc167804460)

[Step 12: Login from new\_user account from Kali. 19](#_Toc167804461)

# OpenSSH

A picture containing text

Description automatically generated

OpenSSH is the premier connectivity tool for remote login with the SSH protocol. It encrypts all traffic to eliminate eavesdropping, connection hijacking, and other attacks. In addition, OpenSSH provides a large suite of secure tunnelling capabilities, several authentication methods, and sophisticated configuration options.

OpenSSH (also known as OpenBSD Secure Shell[a]) is a suite of secure networking utilities based on the Secure Shell (SSH) protocol, which provides a secure channel over an unsecured network in a client–server architecture.[4][5]

OpenSSH started as a fork of the free SSH program developed by Tatu Ylönen; later versions of Ylönen's SSH were proprietary software offered by SSH Communications Security.[6] OpenSSH was first released in 1999 and is currently developed as part of the OpenBSD operating system.

OpenSSH is not a single computer program, but rather a suite of programs that serve as alternatives to unencrypted protocols like Telnet and FTP. OpenSSH is integrated into several operating systems, namely macOS and most Linux operating systems, while the portable version is available as a package in other systems

Reference: <https://en.wikipedia.org/wiki/OpenSSH>

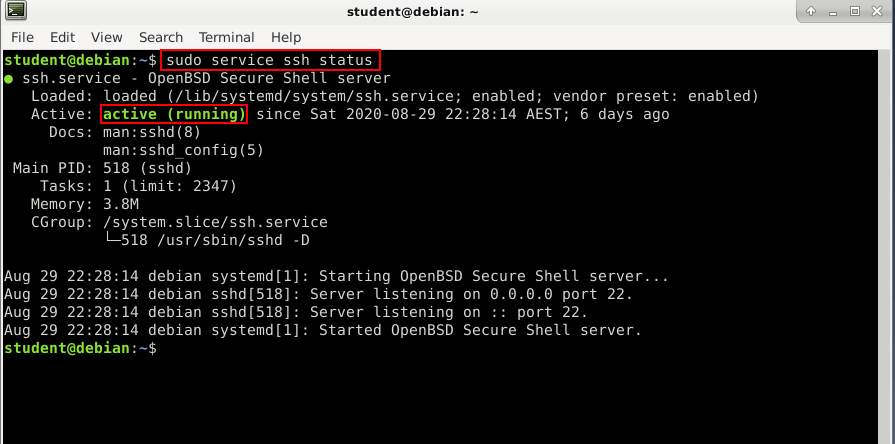
Web: <https://www.openssh.com>

# Task 1: Indenifying status of SSH server

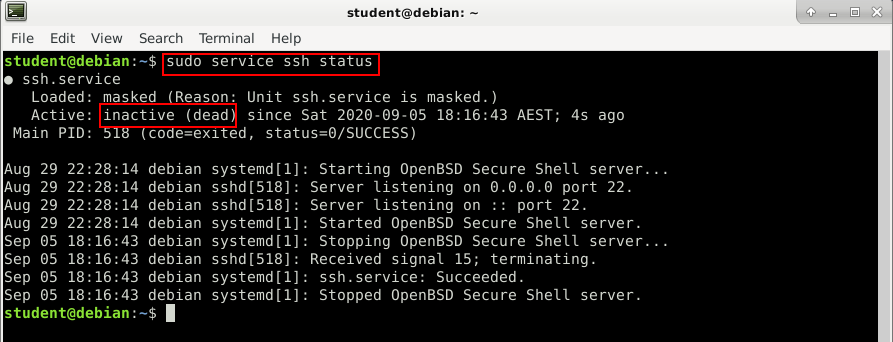
You can check do you have installed SSH server on your linux machine by

**sudo service ssh status**

**If you have SSH already installed, you should see similar output on your screen and you can go to** [**Step 3**](#_Step_3:_Generate)



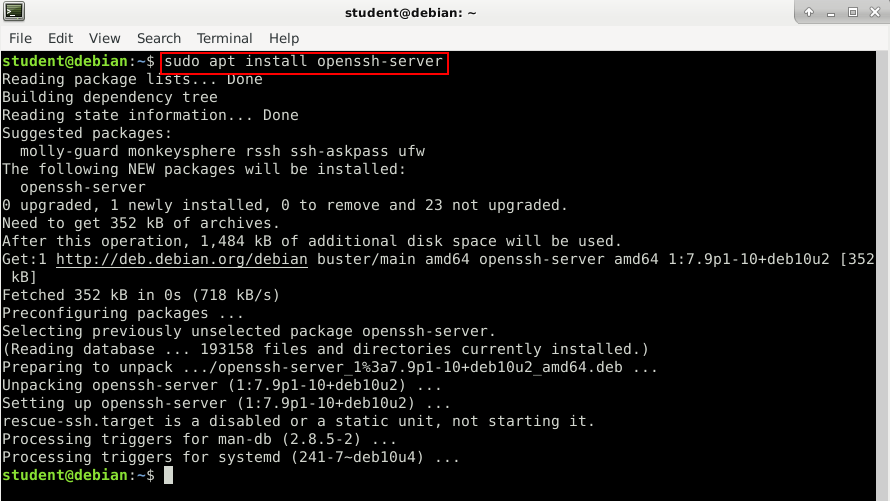
If you see output similar to the below screenshot, you need to install SSH server.



# Task 2: Installing SSH server if it is required

OpenSSH server can be instaled on Debian based distributionsusing:

**sudo apt install openssh-server**



If you get prompted, do you want to downlaod files press Y.

# Task 3: Generate key-pair on Debian

To generate SSH keys use command below:

**ssh-keygen -t rsa -b 2048**

After issuing this command press enter when prompted.

A computer screen shot of a program

Description automatically generated

This command will generate2048 bits RSA key-pair

Password will make your private RSA key secure as it will be encrypted.

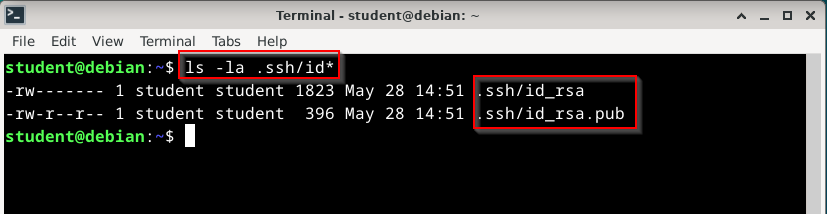
Best practice is to use password however in this case we will leave password blank.

# Task 4: Generate key-pair on Kali

Repeat all steps on Kali

# Task 5: Verifying the key has been created

**ls -la ~/.ssh/id\_\***



**Id\_rsa is your private key. It should not be shared EVER.**

**Id\_rsa.pub is your public key. It will be sent to ssh server. And it will be used for encryption**

Examine contents of keys using tool of your choice:

A black screen with white text

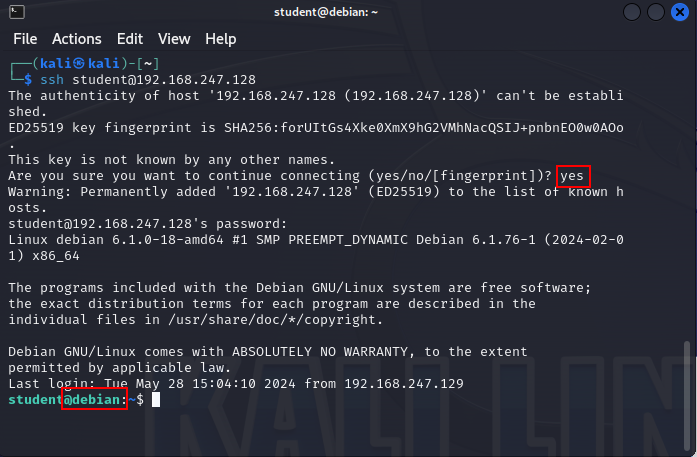
Description automatically generated

Why is bad idea to share screenshots like this?

Answer:

# Task 6: Connecting to SSH server using username and password

In you Kali VM, issue following command to connect to SSH server running on Debian **(Use the IP of your Debain VM)**



Note: As this is self signed key we have no means to verify it’s authenticity. As we are administrators and we know where we are connected to it is ok. Otherwise this will alert you if other isde can’t be authenticated via CA.

Type yes and press enter

You will be prompted for password.

Note: Prompt has changed. Now it shows @debian

# Task 6: Using RSA authentication to connect to SSH server.

Best practice is to disable username and password when connecting to SSH server and use your RSA key to login. To acomplish this we need to copy our public key to SSH server.

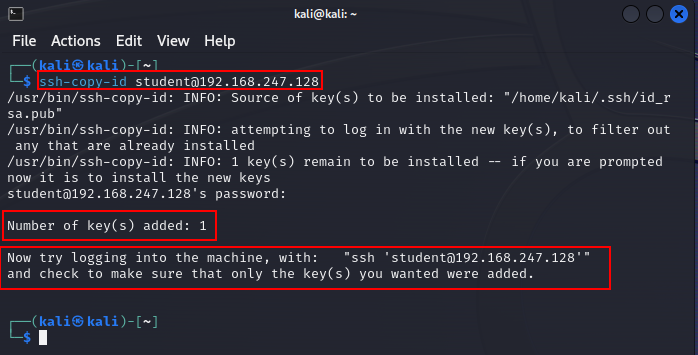
### Step 1: Make sure you have closed pervious SSH connection.

A screenshot of a computer

Description automatically generated

### Step 2: Copying public key to SSH server using ssh-copy-id

**ssh-copy-id student@<ip of ssh server>**

****

**Note: If you see error as below you need to generate RSA keys on Kali. Refer to** [**Task 3**](#_Step_3:_Generate)**.**

A screenshot of a computer

Description automatically generated

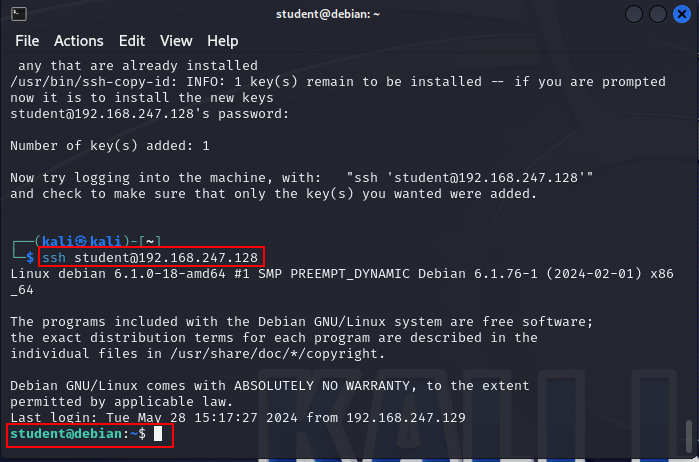
This command will read content of the id\_rsa.pub key and it will store it on the SSH server in file authorized\_keys. That key is located on following locaton **/home/USERNAME/.ssh/authorized\_keys**

In this case full path will be **/home/student/.ssh/authorized\_keys**

This is important because sometimes we need to manually add key to this file.

### Step 3: Login to SSH server

**ssh student@<ip of ssh server>**



Note: This time you are not prompted with password as your RSA public key has been stored in SSH server as a proof of identity.

While forming the connection client signs packet with his private key and the server verifies that it is valid signature using public key that is stored.

Compare public key that has been inserted in Debia’sn authorised keys and Kali’s public key.

A screenshot of a computer

Description automatically generated

On Kali issue following command:

**cat .ssh/id\_rsa.pub**

Compare output from command with content of Debian authorised\_keys file

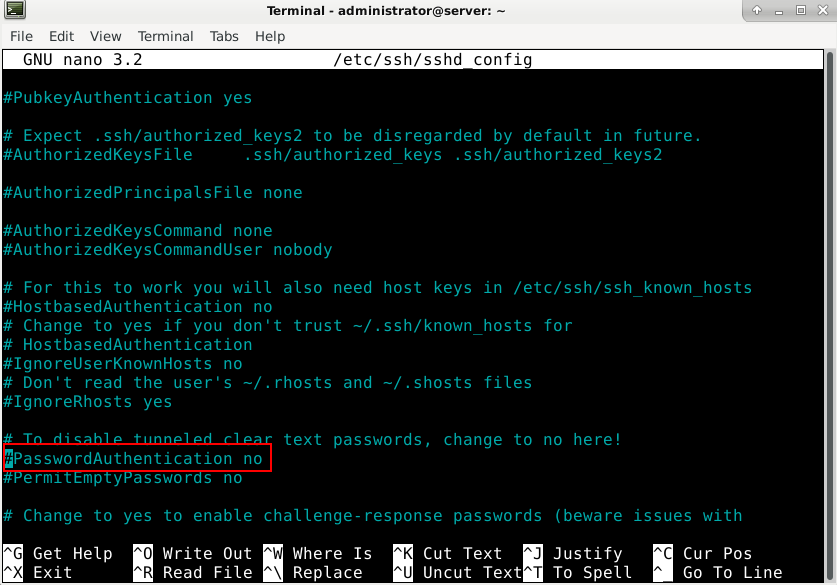
# Task 7: Disabling username - password login.

In oder to disable username and password as authentification method and allow only RSA, we need to edit locating configuration file of SSH server loacted at following path /etc/ssh/sshd\_config

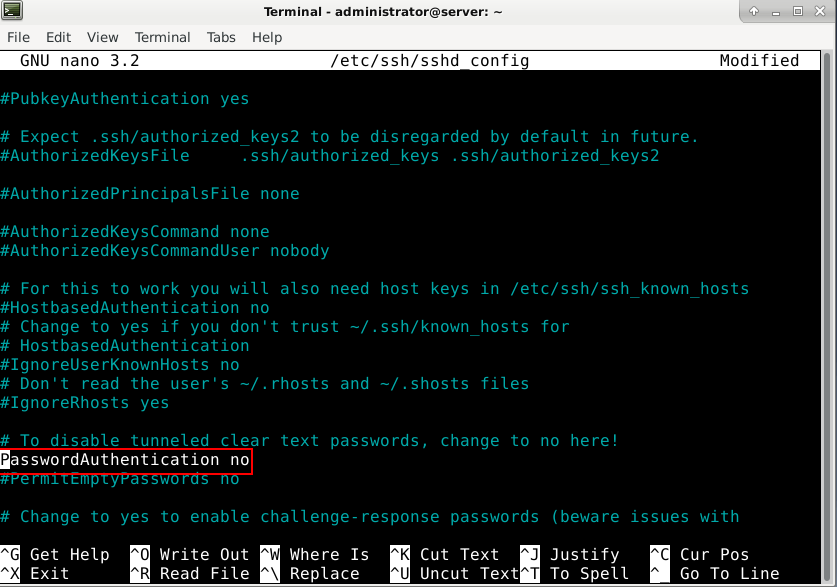
**sudo nano /etc/ssh/sshd\_config**

Locate line

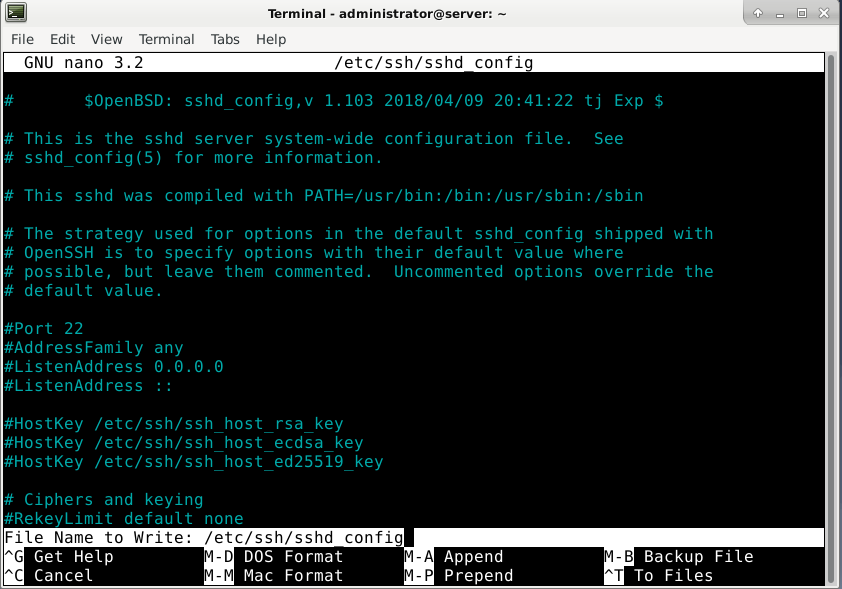
#PasswordAuthentication yes



Remove comment **#**



Save the file using **ctrl + o**



And press enter. Press ctrl + x to close the nano.

Restart SSH service using

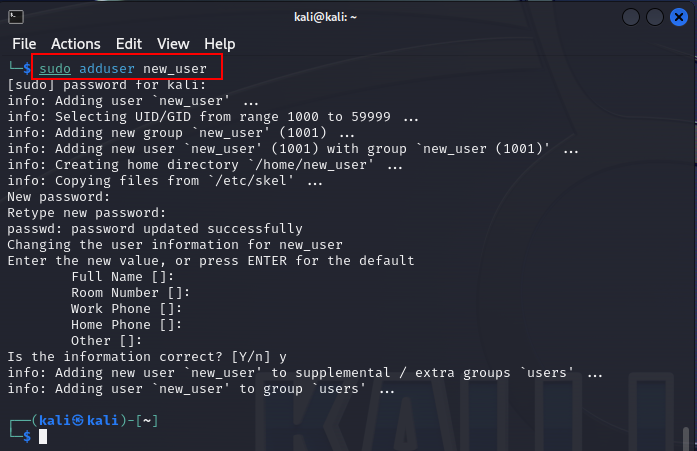
**sudo systemctl restart ssh**

# Task 8: Creating new user on Kali and adding it to sudo group

### Step 1: Create a new user.

**sudo adduser new\_user**

Password for new user is student



### Step 2: Add user new\_user to sudo group

**sudo adduser new\_user sudo**

A screenshot of a computer

Description automatically generated

### Step 3: Change to new\_user

**su new\_user**

**A computer screen shot

Description automatically generated**

**Go to the home directory of new\_user.**

**A screen shot of a computer

Description automatically generated**

# Task 9: Login to the SSH server from the new user account

**ssh student@<ip of ssh server>**

As we can see, permission is denied because we haven’t stored RSA key in the SSH server authorised key file.

# Task 10: Generate RSA key for new\_user.

Step 1 - Repeat steps from [**Task 3**](#_Step_3:_Generate)

Note: Make sure that you are generating RSA for new\_user.

# Task 11: Copy RSA key of new\_user from Kali to authorized\_keys files located on SSH server(Debian).

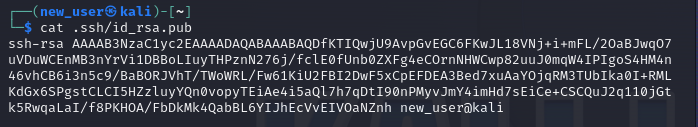
**Ssh-copy-id command will not work as PasswordAuthentication has been disabled in** [**Task 7**](#_Step_7:_Disabling)**. Refer to the screenshot below.**

**A screenshot of a computer

Description automatically generated**

### Step 1: First we need to read the key, then copy it.

**cat .ssh/id\_rsa.pub**

****

### Step 2: Select the key and copy it

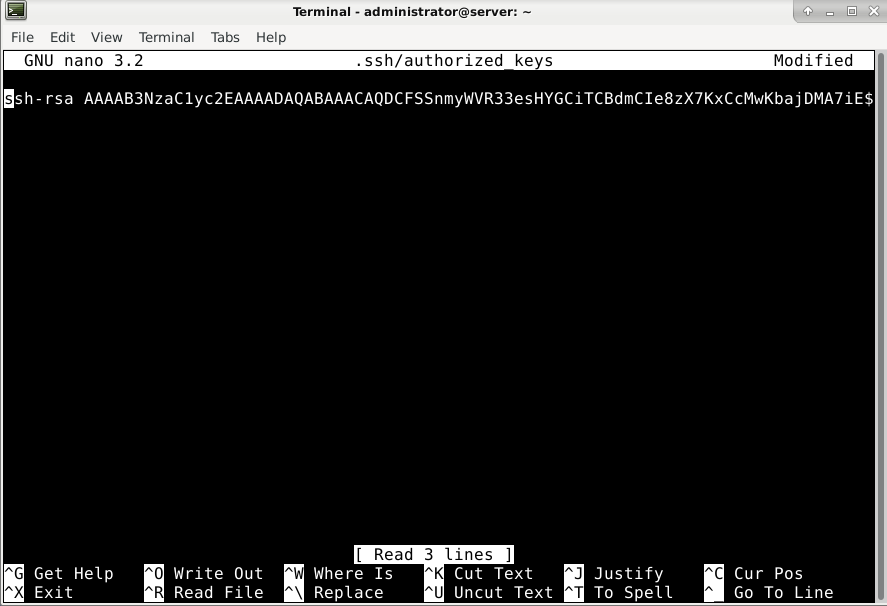
**A white text box

Description automatically generated**

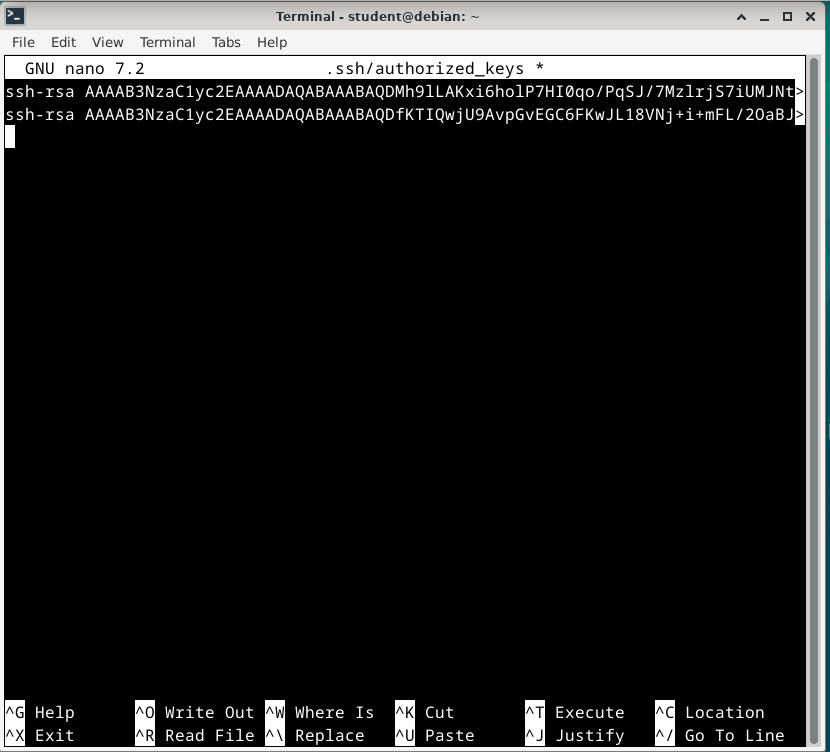
### Step 3: We need to add the key we copied from Kali to authorized\_keys file in SSH server(Debian).

**Note: This command is to be executed on Debian!**

**sudo nano .ssh/authorized\_keys**



As you can see the key for user Kali is in this file. Now we need to paste key for new\_user

****

**Save the file and close it.**

# Step 12: Login from new\_user account from Kali.

Go back to your Kali machine and try to connect again to SSH server.