

12 Project network security

Names: Omar sherif Hosny 221010339

Mohamed ElSayed 221010750

To Eng : Abdelrhman Solyman

Part 2

1. Introduction

Secure Shell (SSH) is a cryptographic network protocol used for securing remote login and other secure network services over an insecure network. This project focuses on configuring SSH key-based authentication between two Linux Mint virtual machines (Client and Server).

2. Steps and Execution

Step 1: Install OpenSSH Server and Client

On both Client and Server VMs, install OpenSSH server and client:

```
sudo apt update
```

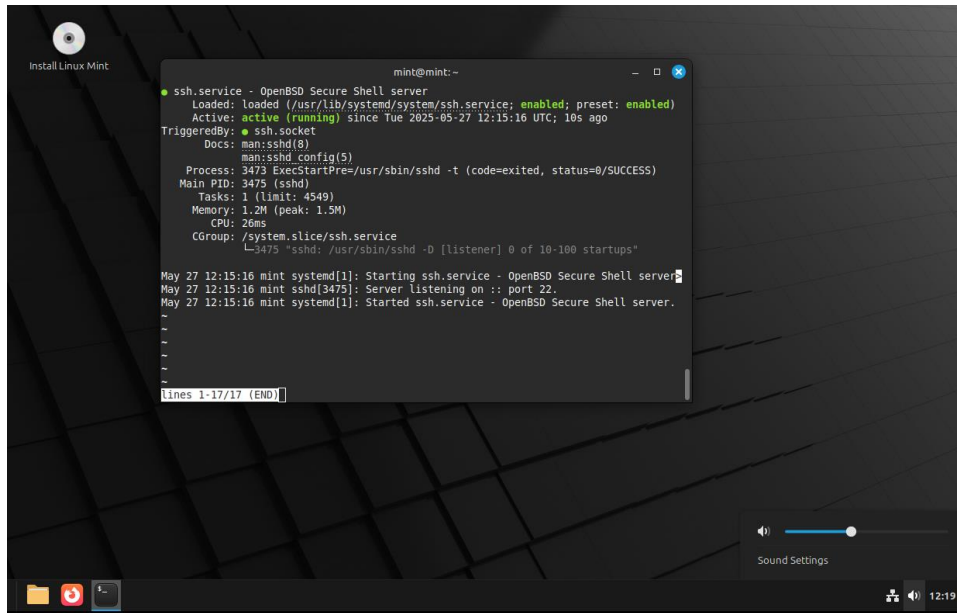
```
sudo apt install openssh-server openssh-client
```

Ensure the SSH service is active on the server:

```
sudo systemctl enable ssh
```

```
sudo systemctl start ssh
```

This commands make us to start use ssh and it make runing in server and client



Step 2: Create a User on the Server

A new user was created on the server with the following command:

```
sudo adduser omar221010339
```

During the process, a password was set, and optional fields (Full Name, Room Number, etc.) were left empty.

```
mint@mint:~  
err: To avoid ambiguity with numerical UIDs, usernames which  
      consist of only digits are not allowed.  
mint@mint:~$ sudo adduser omar221010339  
info: Adding user `omar221010339' ...  
info: Selecting UID/GID from range 1000 to 59999 ...  
info: Adding new group `omar221010339' (1001) ...  
info: Adding new user `omar221010339' (1001) with group `omar221010339 (1001)' .  
..  
info: Creating home directory `/home/omar221010339' ...  
info: Copying files from `/etc/skel' ...  
New password:  
Retype new password:  
passwd: password updated successfully  
Changing the user information for omar221010339  
Enter the new value, or press ENTER for the default  
Full Name []: omar  
Room Number []:  
Work Phone []:  
Home Phone []:  
Other []:  
Is the information correct? [Y/n] y  
info: Adding new user `omar221010339' to supplemental / extra groups `users' ...  
info: Adding user `omar221010339' to group `users' ...  
mint@mint:~$
```

Step 3: Connect to the Server Using Password Authentication

From the client machine, an SSH connection was established using the newly created username and password:

```
ssh omar221010339@<server-ip>
```

```
omar@192.168.13.158: Permission denied (publickey,  
mint@mint:~$ ssh omar221010339@192.168.13.158  
omar221010339@192.168.13.158's password:  
omar221010339@mint:~$
```

Step 4: Generate SSH Key Pair on Client

On the client machine, an ed25519 key pair was generated with:

```
ssh-keygen -t ed25519 -C "omar221010339@client"
```

Default save location was accepted (~/.ssh/id_ed25519), and passphrase was skipped.

```
mint@mint:~$ ssh-keygen -t ed25519 -C "omar@client"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/mint/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/mint/.ssh/id_ed25519
Your public key has been saved in /home/mint/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:x+DDKuwaFdI3uQGpANzCMdR65Y1eI/3SnBQ4a9xREDw omar@client
The key's randomart image is:
+--[ED25519 256]--+
| .+ .          oo+. |
| 0 +. .        o E  |
| =..o o.  + + + |
| .o. 00000 = 0 |
| .....S.o. = . |
| 0 0.= + . = |
| . = = * . |
| 0 + . + |
| ... . |
+-----[SHA256]-----+
mint@mint:~$
```

RSA

```
omar221010339@mint: ~  
omar221010339@mint:~$ ssh-keygen -t rsa -f "omar221010339"  
Generating public/private rsa key pair.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in omar221010339  
Your public key has been saved in omar221010339.pub  
The key fingerprint is:  
SHA256:uy4weAsh1huUJacLIykJsYfEvSpa6RFwbcxFS+QtKyw omar221010339@mint  
The key's randomart image is:  
+---[RSA 3072]-----+  
|+0.=0*=|  
|+*.+0o o|  
|B=+0. + |  
|+0==. o |  
|..E*= . S|  
|..*0+. . |  
|00 + + . |  
|. . . . .|  
|      oo |  
+----[SHA256]-----+  
omar221010339@mint:~$
```

Step 5: Copy Public Key to Server

The public key was copied to the server's `authorized_keys` using:

```
ssh-copy-id omar221010339@<server-ip>
```

Step 6: SSH Login Without Password

The SSH connection was tested again, and it logged in successfully without prompting for a password:

```
ssh omar221010339@<server-ip>
```

```
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'omar221010339@192.168.13.158'"
and check to make sure that only the key(s) you wanted were added.

mint@mint:~$ ssh omar221010339@192.168.13.158

Last login: Tue May 27 14:48:11 2025 from 192.168.13.160
omar221010339@mint:~$
```

Step 7: Disable Password Authentication on Server

To enforce key-based authentication only, password login was disabled on the server:

```
sudo nano /etc/ssh/sshd_config
```

Modified line: PasswordAuthentication no

Then, SSH service was restarted: `sudo systemctl restart ssh`

```
AuthorizedPrincipalsFile none
AuthorizedKeysCommand none
AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
KbdInteractiveAuthentication no

# rberos options
# rberosAuthentication no
# KerberosOrLocalPasswd yes
# KerberosTicketCleanup yes
# KerberosGetAFSToken no

# GSSAPI options
# GSSAPIAuthentication no
# GSSAPICleanupCredentials yes
# GSSAPIStrictAccepterCheck yes
# GSSAPIKeyExchange no

# Set this to 'yes' to enable PAM authentication, account processing,
# and session processing. If this is enabled, PAM authentication will
# be allowed through the KbdInteractiveAuthentication and
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication via KbdInteractiveAuthentication may bypass
# the setting of 'PermitEmptyLogin: Prohibit password'.
# If you just want the PAM account and session checks to run without
# PAM authentication, then enable this but set PasswordAuthentication
# and KbdInteractiveAuthentication to 'no'.
UsePAM yes

AllowAgentForwarding yes
```

Step 8: SSH Verbose Log

To analyze the SSH connection, the following command was used:

```
ssh -v omar221010339@<server-ip>
```


The verbose log shows stages like key exchange, authentication, and session establishment

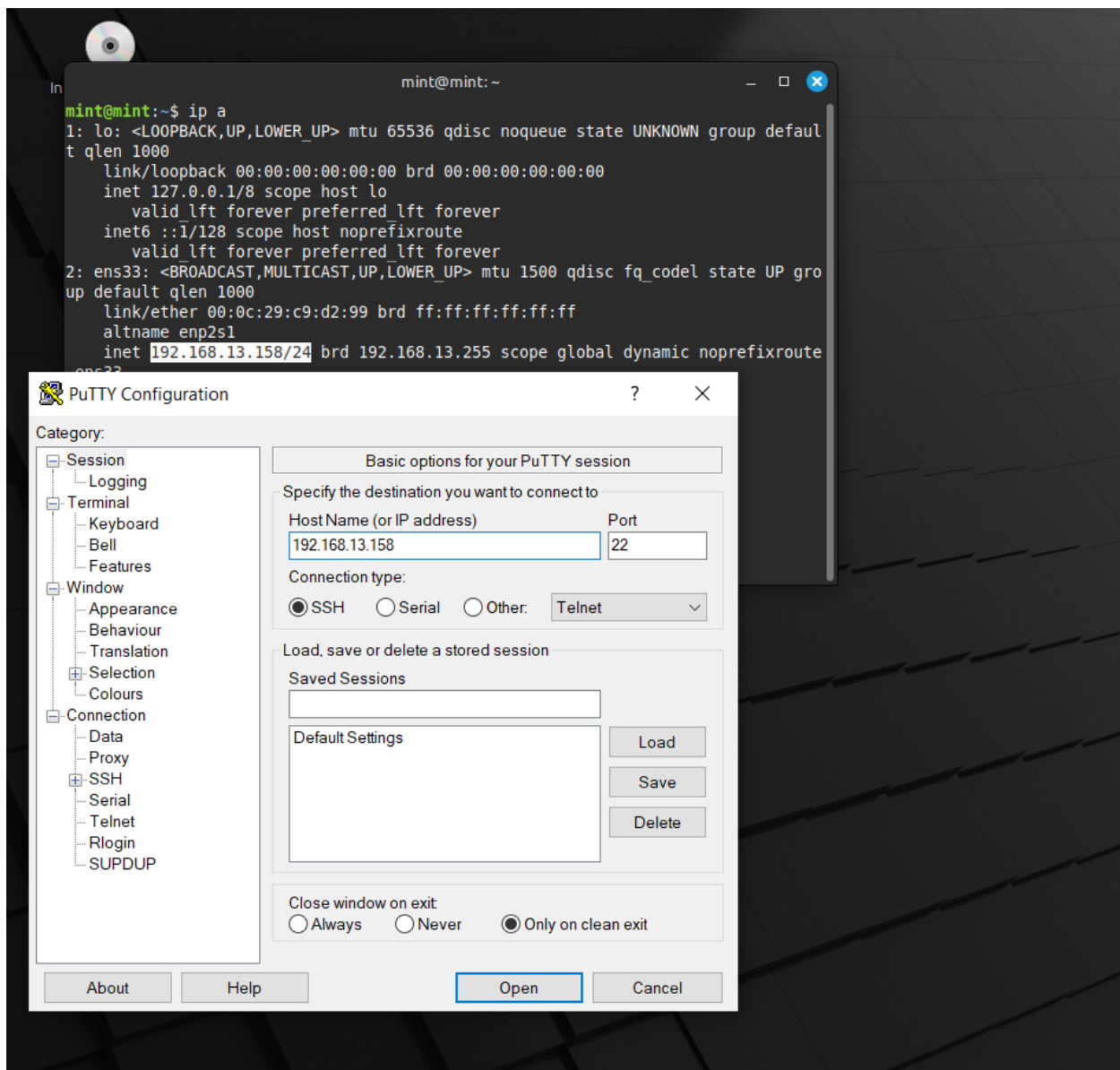
```
mint@mint:~$ ssh -vvv omar221010339@192.168.13.158
OpenSSH 9.6p1 Ubuntu-3ubuntu13.5, OpenSSL 3.0.13 30 Jan 2024
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: /etc/ssh/ssh_config line 19: include /etc/ssh/ssh_config.d/*.conf matched no files
debug1: /etc/ssh/ssh_config line 21: Applying options for *
debug2: resolve_canonicalize: hostname 192.168.13.158 is address
debug3: expanded UserKnownHostsFile ~/.ssh/known_hosts -> '/home/mint/.ssh/known_hosts'
debug3: expanded UserKnownHostsFile ~/.ssh/known_hosts2 -> '/home/mint/.ssh/known_hosts2'
debug3: channel_clear_timeouts: clearing
debug3: ssh_connect_direct: entering
debug1: Connecting to 192.168.13.158 [192.168.13.158] port 22.
debug3: set_sock_tos: set socket 3 IP_TOS 0x10
debug1: Connection established.
debug1: Identity file /home/mint/.ssh/id_rsa type -1
debug1: Identity file /home/mint/.ssh/id_rsa-cert type -1
debug1: Identity file /home/mint/.ssh/id_ecdsa type -1
debug1: Identity file /home/mint/.ssh/id_ecdsa-cert type -1
debug1: Identity file /home/mint/.ssh/id_ecdsa-sk type -1
debug1: Identity file /home/mint/.ssh/id_ecdsa-sk-cert type -1
debug1: Identity file /home/mint/.ssh/id_ed25519 type 3
debug1: Identity file /home/mint/.ssh/id_ed25519-cert type -1
debug1: Identity file /home/mint/.ssh/id_ed25519-sk type -1
debug1: Identity file /home/mint/.ssh/id_ed25519-sk-cert type -1
debug1: Identity file /home/mint/.ssh/id_xmss type -1
debug1: Identity file /home/mint/.ssh/id_xmss-cert type -1
debug1: Identity file /home/mint/.ssh/id_dsa type -1
debug1: Identity file /home/mint/.ssh/id_dsa-cert type -1
debug1: Local version string SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.5
debug1: Remote protocol version 2.0, remote software version OpenSSH_9.6p1 Ubuntu-3ubuntu13.11
debug1: compat_banner: match: OpenSSH_9.6p1 Ubuntu-3ubuntu13.11 pat OpenSSH* compat 0x04000000
debug2: fd 3 setting O_NONBLOCK
debug1: Authenticating to 192.168.13.158:22 as 'omar221010339'
debug3: record_hostkey: found key type ED25519 in file /home/mint/.ssh/known_hosts:1
debug3: record_hostkey: found key type RSA in file /home/mint/.ssh/known_hosts:2
debug3: record_hostkey: found key type ECDSA in file /home/mint/.ssh/known_hosts:3
debug3: load_hostkeys: file: loaded 3 keys from 192.168.13.158
debug1: load_hostkeys: fopen /home/mint/.ssh/known_hosts2: No such file or directory
debug1: load_hostkeys: fopen /etc/ssh/ssh_known_hosts: No such file or directory
debug1: load_hostkeys: fopen /etc/ssh/ssh_known_hosts2: No such file or directory
debug3: order_hostkeyalgs: have matching best-preference key type ssh-ed25519-cert-v01@openssh.com, using HostKeyAlgorithms verbatim
debug3: send packet: type 20
debug1: SSH2_MSG_KEXINIT sent
debug3: receive packet: type 20
```

```
omar221010339@mint: ~
debug3: Ignored env DBUS_SESSION_BUS_ADDRESS
debug3: Ignored env _
debug2: channel 0: request shell confirm 1
debug3: send packet: type 98
debug3: client_repledge: enter
debug1: pledge: fork
debug2: channel_input open_confirmation: channel 0: callback done
debug2: channel 0: open confirm rwindow 0 rmax 32768
debug3: receive packet: type 99
debug2: channel_input_status_confirm: type 99 id 0
debug2: PTY allocation request accepted on channel 0
debug2: channel 0: rcvd adjust 2097152
debug3: receive packet: type 99
debug2: channel_input_status_confirm: type 99 id 0
debug2: shell request accepted on channel 0

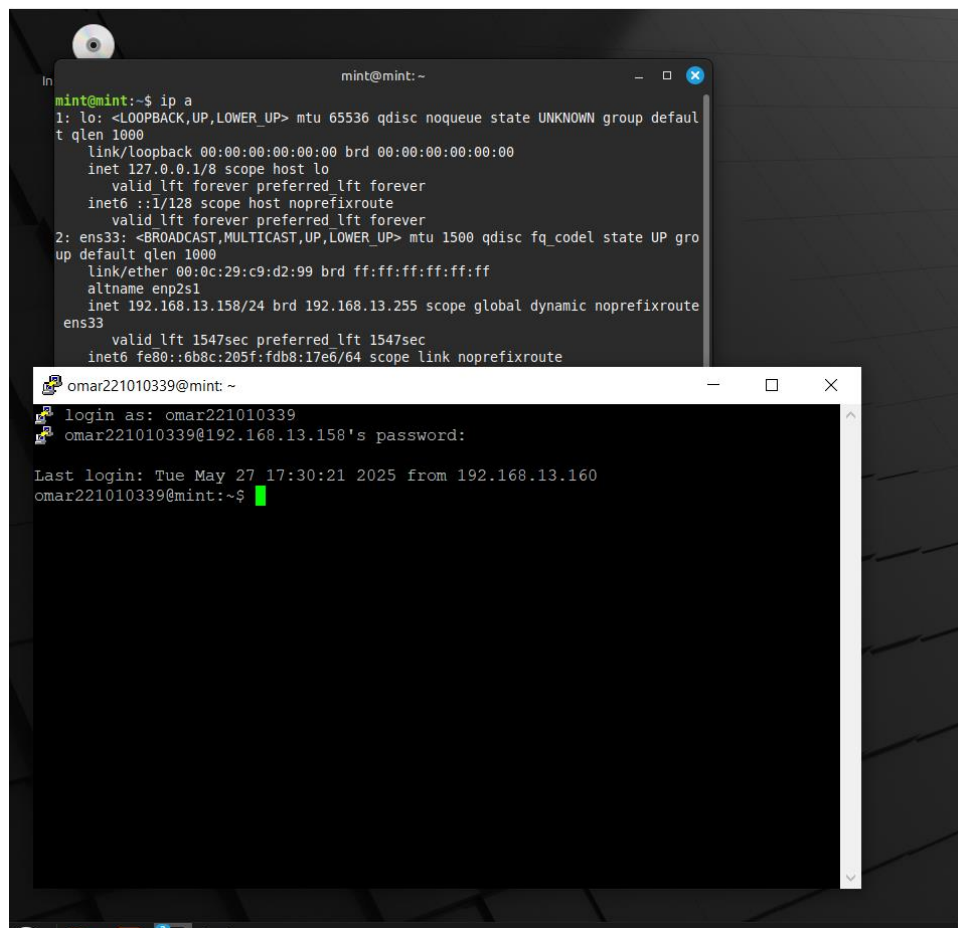
Last login: Tue May 27 16:30:15 2025 from 192.168.13.160
omar221010339@mint:~$ debug2: client_check_window change: changed
debug2: channel 0: request window-change confirm 0
debug3: send packet: type 98
omar221010339@mint:~$ debug2: client_check_window change: changed
debug2: channel 0: request window-change confirm 0
debug3: send packet: type 98
omar221010339@mint:~$
```

Putty configuration

First download it in windows after that know ip of server by use “ip a” then write ip in putty and port 22 and make connection type ssh then press open



After press open we found login as we write client name and password then we will be client



COMMANDS USE IN PART 2:

1-sudo apt update
2- sudo apt install openssh-server openssh-client
3-sudo systemctl enable ssh
4-sudo systemctl start ssh
5-sudo adduser omar221010339
6-ssh-keygen -t ed25519 -C "omar221010339@client"
7-ssh-copy-id omar221010339@<server-ip>
8-ssh omar221010339@<server-ip>
9-sudo nano /etc/ssh/sshd_config
10-sudo systemctl restart ssh
11-ssh -v omar221010339@<server-ip>

Server IP: <192.168.13.158>

Client IP: <192.168.13.160>

3. Conclusion

The SSH key-based login was successfully configured and tested. Compared to password-based authentication, key-pair login provides stronger security and avoids brute-force attacks. All steps were implemented and verified through terminal output and screenshots.

