Experiment No. 2
setup SSH passwordless logins for two or more Linux based machines and execute commands on a remote machine.
Date of Performance:18/01/2024
Date of Submission:18/04/2024

Aim: To set up SSH passwordless logins for two or more Linux based machines and execute commands on a remote machine.

Objective: To execute SSH passwordless logins for Linux based machines

Theory:

SSH (Secure Shell) is a commonly used protocol for securely connecting to and managing remote systems. SSH passwordless logins allow users to access a remote system without having to enter a password each time they connect. This is particularly useful when managing multiple systems, as it can save time and simplify the login process. Here's how to set up SSH passwordless logins for two or more Linux based machines:

Generate SSH keys: The first step is to generate SSH keys on the machine that will be connecting to the remote system. This is done using the 'ssh-keygen' command. The command will create a public key and a private key. The public key will be copied to the remote system.

Copy the public key: The public key needs to be copied to the remote system that the user wants to log in to without a password. This is done using the 'ssh-copy-id' command. The command will prompt the user for their password on the remote system.

Test the connection: Once the public key has been copied to the remote system, the user can test the connection by using the 'ssh' command. If the connection is successful, the user will be logged in to the remote system without having to enter a password.

Repeat steps for additional machines: To set up passwordless logins for additional machines, the user must repeat the same steps for each machine

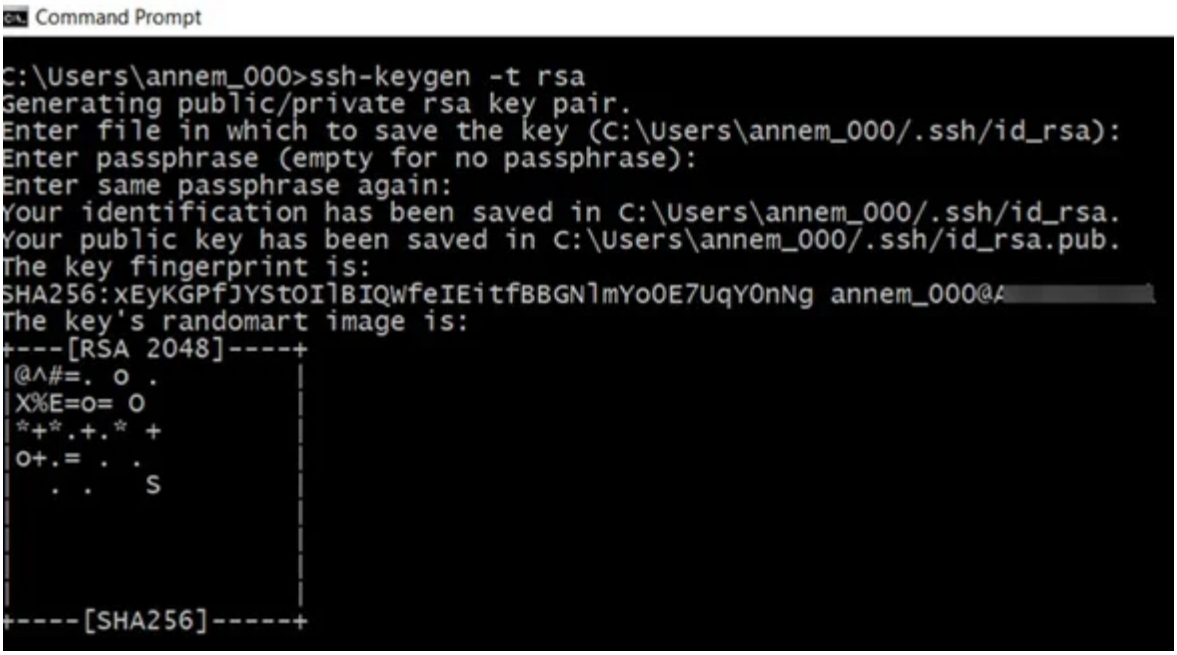
OUTPUT:

```
ssh-keygen -t rsa
```

```
Enter file in which to save the key.
(C:\Users\annem_000\.ssh\id_rsa):
```

```
Enter Passphrase (empty for no passphrase):
```

```
ssh-copy-id annem_000@197.168.100.114
```



Conclusion: SSH passwordless logins offer a convenient and secure way to access remote systems without the need for constantly typing passwords. By generating SSH keys and exchanging public keys between systems, users can authenticate seamlessly, enhancing user experience and workflow efficiency. This method not only reduces the risk of password-based attacks but also streamlines administrative tasks, particularly in automated scripts and batch processes. However, it's crucial to safeguard private keys and implement proper key management practices to maintain the integrity and security of the SSH infrastructure..