Implementing a Secure SSH Server in Linux

Scenario

The Secure Shell (SSH) protocol is a common protocol for establishing remote connections to Linux, Unix, and other Unix-like systems. SSH functionality can be added to Windows, too. In this activity, you will configure several security settings on a remote destination SSH server named LX1. You'll be testing connectivity from the Kali VM.

Objectives

This activity is designed to test your understanding of and ability to apply content examples in the following CompTIA Security+ objectives:

- 3.3 Given a scenario, implement secure network designs.
- 4.1 Given a scenario, use the appropriate tool to assess organizational security.

Lab

- Kali VM
- LX1 VM
- pfSense VM

Task 1

Remotely connect by using SSH with basic password authentication

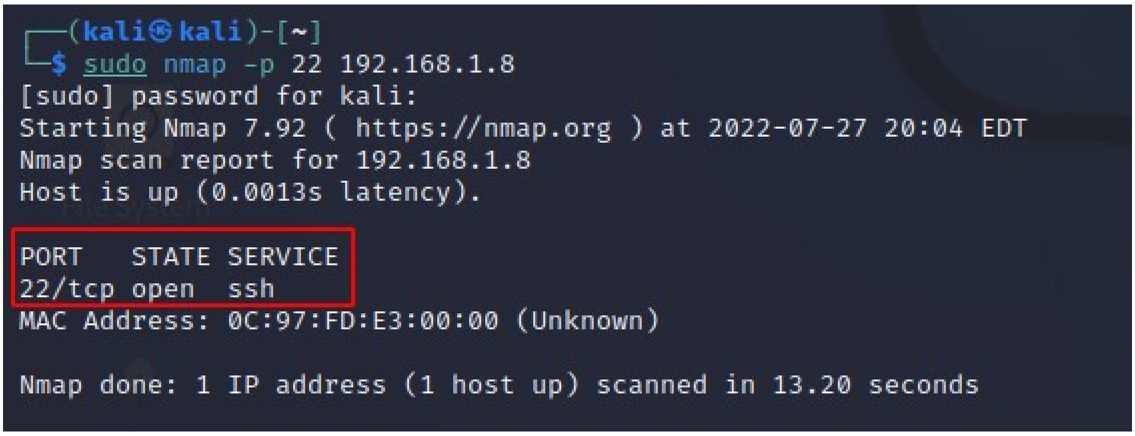
In this activity, the Kali VM will remotely connect to the LX1 VM by using the Secure Shell (SSH). In this case, the Kali VM is the SSH client, and the LX1 VM is the SSH server.

1. Log on to the **Kali** VM with the **kali** username and the **Pa$$w0rd** password.

2. From the top bar, select the **Terminal** Emulator icon.

3. Run the following command to check the status of port 22 on the remote LX1 SSH server:

```
sudo nmap -p 22 192.168.1.8
```



```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -p 22 192.168.1.8
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-27 20:04 EDT
Nmap scan report for 192.168.1.8
Host is up (0.0013s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 0C:97:FD:E3:00:00 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 13.20 seconds
```

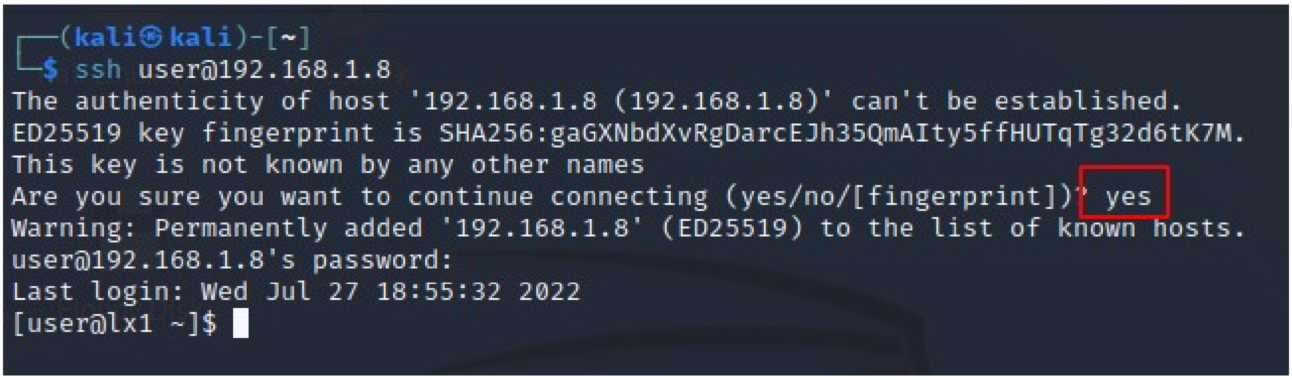
Figure 1.1 – Status of the SSH service on LX1 VM.

NOTE: The remote LX1 server is running the CentOS 7 Linux distribution. This distribution is derived from Red Hat Enterprise Linux (RHEL).

4. Run the following command to test SSH connectivity to the CentOS SSH server:

```
ssh user@192.168.1.8
```

5. Enter **yes** if prompted to confirm the connection. This is the host key, which validates the identity of the server.



```
┌──(kali㉿kali)-[~]
└─$ ssh user@192.168.1.8
The authenticity of host '192.168.1.8 (192.168.1.8)' can't be established.
ED25519 key fingerprint is SHA256:gaGXNbdXvRgDarcEJh35QmAIty5ffHUTqTg32d6tK7M.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.8' (ED25519) to the list of known hosts.
user@192.168.1.8's password:
Last login: Wed Jul 27 18:55:32 2022
[user@lx1 ~]$ █
```

Figure 1.2 – Connecting to LX1 via SSH.

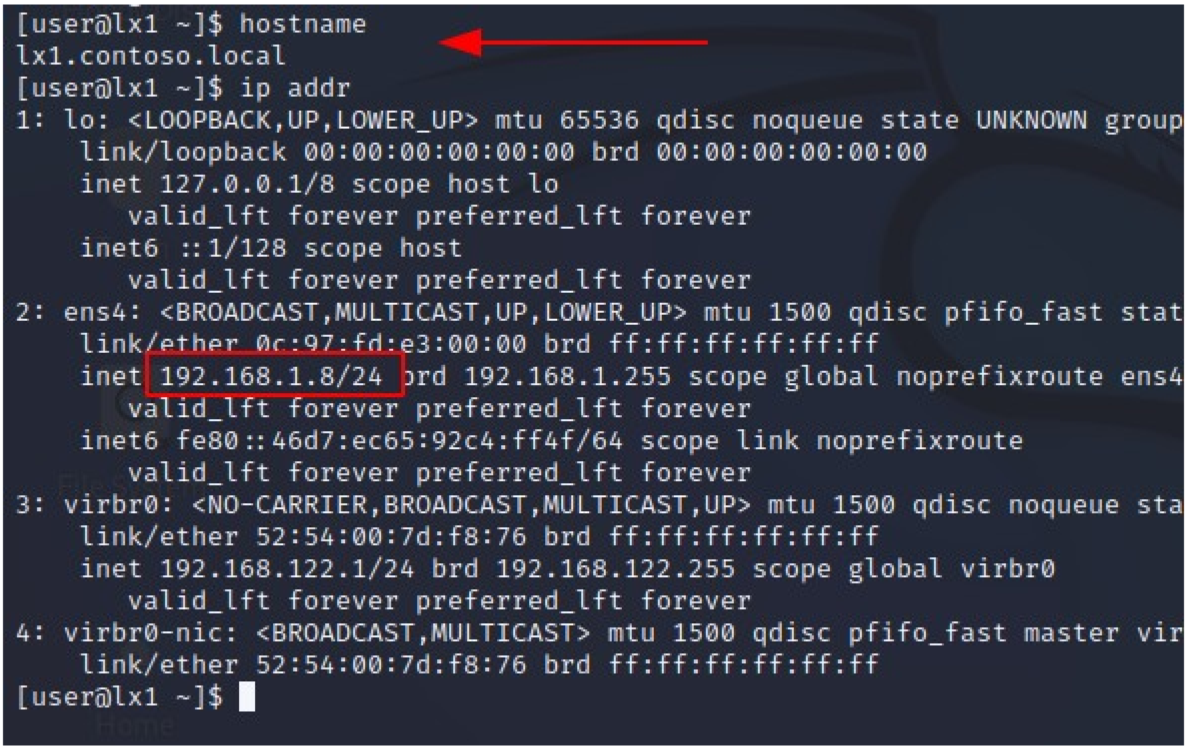
6. Enter **Pa$$w0rd** when prompted for the password. This is the password for a user account with local logon that is authenticated by the SSH server.

NOTE: Most Linux systems are configured to permit SSH connections through the firewall by default.

7. Run the following commands to verify that you are connected to the remote SSH server:

```
hostname
```

```
ip addr
```



```
[user@lx1 ~]$ hostname
lx1.contoso.local
[user@lx1 ~]$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast stat
    link/ether 0c:97:fd:e3:00:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.8/24 brd 192.168.1.255 scope global noprefixroute ens4
       valid_lft forever preferred_lft forever
    inet6 fe80::46d7:ec65:92c4:ff4f/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue sta
    link/ether 52:54:00:7d:f8:76 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
       valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master vir
    link/ether 52:54:00:7d:f8:76 brd ff:ff:ff:ff:ff:ff
[user@lx1 ~]$ 
```

Figure 1.3 – Displaying hostname information and IP address of LX1 server.

You should see a hostname of LX1 and an IP address of 192.168.1.8. Many Linux distributions also display the current user and hostname of the server in the command prompt. Observe how the prompt currently displays the **user@lx1**.

NOTE: The loopback address (127.0.0.1) is configured on each TCP/IP host, and the virbr0 configuration represents and additional network adapters installed on the virtual machine.

8. Run the following command to prove you have administrative privileges on the remote server:

```
sudo systemctl restart rsyslog
```

9. Use the **touch** command to create a new file named **kali-ssh-test** in the root user’s home directory on the remote CentOS server.

```
touch kali-ssh-test
```

10. Type **exit** to disconnect from the remote CentOS server.

Task 2

Configure the CentOS SSH server

You will configure increased security settings for the SSH server.

1. Switch to the **LX1** SSH server.

2. The VM’s privacy screen is probably enabled. Click anywhere on the screen with the mouse and press **ENTER**. A login prompt should appear.

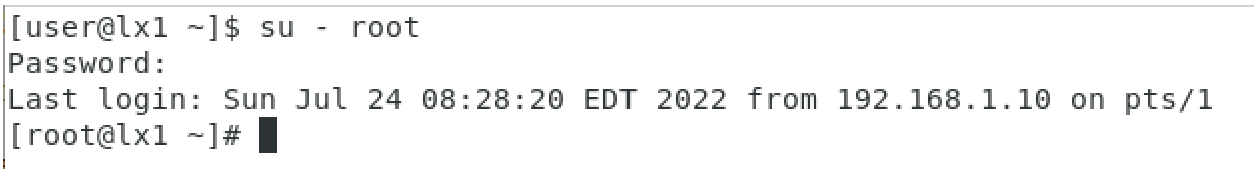
3. Sign in with the pre-configured **User** account and a password of **Pa$$w0rd**.

4. Right-click the desktop, and select **Open Terminal**.

5. Run the following command to elevate your credentials to root:

```
su - root
```

6. Type **Pa$$w0rd** when prompted.



```
[user@lx1 ~]$ su - root
Password:
Last login: Sun Jul 24 08:28:20 EDT 2022 from 192.168.1.10 on pts/1
[root@lx1 ~]#
```

Figure 2.1 – Changing user account to root.

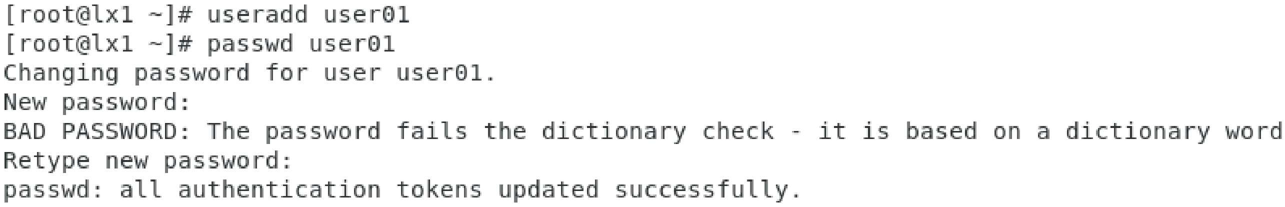
TIP: Note that there is a space on each side of the dash.

7. Run the following commands to create a new user and set a password:

```
useradd user01
```

```
passwd user01
```

8. Set the password as **Pa$$w0rd** when prompted. You may **ignore any warnings** about the password quality.



```
[root@lx1 ~]# useradd user01
[root@lx1 ~]# passwd user01
Changing password for user user01.
New password:
BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word
Retype new password:
passwd: all authentication tokens updated successfully.
```

Figure 2.2 – Creating a user account and updating the password.

TIP: Recall that Linux does not print any characters to the display when you enter a password. It may appear as though the password is not receiving your keystrokes, but it is.

9. Run the following command to backup the current **/etc/ssh/sshd_config** configuration file:

```
cp /etc/ssh/sshd_config ~/sshd_config_old
```

NOTE: It is a good practice to back up a configuration file before making changes to it so that you can easily roll back to a known configuration.

10. Run the following command to open the SSH configuration file with the Vim text editor:

```
vim /etc/ssh/sshd_config
```

11. Review the following list of the required SSH configurations defined by your company’s written security policy:

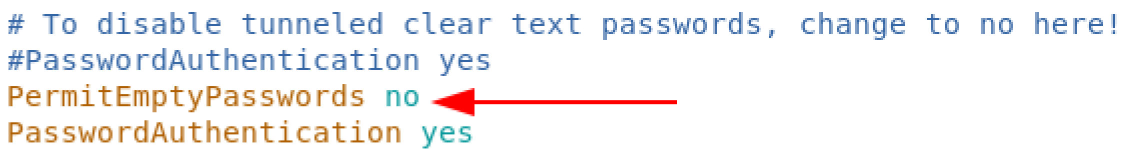
- Empty passwords are not allowed for SSH authentication. **Disable the empty password** option.
- Idle SSH connections should be disconnected after **five minutes** (300 seconds). The system should check twice before disconnecting.
- The following banner message should be configured: “**Warning! Authorized use only!**”. The message is set in the **/etc/issue.net** file and called with the **/etc/ssh/sshd_config** file.

12. In the text editor, edit the **/etc/ssh/sshd_config** file to match the required settings. You will need to uncomment some lines and then edit them. Other lines you may simply edit.

TIP: The # hash character identifies comment lines. Comments are ignored by the system and used to provide administrators with examples, explanations, and additional information within the body of a configuration file or script.

Here are the specific lines to edit:

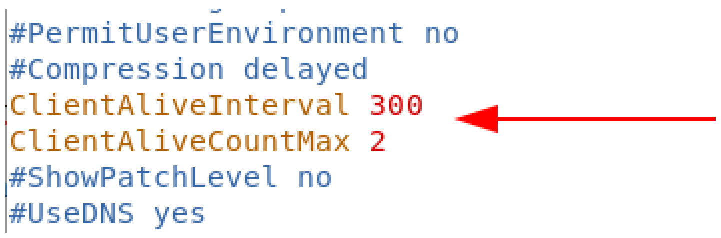
PermitEmptyPasswords no



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

Figure 2.3 – Uncommenting and changing values.

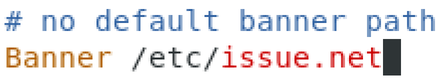
ClientAliveInterval 300
ClientAliveCountMax 2



```
#PermitUserEnvironment no
#Compression delayed
ClientAliveInterval 300
ClientAliveCountMax 2
#ShowPatchLevel no
#UseDNS yes
```

Figure 2.4 – Uncommenting and changing values.

Banner /etc/issue.net



```
# no default banner path
Banner /etc/issue.net
```

Figure 2.5 – Uncommenting and changing values.

13. In Vim, save your changes by pressing **ESC**, and then typing **:wq**.

14. Now, use Vim to edit the **/etc/issue.net** file. Remove all of the existing content in the file, and then add the following warning:

```
Warning!

Authorized use only!
```

15. To save changes and quit Vim, press **ESC**, and then type **:wq**.

16. In the command prompt window, run the following command to restart SSH service:

```
systemctl restart sshd
```

NOTE: Recall that each time you change a configuration file, you must restart the service for the change to take effect.

Task 3

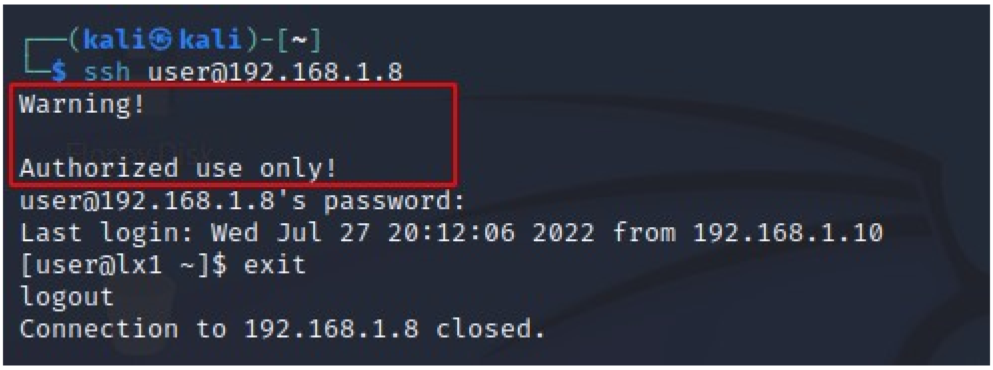
Test the new security configurations

You will connect to the LX1 SSH server from the Kali VM to test the security configurations you implemented above.

1. Switch to the **Kali** SSH client VM.

2. Attempt the SSH connection by using the **user01** credentials. Use **Pa$$w0rd** as the password.

Do you see the warning banner?



```
┌──(kali㉿kali)-[~]
└─$ ssh user@192.168.1.8
Warning!

Authorized use only!
user@192.168.1.8's password:
Last login: Wed Jul 27 20:12:06 2022 from 192.168.1.10
[user@lx1 ~]$ exit
logout
Connection to 192.168.1.8 closed.
```

Figure 3.1 – SSH connection displaying the warning banner.

3. Type **exit** to disconnect from the remote **LX1** SSH server.

Task 4

SSH Key-based authentication

Configure key-based authentication for the SSH server. Key-based authentication is more secure and it alleviates the need to remember passwords. A private key is stored on the client computer and a related public key is copied to the remote SSH server. During the authentication process, the two keys are compared instead of a password being submitted.

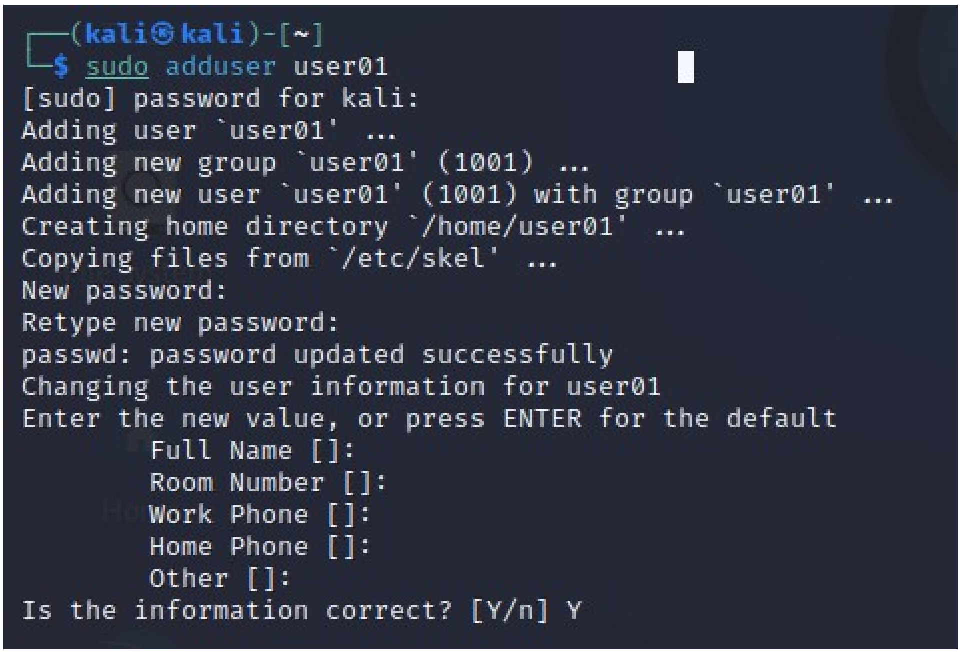
1. On the **Kali** VM, confirm that the command prompt is **kali@kali** to ensure that you do not currently have an SSH connection to the remote **LX1** SSH server.

TIP: Type **exit** to disconnect an SSH session.

2. Run the following commands to create a new user named user01 with Pa$$w0rd as the password.

```
sudo adduser user01
```

3. Type **Pa$$w0rd** twice when prompted. Press **ENTER** for all other user account options.



```
┌──(kali㉿kali)-[~]
└─$ sudo adduser user01
[sudo] password for kali:
Adding user `user01' ...
Adding new group `user01' (1001) ...
Adding new user `user01' (1001) with group `user01' ...
Creating home directory `/home/user01' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for user01
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
```

Figure 4.1 – Adding a new user to Kali VM.

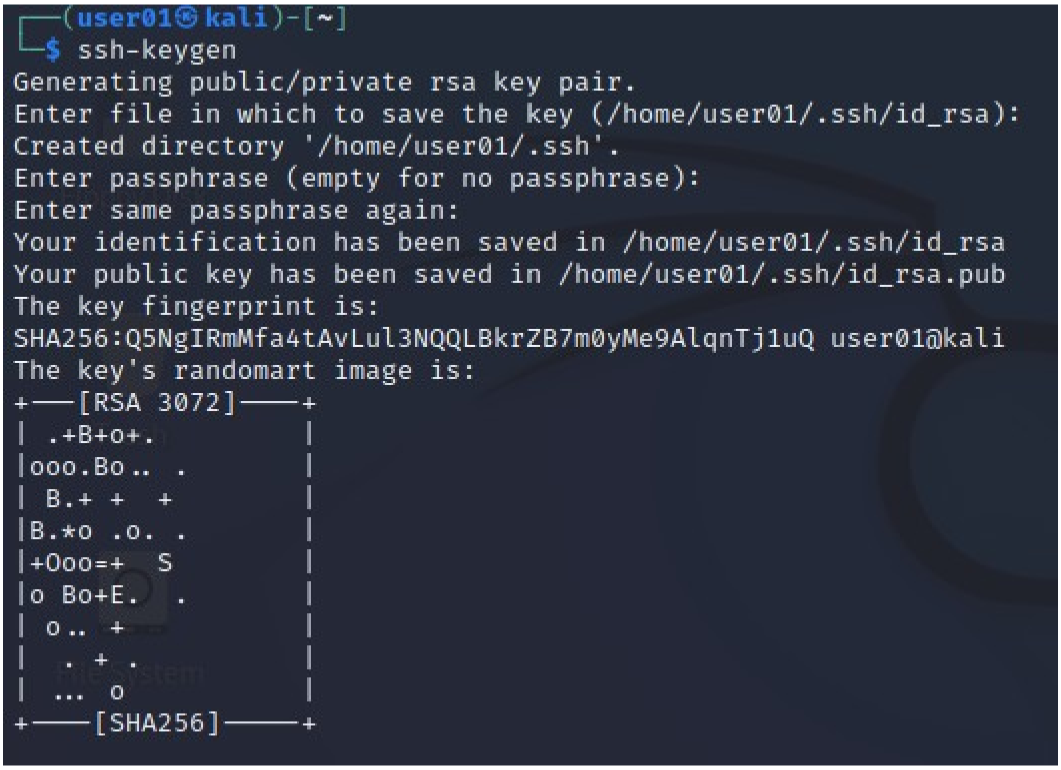
4. Run the following command to switch to the new user01 user:

su - user01

5. Run the following command to generate an SSH key pair that identifies **user01**:

```
ssh-keygen
```

6. Press **ENTER** three times to accept the default settings.



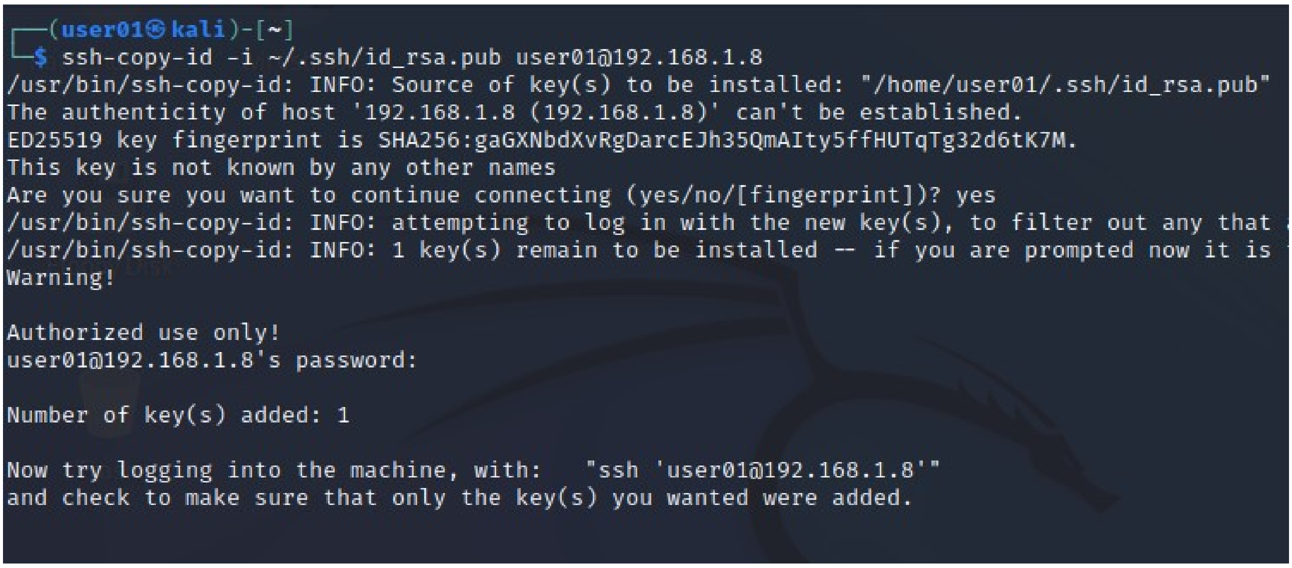
```
┌──(user01㉿kali)-[~]
└─$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user01/.ssh/id_rsa):
Created directory '/home/user01/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user01/.ssh/id_rsa
Your public key has been saved in /home/user01/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:Q5NgIRmMfa4tAvLul3NQQLBkrZB7m0yMe9AlqnTj1uQ user01@kali
The key's randomart image is:
+---[RSA 3072]----+
| .+B+o+.         |
|ooo.Bo.. .       |
| B.+ +  +        |
|B.*o .o. .       |
|+Ooo=+  S        |
|o Bo+E.  .       |
| o.. +           |
|  . + .          |
|  ... o          |
+----[SHA256]-----+
```

Figure 4.2 – Generating a SSH hey pair.

7. Run the following command to copy the public hey to the SSH server.

```
ssh-copy-id -i ~/.ssh/id_rsa.pub user01@192.168.1.8
```

8. Enter **yes** when prompted. Enter **Pa$$w0rd** when prompted.

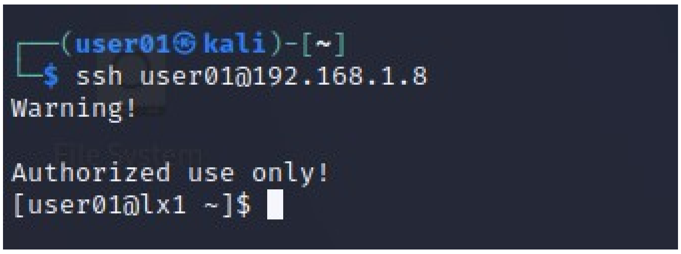


```
┌──(user01㉿kali)-[~]
└─$ ssh-copy-id -i ~/.ssh/id_rsa.pub user01@192.168.1.8
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/user01/.ssh/id_rsa.pub"
The authenticity of host '192.168.1.8 (192.168.1.8)' can't be established.
ED25519 key fingerprint is SHA256:gaGXNbdXvRgDarcEJh35QmAIty5ffHUTqTg32d6tK7M.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is
Warning!

Authorized use only!
user01@192.168.1.8's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'user01@192.168.1.8'"
and check to make sure that only the key(s) you wanted were added.
```

Figure 4.3 – Copying the public ssh key to the remote SSH server.

9. Run the following command to connect to the SSH server and confirm that key-based authentication works:

```
ssh user01@192.168.1.8
```



```
┌──(user01㉿kali)-[~]
└─$ ssh user01@192.168.1.8
Warning!

Authorized use only!
[user01@lx1 ~]$
```

Figure 4.4 – SSH key-based authentication.

10. Type **exit** to terminate the SSH connection.

Task 5

Review the SSH log file

You will display SSH log file entries on the LX1 SSH server to understand which users are remotely connecting to your server.

1. Switch to the **LX1** VM. If the VM's privacy screen is enabled, press **ENTER**. Sign in as the pre-configured user with a password of **Pa$$w0rd**.

2. If a terminal window is not already open, right-click on the desktop, and select **Open Terminal**.

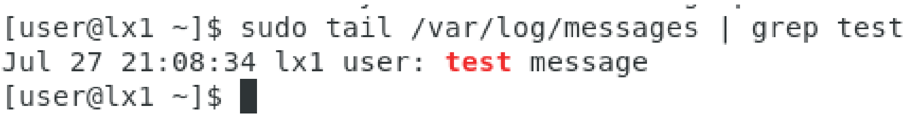
3. Run the following command to generate an entry in the **/var/log/messages** log file:

```
logger “test message”
```

4. Run the following commands to view the lines at the bottom of the **/var/log/secure.log** file:

```
sudo tail /var/log/messages | grep test
```

You should see the “**test message**” string.

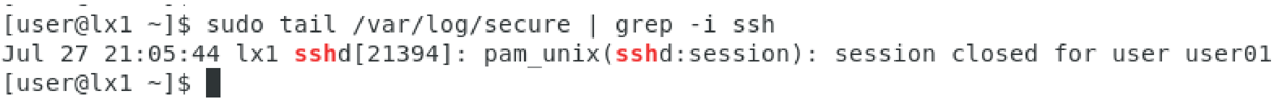


```
[user@lx1 ~]$ sudo tail /var/log/messages | grep test
Jul 27 21:08:34 lx1 user: test message
[user@lx1 ~]$
```

Figure 5.1 – Output from the messages file showing the logger “test message”

5. Run the following command to view SSH events in the log file:

```
sudo tail /var/log/secure | grep -i ssh-copy
```



```
[user@lx1 ~]$ sudo tail /var/log/secure | grep -i ssh
Jul 27 21:05:44 lx1 sshd[21394]: pam_unix(sshd:session): session closed for user user01
[user@lx1 ~]$
```

Figure 5.2 – Inspecting the secure log file.

NOTE: The most recent log entries are written at the bottom of the files, so **tail** is a great tool to use to see the most current information. The **head** command displays the entries at the top of the file.

Task 6

Prevent root from authenticating over SSH

The final settings you will implement prevent the root user from accessing the server remotely via SSH and require key-based authentication only.

1. Review the following list of the required SSH configurations defined by your company's written security policy.:

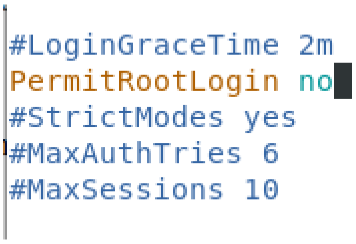
- Root user accounts should not be able to authenticate via SSH. **Disable root user access** over SSH.
- Key-based authentication is required and password authentication should be explicitly denied. **Disable passwords** over SSH.

2. On the **LX1** VM, use the Vim text editor to open **/etc/ssh/sshd_config** file.

```
sudo vim /etc/ssh/sshd_config
```

3. Uncomment the following line by deleting the **#** character, and then change the entry from "**yes**" to "**no**" to prevent the root user from signing via SSH:

```
PermitRootLogin no
```

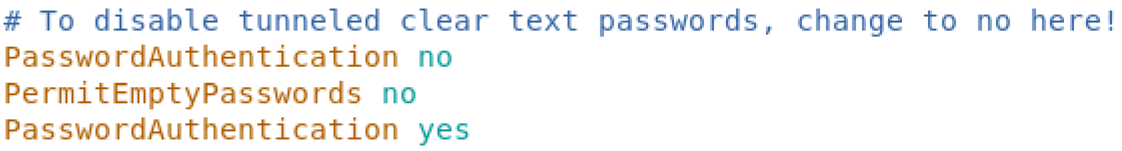


```
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

Figure 6.1 – Uncommenting and changing values.

4. Change the entry on the following line from "**yes**" to "**no**" to prevent password-based authentication:

```
PasswordAuthentication no
```



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

Figure 6.2 – Uncommenting and changing values.

TIP: The SSH connection may be established with a standard user account, such as user01. Once the connection is made and you are "on" the remote server, you may use the sudo su – root command to elevate your privileges, if necessary.

5. Select **ESC**, and then type **:wq** to save your changes and exit Vim.

6. Run the following command to restart the SSH service, causing it to re-read the configuration file and apply the new settings:

```
sudo systemctl restart sshd
```

7. Switch to the **Kali** VM.

8. Attempt the SSH connection by using the **root** credentials.

This attempt fails, because the account is root and signs in by using password based authentication.

9. Attempt the SSH connection by using the **user01** credentials.

This attempt succeeds, because the account is not root and signs in by using key-based authentication.