

SSH Server Hardening

Objectives The objective of this lab is to practice hardening an SSH server, using key authentication, and observe the benefits of the security methods used in the SSH configurations.

Background SSH is a network protocol that provides a secure way to access remote machines over an unsecured network. It was made as a replacement for Telnet and other insecure remote login protocols, it encrypts all traffic to prevent someone seeing data in the connection, and other attacks. Hardening an SSH server is necessary for network defense because it reduces attack risk and strengthens overall system security.

Procedures

First what i did was confirm the IP addresses of the server machine and the client machine using the “ip a” command. Which was:

1. Server IP: **192.168.192.132**
2. Client IP: **192.168.192.133**

I used “ping 192.168.192.132” from the client to test connectivity which resulted in a successful connection

I then used “ssh midterm@192.168.192.132” from the client. To connect to the server which then prompted me to accept the server’s key fingerprint and enter a password. The initial key provided is from the server. **Figure 1.**

```
midterm@Client:~$ ssh midterm@192.168.192.132
The authenticity of host '192.168.192.132 (192.168.192.132)' can't be established.
ECDSA key fingerprint is SHA256:aBq7eYI0d9WYmVs+G8sG+Ze84d4EwTD5JbFEw8UtYiQ.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.192.132' (ECDSA) to the list of known hosts.
midterm@192.168.192.132's password:
Welcome to Ubuntu 18.10 (GNU/Linux 4.18.0-25-generic x86_64)

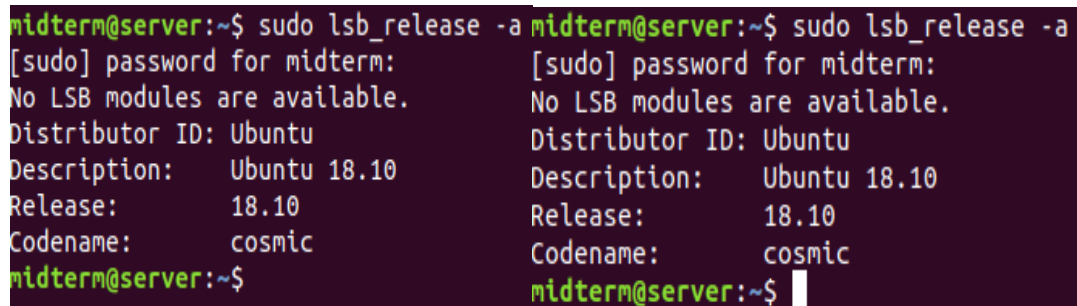
 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.

Last login: Mon Oct 14 20:07:00 2019 from 192.168.119.132
midterm@server:~$
```

Figure 1: Server Connection

I then verified the OS versions of each of the machines using the command “sudo lsb_release -a” as seen in **Figure 2**



```
midterm@server:~$ sudo lsb_release -a
[sudo] password for midterm:
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 18.10
Release:        18.10
Codename:       cosmic
midterm@server:~$

midterm@server:~$ sudo lsb_release -a
[sudo] password for midterm:
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 18.10
Release:        18.10
Codename:       cosmic
midterm@server:~$
```

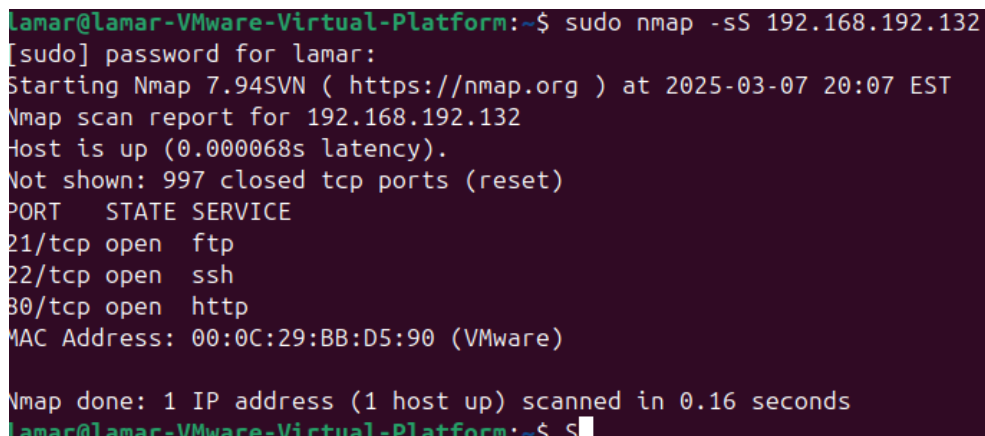
Figure 2: OS Versions

Some Hardening Techniques include:

- 1- Disabled root login.
- 2- key-based authentication.
- 3- Changed SSH listening port.
- 4- Disabled unused authentication methods.
- 5- Limited user access to SSH.
- 6- Configured firewall rules to restrict SSH access.

Screenshot: [Insert Screenshot Here]

Then using “nmap -sS 192.168.192.132” i scanned for any open ports and services as seen in **Figure 3** there are 3 ports and services running.



```
lamar@lamar-VMware-Virtual-Platform:~$ sudo nmap -sS 192.168.192.132
[sudo] password for lamar:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-07 20:07 EST
Nmap scan report for 192.168.192.132
Host is up (0.000068s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 00:0C:29:BB:D5:90 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.16 seconds
lamar@lamar-VMware-Virtual-Platform:~$
```

Figure 3: Ports Running

From the scan it shows Port 21, 22 and 80 are up and running. Next what i did was removed any unnecessary services with the command:

```
sudo apt-get purge vsftpd apache2 telnet.
```

```
midterm@server:~$ sudo apt-get purge apache2 vsftpd telnet
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0 libncursesw5 libtinfo5
Use 'sudo apt autoremove' to remove them.
The following packages will be REMOVED:
  apache2* telnet* vsftpd*
0 upgraded, 0 newly installed, 3 to remove and 0 not upgraded.
After this operation, 1,038 kB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 169006 files and directories currently installed.)
Removing apache2 (2.4.34-1ubuntu2.3) ...
Removing telnet (0.17-41) ...
Removing vsftpd (3.0.3-11) ...
Processing triggers for ufw (0.36-0ubuntu0.18.10.1) ...
Processing triggers for man-db (2.8.4-2) ...
(Reading database ... 168893 files and directories currently installed.)
Purging configuration files for apache2 (2.4.34-1ubuntu2.3) ...
dpkg: warning: while removing apache2, directory '/var/www/html' not empty so not removed
Purging configuration files for vsftpd (3.0.3-11) ...
Purging configuration files for telnet (0.17-41) ...
Processing triggers for ufw (0.36-0ubuntu0.18.10.1) ...
Processing triggers for systemd (239-7ubuntu10.14) ...
midterm@server:~$
```

Figure 4: Uninstall of Services

Then I verified the removal of them as seen in **Figure 5** with command

```
dpkg --get-selections | grep
```

```
midterm@server:~$ dpkg --get-selections | grep 'apache|vsftpd|telnet'
ii  apache2-bin      2.4.34-1ubuntu2.3      amd64      Apache HTTP Server (modules and other binary files)
ii  apache2-data     2.4.34-1ubuntu2.3      all        Apache HTTP Server (common files)
ii  apache2-utils    2.4.34-1ubuntu2.3      amd64      Apache HTTP Server (utility programs for web servers)
midterm@server:~$
```

Figure 5: Verifying Uninstall

I then updated the system to harden the security of the OS with the command and removed the

```
sudo apt-get update
```

In order to harden the server security i started with creating a backup with “sudo cp /etc/ssh/sshd_config sshd_config.backup.”. I then created a private key on the client machine (**Figure 6**) and it created a randomart image for the machine.

```

midterm@server:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/midterm/.ssh/id_rsa): abc123
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in abc123.
Your public key has been saved in abc123.pub.
The key fingerprint is:
SHA256:Z3+0d0at66QEIMCVn3bjv99lxT3hu2N9954fiC254+Q midterm@ubuntu
The key's randomart image is:
+---[RSA 2048]---+
|      . . . . .      |
|      . . .          |
|      . . . . .      |
|      .+.o   ..o     |
|      .So+.   oo*     |
|      o.o+o.==       |
|      *oo+o*         |
|      oo++ *o        |
|      .E++*=@        |
+---[SHA256]-----+
midterm@server:~$

```

Figure 6: Private Key

What i did next was share the private key with the server machine using the command “ssh-copy-id [midterm@192.168.192.132](#)” (**Figure 7**) then i logged in the server on the client machine and instead of asking for a passphrase for the server it asked for my private key instead (**Figure 8**).

```

midterm@server:~$ ssh-copy-id midterm@192.168.192.132
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/midterm/.ssh/id_rsa.pub"
The authenticity of host '192.168.192.132 (192.168.192.132)' can't be established.
ECDSA key fingerprint is SHA256:a8q7eYI0d9WyMvS+G8sG+Ze84d4EwTD5JbFEw8UtiQ.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
midterm@192.168.192.132's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'midterm@192.168.192.132'"
and check to make sure that only the key(s) you wanted were added.

```

Figure 7: Copy Private Key

```
midterm@server:~$ ssh midterm@192.168.192.132
Enter passphrase for key '/home/midterm/.ssh/id_rsa':
Welcome to Ubuntu 18.10 (GNU/Linux 4.18.0-25-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.

Your Ubuntu release is not supported anymore.
For upgrade information, please visit:
http://www.ubuntu.com/releaseendoflife

New release '20.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Mar  7 16:45:50 2025 from 192.168.192.131
```

Figure 8: Logging Into Server

I then went into the `/etc/ssh/sshd_config` disabling password authentication by setting `PasswordAuthentication` to `no` as seen in . I restarted the `ssh` service to apply the changes and then relogged in with no password for the server prompted other than the private key passphrase.

```
midterm@server:~$ ssh midterm@192.168.192.132
Enter passphrase for key '/home/midterm/.ssh/id_rsa':
Welcome to Ubuntu 18.10 (GNU/Linux 4.18.0-25-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.

Your Ubuntu release is not supported anymore.
For upgrade information, please visit:
http://www.ubuntu.com/releaseendoflife

New release '20.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Mar  7 16:45:50 2025 from 192.168.192.131
```

Figure 9: Relogging in with Updated Config

I went into the config and changed SSH port to 2223 in `sshd` config as seen in **Figure 10**. I then relogged with the new port to test if it was functioning as intended (**Figure 11**).

```
Port 2223
#AddressFamily any
#ListenAddress 0.0.0.0
```

Figure 10: Changing Port

```
midterm@server:~$ ssh midterm@192.168.192.132 -p 2223
Enter passphrase for key '/home/midterm/.ssh/id_rsa':
Welcome to Ubuntu 18.10 (GNU/Linux 4.18.0-25-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.

Your Ubuntu release is not supported anymore.
For upgrade information, please visit:
http://www.ubuntu.com/releaseendoflife

New release '20.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Mar  7 17:52:35 2025 from 192.168.192.132
midterm@server:~$
```

Figure 11: New Port Login

Some extra things I did to increase the security on the ssh server was go into the config and change the max authentication tries to 3 and change the PermitEmptyPassword to no (Figure 12)

```
PermitEmptyPasswords no
#StrictModes yes
MaxAuthTries 3s
#MaxSessions 10
```

Figure 12: Additional Security Changes

Conclusion

This lab showed the strategies of how to harden an SSH server to enhance network security. By using key authentication and removing unnecessary services. some of these strategies included changing config settings, adding firewall rules and using nmap to see the open ports that are being unnecessarily used.