**S6 L3**

**AMATO SARA**

**Esercizio programmazione per Hacker**

**Simulazione di Attacco DoS (UDP Flood)**

In questo esercizio, ho eseguito una simulazione di un attacco Denial of Service (DoS) basato su UDP Flood. L’obiettivo dell’attacco è quello di inondare la macchina target (Windows 10) con un elevato numero di pacchetti UDP per verificare l’impatto sul sistema e osservare come un tale attacco potrebbe compromettere le risorse di rete e di sistema.

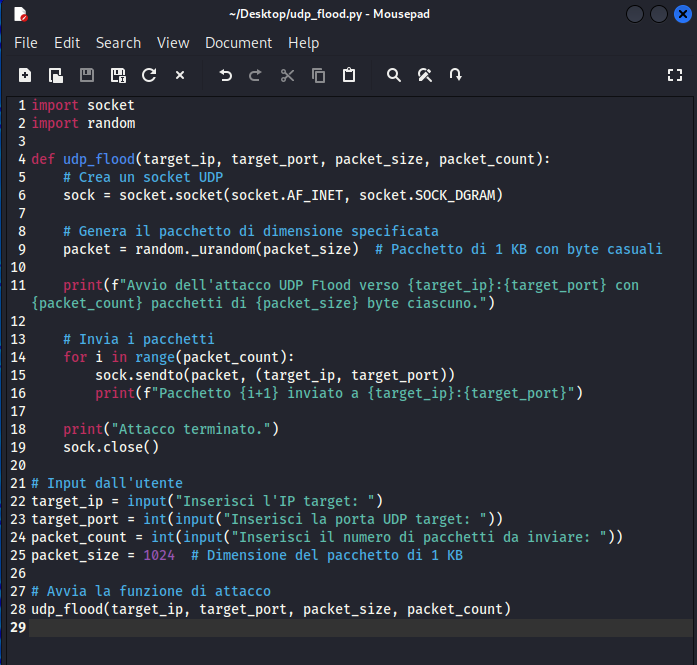
**Sviluppo dello Script UDP Flood**

Ho utilizzato un semplice script Python per inviare pacchetti UDP alla macchina target. Lo script è stato configurato per:

- Target IP e Porta UDP: Richiedere l’input dell’IP della macchina target e della porta UDP di destinazione.

- Generazione del Payload: Inviare pacchetti UDP di 1 KB di dimensione ciascuno, costruiti con byte casuali.

- Numero di Pacchetti: Richiedere il numero di pacchetti da inviare, impostando così la durata e l'intensità dell'attacco.



Importazione delle Librerie:

import socket

import random

- Viene utilizzata la libreria `socket` per creare e gestire la connessione UDP.

- La libreria `random` serve per generare i byte casuali nel pacchetto.

Input dell'IP e della Porta Target:

target\_ip = input("Inserisci l'IP target: ")

target\_port = int(input("Inserisci la porta UDP target: "))

`target\_ip` e `target\_port` accettano l’indirizzo IP e la porta UDP della macchina da attaccare.

Creazione del Pacchetto da 1 KB:

packet = random.\_urandom(1024)

- Qui viene creato un pacchetto di 1024 byte (1 KB) con byte casuali. La funzione `random.\_urandom(1024)` genera un blocco di dati pseudo-casuali, simulando traffico "rumoroso".

Numero di Pacchetti da Inviare:

num\_packets = int(input("Inserisci il numero di pacchetti da inviare: "))

- L’utente inserisce il numero totale di pacchetti da inviare, determinando così la durata e l’intensità dell'attacco.

Creazione del Socket UDP:

udp\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

- Questo crea un socket di tipo UDP (`SOCK\_DGRAM`), che verrà utilizzato per inviare i pacchetti verso la macchina target. `AF\_INET` specifica l’uso dell’IPv4.

Invio dei Pacchetti in Loop:

for \_ in range(num\_packets):

udp\_socket.sendto(packet, (target\_ip, target\_port))

print(f"Pacchetto inviato a {target\_ip}:{target\_port}")

- Il loop `for` invia il pacchetto alla macchina target `num\_packets` volte.

- `udp\_socket.sendto(packet, (target\_ip, target\_port))` invia il pacchetto alla coppia (IP, Porta) specificata.

- `print` conferma l'invio di ogni pacchetto.

Input dell'IP e della Porta: Il programma chiede all’utente di inserire l’indirizzo IP e la porta della macchina target. Questi valori sono necessari per indirizzare i pacchetti UDP alla macchina bersaglio corretta.

Generazione dei Pacchetti: Lo script genera pacchetti UDP di 1 KB (1024 byte) riempiendoli di byte casuali. Questo è fatto per simulare traffico realistico e “riempire” la rete.

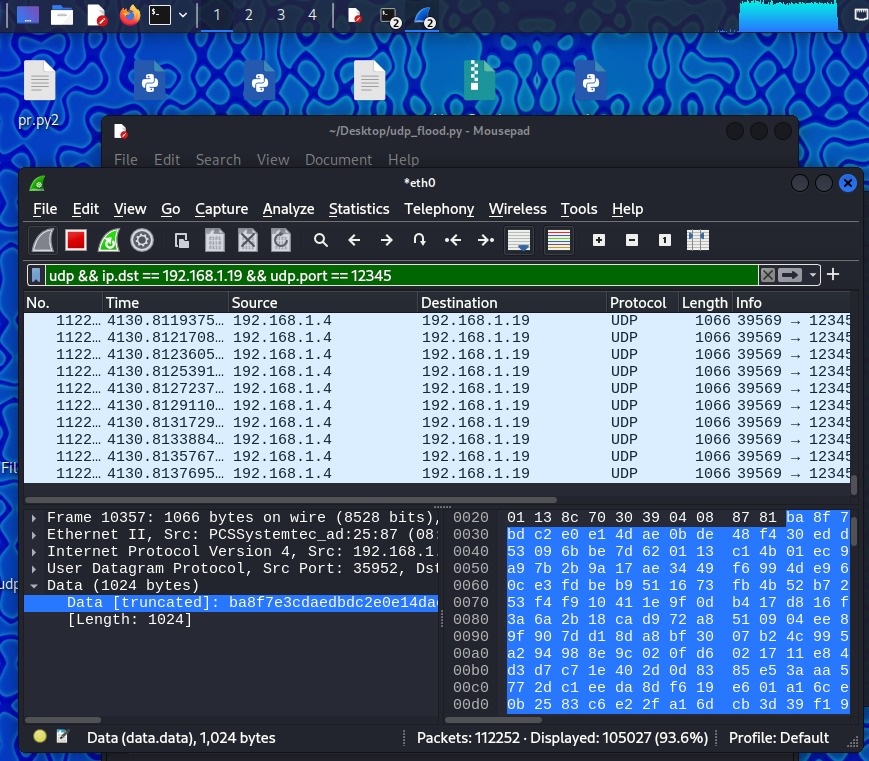
Invio dei Pacchetti: Dopo aver generato il pacchetto, lo script inizia a inviarlo rapidamente al target, creando un effetto di sovraccarico sulla rete e sulla macchina bersaglio.

Ciclo del Numero di Pacchetti: Lo script chiede anche quanti pacchetti inviare. In questo modo, l’utente può impostare la durata e l’intensità dell’attacco.

Ogni passaggio contribuisce a generare un traffico intenso verso la macchina target, creando il tipico effetto di "flood" che satura la connessione o i servizi sulla macchina ricevente.

**Esecuzione e Analisi**

Dopo aver avviato lo script su Kali Linux, i pacchetti UDP sono stati inviati rapidamente alla macchina Windows 10. Wireshark ha catturato il traffico in arrivo, mostrando il flusso continuo di pacchetti UDP. Ho utilizzato un filtro per isolare i pacchetti diretti alla porta selezionata, analizzando i dettagli del traffico in ingresso.



Durante l’attacco, ho osservato un notevole aumento nell’utilizzo delle risorse di sistema sulla macchina target, in particolare per quanto riguarda la rete. Questo tipo di attacco dimostra come un UDP Flood possa saturare la rete e potenzialmente rendere i servizi non disponibili.

L’esercizio ha fornito una comprensione pratica degli attacchi DoS e dell’utilizzo di UDP Flood per simulare una situazione di traffico anomalo e ingente. Utilizzando strumenti come Wireshark, è stato possibile analizzare l’effetto dell’attacco in tempo reale e identificare la natura del traffico sospetto.