Report S6 – L3 - Attacchi DoS (Denial of Service)
Simulazione Controllata di Traffico UDP

Introduzione

In questo esercizio ho approfondito il funzionamento degli **attacchi Denial of Service (DoS)**, con particolare attenzione al traffico **UDP** e al suo impatto sulla disponibilità dei servizi.

L'obiettivo dell'attività è stato quello di comprendere come l'invio ripetuto di pacchetti UDP possa contribuire alla **saturazione delle risorse di rete**, analizzando al tempo stesso il traffico generato tramite strumenti di monitoraggio.

Per svolgere l'esercizio ho realizzato un **programma Python** in grado di:

- richiedere un **indirizzo IP target**,
- richiedere una **porta UDP di destinazione**,
- generare un **payload casuale di dimensione pari a 1 KB**,
- inviare un **numero definito di pacchetti UDP**.

L'attività è stata eseguita in un **ambiente di laboratorio controllato**, utilizzando l'indirizzo di loopback 127.0.0.1, al fine di evitare impatti su sistemi reali e garantire un contesto sicuro e didattico.
Le evidenze tecniche sono state raccolte tramite output applicativo e analisi del traffico con **Wireshark**.

Verifica dei Requisiti del Programma

1. Input dell'IP Target

Il programma sviluppato richiede all'utente di inserire l'indirizzo IP della macchina target. Nel codice del client viene utilizzata una richiesta di input che permette all'utente di specificare l'IP di destinazione:

target_ip = input("Inserisci IP target (solo 127.0.0.1): ").strip()

L'indirizzo IP viene quindi fornito manualmente dall'utente.
Per lo svolgimento dell'esercizio ho scelto di limitare l'invio dei pacchetti all'indirizzo 127.0.0.1, al fine di operare in un ambiente di laboratorio controllato e prevenire impatti su sistemi reali.
Questa scelta non compromette il requisito dell'esercizio, in quanto l'input dell'IP è comunque presente ed è chiaramente documentato e motivato all'interno del report.

---

2. Input della Porta Target

Il programma richiede all'utente di inserire la porta UDP della macchina target.
Nel client l'inserimento della porta avviene tramite input diretto dell'utente, con successiva validazione del valore inserito:

target_port = int(input("Inserisci porta UDP target: ").strip())

La porta viene fornita manualmente, verificata affinché rientri nell'intervallo valido (1–65535) e utilizzata effettivamente durante l'invio dei pacchetti tramite la funzione sendto().
Questo garantisce il pieno rispetto del requisito relativo all'input della porta UDP target.

---

3. Costruzione del Pacchetto (1 KB con Byte Casuali)

Il programma prevede la costruzione di pacchetti UDP di dimensione pari a 1 KB ciascuno.
Nel codice viene definita esplicitamente la dimensione del payload e vengono generati byte casuali utilizzando il modulo random:

PACKET_SIZE = 1024

payload = bytes(random.getrandbits(8) for _ in range(PACKET_SIZE))

Ogni pacchetto ha quindi una dimensione di 1024 byte, equivalente a 1 KB, e il contenuto è composto da byte generati in modo casuale, come suggerito dalle specifiche dell'esercizio.
Il payload così creato viene effettivamente inviato tramite protocollo UDP, soddisfacendo pienamente il requisito di costruzione del pacchetto.

---

4. Numero di Pacchetti da Inviare

Il programma richiede all'utente di specificare il numero di pacchetti da 1 KB da inviare.
Questo valore viene acquisito tramite input e utilizzato per controllare il numero di iterazioni del ciclo di invio:

count = int(input("Quanti pacchetti da 1KB vuoi inviare?: "))

Il numero di pacchetti viene quindi definito dall'utente e utilizzato all'interno di un ciclo for, dove ogni iterazione comporta l'invio di un singolo pacchetto UDP da 1 KB.
In questo modo il programma rispetta il requisito relativo alla gestione del numero di pacchetti da inviare.

Conclusione sui Requisiti

Sono presenti l'input dell'IP target, l'input della porta UDP, la costruzione di pacchetti da 1 KB con contenuto casuale e la possibilità di definire il numero di pacchetti da inviare, con un'implementazione coerente, verificabile e adatta a un contesto formativo di laboratorio.