

S3-L5

Traccia:

Scrivi un programma in Python che simuli un UDP flood (invio massivo di richieste UDP verso una macchina target in ascolto su una data porta UDP).

Di cosa necessitiamo:

- Script in python tramite il quale il client invierà la richiesta
- Server in ascolto su una data porta (la bindiamo) per visualizzare i pacchetti generati e ricevuti.

Walkthrough

“Accendiamo” il server con IP statico e Porta bindata:

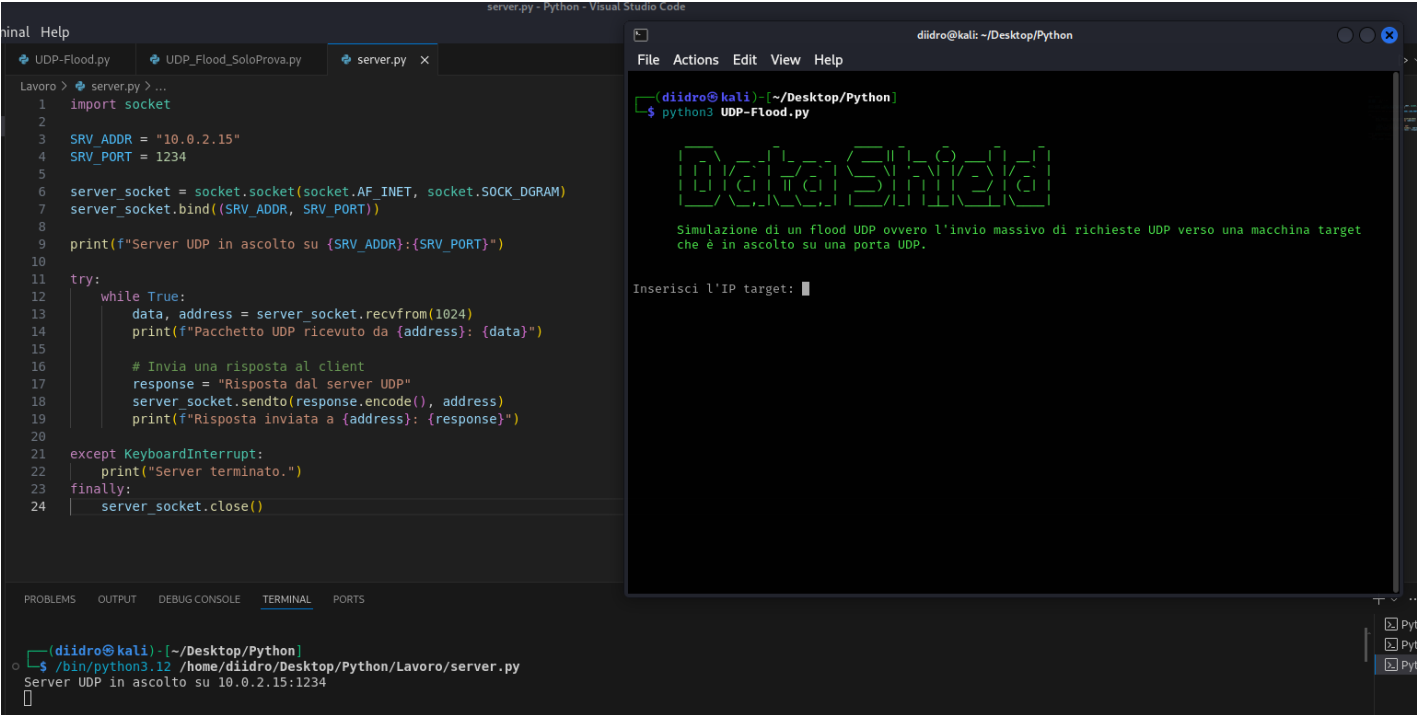
```
Lavoro > server.py > ...
1  import socket
2
3  SRV_ADDR = "10.0.2.15"
4  SRV_PORT = 1234
5
6  server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
7  server_socket.bind((SRV_ADDR, SRV_PORT))
8
9  print(f"Server UDP in ascolto su {SRV_ADDR}:{SRV_PORT}")
10
11  try:
12      while True:
13          data, address = server_socket.recvfrom(1024)
14          print(f"Pacchetto UDP ricevuto da {address}: {data}")
15
16          # Invia una risposta al client
17          response = "Risposta dal server UDP"
18          server_socket.sendto(response.encode(), address)
19          print(f"Risposta inviata a {address}: {response}")
20
21  except KeyboardInterrupt:
22      print("Server terminato.")
23  finally:
24      server_socket.close()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

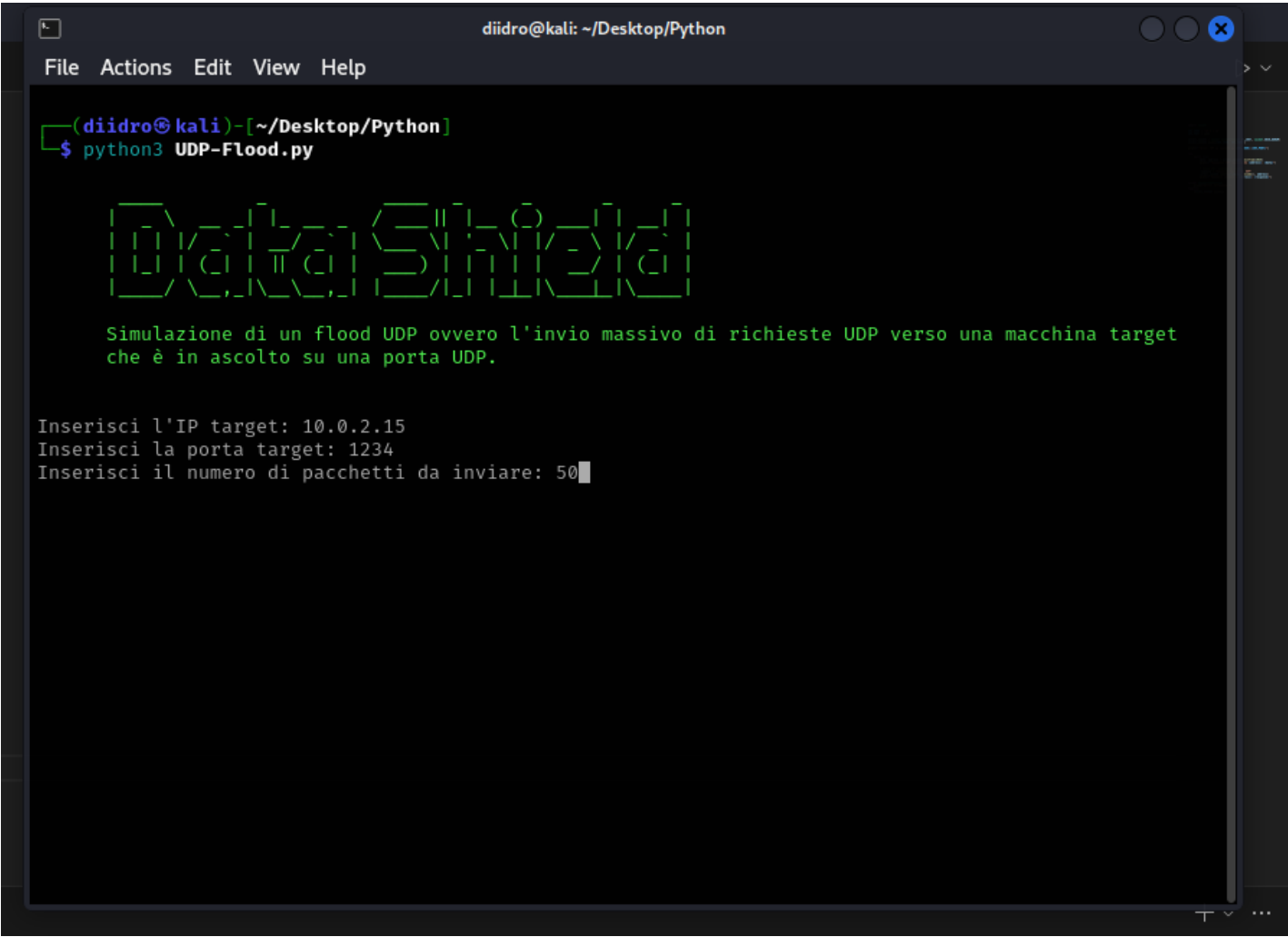
```
(diidro@kali) - [~/Desktop/Python]
$ /bin/python3.12 /home/diidro/Desktop/Python/Lavoro/server.py
Server UDP in ascolto su 10.0.2.15:1234
```

Lanciamo lo Script per l'UDP flood:

NB: Per testare la connettività è buona norma effettuare un ping prima.



Inseriamo i dettagli del target (se non fosse stato biddato avremmo dovuto trovare le porte aperte dello stesso target):



Controlliamo i pacchetti ricevuti (generati randomicamente) lato server:

