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

ES:
Ambiente: Per l'esercizio in questione ho utilizzato due VM Kali Linux, una come attaccante ed una come bersaglio. Le due macchine vanno impostate nella stessa rete (nel mio caso una lan) in modo tale da poter comunicare tra loro.
Ho utilizzato le seguenti impostazioni:

-Macchina "attaccante":

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.1.100/24
netmask 255.255.255.0
gateway 192.168.1.1
```

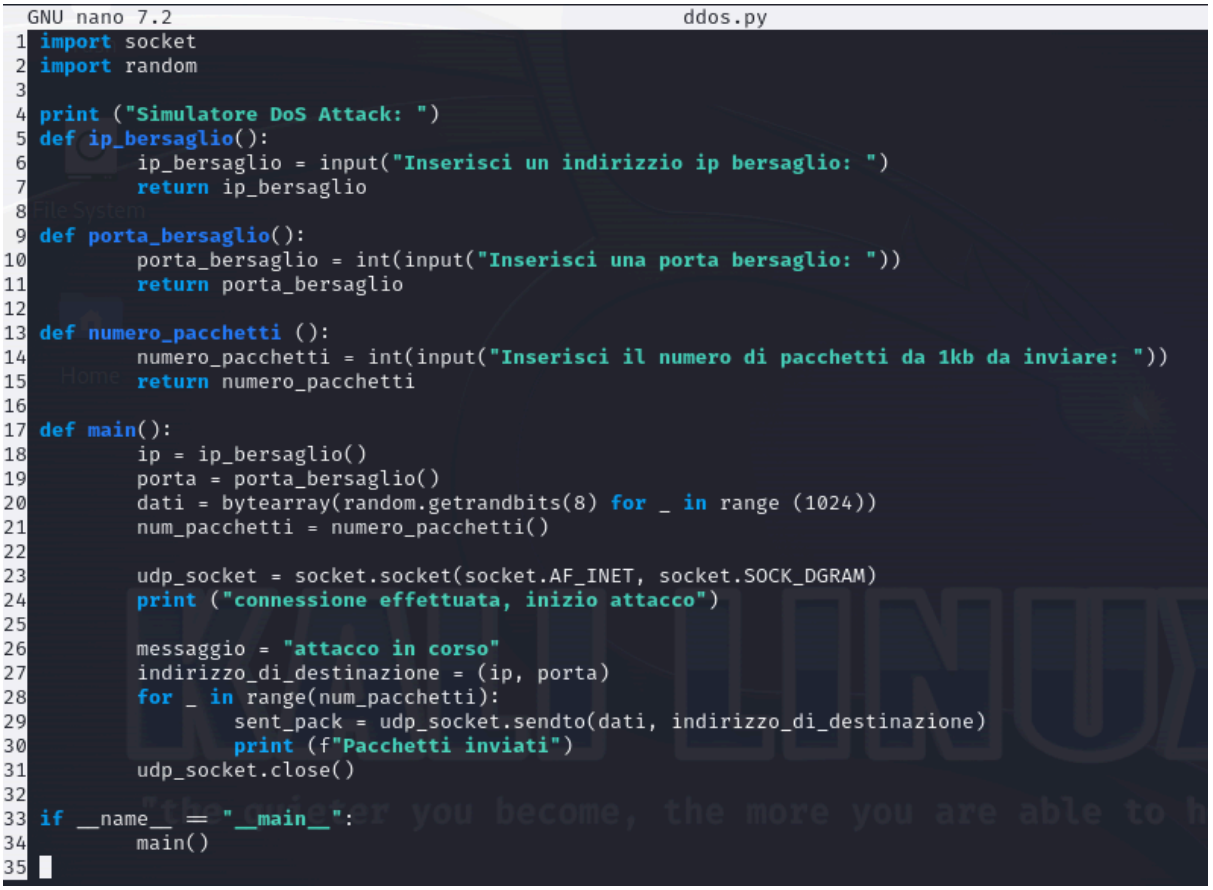
-Macchina "bersaglio":

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.1.110/24
netmask 255.255.255.0
gateway 192.168.1.1
```

Svolgimento:

Come al solito, per programmare in kali utilizziamo l'editor "nano". Importiamo le librerie "random" per la creazione di pacchetti random e "socket" per la creazione e l'invio di pacchetti UDP.
Le funzioni e i comandi che andremo ad utilizzare sono i seguenti:
-**input()** per poter inserire in interfaccia utente i dati per la destinazione e per la grandezza dei pacchetti;
-**random.getrandbits(8)** per far si che il programma crei dei pacchetti di 8 bit;
-**for _ in range()** per far si che il programma crei un determinato numero di pacchetti in base al range stabilito (1kb);
-**udp_socket()** per la creazione di un socket;
-**udp_socket.sendto()** per stabilire la destinazione dei pacchetti e inviarli.

Questo è il codice ottenuto:

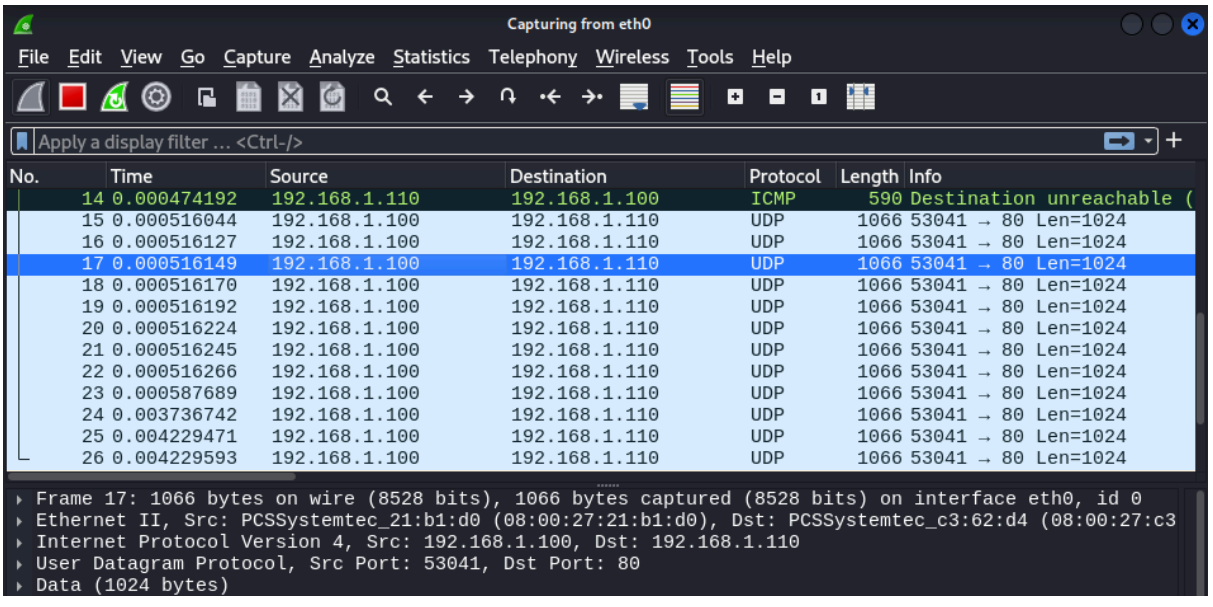


Risultato:

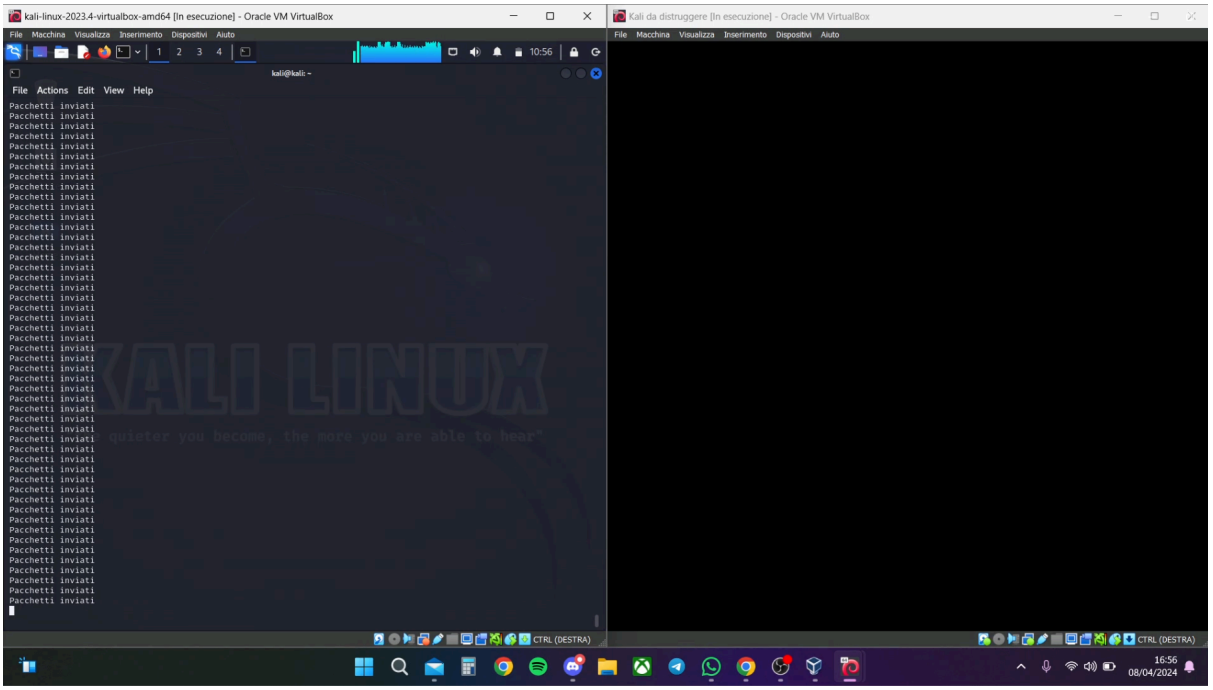
Lancio programma, inserimento dati ed invio pacchetti:

```
┌──(kali㉿kali)-[~]
└─$ python ddos.py
Simulatore DoS Attack:
Inserisci un indirizzio ip bersaglio: 192.168.1.110
Inserisci una porta bersaglio: 80
Inserisci il numero di pacchetti da 1kb da inviare: 999999999999999
connessione effettuata, inizio attacco
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
Pacchetti inviati
```

Sniffing dei pacchetti da wireshark bersaglio:



Crash di Kali Linux bersaglio:



Conclusioni e considerazioni:

In questo caso ho impiegato più tempo del previsto, la maggior parte del quale per capire quali funzioni e comandi avrei dovuto utilizzare.
Inizialmente oltretutto, i pacchetti non venivano sniffati da wireshark in quanto non avevo inserito nella maniera corretta le due macchine all'interno della stessa rete.
Per poter far crashare kali poi, ho dovuto effettuare un downgrade delle specifiche tecniche portandole al minimo (1gb ram e 1 cpu). Questo perché con le impostazioni base, la grandezza dei pacchetti avrebbe impiegato tantissimo tempo a saturare la ram e la cpu bersaglio. Se avessi aumentato la grandezza dei pacchetti invece, molto probabilmente avrei fatto crashare la macchina attaccante.