Avance 3

Red Team:

Proyecto > packet_attack.py > ...

```python
#!/usr/bin/env python3
from scapy.all import IP, TCP, send, RandShort
import time

def syn_probe(target_ip, target_port, count=5, delay=1.0):
    print(f"[INFO] Enviando {count} paquetes SYN a {target_ip}:{target_port}")

    for i in range(1, count + 1):
        sport = RandShort()  # Puerto de origen aleatorio (simula puertos efímeros)

        # Paquete SYN: primera fase del handshake TCP
        pkt = IP(dst=target_ip) / TCP(sport=sport, dport=target_port, flags="S")

        send(pkt, verbose=False)
        print(f"SYN #{i} enviado (sport={sport})")

        time.sleep(delay)  #Permite observar cada paquete

    print("Prueba terminada. Revisar sniffer/logs por parte del Blue Team.\n")


if __name__ == "__main__":
    print("Red Team-packet_attack\n")

    target_ip = input("IP objetivo: ").strip()
    target_port = int(input("Puerto destino: ").strip())

    count_in = input("Cantidad de SYN (Recomendado 5): ").strip()
    delay_in = input("Delay entre paquetes (Recomendado 1.0): ").strip()

    count = int(count_in) if count_in else 5
    delay = float(delay_in) if delay_in else 1.0

    syn_probe(target_ip, target_port, count, delay)
```

El usuario debe ejecutar packet_attack.py desde una terminal en su máquina local, ingresar la dirección IP pública de la máquina objetivo y el puerto permitido en su configuración de red, y proporcionar la cantidad de paquetes SYN que desea enviar junto con el tiempo de espera entre cada uno. Una vez ingresados los datos, el script enviará los paquetes SYN hacia el destino especificado, lo que permite generar tráfico controlado que puede observarse desde la máquina víctima mediante herramientas como tcpdump o un sniffer defensivo.

```
PS C:\Workspace\Proyecto> python .\packet_attack.py
Red Team-packet_attack

IP objetivo: 20.110.88.57
Puerto destino: 22
Cantidad de SYN (Recomendado 5): 5
Delay entre paquetes (Recomendado 1.0): 1.0
[INFO] Enviando 5 paquetes SYN a 20.110.88.57:22
SYN #1 enviado (sport=25984)
SYN #2 enviado (sport=52431)
SYN #3 enviado (sport=7003)
SYN #4 enviado (sport=21830)
SYN #5 enviado (sport=36634)
Prueba terminada. Revisar sniffer/logs por parte del Blue Team.

PS C:\Workspace\Proyecto>
```

```
azureuser@PruebasProyecto:~$ sudo tcpdump -n "tcp[tcpflags] & tcp-syn != 0"
00:10:57.240197 IP 168.63.129.16.32526 > 172.16.0.4.42526: Flags [S.], seq 1721431841, ack 430017009, win 65535, options [mss 1460,nop,wscale 8,sackOK,TS val 3592808480 ecr 2653705570], length 0
00:10:57.954290 IP 172.16.0.4.22 > 186.64.212.47.43643: Flags [S.], seq 399491092, ack 1, win 64240, options [mss 1460], length 0
00:10:58.914298 IP 172.16.0.4.22 > 186.64.212.47.34821: Flags [S.], seq 1095373008, ack 1, win 64240, options [mss 1460], length 0
00:10:59.938283 IP 172.16.0.4.22 > 186.64.212.47.4985: Flags [S.], seq 2853483615, ack 1, win 64240, options [mss 1460], length 0
00:11:00.962278 IP 172.16.0.4.22 > 186.64.212.47.40933: Flags [S.], seq 2560873360, ack 1, win 64240, options [mss 1460], length 0
00:11:03.241642 IP 172.16.0.4.59914 > 168.63.129.16.80: Flags [S], seq 4089953730, win 64240, options [mss 1460,sackOK,TS val 2653711572 ecr 0,nop,wscale 7], length 0
```

En la captura obtenida mediante tcpdump se observa tráfico desde 186.64.212.47 (IP Publica del atacante) hacia 172.16.0.4 (IP Privada de la VM) seguido inmediatamente por respuestas con el flag [S.], lo que indica que la VM reconoció los SYN y respondió con el segundo paso del handshake TCP. Este comportamiento valida que los paquetes generados por Scapy llegaron a la máquina objetivo.