PROGRAMMING EXERCISE

BYTE REBELS

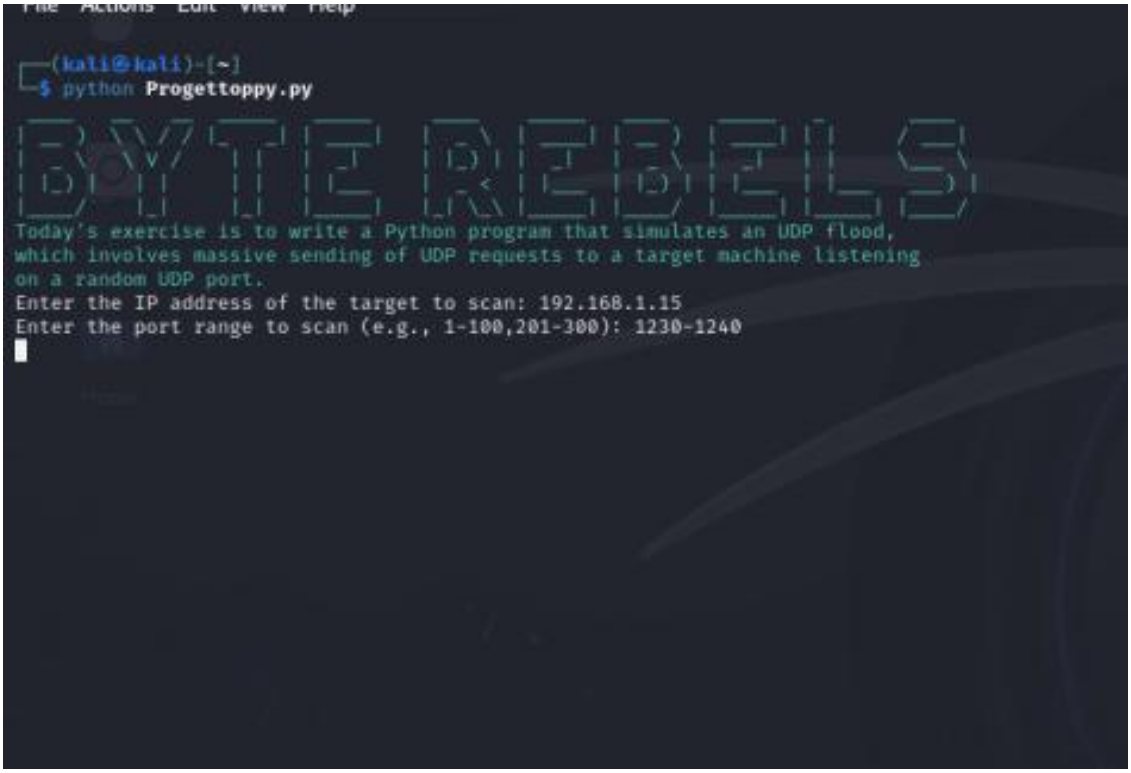
UDP FLOOD

CANNAVACCIUOLO DAVIDE
DI MAIO PAOLO
FORLENZA SIMONE
RUSSO FEDERICO - LEADER
TIZZI FEDERICO
VAN ZWAM ARJEN

BYTE REBELS

ENG

Today's exercise is to write a **Python** program that simulates a **UDP flood**, that is the mass sending of requests to a target machine that is listening on a random UDP port.



UDP FLOOD CLIENT .PY

We projected this program with two main functionalities, port scanner and UDP Packet Sender.
Port Scanner: Allows to scan ports on a given ip address to determine which ports are open
and which are closed or are not giving response. The user can specify a range of ports to scan
and the target IP address . Then we will move on to the actual function of the project, the flood.
The program uses thread to run the scan in a more efficent way.

```
Port 1231 on 192.168.1.15 is CLOSED or unresponsive
Port 1232 on 192.168.1.15 is CLOSED or unresponsive
Port 1233 on 192.168.1.15 is CLOSED or unresponsive
Port 1234 on 192.168.1.15 is OPEN
Port 1235 on 192.168.1.15 is CLOSED or unresponsive
Port 1236 on 192.168.1.15 is CLOSED or unresponsive
Port 1237 on 192.168.1.15 is CLOSED or unresponsive
Port 1238 on 192.168.1.15 is CLOSED or unresponsive
Port 1239 on 192.168.1.15 is CLOSED or unresponsive
Port 1240 on 192.168.1.15 is CLOSED or unresponsive
```

03 ENG PORT SCANNER

What is it for?

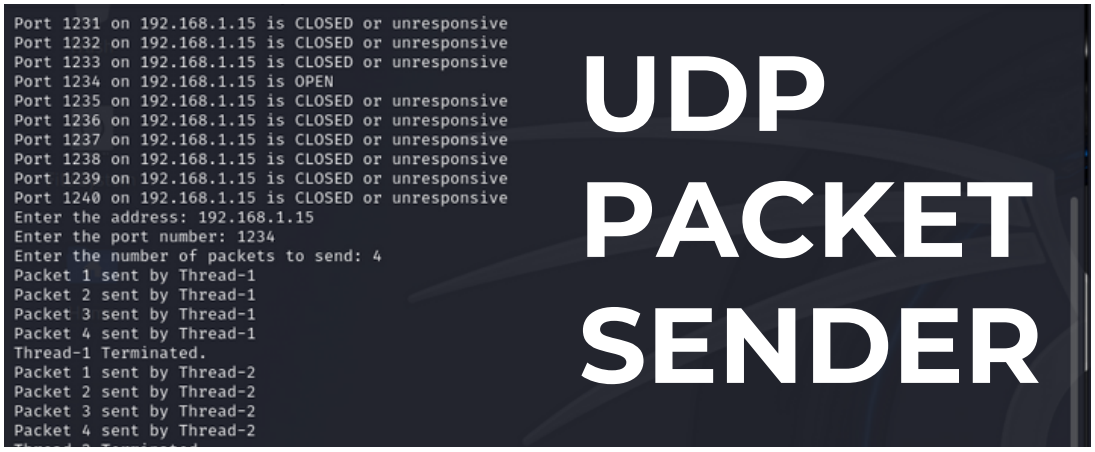
Allows you to perform a port scan on a given **IP** address to determine which ports are open and which are closed or unresponsive.
Furthermore, the user can specify a range of ports to scan and the **IP** address of the target. The program uses threads to scan *more efficiently*.

Usage

Subsequently the user enters the range of ports to scan specifying the lower port and the higher port separeted by a dash(-). For example 1-1000 The script runs the scan of the ports and prints the results in numerical order, giving a feedback.

Scope

Find the victim's open doors



04 ENG

What is it for?

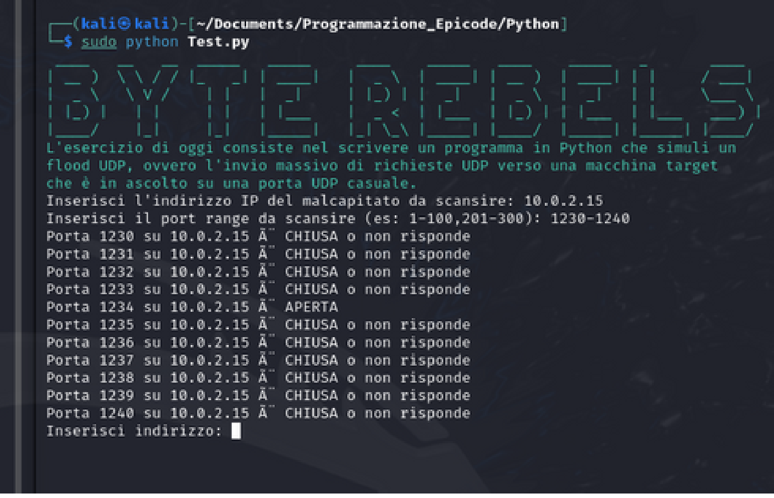
It allows to send a certain amount of **UDP packets** to a given IP address and associeted port. The user can specify the destination IP(*victim*) , the port and the amount of packets to send. This feature also uses thread to send the packets simoultaneously.

Usage

After having identified the port using the scanner, the user is prompted to enter the "victim" IP address to send the **UDP packets**.
So the user enters the port number and number of packets to send.
The script runs a certain amount of threads to send the UDP packet **simoultaneously**.

Scope

Clogging the target IP with fake packets



05
ENG

ANALYSES: Client - Flood.py

1. The program starts by asking for the target IP address.
2. Entering the range of ports to scan.
3. Enter the victim IP address.
4. Insert the previously acquired port.
5. Entering the amount of UDP packets to send.

ANALYSES: Server.py

We simply created a mini server that listens for UDP on its port (we assumed 1234).

PILLOLE DI SCRIPT

IMPORT

We imported the various libraries for the correct running.

DEF FUNCTIONS

We defined the functions *sendudppackets*, *port_scan_range*, *port_scanner* and obviously the *main*.

NOTIFY

s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

SOCK_**DGRAM** stands for the **UDP** protocol, not TCP

data = bytearray(random.getrandbits(8) ***for*** *_ in range(1024))*

So: 8 is 1 Byte, **for** _ in range(1024): to get up to 1024 byte The use of sockets in several points has been deliberately introduced to make it quicker and efficient.