COMP 7005
Assignment 2

Design

Daryush Balsara
A01265967
Nov, 1 2024

Purpose

- Build a basic port scanner that mimics the behaviour of hping3 using scapy. This scanning technique is known as a TCP SYN scan and is commonly used because it provides basic information about the target's services without fully opening a TCP connection.

Functions

port_scanner.py

parse_agrs()	Parses command-line arguments
port_scan()	The range of ports to scan on
main()	Parses the arguments and executes the script

Variables

NA

Pseudo Code

parse_args()

Parameters

Parameter	Type	Description
na	na	na

Return

Value	Reason
parse_args	Namespace containing the parsed command-line arguments, allowing the rest of the program to access the file paths provided by the user

```
FUNCTION parse_args:
    CREATE an argument parser with a description
    ADD a positional argument 'ip' of type ip_address with help message
    ADD an optional argument '-s' for start port, type integer, default value 1, with help message
    ADD an optional argument '-e' for end port, type integer, default value 65535, with help
message
    ADD an optional argument '-d' for delay, type integer, default value 0, with help message

    RETURN the parsed arguments
```

port_scan()

Parameters

Parameter	Type	Description
target_ip	ip_address	The IP of the target machine you want to scan
target_port	int	The port(s) of the target machine you want to scan

Return

Value	Reason
filtered	Return default as filtered if you don't get a response or it is unknown

```
FUNCTION port_scan(target_ip, target_port):
    CONVERT target_ip to string and assign to target_ip_str
    SEND a SYN packet to target_ip_str at target_port and wait for a response, assign to resp

    IF resp is None:
        RETURN "filtered"

    IF TCP is in resp:
        IF resp[TCP].flags equals 18:  // SYN-ACK
            RETURN "open"
        ELSE IF resp[TCP].flags equals 4:  // RST
            RETURN "closed"

    RETURN "filtered"
```

main()

Parameters

Parameter	Type	Description

na	na	na

Return

Value	**Reason**
na	na

```
FUNCTION main:
    CALL parse_args and assign the result to args

    SET ports_to_scan as a range from args.start to args.end + 1

    FOR EACH port in ports_to_scan:
        IF args.delay is greater than 0:
            CALL time.sleep with delay converted from milliseconds to seconds

        CALL port_scan with args.ip and port, assign result to scan

        IF scan equals "open":
            PRINT "Port {port} on {args.ip} is open"
        ELSE IF scan equals "closed":
            PRINT "Port {port} on {args.ip} is closed"
        ELSE IF scan equals "filtered":
            PRINT "Port {port} on {args.ip} is filtered"
```