CSC 251 ♦ NETWORK SECURITY

FINAL COURSE PROJECT: TCP PORT SCANNER
DUE 6:00 PM ON FRIDAY, 21 MAY 2021
PLEASE SUBMIT YOUR WORK ELECTRONICALLY VIA MOODLE
GRADING CRITERIA (30 PTS)

Network Scanners and Port Scanners are essential tools while trying to understand the layout of a network and the services that a specific host is running. They are often used for network diagnostics, but also as a precursor to launching an attack, since they identify potentially vulnerable services. In this project, you are called to design Port Scanner.

You can use whatever language and protocols you want to implement the scanning node. Obviously, you should implement everything by yourself. Do NOT use a ready-made port scanner. ANY code fragments that you find and use from the web, you should document them in your report.

Please adhere to the ethics considerations:

- You may wish to develop your program on your own Unix-based system (e.g., Linux, macOS, WSL on Windows 10) and scan localhost (127.0.0.1).
- It is not cool to scan hosts on the Internet when you do not have permission to do so. Since port scanners are sometimes used to prepare for an attack, network administrators build tools to detect their use (see the next part of this assignment). Hence, by scanning a host, you may cause an alarm to be raised. Even if the target machine is not being monitored for probes, routers along the path from the scanner to the target may detect the "attack".

The required features are as follows:

- Check whether an IP address is alive (e.g., 127.0.0.1)
- Probe(Scan) an IP address for a given set of ports using any of the following scanning modes:
    - Normal Port Scanning (full TCP connect to the remote IP address)
        - When this mode is requested, you should also grab the banner sent by the server
    - TCP SYN Scanning (only send the initial SYN Packet and then send RST when client responds with SYN|ACK)
    - TCP FIN Scanning
- Any of the above host/port scanning methods must also be able to be done sequentially or in random order
    - Probe all 2^{16} TCP ports on a targeted host
    - Scan the ports in order (i.e., from 0 to 65,535)

    - Scan in random order (e.g., instead of first scanning port 1, then port 2, then port 3, etc., randomize the order of ports)
- For each open port, port scanner should report both the port number and the service that normally runs on that port
    - The service can be found by using the getservbyport() and socket.getservbyport() calls in C and Python, respectively
    - Report how long it took to conduct the command
    - Report the number of ports that were found to be closed/open

The design should follow:

- The command-line usage for the program should be:
    - python port_scanner.py target
    - target is the hostname of IP address of the machine that is to be scanned
- The following is a sample output:

```
$ python port_scanner.py 172.16.48.130

Starting port scan                  at 2011-01-21 01:30 PST
Interesting ports on 172.16.48.130:
Not shown: 992 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
111/tcp  open  rpcbind
2049/tcp open  nfs
3306/tcp open  mysql
5000/tcp open  upnp
6000/tcp open  X11
8000/tcp open  http-alt

scan done!1 IP address (1 host up) scanned in 0.23 seconds
```

Submission:

- All files necessary to run your code
- A README.md file including:
    - A list of all files required included in your project
    - An explanation of how to run your projects
- A discussion of at least one major challenge, and how you overcame it.
- If you work in a group,
    - a description of your specific contributions to the project