COSC235 - Homework 3*
Assigned October 2nd, 2014; Due 11:59pm on October 16th, 2014

Prof. Micah Sherr

Note: This homework assignment is worth 75 points.

1 A Port Scanner {30 points}

In this programming assignment, you will be building a port scanner. A port scanner is a software program that probes a target computer for open ports—i.e., ports that have services listening on them. They are often used for network diagnostics, but also as a precursor to launching an attack, since they identify potentially vulnerable services.

Your port scanner, PortScan, will probe all $2^{16}$ TCP ports on a targeted host, and report the ports that accept connections. Your scanner should not require superuser (root) privileges, and can attempt to establish full TCP connections to the tested ports.

Your scanner should scan the ports in order (i.e., from 0 to 65535) as quickly as possible. That is, you should not pause or sleep between probes.

For each open port, PortScan should report both the port number and the service that normally runs on that port. The latter can be found by using the `getservbyport()` and `socket.getservbyport()` calls in C and Python, respectively.

PortScan should also report how long it took to probe all ports, the number of ports that were found to be open, and the scan rate (ports scanned per second).

The command-line usage for PortScan should be:

```
PortScan target
```

or

```
python PortScan.py target
```

for C/C++ and Python, respectively, where target is the hostname or IP address of the machine that is to be scanned.

*Last revised on October 1, 2014.*
IMPORTANT NOTICE REGARDING COMPUTER ETHICS. 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”. You are strictly forbidden to run PortScan against any machine except for netid-alice-HW1, netid-bob-HW1, or any machines announced by the teaching staff as being an appropriate target. Nor should you run PortScan from any machine other than netid-alice-HW1 and netid-bob-HW1.

2 A Port Scanner Detector {30 points}

For the second part of the homework, you will build PSDetect, a port scanner detector. PSDetect will use the pcap library (libpcap for C1 or pycapy2 for Python) to listen to incoming connections, and report the presence of a scanner if a single machine attempted to connect to 15 or more consecutive ports within a 5 second window. PSDetect should therefore be able to detect when PortScan is used.

PSDetect should listen on all network interfaces, and should take no arguments. It should not produce any output until a scanner is detected. When a scanner is detected, it should print out the message3:

Scanner detected. The scanner originated from host A.B.C.D.

where A.B.C.D should be replaced with either the IP address or the hostname of the machine that attempted to connect to 15 or more consecutive ports within a 5 second window.

PSDetect should only terminate when the user presses CTRL-C.

The difficult part of this assignment is obtaining the IP header of captured packets. pcap functions at the data link layer and will return to you Ethernet frames. You’ll need to access the part of those frames that correspond to the IP headers. This is difficult in C/C++, and you can borrow code from http://www.tcpdump.org/sniffex.c. In Python, you can use the dpkt library that is installed on your machines; the page for pycapy has an example that will help you greatly.

PSDetect will require superuser (root) privileges. You will need to run it via either “sudo PSDetect” or “sudo python PSDetect.py”.

---

1see http://www.tcpdump.org/pcap.html
2see https://code.google.com/p/pypcap/
3It does not have to print this message in red.
3  A Port Scanner Detector Evader (yes, I’m having fun with these names)  
{15 points}

Next, you will modify PortScan to evade the PSdetect port scanner. This port scanner will be called PortScanToo. PortScanToo should operate roughly as quickly as PortScan (i.e., the difference in timing between the two port scanners should be negligible\(^4\)). Unlike PortScan, PortScanToo does not have to scan ports in sequential order. Also, unlike PortScan, PortScanToo should not be detected by PSdetect.

Note that the same ethics warning/requirement pertaining to PortScan (see above) also applies to PortScanToo.

The output for PortScanToo should be identical to that of PortScan.

The command-line usage for PortScanToo should be:

PortScanToo target

or

python PortScanToo.py target

for C/C++ and Python, respectively, where target is the hostname or IP address of the machine that is to be scanned.

Submission Instructions

Include the source code to all three programs in a single compress tarball (.tar.gz file), and upload that file to Blackboard. Include compilation instructions, if your code requires it.

To upload your assignment, navigate to the COSC235 course, click the “Assignments” link on the left hand side, and select “hw3”. Look for the “Attach File” section and upload your submission. Be sure to hit the “Submit” button when done. Upload your assignment before 11:59pm on October 16th.

Please post questions (especially requests for clarification) about this homework to Piazza.

---

\(^4\)What’s negligible? Let’s say that PortScanToo should impose less than a 1% increase in the average time it takes to conduct a scan.