CS114 - Homework 2[*]

Assigned March 28th, 2023; Due 11:59pm on April 27th, 2023

Prof. Daniel Votipka

1 A Port Scanner {30 points}

IMPORTANT: Due to the logistics of this assignment and issues with important library availability, we will be using python version 2.7 for this assignment (instead of 3.x like the prior assignments). This assignment will also be manually graded (but still uploaded to Gradescope).

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 *socket.getservbyport()* call. Your program should not crash if no service is defined for a particular open port (i.e., the case in which *socket.getservbyport()* returns an exception). Instead, if there is no associated service, your program should report ”NA” as the port’s service.

Additional requirements and hints

- Your program **must** be called `PortScan.py`.
- `PortScan` should also report how long it took to probe all ports and the scan rate (time per second).

[*] Last revised on April 12, 2023.

- All the output from your program should be saved to a file named `scanner.txt`. Here is an example of what the output should look like:

  ```
  53 (dns) was open
  443 (https) was open
  8213 (NA) was open
  time elapsed = 500s
  time per scan = .0001s
  ```

- `PortScan` should have the following command-line options:

  ```
  python PortScan.py target
  ```

  where `target` is the hostname or IP address of the machine that is to be scanned.

- To test your scanner, we've created a script that will randomly open and listen on 10 ports. You can download a copy of the script here for testing. You can run this script using the following command:

  ```
  python PortOpener.py
  ```

- If you want to make sure the ports you're scanning for are actually open, you can use the linux command *netstat*. This command will tell you which ports are currently open and listening. Here is the exact command-line options for looking at open tcp ports:

  `netstat -anp tcp` (Mac)

  `netstat -anp` (Linux)

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 a machine you personally own, or any machines announced by the teaching staff as being an appropriate target. Nor should you run `PortScan` from any machine other than a machine you personally own.**

Rubric

The following is a rough rubric for evaluating `PortScan.py`.

- Doesn't compile **[Deduction: 30pts]**
- Incorrect port identified **[Deduction: 2pts]** (per port)
- Service information not reported for ports **[Deduction: 5pts]**
- Timing information not reported **[Deduction: 5pts]**

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 pcapy library 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 minute window**. PSDetect should therefore be able to detect when PortScan is used.

PSDetect should listen on the loopback interface, and should take no arguments. It should not produce any output until a scanner is detected. When a scanner is detected, it should write the following message to detector.txt[1]:

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

where A.B.C.D should be replaced with the IP address of the machine that attempted to connect to 15 or more consecutive ports within a 5 minute window. Each new scanner found should have a message like the one above printed on its own line.

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. You can use the dpkt library to handle packet processing and pcapy to sniff traffic sent on the loopback interface (typically lo0).

PSDetect will require superuser (root) privileges. You will need to run it via "sudo python PSDetect.py".

Hints

- You should test with scanning traffic from multiple sources (this is how your program will be evaluated by our grading script). If you only have one computer, then you can spoof the source IP address. Scapy is a really useful python library that will allow you to modify raw packets and do things like change the source IP address. Here is a list of simple scapy commands that should give you some good direction.

Rubric

The following is a rough rubric for evaluating PSDetect.py.

- Doesn't compile **[Deduction: 30pts]**
- Incorrect IP identified **[Deduction: 3pts]**(per port)

[1] It does not have to print this message in red.

3 A Port Scanner Detector Evader (yes, I'm not good at naming things) {15 points}

Next, you will modify `PortScan` to evade `PSDetect`. 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[2]). 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`.

Additional requirements and hints

- The output for `PortScanToo` should be identical to that of `PortScan`, but should be written to `scannertoo.txt`.
- The command-line usage for `PortScanToo` should be:

  `PortScanToo.py target`

  where `target` is the hostname or IP address of the machine that is to be scanned.

Rubric

The following is a rough rubric for evaluating `PortScanToo.py`.

- Doesn't compile **[Deduction: 15pts]**
- Incorrect port identified **[Deduction: .2pts]** (per port)
- `PortScanToo` is detected by `PSDetect` **[Deduction: 8pts]**
- Timing information not reported **[Deduction: 5pts]**
- `PortScanToo` adds more than 1% slower than `PortScan` per scan **[Deduction: 5pts]**

[2] 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.

Submission Instructions

Submit your solutions as three separate files (*PortScan.py*, *PSDetect.py*, and *PortScanToo.py*) to Gradescope.

Unfortunately, it is not possible to make the second two parts work in Gradescope and Gradescope does not support python 2.7. So, I will be running the grading script locally on my machine on your submission and will update your score manually. To ensure that I get you scores as soon as possible, please email me whenever you make a submission to Gradescope that you would like to be graded. Otherwise, you will have to wait for the next time I check for new submissions.

Upload your assignments before 11:59pm on April 27th.

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