CMPT 479/980:
Systems and Network Security
Spring 2020

Assignment 2

Network Analysis

Due date: March 17, 2020

Instructor: Khaled Diab

The goal of the assignment is to:

(a) Implement a network discovery tool.
(b) Analyze network traffic from a captured file.

1. Prerequisites

You will use `scapy` and Wireshark for this assignment.

1.1 Software

(a) Install `scapy`: https://scapy.readthedocs.io/en/latest/installation.html
(b) Install Wireshark: https://www.wireshark.org/#download

The course VM is loaded with scapy 2.2.0 and Wireshark 2.6.10.

1.2 Resources

Both `scapy` and Wireshark have extensive online resources (e.g., documentation, user guides, forums, conferences, etc.). If you are not familiar with these tools/APIs, it is recommended to explore some (unofficial) resources before your start.

- Scapy: https://scapy.readthedocs.io/en/latest/usage.html
- Scapy cheat sheet (unofficial): https://blogs.sans.org/pen-testing/files/2016/04/ScapyCheatSheet_v0.2.pdf
- Wireshark: https://www.wireshark.org/docs/wsug_html_chunked
- Wireshark cheat sheet (unofficial): https://www.comparitech.com/net-admin/wireshark-cheat-sheet/

2. Tasks

Task 1: Implementing `traceroute` (50%)

We discussed the implementation of `traceroute` in the class. `traceroute` is a networking tool that is used to explore the route from a source to destination.

Your **task** is to implement a command line program that implements `traceroute` using `scapy`. Your program should take one input, which is the destination IP address. Specifically, your program should be used as follows.

```
./traceroute.py <IP_Address>
```

The expected output of your program should be:

```
Traceroute 1.1.1.1
1 hops away 10.0.2.1, RTT=1ms
2 hops away 10.1.192.2, RTT=2ms
3 hops away X.X.X.X, RTT=2ms
Request timeout to Y.Y.Y.Y
…
8 hops away 1.1.1.1, RTT=7ms
```

Your program should implement the following functionalities:

(1) Calculating RTT of each hop along the route
(2) Setting each request timeout to 30 seconds
(3) Handling a request timeout: your program should continue sending a request to the next hop. Notice that a request timeout is not the TimeExceeded response you should receive during normal operations
(4) Your program should terminate with error when: (i) the number of hops exceeds 30, or (ii) sending requests to 10 different routers results in timeouts.

Questions

(1.a) Compare your implementation versus `traceroute` with at least five different public name servers and web servers. List your observations with proper screenshots.

`scapy` *has its own* `traceroute` *implementation.*
You should not use `scapy`*'s implementation for the purpose of this task.*

Task 2: Analyzing Network Traffic (50%)

An Information Security Officer at Pandora University received an email from Stephen Grant, a lecturer, reporting that he has been receiving harassing emails at stephengrant@yahoo.com. Based on the body of the emails, the Security Officer suspects that they were sent by a student in Stephen's class. The student list of the class is: Amy Hightower, Jane Ford, Tuck Gorge, Terry Gao, Kelsey Smith, Mike Hunt, Ava Martin, Sadie Clarke, Sophie Russel, Anas Salah, Jenny Wilson.

With the help of the IT department, the Security Officers placed a sniffer on the campus network. *Two emails* were received since the sniffer was placed. The Security Officers in charge believe they have enough information to find the harasser.

The network traffic can be found here: https://vault.sfu.ca/index.php/s/tqFEEF43NdTfv1R

Your **task** is to analyze the provided network traffic and provide answers for the following questions.

Questions

(2.a) What are the message bodies of the two emails?

(2.b) Sending the email was done through a web page. After sending the first email, the mail server sent an HTML response to the harasser. Construct that HTML page and name it `response.html`.

(2.c) Who sent the two emails?

(2.d) What is the evidence? List all evidence you found. Also, explain the procedure you followed to find the harasser (with screenshots).

Hint. Novice harassers may leave traces in the network before or after attacking their victims (e.g., posting in online forums, searching online about harassment etc.).

3. Submission

You are required to submit:

(1) your traceroute program from Task 1: `traceroute.py`

(2) an HTML page `response.html` from Task 2.

(3) a PDF document answering the questions from Section 2.

The files should be compressed in a single (.zip) archive. The code should run without any errors.

4. Policy

- Late submissions will not be graded.
- Make sure that your code is well-organized with sufficient comments.
- Any form of cheating will not be tolerated. Particularly, copying code from other students or from other sources such as the Web.
- You can discuss the assignment with other students. However, the actual coding must be your own.

5. Environment Setup

You may use a virtual machine (VM) to complete this assignment. We created a VM with a preinstalled Ubuntu 16.04 (32-bit) and the required software to run the code and analyze network traffic. You can create your own VM. However, it is your responsibility to ensure that your submitted code for Task 1 works on *our* VM.

 *We will run your submitted code using the given VM.*

Virtual Machine

If you are not familiar with VMs, refer to online articles about virtualization (e.g., https://en.wikipedia.org/wiki/Virtual_machine). In summary, virtualization allows you to run a guest OS on top of your host OS in isolation. Virtualization needs two main components (when it comes to this assignment): (1) Hypervisor, which is the software that allows you to run the guest OS, and (2) VM Image, which contains the OS and installed packages.

Setup. We recommend using VirtualBox (https://www.virtualbox.org/) as the hypervisor. It is a free software and easy to install and use. For the VM Image, we prepared an Ubuntu-based image with the required dependencies to compile and run the code. The setup has three simple steps:

1. Download and install VirtualBox
2. (Optional) Install VirtualBox Guest Additions: https://www.techjunkie.com/ova-virtualbox/
3. Download the VM Image: https://vault.sfu.ca/index.php/s/pq2sVjmUlmfBWwl
4. Import the VM Image: https://techantidote.com/how-to-import-ova-file-into-virtualbox/

The Image is based on Ubuntu 16.04 LTS (32-bit), which requires 2 GHz dual core processor or better, 4 GB system memory, and 25 GB of free hard drive space.

Login. After importing the provided VM Image, you can login to the guest OS using these credentials:
Username: `sfu`
Password: `ufs`
Note that this user has administrative rights.