Cybersecurity Lab II

Lab 8

Network Analysis

The goal of this lab is to:

(a) Implement a (simple) 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

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` (25%)

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
…
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.

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 TLS Traffic (45%)

On this list of top 1000 websites:
https://gist.github.com/bejaneps/ba8d8eed85b0c289a05c750b3d825f61
Choose the website that corresponds to the last 3 digits of your SFU ID and visit it in HTTPS. (If it does not have HTTPS or cannot load, write this in the report and visit the next number.)

We want to understand what someone can see by eavesdropping on HTTPS traffic. Using tcpdump, capture the network traffic. Submit the traffic trace. In your lab report, answer the following questions, and explain how you arrived at these answers.

Note: It is possible that some questions should be answered by "the trace does not contain the answer" because the eavesdropper cannot see the information. In fact, different websites can reveal different sets of information!
(2.a) What is the TLS version being used?
(2.b) For the first TLS session, what is the public encryption scheme, signature scheme, and hash algorithm used by the website?
(2.c) For the first TLS session, what is the HTTP version being used?
(2.d) What is the master secret of this session?
(2.e) What is the domain name of the website?
(2.f) How many total TCP connections have been established to load the webpage?
(2.g) How long did it take to lo

Use the following materials to help you:

- The Illustrated TLS Connection:
  - https://tls12.xargs.org/
  - https://tls13.xargs.org/
- RFCs:
  - TLS 1.2: https://datatracker.ietf.org/doc/html/rfc5246
  - TLS 1.3: https://datatracker.ietf.org/doc/html/rfc8446

Task 3: Identifying Traffic (30%)

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

(3.a) What are the message bodies of the two emails?
(3.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`.
(3.c) Who sent the two emails?
(3.d) What is the evidence? List all evidence you found. Also, explain the procedure you followed to find the harasser (with screenshots).

Your report should include a detailed description of your analysis with proper screenshots.

3. Submission

You need to submit:
(1) The source code that you developed for Task 1.
(2) The trace in Task 2.
(3) Any scripts you wrote (if any) to answer Task 3.
(4) The HTML page `response.html` for Task 3.
(5) A detailed lab report.
The files should be compressed in a single (.zip) archive. The code should run without any errors.