Lab 4: Wireshark and more HTTP APIs

50.012 Networks

Hand-out: October 6
Hand-in: October 13

1 Objectives

- Use Wireshark to look at TCP and HTTP communication
- Write a requests client to communicate with an HTTP API

2 Notes & Setup

- You can collaborate with another student, please hand in code individually with both authors noted in the header
- This exercise requires an HTTP API to talk to. You can use the one that you wrote in Lab 3, or ideally use the API that one of the other students of this class wrote.

3 Wireshark

- Wireshark is a very useful tool to analyze network traffic. It is available for all major OS
- It can be used to capture traffic on one of your interfaces live, or load recorded traces
- Please go through the following good Wireshark tutorial on TCP sessions: `http://packetlife.net/blog/2010/jun/7/understanding-tcp-sequence-acknowledgment-numbers/`
  - The referenced pcap file is in the eDimension folder
  - To start wireshark, just type `wireshark &` in your terminal
- The TCP handshake was not yet discussed in all details in the lecture, but is already in the slide set for L6. Have a look at it (slide 31 and 32)

4 Setting up the HTTP API

- Start up the HTTP API that you selected, and make sure it works by sending *curl* requests
- Using the Python *requests* library, write a small client to talk to the API
  - Use tutorial here: `http://docs.python-requests.org/en/v2.0-0/user/quickstart/`

  - The client should query for some data on the API, and react based on the response
  - The client should create some data on the API using POST
  - The client should also Update some data on the API using PUT
  - You can even think about querying multiple HTTP APIs and synchronize information between them (exact details depend on the APIs of course).
- Alternatively, you can extend your API from last week to query other APIs on its own, when certain resources are requested by the user.
  - The data from the other API should then be combined with this API's data to form a response to the user

5 Wireshark on HTTP and DNS traffic

- Start wireshark, this time as *root* with `sudo wireshark`
- Start a live capture on your *loopback (localhost)* interface. You will probably already see some random DNS traffic or similar.
- Now use your python script to do the automatic interaction with the http API. You should see the HTTP traffic in detail
- Now switch the capture interface to your *wlan0* device
- Use your browser to connect to a site on the internet, and observe the traffic created in Wireshark
  - Can you identify the DNS call made before the HTTP connection is established?
  - Can you identify the first HTTP GET request, and the related TCP connection?
  - Is the TCP connection terminated after the first GET request, or is it used afterwards?

6 What to Hand in

6.1 eDimension submission:

- You will submit in the complete client/server code via eDimension. You can collaborate with a friend on the code, in that case please state both your names in the comments at the start of the file. Both students will then submit the file individually.
- Please add a header at the beginning that comments on the implementation details.

6.2 Checkoff:

- Demo your Python code to the TA and explain it. In particular:
  - Explain which API paths are called by your script
  - Show how your script performs the required operations, and processed the results
  - Answer some questions on TCP/HTTP traffic in Wireshark, and HTTP keep alive