Homework 5: The Final Homework

Due: Monday, December 9 @ 11:59 pm EST

Overview and instructions

This homework has 3 short problems. You can write your responses in your own document or add annotations to this one.

Note on collaboration

You are welcome (and encouraged!) to collaborate with your peers, but the solutions you write down must be **your own work** (ie, written by you). You are responsible for independently understanding all work that you submit—after discussing a problem as a group, you should ensure that you are able to produce your own answers independently to ensure that you understand the problem. For more information, please see the course Collaboration Policy.

In your submission, we ask that you include a brief *collaboration statement* describing how you collaborated with others on each problem—see the next section for details.

How to submit

You will submit your work in PDF form on Gradesope. Your PDF should conform to the following requirements:

- Please **do not** include any identifying information (name, CS username, Banner ID, etc.) in your PDF, since all homeworks are graded anonymously
- Each problem (where "problem" is one of the Problems 1–3) should start on a separate page. When you submit on Gradescope, you will be asked to mark which pages correspond to which problem
- At the start of each problem, write a brief *collaboration statement* that lists the names and CS usernames of anyone you collaborated with and what ideas you discussed together
- If you consulted any outside resources while answering any question, you should cite them with your answer

1 DNS: by the numbers

Relevant lectures: Lectures 17–18

I issued two DNS queries to my system's local DNS server for `www.google.com` using `dig`[1]. The two queries were made a few seconds apart. Here are the responses:

Query 1:

```
;; QUESTION SECTION:
;www.google.com.   IN A

;; ANSWER SECTION:
www.google.com.     300 IN  A   172.217.12.164
www.google.com.     300 IN  A   172.217.30.100

;; Query time: 44 msec
```

Query 2:

```
;; QUESTION SECTION:
;www.google.com.   IN A

;; ANSWER SECTION:
www.google.com.     294 IN  A   172.217.30.100
www.google.com.     294 IN  A   172.217.12.164

;; Query time: 1 msec
```

Based on this information, answer the questions below. Your answers for each part should be short—one sentence should be sufficient, at most.

a) How much time elapsed between queries 1 and 2, and how do you know?

Six seconds, based on the difference in TTL. 4pts total: +2 mentions TTL, +2 correct time difference

b) Why did the second query take so much less time than the first one?

Response was cached by the DNS server 4pts total: full credit if mentions response was cached, partial credit as applicable

c) Google uses a maximum TTL value of 300 (ie, 5 minutes). `cs.brown.edu` uses a maximum of 86400 (1 day). Speculate: why might a cloud company like Google want a very short TTL?

A lower TTL reduces the amount of time the record is cached, which allows Google's DNS servers to update records more quickly, which allows Google to be more responsive in how it directs users to its servers.

4pts total. +1 mentions that shorter TTL reduces time in cache, +3 reduced caching time leads to more flexibility for Google (can send users to different servers more quickly). Partial credit given for other reasonable speculations.

[1] For examples of `dig`'s output, see the lecture notes.

2 TLS and PKI: Is this website sus?

Relevant lectures: Lectures 23–24

It's the year 2024 and you're using a browser that has exactly two root CA certificates installed[2]:

- `VeriTrust Root CA`
- `TotallySecure Root CA`

Using this browser, you connect to `https://very-secure.website`. When establishing a TLS connection, `very-secure.website` sends back the following certificates, which are checked by your browser. Here's a listing of what the certificate info might look like (with cryptographic details omitted, which aren't relevant to the problem):

```
----- BEGIN CERTIFICATE 1 -----
Common Name:  very-secure.website
Validity period:
   Issued  on:   1  January 2022 12:34
   Expires on:  31 December 2022 23:59
Issued By:  AwesomeTrust Root CA
Public Key: ...
Signature:  ...
  [Browser signature check:  Valid signature from "AwesomeTrust Root CA"]
----- END CERTIFICATE 1 -----

----- BEGIN CERTIFICATE 2 -----
Common Name:  AwesomeTrust Root CA
Validity period:
   Issued  on:   1  January 2020 00:00
   Expires on:  31 December 2030 23:59
Issued By:  AwesomeTrust Root CA
Public Key: ...
Signature:  ...
  [Browser signature check:  Valid signature from "AwesomeTrust Root CA"]
----- END CERTIFICATE 2 -----
```

Based on this information, answer the questions below. For each part, your answers may be short (no more than 1–2 sentences).

a) After checking the certificates above, your browser displays a warning that the site cannot be trusted. **Based on only the certificate info shown here, why should your browser <u>not</u> trust this site?** List any reasons you see for why the certificate check should fail. You can assume any info *not* shown is valid and out of scope (e.g., cryptography is correct and up to modern standards, certificate has not been revoked).

There are two problems: 1) Certificate 1 is expired, 2) The Root CA "AwesomeTrust" is not considered trusted by the system. (4pts total, +2 each)

b) Your friend (who doesn't have any background in CS or security) encounters the same error and asks you if it's okay to click past the warning and connect to the site anyway. **As someone who now has background on how TLS works, how would you respond? Why is this website sus?** More concretely: explain in your own words why this site should not be trusted. Your response should be *concise and targeted to an audience with only a general computing background*.

[2] These are not real CA names :)

4pts total: +2 Recognizes that site cannot be trusted and proceeding is risky; +2 Reasonable explanation for why site cannot be trusted

3 Quick tutorial: Exploring web traffic with HTTP

Relevant lectures: Lectures 17–19

This problem is a short tutorial on using Wireshark to explore HTTP traffic.

To complete the tutorial, follow the instructions here:
`https://hackmd.io/@csci1680/tutorial-wireshark-http`

At the end of the tutorial, you'll be asked to write down some observations and include some info you found while observing traffic. To receive credit for this problem, just include these items in your submission.

Note: Tutorials are meant to be short problems where you gain some hands-on practice with some networking tools or protocols that we don't otherwise have a chance to cover in our assignments. These are mostly for your benefit, are graded on completion, and should not take more than 30 minutes to complete.

If you find yourself getting stuck on part of a tutorial, please come to hours or post on Ed to let us know. We can help resolve issues and clarify instructions. Tutorials are a new part of the course, and we hope they are fun and interesting, without adding too much work!

8 pts, graded on completion of each part: +4 includes an image, +4 includes login credentials.