Homework 4: The Final Homework

Due: Friday, December 8 @ 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 18–19

I issued two DNS queries to my system's local DNS server for `www.google.com`. 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 zone?

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). Why might 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. +2 mentions that shorter TTL reduces time in cache, +2 reduced caching time leads to more flexibility for Google (can send users to different servers more quickly)

2 DNS: more than just a key-value store

Relevant lectures: Lectures 18–19, end of lecture 20

Briefly consider the following questions about how DNS can be used for more than just simple IP address lookups. Your responses should be short, at most 1–2 sentences each.

a) If you run the authoritative DNS server for a website that has lots of geographically-distributed webservers, how could you use DNS to send users to the webserver closest to them?

Based on the IP of the host performing the query, can respond with IPs of servers that are "closest" to the host (where "closest" could be determined based various metrics like geographic location, latency, etc.)

4 pts total:

(a) +2 use IP of host making query

(b) +2 map IP to "closest" webserver

b) Imagine you control an ISP. How could you attempt to use DNS to block access to sites you don't want your customers to access? If you are an ISP, you can do the following:

- Run your own DNS server that gets advertised to customers that blocks domains you don't like (ie, respond with error, or redirect to wrong server)
- Intercept/modify DNS traffic sent to your customers (ie, DNS hijacking, or outright blocking of DNS traffic)

4 pts total. Full credit for any reasonable answer for how ISP could affect DNS traffic (like those above). Partial credit if significant details missing.

3 TLS scenarios

Relevant lectures: lecture 24, start of lecture 25 (Tuesday, December 5).

Suppose that Blue University acquires a TLS certificate for its website, `blue.university`, from a CA called AwesomeTrust.

Consider the following scenarios, which examine how different parts of a public key infrastructure are affected if a certain private key is compromised. For each part, consider what an adversary could do if they manage to obtain the private key in question. Specifically:

- Who could they impersonate?
- Who would believe them? (In other words, if the attacker tries to impersonate site S, what set of users would consider the attacker's site as trusted? Everyone, or just a specific subset?)

Note: Assume the attacker already has the means (via BGP hijacking, DNS spoofing, etc.) to redirect users to a fake website. For this problem, just **focus on how the TLS authentication process would be affected if they have the key**.

For each part, your answer should be short—one sentence should be sufficient.

a) Suppose the attacker obtains the private key for the `blue.university` webserver. What capabilities do they have? Give your answer by considering the two bullet points above and briefly explain your reasoning.

You can impersonate `blue.university` for anyone who connects to the website. (2pts, +2 for identifying who to impersonate, +2 for identifying who would believe you; half credit if intuition is close but something significant is missing)

b) What if the attacker obtains the AwesomeTrust CA private key instead? Give your answer by considering the two bullet points above and briefly explain your reasoning. Adversary can now sign their own certificates, so they can now impersonate any website for any user (at least until AwesomeTrust gets revoked or removed from browsers).

(4pts, +2 for identifying who to impersonate, +2 for identifying who would believe you; half credit if intuition is close but something significant is missing. Not required to state that AwesomeTrust would get revoked eventually.)

c) Suppose Blue University gives up on AwesomeTrust and decides to *make its own CA* and issue a certificate for `blue.university`. To make this work, the University's IT policy requires all users to install the Blue University CA certificate on their systems.

i) Why do Blue University users need to install the CA certificate? What happens if a browser **doesn't** have the CA certificate installed and connects to `blue.university` anyway? If the CA certificate isn't installed, browsers will flag it as not trusted because they can't verify `blue.university`'s public key. If the CA isn't installed, this will display a warning when the user tries to connect.

(3pts, +2 for recognizing that certificate is not trusted, +1 for noting that browser will display warning; half-credit as applicable)

ii) What happens if the attacker then obtains the Blue University CA private key? Give your answer by considering the two bullet points above and briefly explain your reasoning. You can now sign your own certificates and thus impersonate any website, and any user who has the CA certificate installed (ie, all Blue University students/employees) will believe you.

(4pts, +2 for identifying who to impersonate, +2 for identifying who would believe you; half credit if intuition is close but something significant is missing.)