Week 10

This is CS50. Harvard University. Fall 2015.

Anna Whitney

Table of Contents

1. Announcements

- In recent news, Russia may be planning on attacking undersea cables that carry more than 95% of daily Internet communication. Today's lecture will be about security, mostly digital rather than physical, but it gives you some idea of what kinds of physical threats can exist.

- CrimsonEMS is providing a free EMT training course in Spring 2016, applications due November 30. All of CS50's staff are trained in CPR, and you should consider getting trained too - it could be life or death for someone.

- Last CS50 Lunch Friday 11/13.

- We will be starting a series on AI with Scaz, our CS50 professor at Yale, something to particularly pay attention to if you're interested in doing a robotics or AI-related final project.

- On Wednesday 11/11, Scaz will be at Harvard to talk about Computation for Communication, focusing on how to build systems that use language to communicate.

- Next Monday, 11/16, we will be at Yale, where Scaz will lecture on AI Opponents in Games.

- Quiz 1 will be Wednesday 11/18.

- The final lecture at Yale will be Friday, 11/20 (followed by cake!) while the final lecture at Harvard will be Monday, 11/23 (also followed by cake!)

- Upcoming final project milestones:
    - Status report - a quick sanity check for you and your TF
    - CS50 Hackathon (in Cambridge - buses from Yale)
    - Implementation
    - CS50 Fairs (in Cambridge and New Haven)
- We have special drawing software for our giant touchscreen from Microsoft, because our previous 99 cent drawing app had a lousy user interface, so Bjorn, a developer at Microsoft, created a very simple drawing program with much less clutter.
- David is also using an experimental device for controlling the slides created by another Microsoft developer.

2. Security

- We want to equip you to make better technological decisions, given how widespread misunderstandings of how technology works are, especially among politicians.
- Bertucci's WiFi is labeled "Secure Internet Portal", so it must be secure, right?
    - As it happens, there's no encryption - if there were, you'd have to type in a password.
    - There's actually nothing protecting your information on this WiFi network!
    - Even more worrisome is this, in the Terms and Conditions: "You understand that we reserve the right to log or monitor traffic to ensure that these terms are being followed."
        - This is pretty common language for those access agreements that you blindly click through!
    - Because there's no encryption, someone else on the network could "sniff" or inspect the contents of the packets on the way to your computer. They could see the URL you're trying to access (domain name as well as path).
- What other threats exist along these lines?
    - Smart TVs have voice sensors, Internet connectivity, and often cameras, and these sensors are always on - they're listening for voice commands, but they can't always interpret your commands locally on the TV, so they're sending everything you say to the distributor's servers to detect whether a command was given.

  - This means that *everything you say* in front of your TV could be broadcast to Samsung’s (for example) servers.

  - Google & Apple mitigate this problem by requiring that commands start with "OK Google" or "Hey Siri", but this is imperfect, because it can’t always perfectly detect what was said, and natural language processing is difficult.

  - If you read the terms & conditions of some of these smart TVs, there’s specific language indicating that you might not want to have personal conversations in front of your TV - but who really pays attention to that?

  - We take for granted that our phones aren’t constantly recording us, but they very well could be! We have GPS transponders, microphones, and cameras on us at all times.

- A few months ago, a third-party service to save Snapchats was compromised, releasing 90,000 Snaps to the Internet.

- A few days ago, a ransomware program (a virus that encrypts a user’s data and demands payment for the key) was found to have deleted the encryption keys for the users' data, because the code was buggy, rendering all the victims' data permanently unrecoverable.

- Last week, there was a bug at Harvard, where Harvard’s monitoring and filtering software went haywire and decided that all websites were suspicious.

- On a more political note, currently in the UK, a new surveillance bill is being considered, with a recent article noting that "the new surveillance bill renders the citizen transparent to the state, putting every one of us under suspicion. It would serve a tyranny well."

  - The UK government is proposing to store the domain of every site visited by every person in the UK for a year, and possibly make that data available for security or police purposes.

  - Proponents of the bill claim that this Internet record is "no different from an itemized phone bill", but as John Oliver points out, that’s not exactly reassuring.

- The information the UK government is proposing to collect would come straight out of the HTTP requests that we’ve discussed over the course of the last few problem sets, such as this:

```
GET / HTTP/1.1
```

```
Host: badplace.com
```

- # The specific page could be recorded from an HTTP request as well as the domain, if Internet traffic is being routed through a government server (as is the case explicitly in some nations, and even here has been conducted in secret via major ISPs cooperating with the government).

3. Defenses

- How can we protect ourselves? Well, password-protection is a classic answer. But any of you who did the Hacker edition of problem set 2 should recall that even properly encrypted passwords can be cracked with some effort (often a lot of computer time and a brute-force dictionary attack).
  - # Short, simple passwords are very easy to guess using software that checks lots of likely options.
  - # An attacker can load a huge dictionary into memory and then guess and check against the hash.
  - # **Salting** a password at least ensures that the hashes of similar passwords won't be too similar.
  - # A common solution to complicate passwords is to replace letters with similar-looking numbers, but this doesn't actually help much, because a dictionary attack can just add an extra loop to replace those characters in the dictionary words.
  - # There are better ways of generating passwords[1], but the best defense for passwords is something much more random.
- Sometimes attempts to get passwords don't come in the form of cracking a stolen hash, but instead via **phishing**, or trying to get a user to give you their password by pretending to be a legitimate entity.
  - # There are usually telltale signs in these emails (poor spelling, grammar, Gmail Spam indicator), but they're extremely cheap to send out using botnets (malware that sends spam from many other people's computers), so they only have to work on a tiny number of un-savvy users to be worth sending.

[1] https://xkcd.com/936/

  # It’s generally poor practice to follow a link in your email even if it looks legitimate, instead typing out the URL by hand to make sure it doesn’t go somewhere you don’t intend to go.

- Lots of us use the same password for multiple sites, which is a problem because then if it’s compromised on one site, it’s compromised everywhere. It’s easier for us to remember simple or repeated passwords, but from a security perspective, this is problematic.

  # One solution is to use a **password manager** like LastPass[2] or 1Password[3], a piece of software that generates big, pseudorandom passwords that a human couldn’t possibly remember, which are then encrypted on your hard drive, and all you as the human need to remember is a single "master" password (which should be a long string, like a sentence, but doesn’t need to be random).

  # These long, random passwords are resilient to dictionary attacks, and if they’re different everywhere, one being compromised doesn’t reveal all the others.

  # There are potential downsides of a system like this, though:

    # If your master password is compromised, an adversary could get into all your accounts (though they’d likely still need access to your physical computer).

    # If you’re using a different computer than usual, you won’t be able to log in to any of your accounts, because your passwords are stored on your machine.

  # We have a site license for 1Password, so you can sign up for free.

  # LastPass does have a version where your passwords are stored in the cloud, but then if you log in from a different computer, you could be vulnerable to a keylogger getting your master password.

- Another option is **two-factor authentication**, where to log in, you need something you know (like a password) and something you have (like your phone or another device).

  # Rather than just use a password, you need physical access to the device, which is much less likely to be a vulnerability.

[2] http://lastpass.com

[3] http://agilebits.com/onepassword

  - Mid-semester, Yale implemented two-factor authentication, whereby to log into your Yale account, you must type in not only your password but also a code that is pushed to your phone.
  - The downside of this is if you lose the device, or if you don’t have cell service and can’t receive the message, you can’t log in.
    - These inconveniences can cause users to disable extra security settings altogether, which is problematic.
  - Google provides two-factor authentication for your account, which you can opt into in the security settings[4] of your account.
    - Google does a good job of balancing security vs. convenience - you can set it so you only have to go through the 2-factor authentication process when logging in from a new computer, or every 30 days on your own computer.
  - Similarly, Facebook[5] also provides 2-factor authentication.
  - The code that is sent each time you use 2-factor authentication expires periodically (usually every few seconds to a minute), so if someone grabs the code you use, they still can’t access your account.
- Another, more background approach uses **audits**, where we have a mechanism to at least know if an account has been compromised.
  - For example, Facebook can notify you if someone logs into your account from a different computer than usual.
  - This means that after an account is compromised, you can narrow the window of time in which the attacker has access to your account and can do bad things with it.
- Other types of threats can be more difficult for us as the end users to guard against, like **session hijacking**, in which an attacker copies your cookies to impersonate you on the Internet.
  - If we go to the CS50 Finance[6] site and watch the network traffic using Chrome’s Network tab, we can see that the HTTP request includes a **Set-Cookie** parameter.

[4] http://google.com/settings/security
[5] http://facebook.com/settings?tab=security
[6] http://finance.cs50.net

- - This parameter contains a value called `PHPSESSID` (PHP session id) that indicates to the site who you are.
  - On subsequent requests to the site, there’s instead a **Cookie** parameter, containing the same `PHPSESSID`.
  - This `PHPSESSID` basically functions like a virtual hand stamp, as you might get to allow reentry at a club or amusement park.
  - CS50’s sites are encrypted - we access them using the HTTPS protocol - but if there were an unencrypted site, an adversary could intercept the packets, see these headers, and find out what your `PHPSESSID` value is. They could then pretend to be you by presenting that same `PHPSESSID` to the website.
  - One of the nice features of LastPass and 1Password is that in addition to storing your passwords, they will warn you if you try to log into a site that isn’t properly encrypted.
- Another way that HTTP headers can be used against us as users is that Verizon, AT&T and the like can insert into all HTTP requests from your phone an additional parameter called **X-UIDH** that lets them uniquely identify all your web traffic.
  - This lets them sell your profile of what sites you visit to advertisers.
  - Even if you delete cookies or use incognito mode, you can still be clearly identified, because Verizon or AT&T is acting as the "man in the middle" and adding this information to all your requests.
  - There are ways to opt out of this now, but initially, the only way to avoid this data being injected into all your HTTP headers was to leave the carrier entirely.
- Websites often use padlock icons and the like to indicate that a login is secure, but anyone can put a padlock and the word "secure" on their site - it means literally nothing. What *does* mean something is that a site is using HTTPS, so that’s what you should actually look for.