Introduction

[00:00] (30 seconds)

- Who I am (Keefer)
- What I do (Undergraduate research assistant at UoG School of Computer Science)
    - built an anonymous microblogging platform while in highschool
- Among other things, this talk is about anti-patterns in web development
    - hopefully serve as a primer for the implications of internet security and privacy

History

[00:30] (60 seconds)

- I want to start off with an *admittedly* shitty history of computing - I think it's important to have an understanding of how we got here:

1. A long time ago, a guy named George Boole formalized a set of rules for binary symbolic logic;
2. Then, a little less long ago, digital circuitry was invented that implemented this logic in really clever ways. Suddenly we had computers!
3. Skipping some corporate drama, eventually everyone had a computer in their homes...
4. And their pockets.

This point in the time-line marks a so-called INFORMATION REVOLUTION, and that's *almost* where we are today.

5. Today we are drowning in applications and services all vying for your attention to turn a profit; they do this using content aggregators, targeted advertising, and downright spying. Today you have no privacy, and this is bad.

[Huxley/Orwell slide] (30 seconds)

The internet was designed as a tool to allow people to share content and knowledge with almost no restrictions, but it has also enabled mass surveillance at an unprecedented scale. This has to do with a careful balance of privacy, and power; and now privacy does not practically exist.

I want to reverse this reality.

Privacy Oriented Design

[01:30] (30 seconds)

Definition: *an approach to development that minimizes extraneous information collection and maximizes user-privacy.*

- Everyone should be concerned about each others' right to privacy
- If you take away one's privacy; then you have explicitly created a power imbalance

Design example: the radio [02:00] (30 seconds)

- A simple example of a privacy-oriented design is the radio
- Simple protocol: select a frequency; broadcast message ; wait for response if any
- Favourite way to communicate because:
  - It's anonymous by design; you have to verbally identify who you're speaking to
  - There's no metadata… just your Tx/Rx frequency and your voice
  - Awesome.

Contrast to HTTP request [2:30] (90 seconds)

- Let's talk about the internet! Things get messier here and the protocol is a lot different.
- HTTP requests work kind of like initiating a handshake with a stranger by throwing a pile of papers at them.
- We have a client (like your laptop) which sends a bunch of information at a server and asks it to do something with some small subset of that information
- So to explain what's going on here… You have a request header which details who you are (User-agent) and where you came from (the "Referer")
  - From this truncated UA string, we've already identified that I'm using a Mozilla-like browser, my operating system and processor architecture, and what display server I'm using – for you Linux nerds, no I haven't switched to Wayland yet. And that's not even half of it.
- And then there's a huge mess of stuff that your browser either sends to a server or makes easy to query by a web application:
  - This data may include a list of browser add-ons you have installed, unique browser hashes, details about your display… etc.
- If you're curious about this stuff; then you can visit the Electronic Frontier Foundation's Panopticlick tool – when I ran a test with a fresh Chromium installation, I was uniquely identifiable out of well over a million browsers that have been tested.
- HTTP lets you do a lot more with communications; and you can have private and secure connections…
  - but first you need to hand over a copy of your fingerprints, your home address, and your passport………….. to ever server you connect to…………. ever.

Datensparsemkeit

[04:00] (120 seconds)

- One of the most profound things I've heard in the past year is that "the only safe data set is the null data set"; that is to say that data is so abstract, ubiquitous, and easy to manipulate, that is easy to cause harm with it
- The German word on display here, *Datensparsemkeit,* loosely translates to "the practise of keeping and using only the required data in a set"; it's basically the opposite of *Big Data* and LOGGING ALL THE THINGS, and it's a philosophy that I hope many of you will consider seriously.
- When you collect and store hundreds or trillions of points of data, that starts raising red flags about your intent.
- In an absurd example, Facebook sometimes asks for users to provide a copy of their government ID when they are locked out of their accounts
    - Feels sketchy that deep profiling should suddenly be attached to real identification
- *Why do you want to uniquely identify me?* Why do you want to know my political alignment? *Why do you want to know my activity on the internet?*
    - Are you trying to dissuade or disallow certain speech (a reality across much of the globe right now as political powers shift)... or just sell me shoes?
- Martin Fowler wrote an excellent blog post on this subject back in 2013, which I'll quote here:

  "The default attitude at the moment is that any data you generate is not just freely usable by the capturer but furthermore becomes their valuable commercial property."
    - This is more true than ever today, and perhaps more dangerous as poor assumptions are often made using imprecise metadata and pseudo-science as a proxy for reality

[Quick NGINX example] [06:00] (30 seconds)

- Putting your data on a diet can be tricky in some cases, but for most simple web applications it might be as simple as disabling access logging and toning down your analytics software

User/Host Power Differential

[06:30] (60 seconds)

- Don't lie; don't cheat; don't cause harm; don't take advantage of those who are less fortunate than you
- This is a moral code that most of us are taught when we are young, is common sense, and forms the basis of most social contracts; where in general we expect people to be nice to one another.
- In positions of power however, it is particularly easy to get away with ignoring this moral code
- Consider a scenario:
    - you build a system that people trust with important personal information (CC numbers, passwords, email addresses, networks of contacts, etc)
    - you, as the maintainer and administrator have the responsibility to:
        1. protect this info from incessant attackers to the best of your ability
        2. ensure the integrity of this info
        3. avoid looking at this info personally
        4. protect this info from insiders by implementing levels of access
- Failure to do any of these things is potentially abuse by negligence, kinda like what happened with Equifax last year, and countless other cases of bad security.
- It's also about a lot more than just security; you need a good sense of ethics if you're going to do anything with user-generated content
    - you, as the administrator have the responsibility to:
        1. ensure that you are treating users ethically and providing equal opportunity
        2. ensure fair moderation systems
        3. ensure there is no potential for abuse through moderation
        4. ensure that minority voices are not systemically silenced

Zelda slide [7:30] (10 seconds)

- Reality check: the internet is scary and it's dangerous to do web dev alone.
- Everyone screws up.

Causing accidental harm [7:40] (40 seconds)

- Facebook is perhaps the most famous system of content management
- In 2014 it introduced "Your Year in Review", where certain posts were highlighted using engagement as a proxy for content filtering
- It failed to take into account context, and people were shown painful memories in an inappropriate manner
- On the right, a Facebook user was reminded that their apartment was destroyed in fire… And there were worse examples.
- Lesson: Don't presume that people all have the same world view and experiences; the human experience is dynamic, and problems like this one are ignorant at best

Abusive anti-patterns [08:20] (20 seconds)

- Lying to your users, like telling them that location services are off, when in fact, they are still on.
  - This was an admitted problem with Google Android in 2017
- Aggressive user profiling and tracking (which can be alienating and is invasive)
- Introducing hidden "anti-features", for example: mining Bitcoin in a browser ad
- Treating people as a commodity and selling private information
- Doing really anything without explicit consent

The Censorship Machine

[08:40] (80 seconds)

- You see censorship crop up on social media all the time
  - Combating trolls, removing hate speech, complying with local, national, and international content laws…
- Censorship and filtering need to be implemented carefully
- Consider a scheme which blocks users from making a post which contains certain keywords that are flagged as hate speech
  - On the surface, this seems like a good way to prevent the spread of hateful messages
  - Now consider that people are usually smarter than systems, and can get around this scheme by using alternative spellings, creative acronyms, and other media-types to spread their message
  - This scheme is suddenly rendered ineffective, and all its done is blocked potentially productive conversation on topics like dealing with systemic discrimination or political unrest

- If you've made a communications platform, and it prevents people from communicating controversial ideas, it inherently prevents social action against injustices
- This is a slippery slope.
- Considering major platforms already use mechanisms to spy on your internet activity and censor your speech; we need to tread very carefully

[change slide]

- Issue trackers and venues for obtaining voluntary user feedback are critical for ensuring fairness in online platforms
    - This is something that free / open-source software development philosophies are good at

If I can't disrespect my users…

[10:00] (60 seconds)

Closing Notes

[11:00]