

A Whirlwind Tour

Matthew Trost

CS107e Computer Systems from the Ground Up

May 23, 2022

A Whirlwind Tour

Exposure to important topics...
...with some fun intermissions



- Memory Hierarchy
- Memory Safety
- Trust and Safety
- Where to Go From Here

A Whirlwind Tour

Exposure to important topics...
...with some fun intermissions



- **Memory Hierarchy**
- Memory Safety
- Trust and Safety
- Where to Go From Here

Where can a computer store data?

Registers ← fast but you only get ~16 integers

slow but you get multiple GB → **Memory**

How slow?

1 cycle to access/use data in registers...

Where can a computer store data?

Registers ← fast but you only get ~16 integers

slow but you get multiple GB → **Memory**

How slow?

1 cycle to access/use data in registers...

... >**100 cycles** to access/use data in memory.

The Problem

- We want: lots of memory and access it fast
- We really have: different speed/size tradeoffs
- Need methods to give illusion of large and fast memory

Locality saves the day

- Temporal locality

- The concept that a resource that is referenced at one point in time will be referenced again sometime in the near future.

- Spatial locality

- The concept that likelihood of referencing a resource is higher if a resource near it was just referenced.

Taking Advantage of Locality

How can we exploit locality phenomena?

- Temporal locality

- The concept that a resource that is referenced at one point in time will be referenced again sometime in the near future.

- Spatial locality

- The concept that likelihood of referencing a resource is higher if a resource near it was just referenced.

Hint: what if you had an intermediate data store, faster than memory but slower than registers, with more space than registers but less than memory?
How would you use it?

Marc Snir;

<https://snir.cs.illinois.edu/PDF/Temporal%20and%20Spatial%20Locality.pdf>

Stanford University

Taking Advantage of Locality

- Goal: get performance (speed) of cache, with the capacity (size) of main memory.
- Techniques:
 - **Caches** exploit temporal locality.
 - **Cache Line Size and Prefetching** exploit spatial locality.

Caches

- Caches are memory arrays that store copies of recently/frequently used data (or instructions) close to the processor.
 - Closer to processor -> faster access.
- Many considerations and issues when designing caches.
 - Most decisions trade off between size and speed.
 - What to do when the cache fills up?
 - Take EE180 and EE282 to learn more!
- Caching is everywhere in systems!
 - Not just a hardware thing at the processor design level.
 - Web browsers cache web pages since memory is faster than network.
 - File systems cache files since memory is faster than disk.
 - You keep your most-frequently used notebooks on your desk.

Using the Cache

- CPU needs data at address.
- Is data in cache?
 - Yes? Great. “**Cache hit**”
 - No? Fetch data from main memory, place into cache. “**Cache miss**”
 - If cache is full, evict something already in cache.



Taking Advantage of Locality

- Goal: get performance (speed) of cache, with the capacity (size) of main memory.
- Techniques:
 - Caches exploit temporal locality.
 - **Cache Line Size** and **Prefetching** exploit spatial locality.

Latency Numbers Every Programmer Should Know

L1 cache reference	0.5 ns
Branch mispredict	5 ns
L2 cache reference	7 ns
Mutex lock/unlock	25 ns
Main memory reference	100 ns
Compress 1K bytes with Zippy	3,000 ns = 3 μ s
Send 2K bytes over 1 Gbps network	20,000 ns = 20 μ s
SSD random read	150,000 ns = 150 μ s
Read 1 MB sequentially from memory	250,000 ns = 250 μ s
Round trip within same datacenter	500,000 ns = 0.5 ms
Read 1 MB sequentially from SSD*	1,000,000 ns = 1 ms
Disk seek	10,000,000 ns = 10 ms
Read 1 MB sequentially from disk	20,000,000 ns = 20 ms
Send packet CA->Netherlands->CA	150,000,000 ns = 150 ms

Assuming ~1GB/sec SSD

Numbers from 2012. Jeff Dean; <https://gist.github.com/hellerbarde/2843375>

Multiplying by a Billion...

L1 Cache Hit	0.5s	One heartbeat
L2 Cache Hit	7s	Long yawn
Main Memory Reference	100s	Brushing your teeth
SSD random read	1.7 days	Weekend
Read 1MB sequentially from memory	2.9 days	Long weekend
Disk Seek	16.5 weeks	1.5 quarters at Stanford
Read 1MB seq. from disk	7.8 months	

Two very similar programs...

```
static const int SIZE = 32768;
int matrix[SIZE][SIZE];

int main() {
    for (int i = 0; i < SIZE; i++) {
        for (int j = 0; j < SIZE; j++) {
            matrix[i][j] = 47;
        }
    }
    return 0;
}
```

```
static const int SIZE = 32768;
int matrix[SIZE][SIZE];

int main() {
    for (int i = 0; i < SIZE; i++) {
        for (int j = 0; j < SIZE; j++) {
            matrix[j][i] = 47;
        }
    }
    return 0;
}
```

Two very similar programs...

```
static const int SIZE = 32768;
int matrix[SIZE][SIZE];

int main() {
    for (int i = 0; i < SIZE; i++) {
        for (int j = 0; j < SIZE; j++) {
            matrix[i][j] = 47;
        }
    }
    return 0;
}
```

```
static const int SIZE = 32768;
int matrix[SIZE][SIZE];

int main() {
    for (int i = 0; i < SIZE; i++) {
        for (int j = 0; j < SIZE; j++) {
            matrix[j][i] = 47;
        }
    }
    return 0;
}
```

Violation of Locality

```
static const int SIZE = 32768;
int matrix[SIZE][SIZE];

int main() {
    for (int i = 0; i < SIZE; i++) {
        for (int j = 0; j < SIZE; j++) {
            matrix[i][j] = 47;
        }
    }
    return 0;
}
```

```
✓ ~/coding/cs107e_ta_f21/your_lecture % time ./locality
./locality 2.55s user 1.22s system 99% cpu 3.790 total
```

2.55 seconds

```
static const int SIZE = 32768;
int matrix[SIZE][SIZE];

int main() {
    for (int i = 0; i < SIZE; i++) {
        for (int j = 0; j < SIZE; j++) {
            matrix[j][i] = 47;
        }
    }
    return 0;
}
```

```
✓ ~/coding/cs107e_ta_f21/your_lecture % time ./nonlocality
./nonlocality 22.42s user 1.43s system 98% cpu 24.323 total
```

22.42 seconds

Why not just increase the cache size?

- Really big caches?

Why not just increase the cache size?

- Really big caches? **Too expensive...**

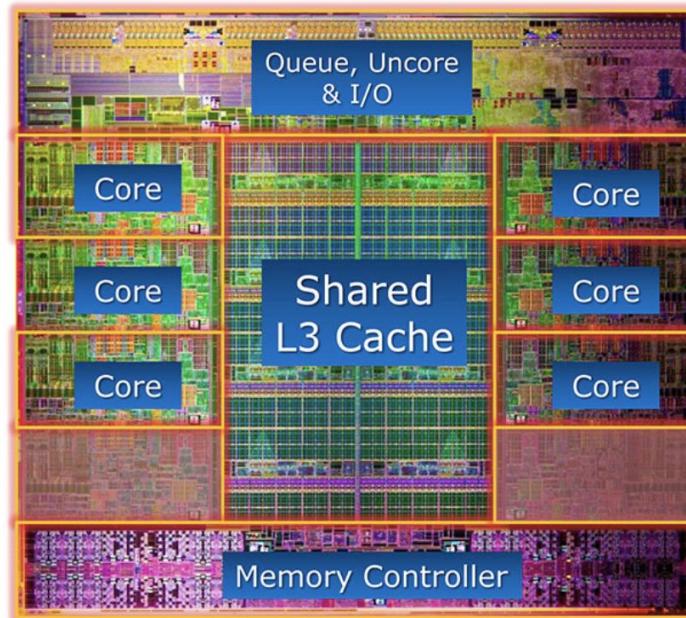
Technology	Access Time	Price per GB
SRAM semiconductor	0.5-2.5ns	\$300
DRAM semiconductor	50-70ns	\$6
Flash semiconductor	5,000-50,000ns	\$0.40
Magnetic disk	5,000,000-20,000,000ns	\$0.02
Magnetic tape	-	\$0.008

(prices from 2018)

Why not just increase the cache size?

- Really big caches? **Where would we put them?**

Intel® Core™ i7-3960X Processor Die Detail



[Hidden Slide] Making the Most of “Downtime”

- Despite our best efforts, CPU will need to wait for main memory (or worse, disk or network I/O!)
- Techniques:
 - **Multithreading**
 - **Speculative Execution**

Intermission: Sneakernet

Intermission: Sneakernet

Never underestimate the bandwidth of a station wagon full of tapes hurtling down the highway.
—Andrew Tanenbaum, 1981

Intermission: Sneakernet

"We have a number of machines about the size of brick blocks, filled with hard drives.

"We send them out to people who copy the data on them and ship them back to us. We dump them on to one of our data systems and ship it out to people."

-informal Google program, ca. 2007

Intermission: Sneakernet



AWS Snowmobile is an Exabyte-scale data transfer service used to move extremely large amounts of data to AWS. You can transfer up to 100PB per Snowmobile, a 45-foot long ruggedized shipping container, pulled by a semi-trailer truck. Snowmobile makes it easy to move massive volumes of data to the cloud, including video libraries, image repositories, or even a complete data center migration. Transferring data with Snowmobile is more secure, fast and cost effective.

A Whirlwind Tour

Exposure to important topics



- Memory Hierarchy
- **Memory Safety**
- Trust and Safety
- Where to Go From Here

Memory Safety

Memory Safety Errors include:

- Buffer overflows
- Double free
- Use-after-free
- Not checking malloc() for NULL
 - Error-checking other system calls is also important
- Using uninitialized memory
- and more...

Memory Safety

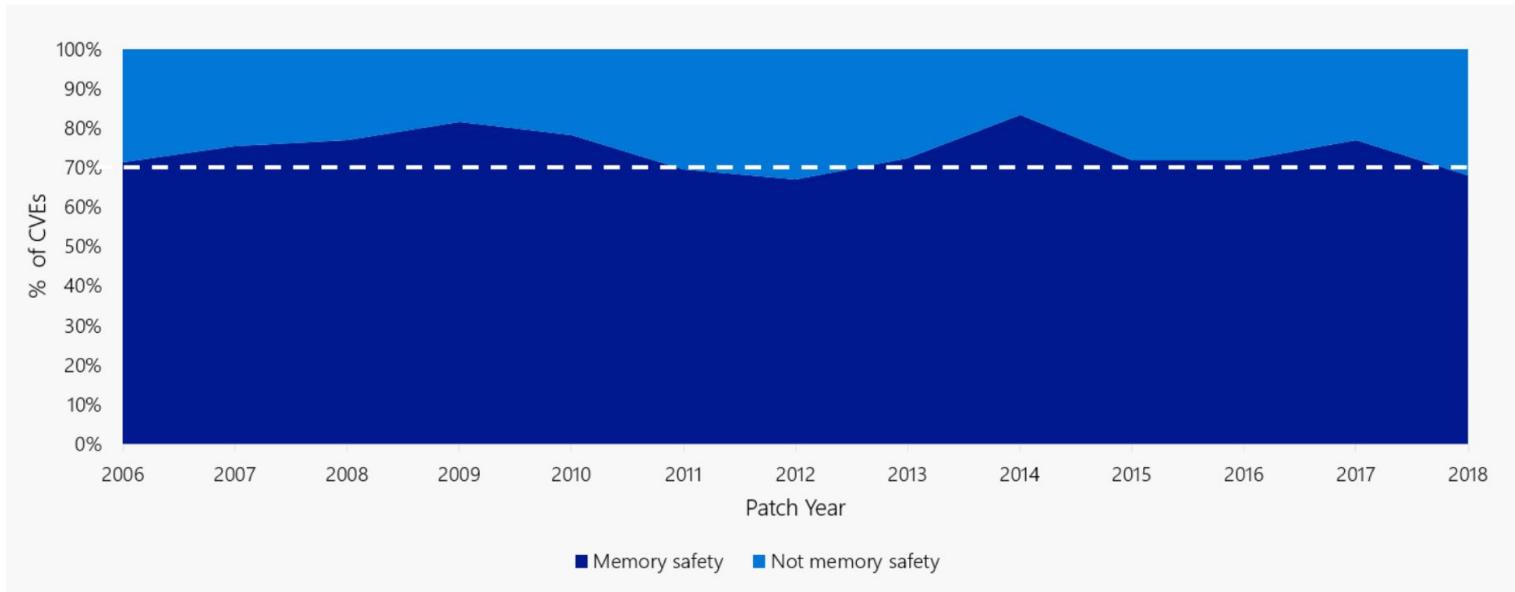


Figure 1: ~70% of the vulnerabilities Microsoft assigns a CVE each year continue to be memory safety issues

<https://msrc-blog.microsoft.com/2019/07/16/a-proactive-approach-to-more-secure-code/>

<https://msrc-blog.microsoft.com/2019/07/18/we-need-a-safer-systems-programming-language/>

Memory Safety

Banned functions in git codebase:

<https://github.com/git/git/blob/master/banned.h>
<https://devblogs.microsoft.com/oldnewthing/2005107-00/?p=36773>

```
11 #define BANNED(func) sorry_##func##_is_a_banned_function
12
13 #undef strcpy
14 #define strcpy(x,y) BANNED(strcpy)
15 #undef strcat
16 #define strcat(x,y) BANNED(strcat)
17 #undef strncpy
18 #define strncpy(x,y,n) BANNED(strncpy)
19 #undef strncat
20 #define strncat(x,y,n) BANNED(strncat)
21
22 #undef sprintf
23 #undef vsprintf
24 #ifdef HAVE_VARIADIC_MACROS
25 #define sprintf(...) BANNED(sprintf)
26 #define vsprintf(...) BANNED(vsprintf)
27 #else
28 #define sprintf(buf,fmt,arg) BANNED(sprintf)
29 #define vsprintf(buf,fmt,arg) BANNED(vsprintf)
30 #endif
31
```

Memory Safety

GETS(3)

Linux Programmer's Manual

GETS(3)

NAME

gets - get a string from standard input (DEPRECATED)

SYNOPSIS

```
#include <stdio.h>

char *gets(char *s);
```

DESCRIPTION

Never use this function.

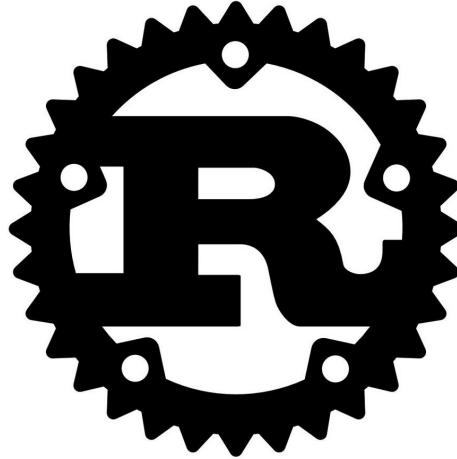
`gets()` reads a line from `stdin` into the buffer pointed to by `s` until either a terminating newline or `EOF`, which it replaces with a null byte ('\0'). No check for buffer overrun is performed (see BUGS below).

RETURN VALUE

`gets()` returns `s` on success, and `NULL` on error or when end of file occurs while no characters have been read. However, given the lack of buffer overrun checking, there can be no guarantees that the function will even return.

A note about Rust...

Rust offers C/C++ level performance with powerful compile-time memory safety checks.



**The Rust
Programming
Language**

[https://stackoverflow.com/questions/36136201/how-does-rust-guarantee-m
emory-safety-and-prevent-segfaults](https://stackoverflow.com/questions/36136201/how-does-rust-guarantee-memory-safety-and-prevent-segfaults)

<https://stanford-cs242.github.io/f18/lectures/05-1-rust-memory-safety.html>

<https://www.rust-lang.org/>

[Hidden Slide] Stack Buffer Overflow Example

- [whiteboard if time] Overwriting saved lr on stack to direct program flow to arbitrary address.
- Take CS155 for more!

Intermission: Internships

- <https://forum.stanford.edu/>
Resume reviews, mock interviews, hosted tech talks, career fairs
- Ask friends for internal referrals
- If you are interested in your professor's research field, ask them what companies exist in the space that you may be able to reach out to.
- In my experience, recruiting is a black box.
- Cracking the Coding Interview
- CURIS, REU, GEI, SGS

A Whirlwind Tour

Exposure to important topics...
...with some fun intermissions



- Memory Hierarchy
- Memory Safety
- **Trust and Safety**
- Where to Go From Here

Trust and Safety

Cybersecurity: Protect against technical attacks that make systems do things they are not designed to do.

- e.g. computer viruses

Trust and Safety

Cybersecurity: Protect against technical attacks that make systems do things they are not designed to do.

- e.g. computer viruses

Trust and Safety: Protect against abuse – systems doing what they were designed to do, but used in a way that harms people.

Unique Aspects of Trust and Safety

1. The study of how people abuse the internet to cause harm.
2. Often using products the way they are designed to work.
3. Crosses between specialties.
Requires understanding of society and humanity.
4. Is dynamic and unpredictable.

Trust and Safety

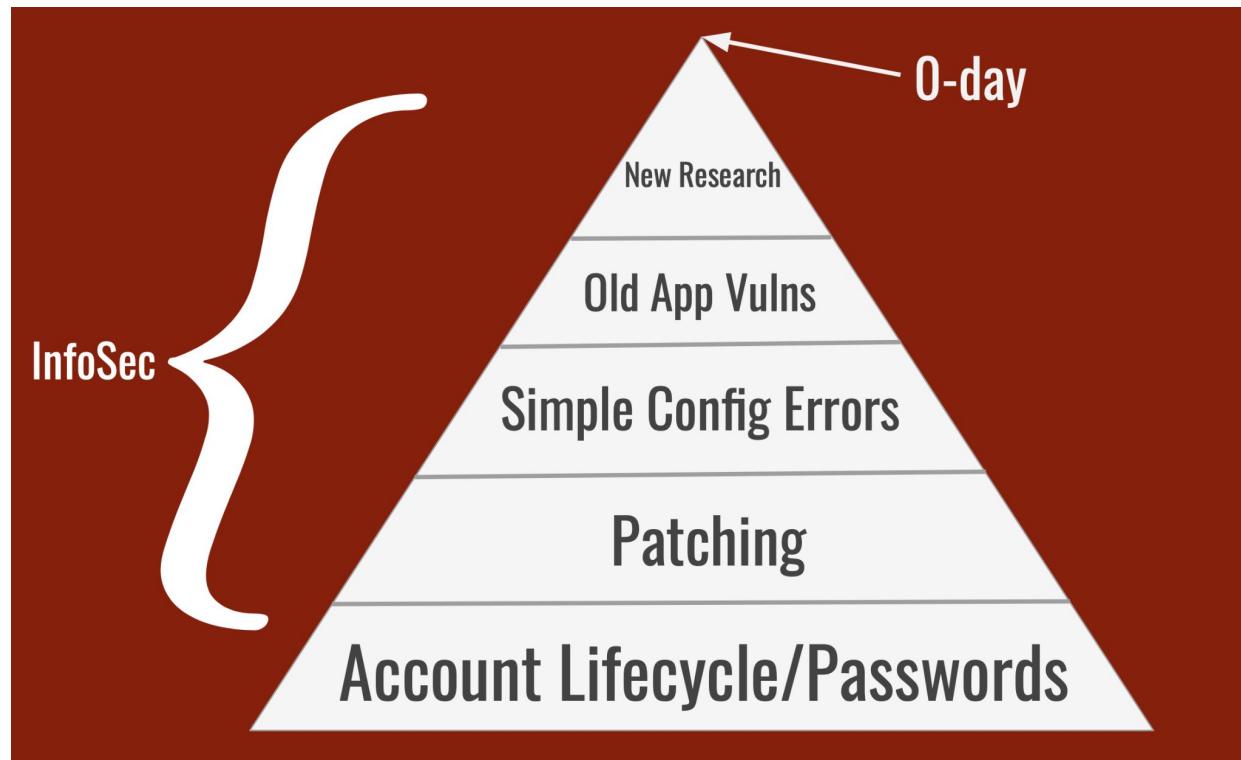
Protect against people using systems *as designed*, but in ways that harm people.

e.g. social media platforms ~~could be~~ **are** used to:

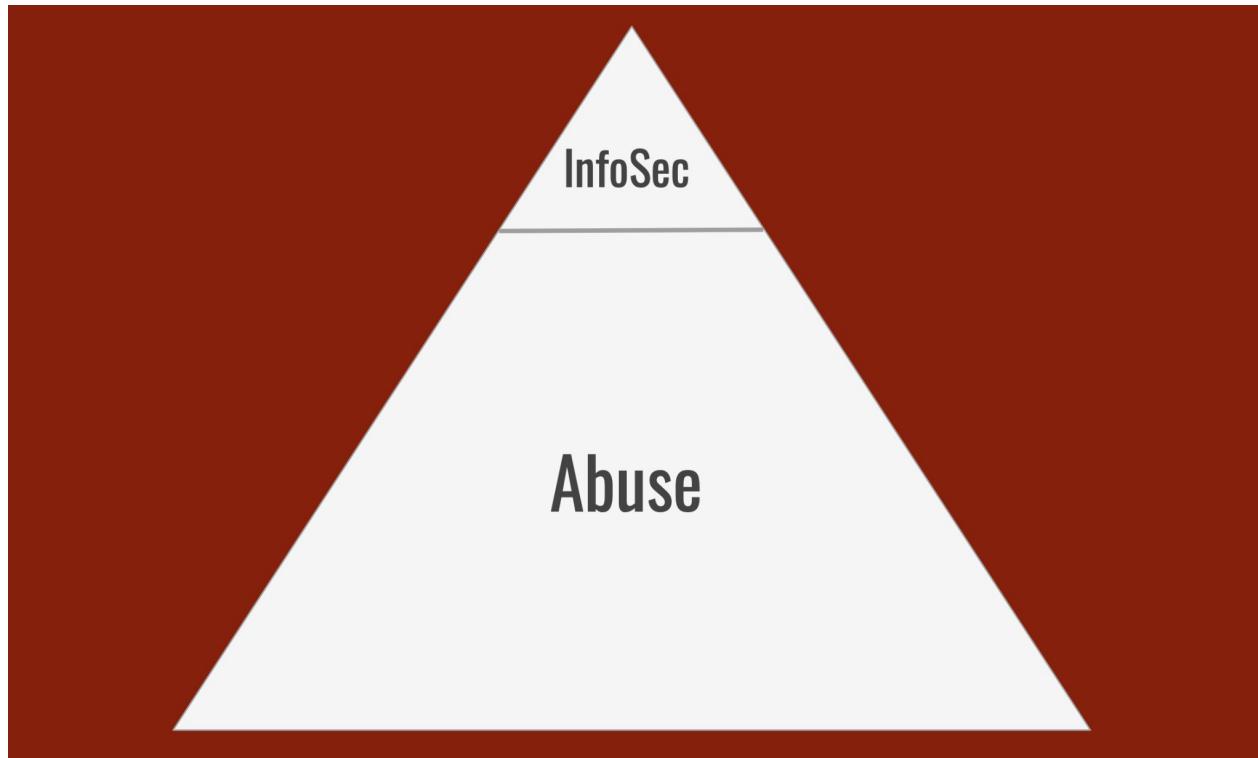
- Spread disinformation (ask me if interested in more)
 - From political influence campaigns...
 - ...to enabling genocide in Myanmar
- Coordinate harassment/violence against individuals
 - Doxxing
- Recruit and radicalize violent extremists
- Disseminate CSAM, conduct sextortion

Trust and Safety: Motivation

Trust and Safety: Motivation



Trust and Safety: Motivation



Trust and Safety: What can we do?

- Understand common ways technologies (esp. Internet tech) are used to cause harm.
- **Consider Trust and Safety concerns when designing and building your products.** Think through impact of new tech on users and other groups.
- Design user interfaces and features to minimize abuse potential.
- Provide support channels (e.g. reporting) to address abuse when it appears.
- Have empathy for a broad cross-section of the people who use your products and the risks they face.

Trust and Safety: Case Study



Feature: Display distance from your match.

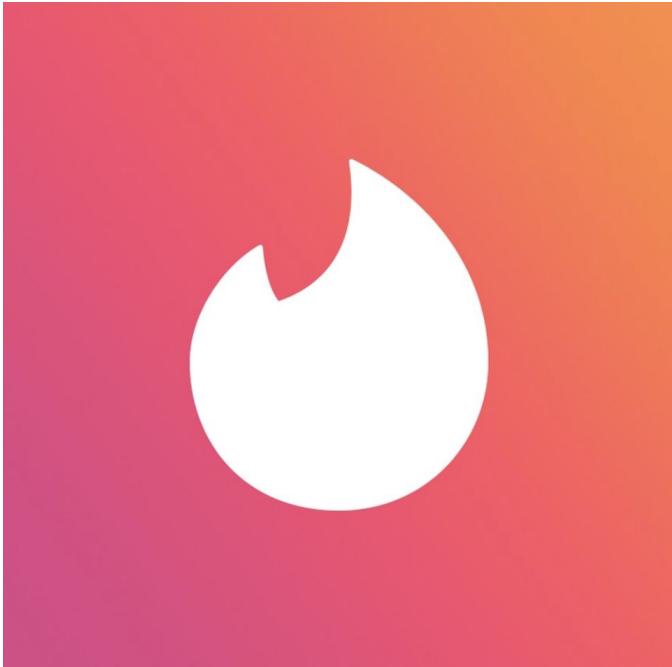
Implementation: Their app **sends their GPS location to your phone**, your app computes the distance, displays rounded to the nearest mile.

Problematic?

<https://robertheaton.com/2018/07/09/how-tinder-keeps-your-location-a-bit-private/>

Stanford University

Trust and Safety: Case Study



Feature: Display distance from your match.

Implementation: Their app **sends their GPS location to Tinder server**, your app does the same, server computes distance and sends distance to phone.

Problematic?

<https://robertheaton.com/2018/07/09/how-tinder-keeps-your-location-a-bit-private/>

Stanford University

Trust and Safety: Case Study



Problematic? Server sends distance with
15 decimal places of precision...

GPS spoofing of your phone's location
+ trilateration =
30m accuracy of your match's location!

Eventually fixed with rounding distances,
so trilateration only works to ~1mile
accuracy.

<https://robertheaton.com/2018/07/09/how-tinder-keeps-your-location-a-bit-private/>

Stanford University

Trust and Safety: Case Study

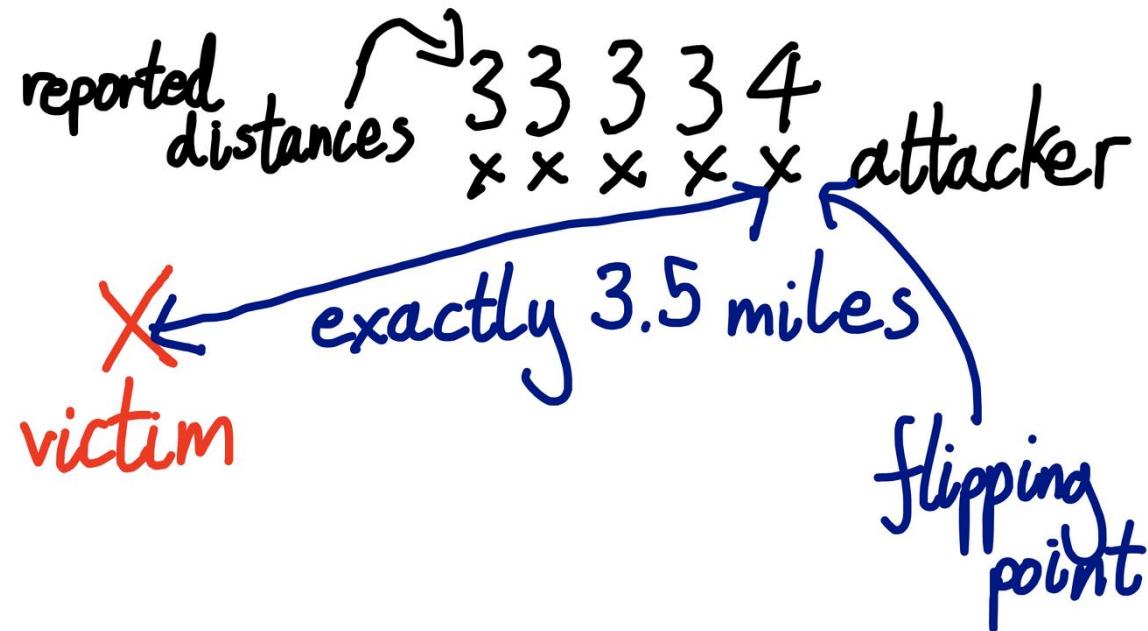


Feature: Display distance from your match.

Implementation: Locations sent to server, server computes distance and sends back rounded response.

Problematic?

Trust and Safety: Case Study



Responsible Disclosure

Disclosure timeline

- 2021-06-15: vulnerability reported to Bumble via HackerOne
- 2021-06-18: Bumble deploy a fix
- 2021-07-21: disclosure agreed via HackerOne

Trust and Safety: Another Example



<https://www.nytimes.com/2022/02/11/technology/airtags-gps-surveillance.html>

<https://support.apple.com/en-us/HT212227>

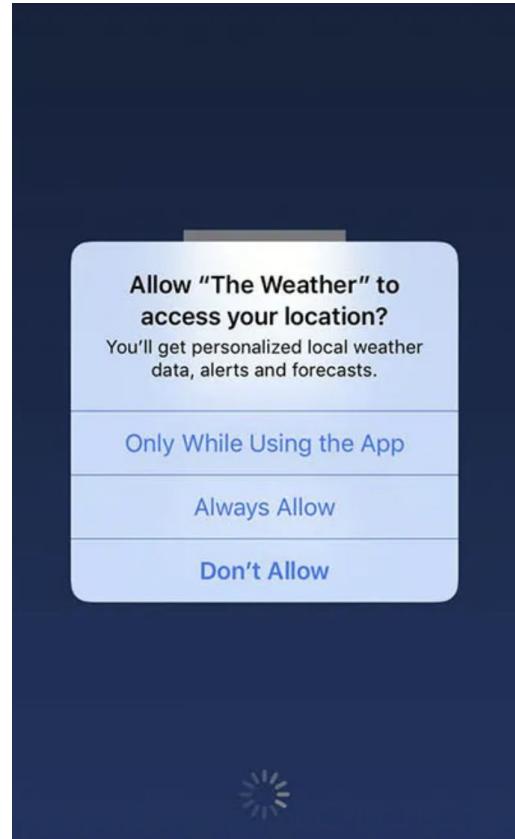


Violations of Trust and Safety



<https://www.technologyreview.com/2019/01/07/1517/the-weather-channel-app-has-been-accused-of-tracking-users-and-then-selling/>

<https://9to5mac.com/2020/08/20/ibm-settles-la-lawsuit-over-the-weather-channel-app-selling-user-location-data/>



Violations of Trust and Safety

- **Gross Negligence:** Equifax (patching), Solarwinds (passwords)
- **Fraud:** Theranos
 - Including lying to patients!
- **Clearly Unethical Behavior:** Uber
 - “Greyball”
 - “Hell”
- **Aggressive Design, Ignoring Warnings:** Zoom

<https://www.cnet.com/tech/tech-industry/uber-to-end-use-of-greyball-tool-to-evade-officials/>

<https://www.engadget.com/2017-09-08-uber-federal-investigation-hell-program.html>

Zoom: <https://www.youtube.com/watch?v=K7hlrw1BUck>

Violations of Trust and Safety

- Encouraging Risky Behavior: Venmo



v Venmo Matt, got a burning crypto question? We can help Get answers about the world of crypto We'...

v Venmo Matt, explore how crypto works on Venmo See how simple it is to get started Explore differen...

v Venmo Matt, curious about crypto? Dive on in Your crypto questions, answered Explore what cryptoc...

v Venmo Earn \$25, just for buying crypto on Venmo Start your crypto journey today We want to help ju...

v Venmo Now you can buy or sell crypto in just a few taps Matt, it's here! Crypto is now on Venmo Wel...

Violations of Trust and Safety

- **Severe Privacy Issues:** Venmo
 - Transactions are public by default.
 - Friends list is also public.
 - Until sometime after May 2021, not possible to hide friends list.
 - Only recourse was to manually unfriend each one.



Is this in the best interests of the user? How could this be abused to cause harm?

Violations of Trust and Safety

- Severe Privacy Issues: Venmo



"BuzzFeed News found President Joe Biden's Venmo account after less than 10 minutes of looking for it, revealing a network of his private social connections, a national security issue for the United States, and a major privacy concern for everyone who uses the popular peer-to-peer payments app."

"While many critics have focused on how the app makes all transactions public by default, Venmo's friend lists are arguably a larger privacy issue. Even if a Venmo account is set to make payments private, its friend list remains exposed. There is no setting to make this information private..."

Violations of Trust and Safety

- Encouraging Risky Behavior: Robinhood



The New York Times

<https://www.nytimes.com/2020/07/08/technology/robinhood-risky-trading.html>

Robinhood Has Lured Young Traders, Sometimes With Devastating Results

Its users buy and sell the riskiest financial products and do so more frequently than customers at other retail brokerage firms, but their inexperience can lead to staggering losses.



By Nathaniel Popper

Published July 8, 2020 Updated Sept. 25, 2021

<https://www.nytimes.com/2020/07/08/technology/robinhood-risky-trading.html>

Stanford University

Violations of Trust and Safety

- Encouraging Risky Behavior: Robinhood



“Mr. Dobatse, now 32, said he had been charmed by Robinhood’s one-click trading, easy access to complex investment products, and features like falling confetti and emoji-filled phone notifications that made it feel like a game.”

“[A]t least part of Robinhood’s success appears to have been built on a Silicon Valley playbook of behavioral nudges and push notifications, which has drawn inexperienced investors into the riskiest trading.”

Violations of Trust and Safety

- **Encouraging Risky Behavior:** Robinhood



In the first three months of 2020, Robinhood users traded:

- 9x shares as E-Trade customers
- 40x shares as Charles Schwab customers
- 88x risky options contracts as Charles Schwab customers

Violations of Trust and Safety

- Encouraging Risky Behavior: Robinhood



"The more often small investors trade stocks, the worse their returns are likely to be, studies have shown. The returns are even worse when they get involved with options, research has found."

"At the core of Robinhood's business is an incentive to encourage more trading. It does not charge fees for trading, but it is still paid more if its customers trade more."

Violations of Trust and Safety

- Encouraging Risky Behavior: Robinhood



Combining this with a poorly-thought-through user experience: tragedy.

Trust and Safety: Stanford Case Study



Trust and Safety: Humans are Cats

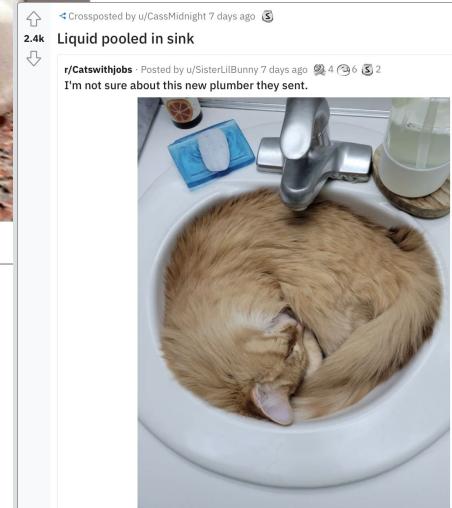
↑ Posted by u/UglyTheater 3 days ago 2 2 5

6.1k Yeah... they're liquid

↓



21 Comments Share Save ...



↑ Posted by u/JacquelinLaughridge 3 months ago 2 2 5 5

11.1k Meet Blossoms!

↓



64 Comments Share Save ...

Trust and Safety: Culture

Behaviors we Tolerate

Culture



Behaviors we do not Tolerate

Intermission: Netflix Chaos Monkey

<https://github.com/netflix/chaosmonkey>

Chaos Monkey **randomly terminates virtual machine instances and containers that run inside of your production environment.** Exposing engineers to failures more frequently incentivizes them to build resilient services.



A Whirlwind Tour

Exposure to important topics...
...with some fun intermissions



- Memory Hierarchy
- Memory Safety
- Trust and Safety
- **Where to Go from Here**

Life after CS107e: Where to Go From Here

Systems Core:

- **CS111**
- CS112/140
- CS140E
- CS240
- CS240LX

Systems:

- CS144 Networking
- **CS149** Parallel Computing

Ethics:

- CS182

Life after CS107e: Where to Go From Here

Other CS:

- **CS166** Data Structures
- CS242 Programming Languages (Rust!)

Computer Architecture:

- EE108 Digital Systems Design
- EE180 Digital Systems Architecture
- **EE282** Computer Systems Architecture

Circuits/Hardware:

- **ENGR40M** Intro to Making
- EE101A Circuits I

Life after CS107e: Where to Go From Here

Security:

- **CS155** Computer and Network Security (technical)
- INTLPOL268 Hack Lab (practical exposure + legal)
- CS255 Cryptography (technical)
- CS356 Topics in Computer and Network Security (papers + project)
- **CS152** Trust and Safety Engineering
- CS253 Web Security

International Security:

- **MS&E193** Technology and National Security
- **POLISCI114S** International Security in a Changing World
- CISAC talks, etc.

Life after CS107e: Where to Go From Here

- Way too many courses – what to do?
- Balance breadth versus depth, biased by your interests
- Coterming is a great way to stick around and take more classes :)
- Talk to us about it! Stay tuned for optional one-on-one course reflection meeting signups with your friendly course staff.

My favorite Bug (anecdotal)

