Announcements

- Course evals are available. Fill them in by 3pm tomorrow to receive an extra 1% towards your final grade.
- Recitations this week will be on a variety of topics, you can attend whichever one you want:
  
<table>
<thead>
<tr>
<th>Day</th>
<th>Time</th>
<th>Location</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tu</td>
<td>12:20</td>
<td>Bard 140</td>
<td>Regular Expressions</td>
</tr>
<tr>
<td>Tu</td>
<td>12:20</td>
<td>Hollister 368</td>
<td>Kooky Data Structures</td>
</tr>
<tr>
<td>Tu</td>
<td>12:20</td>
<td>Olin 216</td>
<td>Sound</td>
</tr>
<tr>
<td>Tu</td>
<td>12:20</td>
<td>Upson 216</td>
<td>Coding Interviews</td>
</tr>
<tr>
<td>Tu</td>
<td>1:25</td>
<td>Hollister 206</td>
<td>Java 9</td>
</tr>
<tr>
<td>Tu</td>
<td>1:25</td>
<td>Hollister 312</td>
<td>Dynamic Programming</td>
</tr>
<tr>
<td>Tu</td>
<td>2:30</td>
<td>Hollister 110</td>
<td>TBA</td>
</tr>
<tr>
<td>Tu</td>
<td>2:30</td>
<td>Olin 165</td>
<td>Collections</td>
</tr>
<tr>
<td>Tu</td>
<td>3:35</td>
<td>Bard 140</td>
<td>Distributed Computing</td>
</tr>
</tbody>
</table>

Security

- Security is about making sure that computers behave correctly
- A **secure system** should:
  1. Do what it is supposed to do
  2. Not do anything else

```java
public class ObjectStore {
    private Object[] objects;

    public ObjectStore(int len){
        objects = new Object[len];
    }

    public Object read(int i){
        return objects[i];
    }

    public void store(int i, Object o){
        objects[i]= o;
    }
}
```
OpenSSL

```
struct {
    HeartbeatMessageType type;
    uint16 payload_length;
    opaque payload[HeartbeatMessage.payload_length];
    opaque padding[padding_length];
} HeartbeatMessage;
```

Heartbleed

What might go wrong

```
public class ObjectStore {
    private Object[] objects;

    public ObjectStore(int len){
        objects = new Object[len];
    }

    public Object read(int i){
        return objects[i];
    }

    public void store(int i, Object o){
        objects[i] = o;
    }
}
```

Memory

```
ObjectStore OS = new ObjectStore(10);
...
store(12, o);
...
```

Skype Vulnerability

```
tmp = tmp + 1;
store tmp to i;
Load 0 from memory
Store 1 to memory
```

What might go wrong

```
Thread 1
Initially, i = 0

Thread 2
```
Copy-on-write (COW)

- Common resource optimization
- When someone copies a file, it doesn't really get copies
- If/when someone modifies the "copy" the original file gets copied and modified

Privilege Escalation

So how do we fix this?

- Testing
- Bug finding tools
- White-hat hacking

Security by Design

- Build secure, trustworthy computer systems/applications/etc.
- Define what the system is supposed to do
- Make sure it does that (and only that)
How do we specify what systems are and are not supposed to do?

What is Privacy?

Use-Based Privacy

- Privacy viewed as restrictions on uses [Cate02]
- Captures modern privacy goals
  - express restrictions in presence of necessary sharing

Policy Language

How do we make systems secure?
Threat Models

Example: Threat Model for Data Privacy

Approaches to security

- Axiomatic security
  - You trust someone else to get it right
- Constructive security
  - E.g., compiler checks, automated proofs

Code

Monitor
Approaches to security

- Axiomatic security
  - You trust someone else to get it right
- Constructive security
  - E.g., compiler checks, automated proofs
- Synthetic security
  - Modify the code to add checks (e.g., monitoring)
- Deterrence through accountability
  - Make sure you'll notice if something goes wrong

Example: Data Privacy from SGX

- Policy enforcement implemented by external monitor that runs on DHDs
  - Monitor can send/receive values from DS
  - Monitor shares values with authorized programs co-located at DH
    - Auth decisions based on credentials
    - Unauthorized values are cryptographically sealed with associated policy to prevent authorized use
    - Monitor maintains taint for each program, automatically derives policies for derived values

![Diagram of data privacy from SGX](image.png)