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

What might go wrong

```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;
    }
}
```
Heartbleed

```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;
    }
}
```

What might go wrong

```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;
    }
}
```

Memory

- Stack
- Heap

Skype Vulnerability

Copy-on-write (COW)

- Common resource optimization
- When someone copies a file, it doesn't really get copied
- 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
  - FindBugs
- 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)

Engineering Security

Attacks are perpetrated by threats that cause incorrect behavior by exploiting vulnerabilities which are controlled by countermeasures.
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

Engineering Security

Attacks are perpetrated by threats that cause incorrect behavior by exploiting vulnerabilities which are controlled by countermeasures.
What are the threats?

Threat Models

Example: Threat Model for Data Privacy

Engineering Security

Attacks are perpetrated by threats that cause incorrect behavior by exploiting vulnerabilities which are controlled by countermeasures.

How do we design countermeasures?
Classes of Countermeasures

Authentication: mechanisms that bind principals to actions
Authorization: mechanisms that govern whether actions are permitted
Audit: mechanisms that record and review actions

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 DHs
  - 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 unauthorized use
  - Monitor maintains taint for each program, automatically derives policies for derived values