Computer Science

- Vision
- Theory
- Networking
- Programming Languages
- Human–Computer Interaction
- Systems
- Machine Learning
- Natural Language Processing
- Graphics
- Scientific Computing
- Databases
- Robotics
- Software Engineering
- Architecture
- Security
Computer Security

- Theory
- Networking
- Programming Languages
- Human–Computer Interaction
- Vision
- Systems
- Machine Learning
- Natural Language Processing
- Graphics
- Scientific Computing
- Databases
- Robotics
- Software Engineering
- Security
- Architecture
- Theory
- Networking
- Programming Languages
- Human–Computer Interaction
- Vision
- Systems
- Machine Learning
- Natural Language Processing
- Graphics
- Scientific Computing
- Databases
- Robotics
- Software Engineering
- Security
- Architecture
Computer 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
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;
    }
}

struct {
    HeartbeatMessageType type;
    uint16 payload_length;
    opaque payload[HeartbeatMessage.payload_length];
    opaque padding[padding_length];
} HeartbeatMessage;
Heartbleed
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;
    }
}
```
ObjectStore OS = new ObjectStore(10);
...
store(12, o);
...
objects[i] = o;
exploit code
Skype Vulnerability
What might go wrong

Initially, $i = 0$

**Thread 1**

```
tmp = load i;
```

Load 0 from memory

```
tmp = tmp + 1;
store tmp to i;
```

Store 1 to memory

**Thread 2**

```
tmp = load i;
```

Load 0 from memory

```
tmp = tmp + 1;
store tmp to i;
```

Store 1 to memory

time
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
- White-hat hacking
So how do we fix this?

AT LEAST

YOU TRIED
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)
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?
Example: Data Privacy

Facebook app now reads your smartphone's text messages? THE TRUTH

Blame Android, says social network

Google Accused of Violating COPPA

Developers need explicit user permission according to Apple guidelines

Lawsuit Claims Disney Is Violating COPPA, Tracking Kids in 42 Apps

Disney believes the class action lawsuit "is based on a fundamental misunderstanding of COPPA principles."

Windows 10 data collection found to violate privacy laws

AccuWeather's iPhone app may track you even if you opt out (update)

The "feature" appears to violate Apple's terms of service.
What is Privacy?
Use-Based Privacy

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

Medical Data

Social Network Data
Policy Language

```
[
  {"curr":"1",
   "states":{"1":{"name":"s1-1",
                      "permissions":{"aggregate":true},
                      "transitions":{"aggregate":"s2-1"},
                      "defaultPermission":false},
               "2":{"name":"s2-1",
                     "permissions":{"fulfill":true},
                     "transitions":{},
                     "defaultPermission":true}}},
  {"curr":"2",
   "states":{"1":{"name":"s1-2",
                      "permissions":{"aggregate":true},
                      "transitions":{"aggregate":"s2-2"},
                      "defaultPermission":false},
               "2":{"name":"s2-2",
                     "permissions":{"fulfill":true},
                     "transitions":{},
                     "defaultPermission":true}}}]
```
Attacks are perpetrated by threats that cause incorrect behavior by exploiting vulnerabilities which are controlled by countermeasures.
What are the threats?
Threat Models

Capabilities, Resources, Motivation
Threat Models

A Crypto Nerd's Imagination:

His laptop's encrypted. Let's build a million-dollar cluster to crack it.

No good! It's 4096-bit RSA!

Blast! Our evil plan is foiled!

What would actually happen:

His laptop's encrypted. Drug him and hit him with this $5 wrench until he tells us the password.

Got it.
Example: Threat Model for Data Privacy
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

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

• Axiomatic security
  • You trust someone else to get it right
Approaches to security

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

String s=5;
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)
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 authorized use
  - monitor maintains taint for each program, automatically derives policies for derived values
Security

[{
  "curr": "1",
  "states": {
    "1": {
      "name": "s1-1",
      "permissions": {
        "aggregate": true
      },
      "transitions": {
        "aggregate": "s2-1"
      },
      "defaultPermission": false
    },
    "2": {
      "name": "s2-1",
      "permissions": {
        "fulfill": true
      },
      "defaultPermission": true
    }
  }
}]

![Diagram of security issues](image)

![Bar chart showing comfort levels](chart)