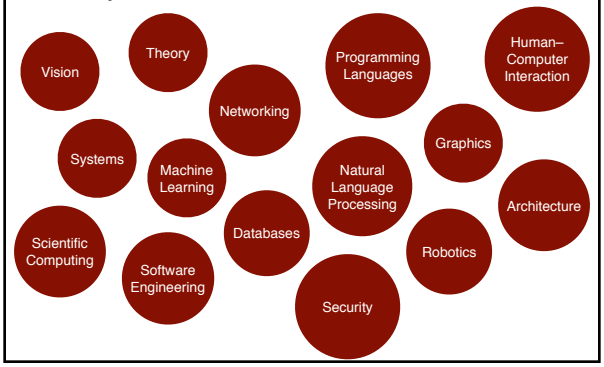
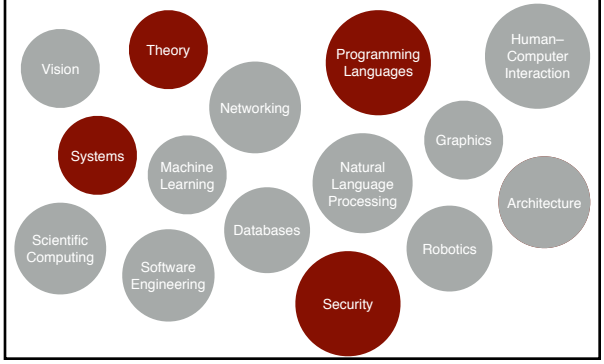
Computer Security

CS 2110 3 May, 2018

Computer Science



Computer Security



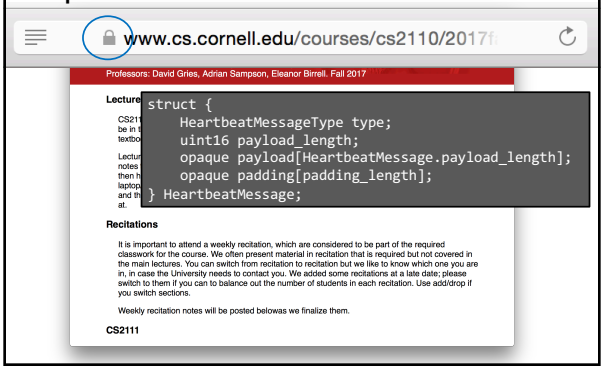
Computer Security

- 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;
      }
}
```

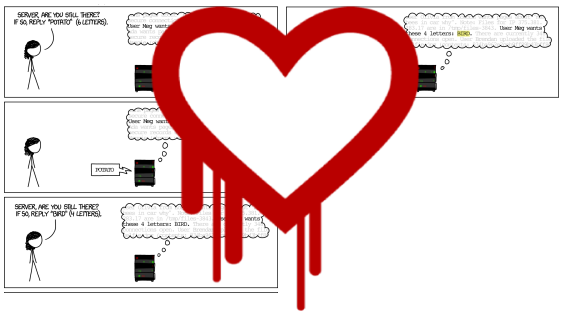

OpenSSL




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

7

Heartbleed



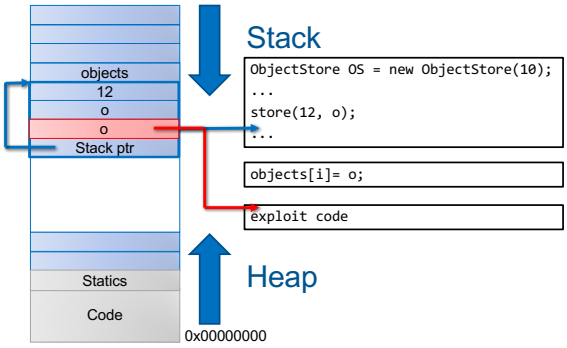
8

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;
      }
}
```

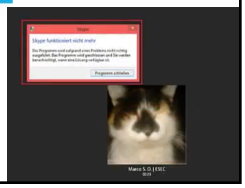
9

Memory



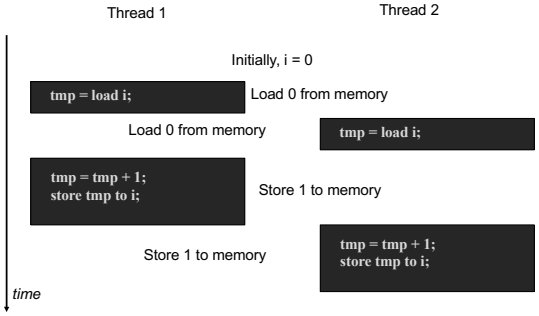
10

Skype Vulnerability



11

What might go wrong



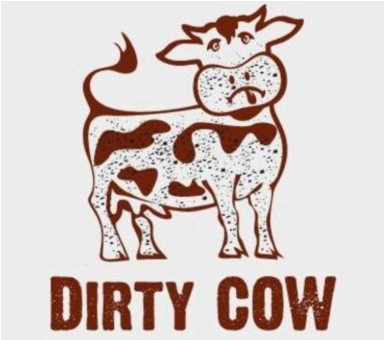
12

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

13

Privilege Escalation



14

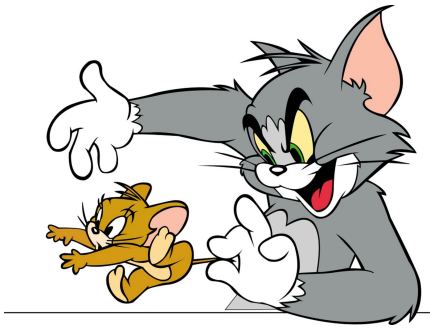
So how do we fix this?

- Testing
- Bug finding tools

FindBugs™

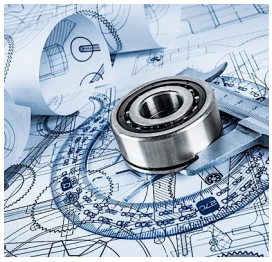
- White-hat hacking

15



16

So how do we fix this?



17

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

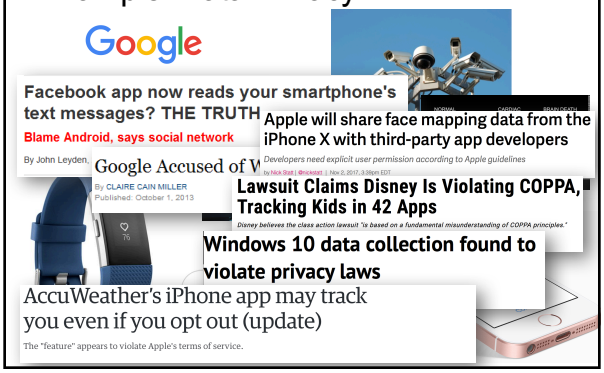
Attacks
are perpetrated by
threats
that cause
incorrect behavior
by exploiting
vulnerabilities
which are controlled by
countermeasures.

19

How do we specify what systems are and are not supposed to do?

20

Example: Data Privacy

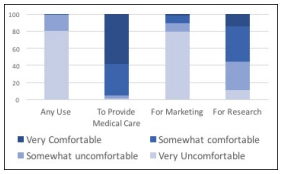


What is Privacy?

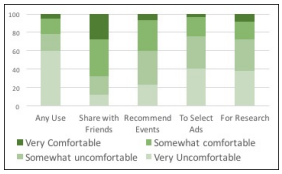
22

Use-Based Privacy

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



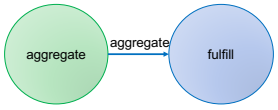
Medical Data



Social Network Data

23

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}}}]
```

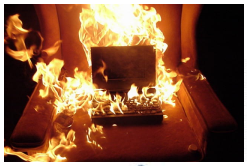
Engineering Security

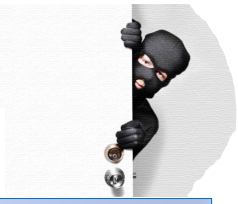
Attacks
are perpetrated by
threats
that cause
incorrect behavior
by exploiting
vulnerabilities
which are controlled by
countermeasures.

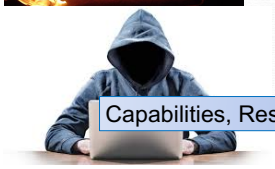
25

What are the threats?

Threat Models

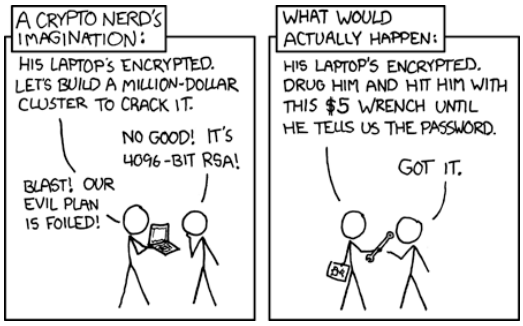






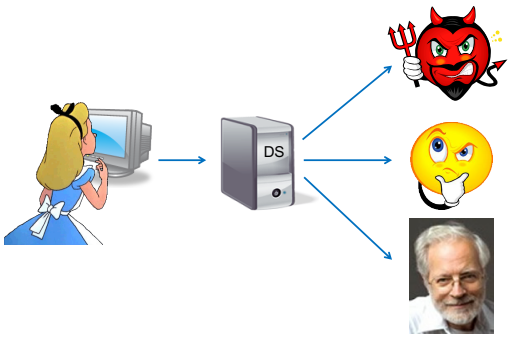
Capabilities, Resources, Motivation

Threat Models



28

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.

30

How do we design countermeasures

Classes of Countermeaures

79 Au Gold 196.967 thentication: mechanisms that bind principals to actions

79 Au Gold 196.967 thorization: mechanisms that govern whether actions are permitted

79 Au Gold 196.967 dit: mechanisms that record and review actions

Approaches to security

- Axiomatic security
  - You trust someone else to get it right

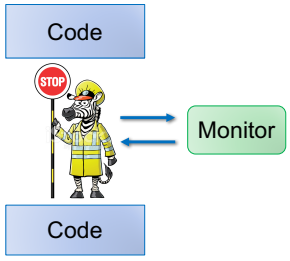
Approaches to security

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

```
35
36   String s=5;
37
```


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)

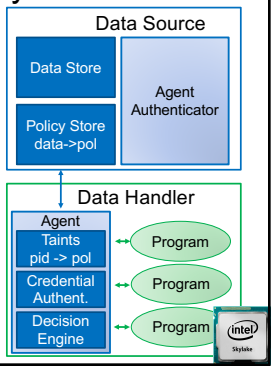


Approaches to security

- 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




37
Security
DIRTY COW
[{"curr":"1",
"states":{"1":{"name":"s1-1",
"permissions":{"aggregate":true},
"transitions":{"aggregate":"s2-1"}
"defaultPermission":false},
"2":{"name":"s2-1",
"permissions":{"fulfill":true},
"tr
"de
n":true}}}]
STOP
intel
Skylake