CSE 409, Fall 2011, Rob Johnson, http://www.cs.stonybrook.edu/~rob/teaching/cse409-fa11/
Alin Tomescu

About security

Definition

What is **security**? Security is computing in front of an adversary! This distinguishes it from **reliability**, like a disk failing…

Goals

There are three widely accepted security goals:

- **Confidentiality** (security mechanisms that keep unauthorized people out of the system)
- **Integrity** (a broad goal, the state of your data and your system is how you want it to be, you don't want to allow people to mess with it)
  - For example, SCADA (supervisory control and data acquisition) refers to industrial control systems: computer systems that monitor and control industrial, infrastructure, or facility-based processes.
  - SCADA controls these huge power plants, chemical plants, mining operations, breaking into them could be catastrophic
- **Availability**

Common mistake in security: Forget about certain security goals: *"Hey let's offer email to everyone without realizing spam will arise."*

The attacker

Who is your adversary? You need a threat model – a definition of what the attacker has as his starting capabilities:

- Most important capability is their **initial authorization level (local vs. remote)**
  - If you're running a web service you could have 3 different types of attackers
    - Attacker who has no account on the server
      - Common goal of an attacker is privilege escalation: get more access
        - That would violate integrity which will most likely compromise confidentiality and availability
    - Attacker who has an unprivileged account on the server
    - Attacker who has a privileged account on the server
- Other capabilities include
  - How much **money** they have
  - How much **time** they have
  - How much **computing power** they have
  - How much **network bandwidth** they have
  - How much **knowledge of your system** they have (probably more than you expect)
    - Do they know your source code?

Basic principles of how to design good secure systems

These are some of the principles used in designing secure systems:

1. Trust is **bad**

a. If you trust x, then your security depends on x.
b. Trust is **transitive**. If you trust x, and x trusts y, then if y fails, by transitivity you fail.
c. Mitigating trust
    i. Authenticate sources
    ii. Build it yourself (not always viable, maybe for govt.)
    iii. Verify
    iv. Redundancy (if you don't trust supplier A and B, buy two devices from both of them, run them in parallel and make sure they always give you the same output)
        1. For instance, some network protocols use two hashing algorithms at the same time so that if one of them is broken the security of the protocol can still rely on the other one.