Introduction to Security
(better late than never!)

Slides borrowed from CS 240 by Marco Canini
And Tadayoshi (Yoshi) Kohno CSE 484
Selected content adapted from D. Boneh.
How Systems Fail

Systems may fail for many reasons, including:

- **Reliability** deals with accidental failures
- **Usability** deals with problems arising from operating mistakes made by users
- **Security** deals with intentional failures created by intelligent parties
  - Security is about computing in the presence of an adversary
  - But security, reliability, and usability are all related
The computer security problem

Two factors:

• **Lots of buggy software** (and gullible users)

• **Money can be made from finding and exploiting vulnerabilities**

1. Marketplace for vulnerabilities

2. Marketplace for owned machines (PPI) ← Pay per install

3. Many methods to profit from owned client machines

current state of computer security
Why own machines:
1. IP address and bandwidth stealing

Attacker’s goal: look like a random Internet user

Use the IP address of infected machine or phone for:

• **Spam** (e.g. the storm botnet)

  Spamalytics: 1:12M pharma spams leads to purchase
  1:260K greeting card spams leads to infection

• **Denial of Service:** Services: 1h (20$), 24h (100$)

• **Click fraud** (e.g. Clickbot.a)
keylog for banking passwords, web pwds., gaming pwds.

Example: SilentBanker (and many like it)

Why own machines:
2. Steal user credentials

User requests login page

Malware injects Javascript

Bank sends login page needed to log in

When user submits information, also sent to attacker

Similar mechanism used by Zeus botnet
Why own machines:
3. Spread to isolated systems

Example: Stuxnet

Windows infection ⇒

Siemens PCS 7 SCADA control software on Windows ⇒

Siemens device controller on isolated network
Challenges: What is “Security”?

• What does security mean?
  – Often the hardest part of building a secure system is figuring out what security means
  – What are the assets to protect?
  – What are the threats to those assets?
  – Who are the adversaries, and what are their resources?
  – What is the security policy or goals?

• Perfect security does not exist!
  – Security is not a binary property
  – Security is about risk management

Current events, security reviews, and other discussions are designed to exercise our thinking about these issues.
Theme 1: Security Mindset

• Thinking critically about designs, **challenging assumptions**
• Being **curious**, thinking **like an attacker**
• “That new product X sounds awesome, I can’t wait to use it!” versus “That new product X sounds cool, but I wonder what would happen if someone did Y with it...”

• Why it’s important
  – **Technology changes**, so learning to **think like a security person** is more important than learning specifics of today
  – Will help you **design better systems/solutions**
  – Interactions with **broader context**: law, policy, ethics, etc.
Example
Example – What Do You See?
Example – What Do You See?
“Security is mostly a superstition” – Helen Keller (1880-1968), American writer and activist

• Security is all about trade-offs
  – Performance
  – Cost
  – Usability
  – Functionality

• The right question is: how do you know when something is secure enough?
  – Manage security risks vs benefits
  – Requires understanding of the trade-offs involved
How to think about trade-offs?

• What are you trying to protect? How valuable is it?
  – Nuclear missile launch station vs. … coffee machine

• In what way is it valuable?
  – May be important only to one person (e.g. private e-mail or passwords)
  – May be important because accurate and reliable (e.g. bank’s accounting logs)
  – May be important because of a service it provides (e.g. Google’s web servers)
High level plan

- **Policy**: the goal you want to achieve
  - e.g. only Alice should read file F
- **Threat model**: assumptions about what the attacker could do
  - e.g. can guess passwords, cannot physically grab file server
    - Better to err on the side of assuming attacker can do something
- **Mechanism**: knobs that your system provides to help uphold policy
  - e.g. user accounts, passwords, file permissions, encryption
- **Resulting goal**: no way for adversary within threat model to violate policy
  - Note that goal has nothing to say about mechanism
Security goals

• Prevent common vulnerabilities from occurring (e.g. buffer overflows)
  – Recover from attacks
• Traceability, accountability and auditing of security-relevant actions
  – Monitoring
• Detect attacks
  – Privacy, confidentiality, anonymity
  – Protect secrets
• Authenticity
  – Needed for access control, authorization, etc.
• Integrity
  – Prevent unwanted modification or tampering
• Availability and reliability
  – Reduce risk of DoS
Classic CIA triad

- **Confidentiality**
  - NO unauthorized disclosure of information
    - E.g. a credit card transaction system attempts to enforce confidentiality by encrypting credit card details over the Internet and in the transaction processing network

- **Integrity**
  - NO unauthorized information modification
    - E.g. traditional Unix file permissions can be an important factor in single system measures for protecting data integrity

- **Availability + Authenticity (non-standard)**
  - Information or system remains available despite attacks
    - High availability systems aim to remain available at all times, preventing disruptions due to power outages, upgrades, hardware failures, Denial of Service (DoS) attacks, …
Confidentiality (Privacy)

- Confidentiality is *concealment of information.*

Eavesdropping, packet sniffing, illegal copying

network
Integrity

- Integrity is prevention of unauthorized changes.

Intercept messages, tamper, release again
• Authenticity is *knowing who you’re talking to*.
Availability

• Availability is **ability to use information or resources.**
THREAT MODELING
Threat Modeling (Security Reviews)

- **Assets**: What are we trying to protect? How valuable are those assets?
- **Adversaries**: Who might try to attack, and why?
- **Vulnerabilities**: How might the system be weak?
- **Threats**: What actions might an adversary take to exploit vulnerabilities?
- **Risk**: How important are assets? How likely is exploit?
- **Possible Defenses**
Example: Electronic Voting

- Popular replacement to traditional paper ballots
**Pre-Election**

**Pre-election**: Poll workers load “ballot definition files” on voting machine.
Active voting: Voters obtain single-use tokens from poll workers. Voters use tokens to activate machines and vote.
Active voting: Votes encrypted and stored. Voter token canceled.
Post-Election

Post-election: Stored votes transported to tabulation center.
Security and E-Voting (Simplified)

- Functionality goals:
  - Easy to use, reduce mistakes/confusion

- Security goals:
  - Adversary should not be able to tamper with the election outcome
    - By changing votes (**integrity**)
    - By voting on behalf of someone (**authenticity**)
    - By denying voters the right to vote (**availability**)
  - Adversary should not be able to figure out how voters vote (**confidentiality**)

Can You Spot Any Potential Issues?

Voter token

Ballot definition file

Poll worker

Interactively vote

Voter

Encrypted votes

Recorded votes

Tabulator
Potential Adversaries

- Voters
- Election officials
- Employees of voting machine manufacturer
  - Software/hardware engineers
  - Maintenance people
- Other engineers
  - Makers of hardware
  - Makers of underlying software or add-on components
  - Makers of compiler
- ...
- Or any combination of the above
What Software is Running?

Problem: An adversary (e.g., a poll worker, software developer, or company representative) able to control the software or the underlying hardware could do whatever he or she wanted.
KEYS TO THE KINGDOM
Photo taken from Diebold's online store. The keys that open every Diebold touch-screen voting machine. Working copies have been made from the photo.
Problem: Ballot definition files are not authenticated.

Example attack: A malicious poll worker could modify ballot definition files so that votes cast for “Mickey Mouse” are recorded for “Donald Duck.”
Problem: Smartcards can perform cryptographic operations. But there is no authentication from voter token to terminal.

Example attack: A regular voter could make his or her own voter token and vote multiple times.
Problem: Encryption key ("F2654hD4") hard-coded into the software since (at least) 1998. Votes stored in the order cast.

Example attack: A poll worker could determine how voters vote.
**Problem:** When votes transmitted to tabulator over the Internet or a dialup connection, they are **decrypted first**; the cleartext results are sent to the tabulator.

**Example attack:** A sophisticated outsider could determine how voters vote.
Tables Often Help!

![Table Image](image-url)
### Example Table 1

<table>
<thead>
<tr>
<th>Attacker “Positions”</th>
<th>Machine Manufacturer</th>
<th>Poll Worker</th>
<th>Voter</th>
<th>Power Company Employee</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voter Privacy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vote Integrity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voting Machine</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Availability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- What can different parties do? Each cell would have an action or actions that these parties might try to do.
- Note that some parties could collaborate.
### Example Table 2

<table>
<thead>
<tr>
<th>Attack Methods</th>
<th>Modify Software</th>
<th>Produce Fake Voter Tokens</th>
<th>Steal Flash Drive</th>
<th>Intercept Network Connections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voter Privacy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vote Integrity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voting Machine Availability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- What different attack methods are there? (Columns)
- Who could mount these different attacks? What are the attack details (the cells)
- How easy is it to implement each of these attack methods?
Table from Paper

<table>
<thead>
<tr>
<th>Table 1: This table summarizes some of the more important attacks on the system.</th>
<th>Voter (with forged smartcard)</th>
<th>Poll Worker (with access to storage media)</th>
<th>Poll Worker (with access to network traffic)</th>
<th>Internet Provider (with access to network traffic)</th>
<th>OS Developer</th>
<th>Voting Device Developer</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vote multiple times using forged smartcard</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td>3.2</td>
</tr>
<tr>
<td>Access administrative functions or close polling station</td>
<td>•</td>
<td>•</td>
<td></td>
<td></td>
<td>•</td>
<td>•</td>
<td>3.3</td>
</tr>
<tr>
<td>Modify system configuration</td>
<td></td>
<td>•</td>
<td></td>
<td></td>
<td>•</td>
<td>•</td>
<td>4.1</td>
</tr>
<tr>
<td>Modify ballot definition (e.g., party affiliation)</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td>4.2</td>
</tr>
<tr>
<td>Cause votes to be miscounted by tampering with configuration</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td>4.2</td>
</tr>
<tr>
<td>Impersonate legitimate voting machine to tallying authority</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td>4.3</td>
</tr>
<tr>
<td>Create, delete, and modify votes</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>4.3, 4.5</td>
</tr>
<tr>
<td>Link voters with their votes</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td>4.5</td>
</tr>
<tr>
<td>Tamper with audit logs</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td>4.6</td>
</tr>
<tr>
<td>Delay the start of an election</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td>4.7</td>
</tr>
<tr>
<td>Insert backdoors into code</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td>5.3</td>
</tr>
</tbody>
</table>
TOWARDS DEFENSES
Approaches to Security

• Prevention
  – Stop an attack

• Detection
  – Detect an ongoing or past attack

• Response
  – Respond to attacks

• The threat of a response may be enough to deter some attackers
Example security mechanisms

- Verifying the identity of a prospective user by demanding a password
  - Authentication
- Shielding the computer to prevent interception and subsequent interpretation of electromagnetic radiation
  - Covert channels
- Enciphering information sent via communication channels
  - Cryptography
- Locking the room containing the computer
  - Physical aspects of security
- Controlling who is allowed to make changes to a computer system
  - Social aspects of security
Whole System is Critical

• Securing a system involves a whole-system view
  – Cryptography
  – Implementation
  – People
  – Physical security
  – Everything in between

• This is because “security is only as strong as the weakest link,” and security can fail in many places
  – No reason to attack the strongest part of a system if you can walk right around it.
Whole System is Critical

• Securing a system involves a whole-system view
  – Cryptography
  – Implementation
  – People
  – Physical security
  – Everything in between

• This is because “security is only as strong as the weakest link,” and security can fail in many places
  – No reason to attack the strongest part of a system if you can walk right around it.
Whole System is Critical

Securing a system involves a whole-system view—

- Cryptography
- Implementation
- People
- Physical security
- Everything in between

This is because "security is only as strong as the weakest link," and security can fail in many places—No reason to attack the strongest part of a system if you can walk right around it.

9/29/2018
CSE 484 / CSE M 584 - Fall 2017
In reality

**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.
Thank you for taking CPSC 416

And attending 8am classes

You took on a **challenging** course
I hope it was a rewarding experience

- December deadlines:
  - Final exam: December 6, 830AM, DMP 110
  - Project code/reports: December 9
  - Project demos: December 10-13