Some administrativa

- HW #0 assigned today, due in a week
  - Substitution cipher
  - Everyone has a unique problem w/a unique solution
  - You can probably do it now, but if not definitely after next class
  - Online turnin documented in the handout

- Please sign up for the class piazza forum
  - Instructions on the Web site
How to think about the security of a system?
- Summarization, attack surface, vulnerabilities, threats, triage

Security goals
- Confidentiality, integrity, authenticity, availability

Examples
- Locks
- Voting machines
How to analyze the security of a system?

- What is the system and what is its value?
- Identify the *attack surface*
  - How can it be attacked?
- Identify potential vulnerabilities
- Identify the threats and adversaries
  - What is the *threat model* you need to protect against?
- Triage
What is the system?

- **Tangible assets**
  - **Hardware**: servers, routers, networks, iPads, etc
  - **Software**: operating systems, applications, databases
  - **Information**: customer records, design documents, source code, etc

- **Intangible components**
  - Value of uptime, latency, etc.
  - Brand/reputation

- What is the value of these components?
Attack surface

- What are the avenues by which someone might try to attack your system?
  - Network interfaces, USB ports, user input, software updates, hardware upgrades, insiders, external dependencies, etc...

- Simple example: a locked gate
Attack trees: think about attack surface like an bad guy

- Log in to UNIX Account
  - No Password Required
  - Learn Password
  - Guess Password
  - Use Widely Known Passwords
    - Find Written Password
    - Get Password From Target
      - Threaten
      - Blackmail
      - Steal
      - Bribe
        - Install Keyboard Sniffer
        - Obtain Sniffer Output File
Vulnerabilities

- Weaknesses that could be exploited to cause damage
  - Default password left intact ("password")
  - Implementation flaws in software
  - Debug interface left open to network
  - Cryptography based on weak keys
- In particular, look to *assumptions* in system
- Known vulnerabilities (Bugtraq, full-disclosure, NVD) vs unknown vulnerabilities (0-days)
Threats

● Actions by adversaries who try to exploit vulnerabilities to damage asset

● Example: voting machines
  ◆ Extract records: find out who voted for who
  ◆ Tampering with data: change outcome of election
  ◆ Spoofing identity: vote as someone else
  ◆ Crash machine: prevent others from voting

● Correspond to confidentiality, integrity, authenticity and availability
Confidentiality

- Concealment of information; privacy

Alice -> Message -> Bob

Eavesdropping, Copying, intercepting

Zurg the Evil
Integrity

- Prevention of unauthorized changes

[Diagram showing Alice sending a message to Bob, with Zurg the Evil intercepting and modifying the message.]
Authenticity

- Identification and assurance of origin

Alice

Zurg the Evil

Falsely assuming another’s identity

Bob
Availability

- Ability to use information or resources desired

- Destroy data, overwhelm net, crash servers

Alice

Bob

Zurg the Evil
Threat model

- What you assume about attacker’s
  - Goals:
    » Kind of damage: confidentiality, integrity, etc
    » Method of damage: Interception, interruption, modification, fabrication
  - Capabilities
    » Have local access? Only remote access?
    » What do they know?
    » What resources are at their disposal?

- You must have a threat model!
Triage

- What combination of threats * vulnerabilities * asset value are the biggest?
  - Likelihood of threat
  - Seriousness of vulnerability
    » Urgency: Secret, disclosed, in the wild
    » Impact: crash, limited info disclosure, arbitrary execution
  - Value of potential loss?
    » Lost revenue from server crash
    » Reputation damage from losing customer e-mail addrs
    » Lost financial credentials and intellectual property

- Imperfect knowledge… but we try
Simple example: locks

- What is the purpose of a lock?

- Let's look at the attack surface of a lock and the vulnerabilities it might have…
Locks

- Worlds oldest (pre-Biblical) and most pervasive form of access control

Egyptian tumbler lock design ~1000 BCE

Modern Cylinder lock
How physical locks work

Shear line
Plug
Driver pins
Bottom pins
courtesy Matt Blaze
How physical locks work
courtesy Matt Blaze
Shared secrets

- There is a shared secret between the lock and the key... its shape

- In fact, it’s a digital code
Bitting codes

- A key can be precisely described with a discrete code
  - Cuts at regular intervals (4-6 cuts)
  - Depth of cuts quantized in standard fashion (typically 6-9 bins)
  - 4-6 digits sufficient to describe most keys
Design assumptions

- If you don’t know the secret code, you can’t open the lock
- The secret code is secret
- If you can’t open the lock, everything is fine
Design assumptions

- If you don’t know the secret code, you can’t open the lock
- The secret code is secret
- If you can’t open the lock, everything is fine
Lock bypass via manipulation

Picking & Raking

Bumping
Picking
Picking

- Two parts
  - Tension wrench used to apply *slight* lateral force on plug
  - Pick used to lift individual bottom pins to the shear line
- Tension causes driver pins to bind above shear line
Picking

Here the tension wrench is twisted slightly to allow the yellow pins to rest on the edge of the shear line.
Raking

- Similar idea, but less finesse...
- Rake pick moved in and out quickly imparts force to bottom pins; driver pins bind
- Quick & easy
Bumping

- Similar idea to raking, but does all pins in parallel; super easy to do
- Max-depth key (bump key) used to impart force to bottom pins who transfer energy to driver pins (think billiards)
Bumping
Some defenses

- Security pins
  - Spool pins, mushroom pins, interlocking pins
    » Shapes that get “stuck” when plug under tension

![Binding Security Pin](image-url)
Some defenses

- Security pins
  - Spool pins, mushroom pins, interlocking pins
    » Shapes that get “stuck” when plug under tension
  - Pin rotation (angled cuts on keys)
Some defenses

- Ancillary locking mechanisms; sidebars (2)
Side issue: master keying

- How do master keys work?
  - Second set of pins (spacers); multiple shear lines
Hmmm.... problem?

- Suppose 6 pins and 10 positions per pin
  - In principal $10^6$ combinations; can’t guess master
  - But what if you have one working key
- Scenario: your key: 557346, master: 232346
  - Make key: 157346; does it work? No, cut groove down one position; at position N it works!
  - If N is not equal to 5, then N is the master cut for that pin
  - Repeat for each pin; six keys are sufficient if all six pins have master pins;
- “Rights Amplification”
Design assumptions

- If you don’t know the secret code, you can’t open the lock
- The secret code is secret
- If you can’t open the lock, everything is fine
Design assumptions

- If you don’t know the secret code, you can’t open the lock
- The secret code is secret
- If you can’t open the lock, everything is fine
Problem

- The bitting code is only secret if the key is kept secure.
- What if I “borrow” your key?
Lock bypass via surreptitious duplication

Field casting

Decoding
The power of decoding

Code key cutting machine

Key Blank + 64678 = Key replica
April 4 2008 95/366 - House keys!
Optical decoding

- Decode keys semi-automatically from photos
- Traditional computer vision problem (photometry)
  - Normalize for scale and rotation
Sneakey: UCSD

- Reference key measured at control points
- User supplies correspondences between target key and reference image
- Image normalized (homographic transform), cut locations identified and cut depths measured (n guesses)
Works really well

- Almost perfectly from up close photos (e.g., cell phone cameras, etc)
- But that’s no fun… what would James Bond do?
Where’s the Key?
One defense: restricted keyways

- Key shape registered to customer and not available for sale to anyone else
One defense: restricted keyways

- But…

3D Printers

Key milling machines
A better approach

- Electronic & mechanical keys
- Challenge/response via RF
  - But own issues; batteries, replay, how to program, etc
Very high security

- Electronic; no battery; self-erase; heavy RF shielding; different combination for each user; unerasable audit log
Design assumptions

- If you don’t know the secret code, you can’t open the lock
- The secret code is secret
- If you can’t open the lock, everything is fine
Design assumptions

- If you don’t know the secret code, you can’t open the lock
- The secret code is secret
- If you can’t open the lock, everything is fine
Taking the big picture

- What is the threat?
  - Capabilities, resources, goals
  - Faster than the bear or faster than the next guy?
- What are all the ways the adversary might get access (the “attack surface”)?
Deterrence

- No physical security is perfect
- For indiscriminate adversary (e.g., burglar) goal is to make cost higher than its worth
  - Go elsewhere
- Deterrence can be indirect
  - Lojack story
- Determined adversary may not be deterrable
  - How to increase risk, response, cost, time
  - E.g. time locks
Also: Denial of service
Worse denial of service
Aside: For those interested...

- Check out
  - Matt Blaze’s work
    » Safecracking for the Computer Scientist
    » Cryptology and Physical Security: Rights Amplification in Master-Keyed Mechanical Locks
    » Notes on Picking Pin Tumbler Locks,
  - MIT Guide to Lockpicking
  - Locksport International (http://locksport.com/)

- However…
  - NEVER pick a lock you do not own
  - ALWAYS know the local law about using such tools
Example: Electronic Voting

- Popular replacement to traditional paper ballots
Pre-election: Poll workers load “ballot definition files” on voting machine.
Active Voting:

Active voting: Voters obtain single-use tokens from poll workers. Voters use tokens to activate machines and vote.
Active Voting

**Active voting:** Votes encrypted and stored. Voter token canceled.
Post-Election

Voter token

Interactively vote

Voter

Poll worker

Ballot definition file

Post-election: Stored votes transported to tabulation center.

Tabulator

Recorded votes

Encrypted votes
Security and E-Voting (Simplified)

Functionality goals:
- Easy to use
- People should be able to cast votes easily, in their own language or with headphones for accessibility
Security and E-Voting (Simplified)

- Functionality goals:
  - Easy to use
  - People should be able to cast votes easily, in their own language or with headphones for accessibility

- Security goals:
  - Adversary should not be able to tamper with the election outcome
    - By changing votes
    - By denying voters the right to vote
  - Is it OK if an adversary can do the above, assuming you can catch him or her or them?
  - Adversary should not be able to figure out how voters vote
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
Can You Spot Any Potential Issues?

Poll worker → Ballot definition file → Voter token → Interactively vote → Voter token → Encrypted votes → Recorded votes → Tabulator

Poll worker

Ballot definition file

Voter token

Interactively vote

Voter

Encrypted votes

Recorded votes

Tabulator
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.
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 the the tabulator.

Example attack: A sophisticated outsider could determine how voters vote.
Practice

- I encourage you to start thinking about the security issues in the goods/services you use.
- Once you start looking at things this way, you can find risks everywhere.

- For next time, start reading Anderson’s chapter on Crypto (link on Web site).
- We start basic crypto.