CS 194-1 (CS 161) Computer Security

Lecture 2

Threat models, security goals, access control

August 30, 2006

Prof. Anthony D. Joseph
http://cs161.org/

Review: What is Computer Security Today?

- Computing in the presence of an adversary!
  - An *adversary* is the security field's defining characteristic
- Reliability, robustness, & FT: random failures
- Security
  - Dealing with/surviving actions of knowledgeable attacker dedicated to causing harm
- Wherever there is an adversary, there is a computer security problem!

Review: Analyze to Learn!

- Study attackers and think about how to break into systems to learn attack tactics
  - Analyze previous successful attacks
- Arms race for solutions…
  - (Some) attackers are intelligent
    - » Attacks will change and get better with time
  - Deploy a new defense, they respond, you build a better defense, they respond, you…
    - » Try to anticipate future attacks
- Security is like a game of chess
  - Except the attackers often get the last move!

Review: Security Evaluation Process

- We need a framework to help you think through the ways that an adversary might penetrate system security?
- Start with *security goals*:
  - What properties do we want the system to have, even when it is under attack?
  - What are we trying to protect from the attacker?
  - Or, to look at it the other way around, what are we trying to prevent?

Goals for Today

- How do we assess threats to a system?
- How do we create a threat model?
- What is access control and what is its role?

Threat Assessment

- Some questions:
  - What kind of threats might we face?
  - What kind of capabilities might we expect adversaries to have?
  - What are the limits on what the adversary might be able to do to us?
- Result is a *threat model*, a characterization of the threats the system must deal with
  - Think: *Who? What? Why?*

Developing a Threat Model

- First decide how much we can predict about kinds of adversaries we will face
  - Sometimes, know very well who the adversary is, and even their capabilities, motivations, and limitations
    - Cold War: US military oriented towards main enemy (Soviets) and focused on understanding USSR's military capabilities/effectiveness/responsiveness
- If we know potential adversary, can craft a threat model that reflects adversary's abilities and options and nothing more
  - However, often adversary is unknown

Thinking Generically

- Must reason more generically about unavoidable limitations of the adversary
  - Silly ex: physics means adversary can't exceed speed of light
- Can usually look at system design and identify what an adversary might do
  - Ex: If system never sends secret info over wireless nets, then don't need to worry about threat of wireless eavesdropping
  - Ex: If system design means people might discuss secrets by phone, then threat model needs to include possible phone co. insider threats: eavesdrop/re-route/impersonate

What to Ignore?

- Good threat model also specifies threats we don't care to defend against
  - Ex: home security – I don't worry about a team of burglars flying a helicopter over my house and rappelling down my chimney
- Why not?
- Many easier ways to break into my house…
- Can classify adversaries by their motivation
  - Ex: financial gain motivation means won't spend more money on attack than they'll gain
  - Burglar won't spend 1,000's to steal car radio
- Motives are as varied as human nature
  - Have to prepare for all eventualities…

Helpful to Examine Incentives

- Ex: Do fast food places make more profit on soft drinks than on food?
  - Would expect some places to try to boost drink sales (e.g., salting french fries heavily)
- Ex: Do customer svc reps earn bonuses for handling more than X calls per hour?
  - Would expect some reps to cut long calls short, or to transfer trouble customers to other depts. when possible
- Ex: Do spammers make money from those who respond, while losing nothing from those who don't?
  - Would expect that spammers send their emails as widely as possible, no matter how unpopular it makes them

Incentives

- As a rule of thumb, organizations tend not to act against their own self-interest (at least not often…)
- Incentives (frequently) influence behavior
  - Exposes the motivations of potential adversaries
- Incentives are particularly relevant when two parties have opposing interests
  - When incentives clash, conflict often follows.
  - In this case it is worth looking deeply at the potential for attacks by one such party against the other

Threat Assessment Summary

- Remember the three W's:
  - *Who* are the adversaries we might face?
  - *How* might they try to attack us, and what are their capabilities?
  - *Why* might they be motivated to attack us, and what are their incentives?
- Given security goals and threat model, last step is performing a security analysis

Administrivia

- Space still available in this class
  - Talk to Michael-David Sasson today
- Three sections on Thursdays in 320 Soda
  - 101. 10:00-11:00
  - 102. 11:00-12:00
  - 103. 3:00- 4:00
- 18 students have final exam conflicts
  - CS 162 and EE 122
- No account forms needed, use named accts (details in HW #1)
- Minor changes to project and HW due dates

BREAK

Security Analysis

- Seeing whether there attacks (within threat model) that successfully violate security goals
  - Often highly technical and dependent on system details
  - We'll show you many security analysis methods
- One analogy:
  - Threat model defines set of moves an adversary is allowed to make
  - System design defines how defender plays game
  - Security goals define success condition: if adversary violates any goal, he wins; otherwise, the defender wins
  - Security analysis is examining all moves and counter-moves to see who has a winning strategy

Another Analogy

- Mystery writers like to talk about means, motive, and opportunity
  - Security evaluation is similar way of thinking
- Threat assessment examines the means and motive
- Security analysis examines what opportunity the adversary might have to do harm

Security Evaluation Summary

- Identify the security goals
  - What are we trying to protect?
- Perform a threat assessment
  - What threats does the system need to protect against?
- Do a security analysis
  - Can we envision any feasible attack that would violate the security goals?
  - May be very technical
- Use same process for new system design
  - Easier to ensure security when you know the security goals you and threats
  - Security analysis helps refine system design

Home Security Analysis Example

- What are my security goals?
  - Protecting assets from theft or tampering (integrity)
  - Protecting my personal safety
    - » Ex: if someone does break in to steal money, I'd much prefer to know, so that I don't surprise them and get shot
  - Ensuring my house and contents remain in full working order whenever I want them (availability)
  - Providing a certain measure of privacy (confidentiality)
  - …

Threat Assessment: Motivation

- Who am I trying to protect against, and what's their motivation?
  - Burglar motivated by financial gain
  - Peeping Tom motivated by curiousity
  - Grudge holder motivated by revenge
  - …

Threat Assessment: Capabilities/Threats

- What are their capabilities (tools, skills, knowledge, access, etc.)? What threats might I face?
  - Burglar has lockpicks and know-how, crowbar, or can cut my phone lines
  - Repairman might have unaccompanied access
  - Peepers might have binoculars or telescope
  - One neighbor might have line-of-sight to my living room window, while another might be blocked by trees
- Threats to ignore?
  - Navy ships here for Fleet Day aren't likely to start shelling my house, …

Security Analysis: Possible Attacks

- All sorts of crazy scenarios!
- A burglar breaks window, grabs stuff, leaves
- I secure the windows, but determined burglar takes chainsaw to the walls and breaks in
- Slightly smarter burglar might look under the flowerpot and find the spare house key, …
- Sneaky burglar throws pebble against window at 3am each morning, setting off alarm and bringing the police, each morning until police decide to ignore obviously unreliable burglar alarm, then burglar is free to break in…

More Possible Attacks

- Neighbor with line-of-sight to my house uses telescope to peer in window
- Someone intent on revenge might leave unpleasant items on my lawn, or – depending on my home's security system – might be able to smash my windows
- Unscrupulous competitor knows I have an important early morning business meeting, so they cut my house's power at night to make my alarm clock fail

Security Example Summary

- This was a trivial example, mostly because you're already familiar with home security
- However, it helps to have a framework when dealing with a complex computer system
  - Structures the security evaluation
  - Defines the process

BREAK

Role of Access Control

- Before closing "back doors" we need to close "front doors"
- Access control: determines access to files & processes in OS
- We will return to these themes throughout the course

Classic Models of Security

- Computer security has its origin in military models of security
- Different levels of secrecy
  - e.g. FOUO/SSI/Secret/Top Secret
- Compartmentalized security
  - e.g. CNWDI (Critical Nuclear Weapons Design Information), COMSEC (Communications Security), ...
  - TS/SCI (Top Secret/Sensitive Compartmented Information)
  - CRYPTO

Corresponding Access Control

- Classic model ® Mandatory Access Control (MAC)
  - » We also use the abbreviation MAC for "message authentication code"
- User controlled security ® Discretionary Access Control (DAC)

Subjects & Objects

- Subjects do things
  - Ex: users, processes …
- Objects have things done to them
  - Ex: files, processes …
- Access types are the things that are done
  - Ex: read, write, append, list, detect, remove, execute …

Read and Write are Different

- Access types can be distinguished by whether they pass information
- Generally "write" passes information
- Generally "read" does not pass information

Access Control Matrix

	File 1	File 2	File 3
Alice	read		read/write
Bob		execute	
Charlie			read

Problems with an Access Control Matrix

- Sparse matrix – many blank entries
- Hard to manage
- Who can manage different entries?
- What if we need to give "temporary rights"?
- Common entries?

Sparse Matrix Representations

- Access Control Lists (ACLs)
  - Objects list subjects and access types
  - Ex: This file can be modified by Alice and read by Charlie
- Capabilities
  - Subjects have particular "permissions"
  - Ex: Bob is allowed to modify files
- Hybrid models also exist

Are ACLs & Capabilities Equivalent?

- In representative power, yes
  - Both are sparse matrix representations of the Access Matrix
- In philosophy, no
  - Often come with particular features & OS philosophy
  - Capabilities often appeal to researchers
  - But capability systems often work poorly
  - Perennial claim: Capability lists are coming back!

Where is an ACL Applied?

- In some systems: on the file
- In some systems: on the directory
- In some systems: a combination

Who Determines Identity?

- In (non-distributed) multi-user systems, usually OS
  - » login
- In distributed systems
  - Sometimes a central authority
  - (trusted third party, e.g., Kerberos)
    - » Single login
  - Sometimes knowledge of a password
  - (e.g., ssh or "guest" file sharing in Windows)
    - » Remote login

Who is Allowed to Modify an ACL?

- In some systems, the "owner" of the file/process/directory
- Example: chmod command in UNIX
  - World access: read/write/execute
    - » For directories: read = list items;
    - » execute = "enter" directory
  - Owner access: read/write/execute

Fine-Grained Control

- But we need options other than "world access" or "owner-only access"
- General ACLs allow arbitrary access, but hard to manage
- Solution: groups

Groups

- A group is a single id such as
  "Berkeley-undergrads"
  "friends of Alice"
  "administrative access"
- A group administrator maintains group membership list

More on UNIX chmod

- World: read/write/execute
- Group: read/write/execute
- Owner: read/write/execute
- Can change owner using chown command

Temporary Access

- This is an area where capabilities systems excel
  - "Transferring a capability"
  - Sometimes like giving a reference
- ACL systems need special mechanism
  - UNIX: "setuid" bit
  - Windows NT/XP: "Run as"

Procedure-Oriented Access Control

- Run a program to determine access
- Ex: Web server access

Monotonic vs. Non-Monotonic

- Classic access control was monotonic
- As we acquire more capabilities, or identities, we get more powerful
  - "root", "super-user", "Administrator"
- But this often causes problems
  - What if "root" password is discovered?

Non-Monotonic Access Control

- With non-monotonic access control, as we gain identities or capabilities, we may lose access
- Ex: Windows file sharing (administrators have crippled access)

Distributed Access Control

- Distributed access control is an active research area
- Ex: who can access an encrypted satellite broadcast?
  - Users join and leave all the time
  - Millions or tens of millions of users
- Solution: "Distributed key distribution"
  - Must be efficient…

Compatibility of Access Control

- ACLs predominate, but each system implements them in their own way
- Systems must "translate" access control
  - SAMBA supports Windows and Unix-like systems
- Continual source of serious errors

Autonomous Access Control

- Each system manages its own access control
- Requires remote login
- Problem: people often access hundreds or thousands of systems, and necessarily reuse login info (passwords)
- Common password problem
- We will revisit these issues in the course

Summary

- Access Control is Central to Security
  - We'll return to access control repeatedly in the course
  - Old area of security, but not well understood
  - Often poorly implemented
  - And we haven't even begun to look at "backdoors"!
- Security analysis framework for a complex computer system
  - Structures the security evaluation
  - Defines the process

Background: Terminology

- An *attack* is an attempt to breach system security – Not all attacks are successful
- A *threat* is a circumstance or scenario with the potential to cause harm to a system
  - An attack usually refers to a specific stratagem whereas threat refers to a broader class of ways that things could go wrong
- A *vulnerability* is an aspect of the system that permits someone to mount a successful attack. Sometimes called a *security hole*
  - A *security weakness* is like a vulnerability, only it is less clear whether it could actually lead to any direct violation of the security goals
  - A weakness might represent a potential vulnerability whose risk is unclear; or, several weaknesses might combine to yield a full-fledged vulnerability

Background: Terminology (cont'd)

- A *security goal* is a goal that is supposed to be achieved by the system; if it fails, the system will be considered insecure
- A *threat assessment* is an attempt to assess the set of all possible threats
- A *threat model* is a characterization of the possible threats, usually produced during a threat assessment
- *24 by 7* refers to the window of time in which systems are most vulnerable to attack
  - (Ok, this one is a joke from http://www.csoonline.com/read/080105/debrief.html)