CS 194-1 (CS 161) Computer Security

Lecture 1

Class Introduction

August 28, 2006
Prof. Anthony D. Joseph
http://cs161.org/

CS 161 (194-1) Basic Facts

- This is a class about computer security
  - 4 units
- This is the second offering of this class
  - It will become CS 161
- To take this class, you need patience, an open mind, and a willingness to work hard and participate

Adding This Class

- If you are an upper division declared major and currently on the waiting list
  - You have a good chance of getting in
  - Talk to Michael-David Sasson this week
- If you want to add and aren't in already
  - Get on the waiting list ASAP!

Berkeley – Leader in Security Research

- TRUST (Berkeley leads consortium)
- DETER (Berkeley leads consortium)
- ACCURATE
- NEST
- Cryptosystems research
- Security and HCI
- Security for NSF, DoD, DHS, USPS, DOE, etc.

Course Staff

- Instructors
  - Anthony Joseph
    - » (adj@cs)
    - » 675 Soda
  - Doug Tygar
    - » (tygar@cs)
    - » 531 Soda and 307B South
- TAs (so far …)
  - Marco Barreno
    - » barreno@cs
  - Todd Kosloff
    - » koslofto@cs




Sections

- Three sections on Thursdays in 320 Soda
  - 101. 10:00-11:00
  - 102. 11:00-12:00
  - 103. 3:00- 4:00
- Attendance and participation are mandatory
  - Part of class participation grade

Grading

- Project (35%)
  - Two parts, three grace days
- Exams (45%)
  - Midterm 1 (October 9, 15%)
  - Midterm 2 (November 6, 15%)
  - Final (December 16, 2006 8-11am, 15%)
    - » Let us know ASAP about conflicts
- Homework (10%)
  - 3-4 assignments – lowest score dropped
- Class participation (10%)

Final Grade

- Final grade = (ethics grade) * (academic grade)
- Ethics grade will normally be 1
- Ways to get a 0 ethics grade:
  - Violate campus computing policy
  - Violate privacy of other people without permission
  - Tamper with data of other people without permission
  - Fail to report a vulnerability or an observation of unethical behavior
  - Unethical behavior may be referred for additional disciplinary action

Class Participation

- Showing up is the first step
- Asking (or answering) questions is good
  - (but don't filibuster)
- Having your cell phone ring in class is bad
  - Taking the cell phone call in class is worse
- Treat students and staff with dignity

Collaborative Work

- Projects will be in groups of four
  - All must be in the same section
- Homeworks are done individually
- You may use the following resources:
  - Instructors, TAs, assigned texts, posted notes
- No consulting others; No "Googling for the answer"
  - Consult with TAs over problem cases
  - Always cite references – plagiarism is not permitted

Academic Dishonesty Policy

- Copying all or part of another person's work, or using reference material not specifically allowed, are forms of cheating and will not be tolerated. A student involved in an incident of cheating will be notified by the instructor and the following policy will apply:

  http://www.eecs.berkeley.edu/Policies/acad.dis.shtml
- The instructor may take actions such as:
  - require repetition of the subject work,
  - assign an F grade or a 'zero' grade to the subject work,
  - for serious offenses, assign an F grade for the course.
- The instructor must inform the student and the Department Chair in writing of the incident, the action taken, if any, and the student's right to appeal to the Chair of the Department Grievance Committee or to the Director of the Office of Student Conduct.
- The Office of Student Conduct may choose to conduct a formal hearing on the incident and to assess a penalty for misconduct.
- The Department will recommend that students involved in a second incident of cheating be dismissed from the University.

Textbooks

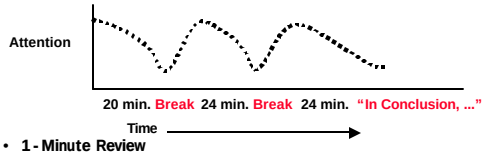
- Computer Security, 2nd ed.(Dieter Gollmann)
  - *Required*

- Security Engineering (Anderson)
  - *Optional*
  - Available in online form

Other Class Resources

- cs161.org
  - Lecture notes, pointers to some readings, and assignments are posted here
- Newsgroup: ucb.class.cs161 (read daily!)
  - This is the place to ask questions and receive answers
- cs161-xx@cory account
  - Every student who is enrolled should get a cs161 account form at end of lecture
  - This account is required for submissions
  - Make sure to log into your new account this week and fill out the questions

Typical Lecture Format



- 1-Minute Review
- 20-Minute Lecture
- 5- Minute Administrative Matters
- 2-Minute Break (stretch)
- 24-Minute Lecture
- 4-Minute Break (water, stretch)
- 24-Minute Lecture
- Instructors will come to class early & stay after to answer questions

Lectures (tentative)

- Aug 28 Overview; intro to computer security, adversaries, security goals
- Aug 30 Threat models, access control, authorization
- Sept 4 No class! Labor Day Holiday.
- Sept 6 Network security intro and networking background
- Sept 11 Symmetric-key cryptography, block ciphers

Lectures (tentative)

- Sept 13 Public-key encryption, modular arithmetic background
- Sept 18 Message authentication, public-key signatures, secret sharing
- Sept 20 Cryptographic protocols, zero knowledge protocols
- Sept 25 Authentication protocols
- Sept 27 Random number generation
- Oct 2 Firewalls, intrusion detection

Lectures (tentative)

- Oct 4 Midterm review
- Oct 9 Midterm 1
- Oct 11 Secure channels, web security
- Oct 16 Implementation flaws, buffer overruns, software security (principles)
- Oct 18 Software security (defensive programming)
- Oct 23 Electronic cash protocols, electronic commerce systems

Lectures (tentative)

- Oct 25 Multi-level security, mandatory access control, trusted computing
- Oct 30 Database security: side channels, inference control
- Nov 1 Midterm review
- Nov 6 Midterm 2
- Nov 8 Worms and viruses, Distributed Denial of Service
- Nov 13 Isolation, sandboxing, language-based security (type- and memory-safe languages)

Lectures (tentative)

- Nov 15 Rights management, spyware, rootkits (case study)
- Nov 20 Watermarking, electronic voting
- Nov 22 No class! Thanksgiving Holiday
- Dec 4 OS security, memory protection, rootkits
- Dec 6 Course Review
- Special Topics?
  - Post your requests!

Class Scope

- How to build secure systems
  - Techniques for designing, implementing, and maintaining secure systems
- How to evaluate the security of systems
  - How to tell whether a system is secure
  - How systems have failed in the past
  - How attackers break into real systems, and how to tell whether a given system is likely to be secure.
- Learn the science of cryptography
  - How to communicate securely over an insecure communications medium

Why is Security such a problem?

- Monoculture computing environment
- Web, e-commerce, & distributed / collaborative applications
- Internet spans national boundaries
- Poor programming practices

Two Security Nightmares

- "The transparent society" by David Brin
  - Will Technology Force Us to Choose Between Privacy and Freedom?
- "Electronic Pearl Harbor"
  - Is this just scare-mongering?
  - Slammer worm took down Bank of America's ATM network, Seattle 911 service, Ohio's Davis-Besse nuclear power plant's safety monitoring system, …
  - Nachi worm invaded Diebold ATMs?
  - Real worries about e-voting validity
  - Millions of CC #s, SS #s leaked

BREAK

What is Computer Security Today?

- Computing in the presence of an adversary!
  - An *adversary* is the security field's defining characteristic
- Reliability, robustness, and fault tolerance
  - Dealing with Mother Nature (random failures)
- Security
  - Dealing with actions of a knowledgeable attacker dedicated to causing harm
  - Surviving malice, and not just mischance
- Wherever there is an adversary, there is a computer security problem!

Adversaries

- Adversaries are everywhere!
- Code Red worm infected 250,000 in less than a week
  - Contained a time-bomb set to attack the White House web server on a specific date
  - Fortunately, attack on White House was diverted
  - Estimates: $2 billion in lost productivity and infected machine clean up
- Estimate: in 2003 viruses cost businesses over $50 billion

Zombie Networks

- 2005 estimate
  - Over 1M computers penetrated and "0wned" by malicious hackers (crackers) for phishing, spamming, identity theft, extortion
- Crackers build zombie networks of 10K-1M compromised machines & sell services
  - Ex: Take down competitor's website for $1K
- Hugely profitable!
  - Massive spamming, ID fraud through phishing
  - Roughly half of all spam is sent by zombies
- How can we secure our machines against folks like this?
  - *That's the subject of this class!*

Computer Security History

- Early history interwoven with military apps
  - First big users of computers
  - First to worry seriously about the potential for misuse
- Terminology has military connotations:
  - *Attacker* who is trying to *attack* computer systems
  - *Defenders* working to protect their system from these *threats*

Analyze to Learn!

- We're going spend a lot of time studying attackers and thinking about how to break into systems
  - Why spread knowledge that will help bad guys be more effective?
- To protect a system, you have to learn how it can be attacked
  - Civil engineers learn what makes bridges fall down so they can build bridges that last
  - Software engineering is similar
- Security is the same and different!
  - Why?

Challenges in Securing Systems

- Similar:
  - Analyze previous successful attacks
- But, deploy a new defense, they respond, you build a better defense, they respond, you…
  - Need to find ways to anticipate kinds of attacks
- Different:
  - Attackers are intelligent (or some of them are)
  - Attacks will change and get better with time
  - Have to anticipate future attacks
- Security is like a game of chess
  - Except the attackers often get the last move!

Reality: Static Systems

- A deployed system is very hard to change
  - Serious consequences if attackers find a security hole in a widely deployed system
- Goal: Predict *in advance* what attackers might do and eliminate all security holes
- Reality: Have to think like an attacker
- Thinking like an attacker is not always easy
  - Can be fun to try to outwit the system
  - Or can be disconcerting to think about what could go wrong and who could get hurt
- What if you don't anticipate attacks?
  - Analog cellular phones in the 80's and 90's

Real-World Example: Analog Cellular

- 1970's: analog cellular had no security
  - Phones transmit ID/billing info in the clear
  - Assumption: attackers wouldn't bother to assemble equipment to intercept info…
- Attackers built "black boxes" to intercept and clone phones for fraudulent calling
  - Where's the best place to intercept?
  - Cellular operators completely unprepared
- Early 90's, US carriers losing >$1B/yr
  - 70% of LD cellular calls placed from downtown Oakland on Fri nights fraudulent
- Problems: huge capital investment/debt, 5–10 yrs & huge replacement cost

Lesson Learned

- Failing to anticipate types of attacks, or underestimating the threat, can be costly
- Security design requires studying attacks
  - Security experts spend a lot of time trying to come up with new attacks
  - Sounds counter-productive (why help the attackers?), but it is better to learn about vulnerabilities before the system is deployed than after
- If you know about the possible attacks in advance, you can design a system to resist those attacks
  - But, anything else is a toss of the dice…

A Process for Security Evaluation

- How do we think about the ways that an adversary might use to penetrate system security or otherwise cause mischief?
- We need a framework to help you think through these issues
- Start with *security goals* or in other words:
  - 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?

Some Common Security Goals

- *Confidentiality*:
  - Private information that we want to keep secret from an adversary (password, bank acct balance, diary entry, …)
  - Anything we want to prevent adversary from learning
- *Integrity*:
  - Want to prevent adversary from tampering with or modifying information
- *Availability*:
  - System should be operational when needed
  - Must prevent adversary from taking the system out of service at inconvenient times

Example: CS161 Grades Database?

- One obvious goal is protecting its integrity
  - Don't want you to be able to give yourself an A+ merely by tampering with grade database
- Federal law and university rules require us to protect its confidentiality
  - No one else can learn what grade you are getting
- We probably also want some level of availability
  - So you can check your grades to date and we can calculate grades at the end of the semester

Security Goals

- Can be simple or detailed
  - Exercise in requirements analysis
  - Specification of what it means for a system to be secure
  - Want goals to be met even when adversary tries to violate them
- Which goals are security goals?
  - Highly application-dependent
  - If someone figures out how to violate this goal, would it be a security breach?
    - » If yes, you've found a security goal!

Security Goals Summary

- "A program that has not been specified cannot be incorrect; it can only be surprising." [Young, Boebert, and Kain]
- Or, a system without security goals has not been specified, and cannot be wrong; it can only be surprising

"In conclusion..."

- Goals of this class
  - Solid foundation in understanding security
  - Key info about building secure systems
  - Introduce range of topics in security
  - Interest some of you in further study
- Adversaries are everywhere
  - Fact of life, plan for them!
- Systems become static after deployment
- Analyze the past to prepare for the future
  - Determine security goals
  - We need a process to drive the analysis…