CIS 551 / TCOM 401

Computer and Network Security

Spring 2008
Lecture 1

Course Staff

- Steve Zdancewic (Instructor)
  - Web: www.cis.upenn.edu/~stevez
  - Office hours: Tues: 9:30 - 10:30 am, and by appointment
  - Office: Levine 511

- TA: To be announced

- Send mail to:
  - cis551@seas.upenn.edu
  - cis551staff@seas.upenn.edu (will be set up soon)

  - Your chances of getting a prompt reply increase if you put cis551 in the subject of your mail.

Course Information

- Course Web Page:
  - www.cis.upenn.edu/~cis551
- News group:
  - upenn.cis.cis551
- Textbook: none
  - Assigned reading: articles and web pages
  - Lecture slides will be available on the course web pages
  - Handouts and notes as appropriate

Prerequisites

- Would like to learn about computer and network security.

- Some programming experience
  - Java
  - C or C++ helpful (but not necessary - you can pick up what you need to know)

- Some computer networks experience
  - Do you know what a protocol stack is?
  - Do you generally understand TCP/IP?
  - TCOM 500

- Note: Undergraduates should take 551 (331 is now merged with this class)

Grading Criteria

- 16% Midterm I - tentative date: Feb. 19th
- 16% Midterm II - tentative date: April 1st
- 25% Final exam
- 40% Course projects (group projects)
- 03% Course participation

- Policies:
  - No individual work on group projects
  - Only “reasonable” regrade requests permitted
  - See course web pages

Student Background…

1. How many of you have programmed in C or C++?
2. How many of you have programmed in Java?
3. How many of you have written shell scripts?
4. How many of you have never done any programming?
5. How many of you can explain how a buffer overflow exploit works?
6. Have any of you written a buffer overflow exploit?
7. How many of you can explain how TCP/IP works?
8. How many of you have set up a wireless network?
9. How many of you have had experienced a virus or worm attack on some computer you care about?
10. Have any of you written a virus or worm?

Student Background…

11. How many of you regularly use SSH or SFTP?
12. How many of you can explain how they work?
13. How many of you have run a packet sniffer or port scanner?
14. How many of you can define the term “Trusted Computing Base”?
15. How many of you have used a debugger?
16. How many of you are Masters students?
17. How many of you are PhD students?
18. How many of you are Undergraduates?

Course Topics

- Software Security / Malicious Code
  - Buffer overflows, viruses, worms, protection mechanisms
- System Security
  - Hacker behavior, intrusion & anomaly detection, hacker and admin tools
- Networks & Infrastructure
  - TCP/IP, Denial of Service, IPSEC, TLS/SSL
- Internet Security
  - Viruses, worms, spam, web security (XSS), phishing
- Basic Cryptography
  - Shared Key Crypto (AES/DES), Public Key Crypto (RSA)
- Crypto Software & Applications
  - Cryptographic libraries, authentication, digital signatures
- Covert Channels

Outline

- Try to answer the questions:
  - What is computer security?
  - What do we mean by a secure program?
- Historical context
  - Basic definitions & background
  - Examples of security
- General principles of secure design
- Focus on one widespread example:
  - Buffer overflows

Software Vulnerabilities

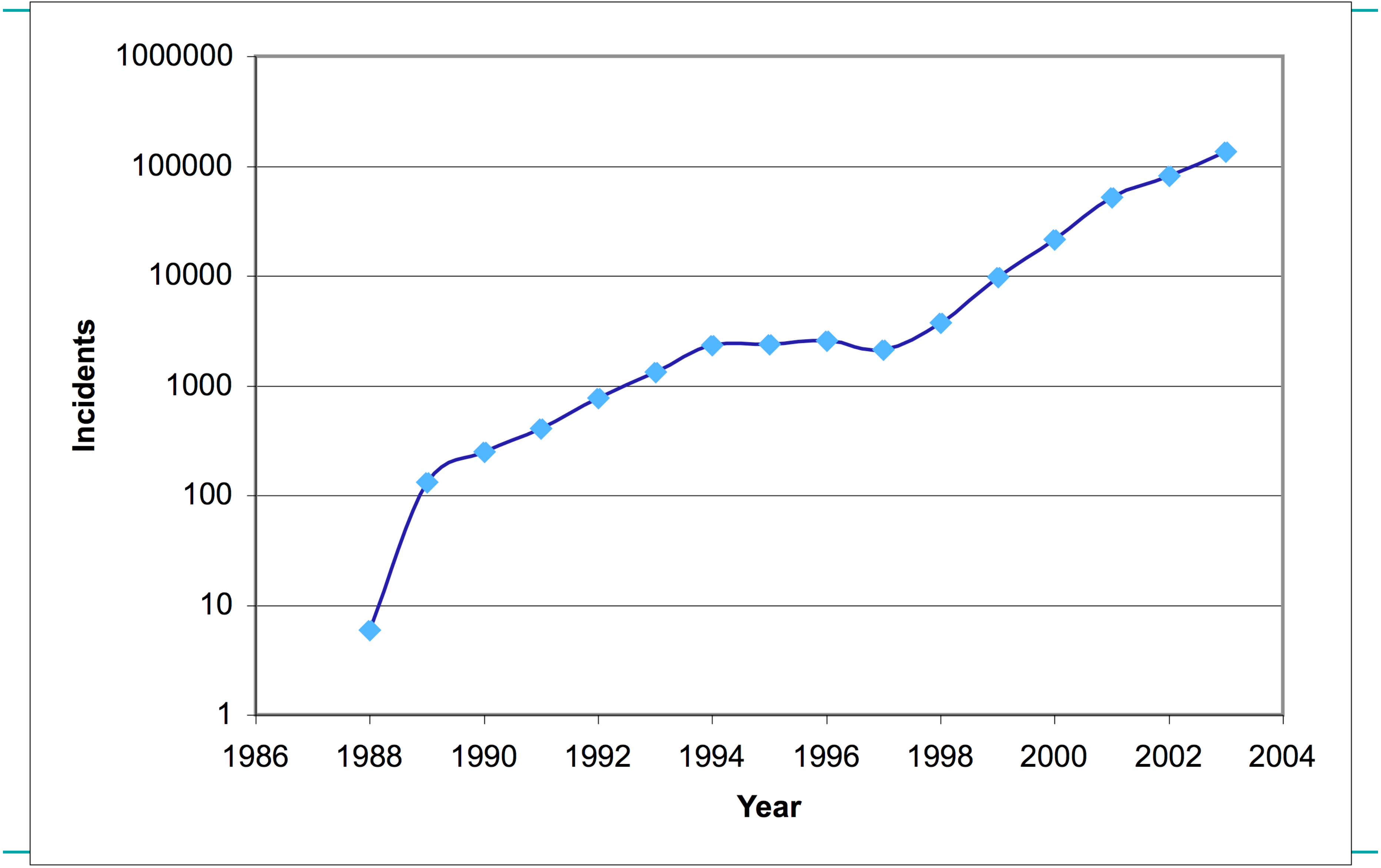
- Every day you read about new software vulnerabilities in the news
  - Buffer overflows
  - Cross-site scripting
  - Format-string vulnerabilities
  - Spam
  - Worms/Viruses
  - Phishing

- Check out www.cert.org for plenty of examples

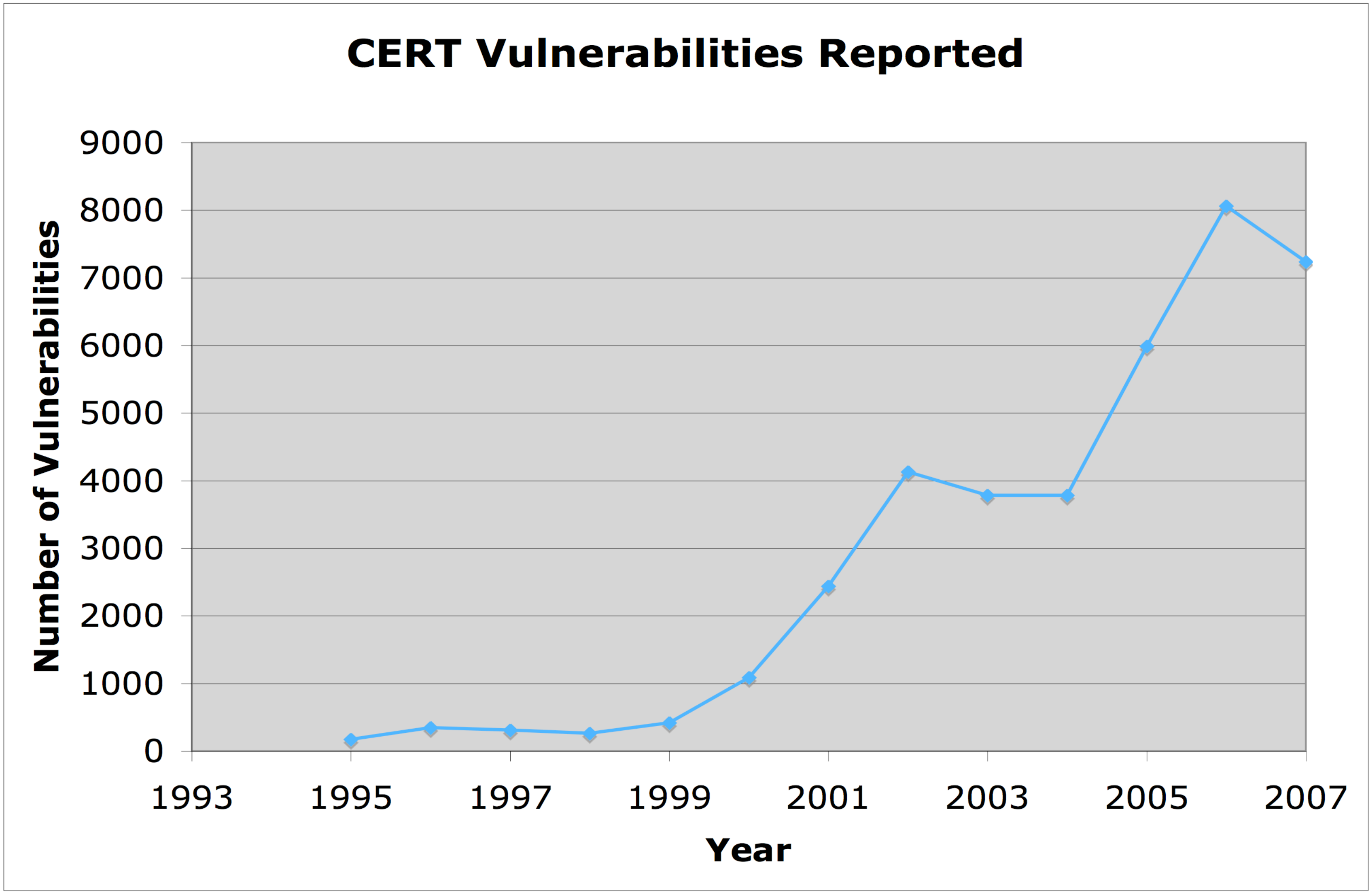
Slashdot Security Headlines in 2008

- US Policy Would Allow Government Access to Any Email
- Lax TSA Website Exposed Travelers' Information
- Coverity Reports Open Source Security Making Great Strides
- 95 Of Every 100 Windows PCs Miss Security Updates
- Identity Theft Skeptic Ends Up As Fraud Victim
- XP/Vista IGMP Buffer Overflow — Explained
- US DHS Testing FOSS Security
- Mass Hack Infects Tens of Thousands of Sites
- Boot Record Rootkit Threatens Vista, XP, NT
- Boeing 787 May Be Vulnerable to Hacker Attack
- Facebook Widget Installs Zango Spyware
- Researchers Say Wi-Fi Virus Outbreak Possible
- Four Root DNS Servers Go IPv6 On February 4th
- Sears Installs Spyware

CERT Incidents



CERT Vulnerabilities



What do we mean by security?

- What does it mean for a computer system to be secure?
- Comments generated from class discussion:
  - Game: no one cheating?
  - Game: the software shouldn't lie
  - It shouldn't do "bad" things
  - No spyware - no "botnet"
  - No bugs ==> fewer security holes

When is a program secure?

- When it does exactly what it should?
  - Not more.
  - Not less.

- But how do we know what a program is supposed to do?
  - Somebody tells us? (But do we trust them?)
  - We write the specification ourselves? (How do we verify that the program meets the specification?)
  - We write the code ourselves? (But what fraction of the software you use have you written?)

When is a program secure?

- 2nd try: A program is secure when it doesn’t do something it shouldn’t.

- Easier to specify a list of “bad” things:
  - Delete or corrupt important files
  - Crash my system
  - Send my password over the Internet
  - Send threatening e-mail to the president posing as me

- But… what if most of the time the program doesn’t do bad things, but occasionally it does? Is it secure?

When is a program secure?

- Claim: Perfect security does not exist.
  - Security vulnerabilities are the result of violating an assumption about the software (or, more generally the entire system).
  - Corollary: As long as you make assumptions, you're vulnerable.
  - And: You *always* need to make assumptions!

- Example: Buffer overflows
  - Assumption (by programmer) is that the data will fit in the buffer.
  - This leads to a vulnerability: Supply data that is too big for the buffer (thereby violating the assumptions)
  - Vulnerabilities can be *exploited* by an *attack.*

When is a program secure enough?

- Security is all about tradeoffs
  - Performance
  - Cost
  - Usabilitity
  - Functionality

- The right question is: how do you know when something is secure enough?
  - Still a hard question
  - Requires understanding of the tradeoffs involved

- Is Internet Explorer secure enough?
  - Depends on context

How to think about tradeoffs?

- What is it that you are trying to protect?
  - Music collection vs. nuclear missile design data

- How valuable is it?

- In what way is it valuable?
  - Information may be important only to one person (e.g. private e-mail or passwords)
  - Information may be important because it is accurate and reliable (e.g. bank’s accounting information)
  - A computer system may be important because of a service it provides (e.g. Google’s web servers)

Historical Context

- Assigned Reading:
  Saltzer & Schroeder 1975
  The Protection of Information in Computer Systems
  - available from course web pages

- Unauthorized information release
  - *Confidentiality*
- Unauthorized information modification
  - *Integrity*
- Unauthorized denial of use
  - *Availability*

- What does “unauthorized” mean?

Example Security Techniques

- Labeling files with a list of authorized users
  - Access control (must check that the user is permitted on access)
- 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 over telephone lines
  - Cryptography
- Locking the room containing the computer
  - Physical aspects of security
- Controlling who is allowed to make changes to a computer system (both its hardware and software)
  - Social aspects of security

Case Study: Buffer Overflows

- First project: Due: 8 Feb. 2007 at 11:59 p.m.
- http://www.cis.upenn.edu/~cis551/project1.html

- Group project:
  - 2 or 3 students per group
  - Send e-mail to TA with your group by Jan. 25th

- Assigned Reading:
  Aleph One (1996)
  Smashing the Stack for Fun and Profit
- This is essentially a tutorial for the project