CS466: Computer Security

Clyde D. Ruby, Ph.D.

Ralph Bunker, Ph.D.

What is a Trojan Horse?

Maharishi University of Management

Computer Science Department

CS 440

Compiler Construction:
The Connection of Name and Form

Clyde D. Ruby, Ph.D.

Course Goals

- Learn the 8 basic principles of computer security
- Learn the 19 sins of software security
- Learn the various ways of controlling access to resources
- Learn key exchange protocols
- Learn public key cryptography including the Public Key Infrastructure
- Learn what software assurance is
- Learn how viruses work
- Learn models used for intrusion detection and vulnerability detection
- Become familiar with a secure system based on principles learned during the course
- Understand the intelligence embodied in this course in terms of SCI

Social networking sites

- Even computer security pros vulnerable to scams

  “Computer security professionals tend … to see potential threats everywhere. It turns out that some aren’t cautious enough, though.”

  …

  “A relatively simple ruse persuaded dozens of prominent security analysts to connect on their social networking Web pages with people who weren’t friends at all. They were fake profiles, purportedly of other well-known security pros. The scam was designed to expose the trust that even some of the most skeptical Internet users display on some of the most insecure sites on the Web.”

Lecture 1

Introduction to Computer Security:

The Unmanifest Field of Pure Consciousness is an Infinite Continuum of Invincibility

Wholeness Statement

- Computer security rests on confidentiality, integrity, and availability. The effectiveness of TM is based on keeping the mantra private and learning it from a teacher trained by Maharishi.

CS466: Computer Security

- Could be the introduction to four other courses or seminars:
  1. Cryptography
  2. Secure Programming (19 deadly sins)
  3. Intrusion detection techniques
  4. Penetration testing

CS466

- Security topics could be included in the following courses taught at M.U.M.
  1. **CS465** Operating Systems (Secure kernel, race conditions)
  2. **CS450** Computer Networks (TCP/IP stack vulnerabilities)
  3. **CS545** Distributed Computing (Web application vulnerabilities)
  4. **CS425** Software Engineering (Fuzz testing, malicious, not just error and mischance)
- Conversely these courses could be prerequisites for CS466 (at least the first three would be useful, plus the Compiler Construction course)

CAEIAE

- Centers of Academic Excellence in Information Assurance Education
  - National Security Agency (NSA) sponsored program
  - Certify degree and certificate programs in information assurance
- Information assurance covers the confidentiality and integrity of information

Masters in Information Security,
James Madison University

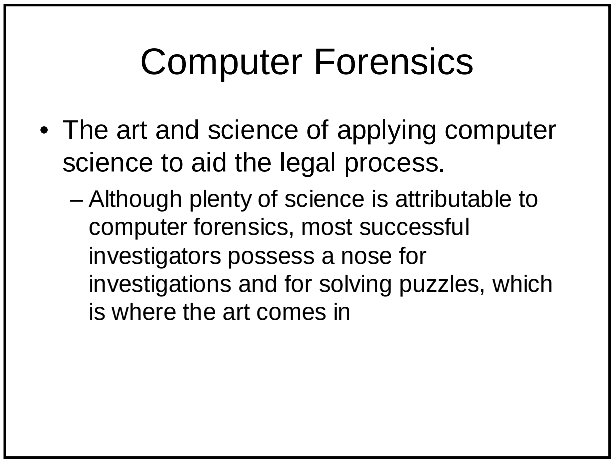
- Operating Systems
- Networks and Network Security
- Distributed Computing and Security
- Secure Software Engineering
- Formal Methods for Information Security
- Ethics, Law and Policy in Cyberspace
- Software Assurance
- Cryptography: Algorithms and Applications
- Computer Forensics
- Secure Operations

Masters of Information Assurance, Iowa State University

- An interdisciplinary program supported by six departments (Math, CS, CE, IE, MIS, PoliSci).
- Core/Required Courses

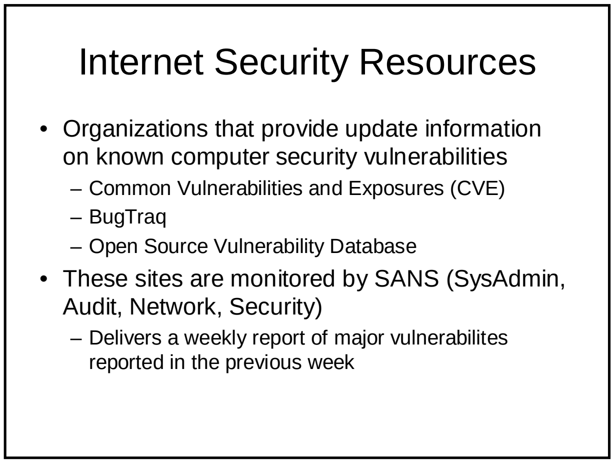
  InfAs 531: Computer Security
  InfAs 534: Ethical and Legal Issues in Computer Security
  InfAs 530: Advanced Computer Networking
  or ComS 586 Computer Network Architecture
  InfAs 533: Cryptography
- Elective Courses (4-5 more courses)

  InfAs 532: Information Warfare
  InfAs 536: Computer and Network Forensics
  InfAs 592: Seminar in Information Assurance
  (others are selected from a list of related CS, CE, MIS, Math, IE, and Political Science courses)
- Thesis/Creative Component (research)



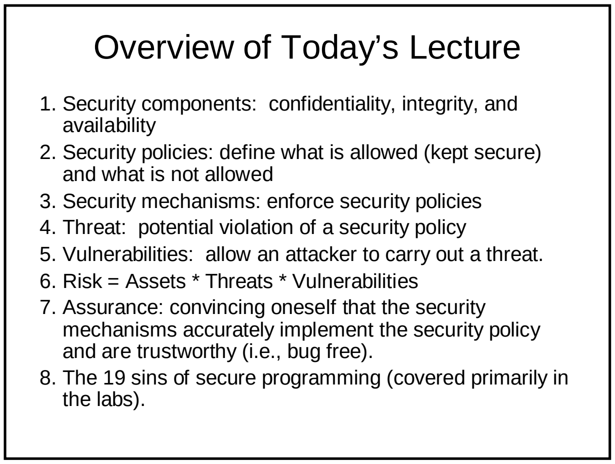
Computer Forensics

- The art and science of applying computer science to aid the legal process.
  - Although plenty of science is attributable to computer forensics, most successful investigators possess a nose for investigations and for solving puzzles, which is where the art comes in



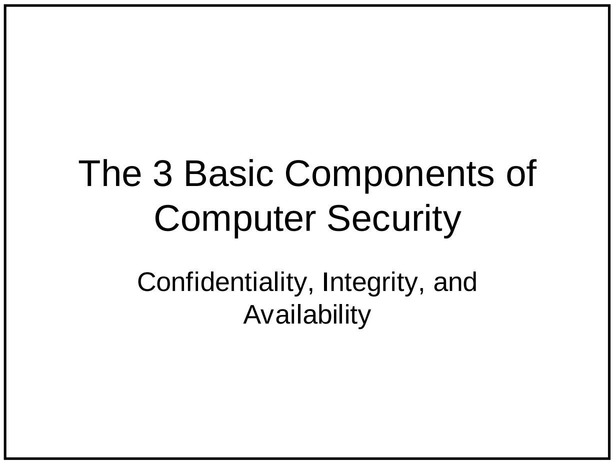
Internet Security Resources

- Organizations that provide update information on known computer security vulnerabilities
  - Common Vulnerabilities and Exposures (CVE)
  - BugTraq
  - Open Source Vulnerability Database
- These sites are monitored by SANS (SysAdmin, Audit, Network, Security)
  - Delivers a weekly report of major vulnerabilites reported in the previous week



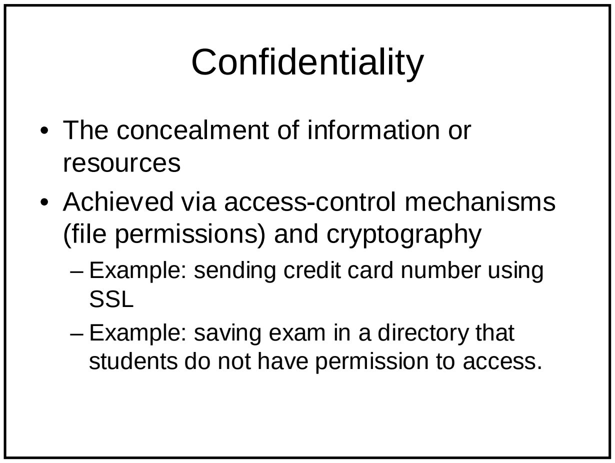
Overview of Today's Lecture

1. Security components: confidentiality, integrity, and availability
2. Security policies: define what is allowed (kept secure) and what is not allowed
3. Security mechanisms: enforce security policies
4. Threat: potential violation of a security policy
5. Vulnerabilities: allow an attacker to carry out a threat.
6. Risk = Assets * Threats * Vulnerabilities
7. Assurance: convincing oneself that the security mechanisms accurately implement the security policy and are trustworthy (i.e., bug free).
8. The 19 sins of secure programming (covered primarily in the labs).



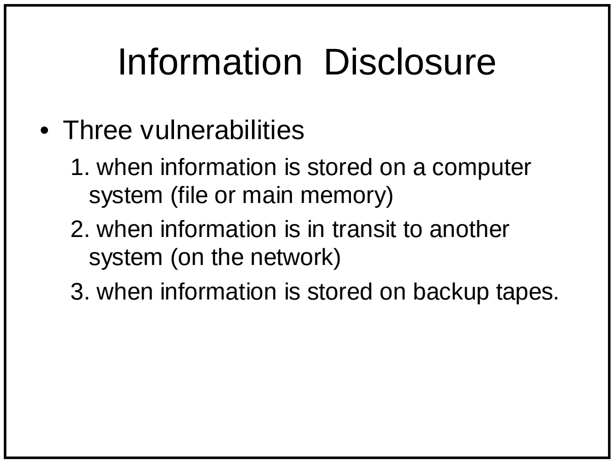
The 3 Basic Components of Computer Security

Confidentiality, Integrity, and Availability



Confidentiality

- The concealment of information or resources
- Achieved via access-control mechanisms (file permissions) and cryptography
  - Example: sending credit card number using SSL
  - Example: saving exam in a directory that students do not have permission to access.



Information Disclosure

- Three vulnerabilities
  1. when information is stored on a computer system (file or main memory)
  2. when information is in transit to another system (on the network)
  3. when information is stored on backup tapes.

Confidentiality Mechanisms

- Ultimately based on trust in the OS
  - OS supports the confidentiality mechanisms
  - E.g., read/write protection of files
- Things we can do
  - don't let hackers know what server or OS we are using
  - conceal the mere existence of data

Integrity

- The trustworthiness of data or resources
  - usually phrased in terms of preventing improper or unauthorized changes
- Can we trust the data/information?
- Includes

  data integrity - content of the information

  source integrity - source of the information
- The source bears on the accuracy and credibility of the information
  - Determines the level of trust we have

Integrity Mechanisms

- Two types of mechanism: prevention and detection
- Prevention attempts to block
  - unauthorized changes to data (hacker changing data)
  - changes to data in unauthorized ways (accountant embezzling data)
- Detection
  - Does not try to prevent violations of integrity
  - Only reports when data has been modified
  - Example: detect that an Internet order was modified between the client and the server
  - Example: detect that a system file has been modified by a virus

Trust

- Confidentiality places its trust in OS
- However, integrity also relies on assumptions about the source of the data and the level of trust in the source

Main Point

1. Trust in the OS and its security system underlies the confidentiality and integrity mechanisms. Pure consciousness underlies all of manifest creation. Having the home of all knowledge in one's awareness makes one at home and confident in any situation.

Availability

- The ability to use the information or resource desired
- Availability security problems can be malicious or inadvertent
  - Power generator explosion
  - Denial of service attack (DoS)
    - An attacker might swamp a server with requests, thereby making it unavailable to legitimate users

How do we know whether an increase in activity is malicious?

- Example: If a TV or radio commentator mentions a web site, its activity will increase, perhaps to a point where the site crashes.
  - Whether deliberate or not, it is still a security problem.

Threats

- *Threat*: A potential violation of security
  - The violation need not occur for there to be a threat
- *Attack*: An action that could cause a threat to be realized
  - Security mechanisms must guard against and/or prepare for attacks
- *Attacker*: The perpetrator of an attack

Examples

- If I have a beat-up old bicycle that nobody wants there is no threat that somebody will steal it
- However, there is a threat that somebody will steal a laptop computer

- Confidentiality, integrity, and availability security mechanisms need to counter threats to the security policies of a system
- Threats recur everywhere in the study of security, so we are going to categorize them and then look at them in more detail through examples

Four Classes of Threats

- *disclosure* - unauthorized access to information (confidentiality)
- *deception* - acceptance of false data (integrity)
- *disruption* - interruption or prevention of correct operation (availability)
- *usurpation* - unauthorized control of some part of a system (confidentiality, integrity, availability)

Some Common Threats

1. Snooping (disclosure)

   Example: interception of information, possibly through passive listening or accessing of communication or files

   Countered by confidentiality mechanisms
2. Modification or alteration (deception, disruption, usurpation)

   Example: man in the middle (active interception)

   Countered by integrity mechanisms.
3. Masquerading or spoofing (deception, usurpation)

   Example: phishing, stealing password

   Countered by integrity mechanisms (authentication).

4. Repudiation of origin (deception)
   Example: customer denies having ordered a product
   Countered by integrity mechanism (signatures)
5. Denial of receipt (deception)
   Example: vendor ships product, but customer denies getting it.
   Countered by integrity and availability mechanisms.
6. Delay (availability)
   Example: delay primary server to force request to secondary server which is controlled by the attacker.
   Countered by availability mechanisms.
7. Denial of service (availability), essentially an infinite delay
   Example: A SYN flood attack
   Countered by availability mechanisms.

Main Point

2. A threat is a potential violation of security. SCI's approach to mitigating threats is to disallow the birth of an enemy (attacker).

Security vs. Mechanism

- Definition 1-1: A *security policy* is a statement of what is, and what is not allowed.
- Definition 1-2: A *security mechanism* is a method, tool, or procedure for enforcing a security policy.
  - Note that the procedure may be non-technical, e.g. presenting a photo ID such as a driver's license when changing a password.
  - Technology cannot enforce all policies

Example

- M.U.M. has a no-copy security policy.
  - The mechanism used is to set file permissions of each student's directory so that only that student can read and write files to it.
  - The lab today will verify that this is the case.

Example

- The following areas might be covered in a security policy for a company
  1. Web surfing
  2. Virus precautions
  3. Personal email
  4. Software installation
  5. The firewall (must not bypass firewall)
  6. Encryption (must encrypt files using a specific encrypting system)
  7. The law (Computer Misuse Act 1990)
  8. Logging and Surveillance
  9. Passwords (how strong they are)
  10. Laptops

Goals of Security

- The security policy defines what is secure and non-secure.
  - implied that secure items have a threat associated with them.
- Three strategies to handle attacks against secure items
  1. Prevention - the attack will fail, e.g., firewall discards request
  2. Detection - accept that an attack will occur but detect it as soon as possible, e.g., Tripwire.
  3. Recovery
     a. Stop the attack, possibly disrupting the system and assess and repair any damage
     b. System continues to function normally while attack is underway.

Assumptions and Trust

- How do we determine if a security policy describes the required level and type of security for a site?
- Example - At MUM, locks are assumed to be secure against lock picking. But in a prison that assumption may not be valid. Assumption is still valid if lock picker is trustworthy.

Assumptions and Trust

- A policy consists of a set of axioms that the policy makers believe can be enforced
- Designers of security policies always make two assumptions
  1. The policy correctly and unambiguously partitions the set of system states into "secure" and "non-secure" states
  2. The security mechanisms prevent the system from entering a "non-secure" state
- Two things can go wrong
  1. The policy does not describe what a "secure" system is.
  2. The mechanisms do not enforce the policy.

Assurance

- The security mechanisms have to be trusted because they are critical to the security of the system.
- The security mechanisms must be shown to be trustworthy;
  - this is done through software assurance.
- *Assurance*: convincing oneself that the security mechanisms accurately implement the security policy and are trustworthy (i.e., bug free).

Security Mechanisms

- Classified as either *secure*, *precise* or *broad*

	Secure	Precise	Broad
non-secure state can be entered	No	No	Yes
a secure state cannot be entered	Yes	No	?

Assumptions

- Trusting that the mechanisms work requires several assumptions
  1. Each mechanism is designed to implement one or more parts of the security policy
  2. The union of the mechanisms implements all aspects of the security policy
  3. The mechanisms are implemented correctly
  4. The mechanisms are installed and implemented correctly.

Example

- **Security policy**: Consider a policy that describes who is allowed to have keys to faculty offices and that it must not be possible for a student to enter a faculty office without the faculty member's knowledge.
- **Initial state**: all faculty offices start out locked and empty, so the system starts out in a secure state.
- **Non-secure state**: if a student manages to get into a faculty member's office without permission.
- **Security mechanisms**: the locks, keys, and the rules for who gets keys to faculty offices.

- Are there any security problems with this system, i.e., how secure is this system?

Assurance

- Trust/trustworthiness cannot be quantified precisely.
  - System specification, design, and implementation can help establish trust/trustworthiness.
  - This aspect of trust is called assurance which is an attempt to quantify trust/trustworthiness by evaluating the specification, design and implementation of the system.

Operational Issues

- Any useful policy and mechanism
  - must balance the benefits of the protection against the cost of designing, implementing, and using the mechanism.
  - This balance can be determined by analyzing the risks (cost) of a security breach and the likelihood of it occurring.

Cost-Benefit Analysis

- If data and resource being protected cost less than the security mechanisms to protect them,
  - then protecting them is not cost-effective.
- Don't use a $l00 lock to protect a $50 bicycle.

Risk Analysis

- If an attack against a resource is unlikely, protecting against it has a lower priority than protecting against a likely one.

1. Risk is a function of the environment (Internet more risky than intranet)
2. Risks change with time (what if modems allowed on intranet)
3. Some risks are quite remote but still exist (modems)
4. Analysis paralysis (at some point you must act even though not completely sure)

Risk Analysis

- Risk = Assets * Threats * Vulnerabilities

- If no threats, then no security problems (risk).
- Similarly, if no vulnerabilities, then no risk.
- If no assets, then …

Laws and Customs

- Security system must not break any laws.
- Example: Restrictions on the export of cryptographic software
- Example: OK for system administrators to read files in the line of duty.
- Example: Cost of prosecuting attackers
- Example: How much search is appropriate at an airport

Human Issues

- Security mechanisms must be configured and used correctly to be effective.
- Two ways this can fail
  - Organizational Problems
  - People Problems

Organizational Problems

- Security provides no direct financial rewards to the user (busy business man phenomenon)
- Management must be willing to allocate resources for security
- Time must be taken to educate employees

People Problems

- Insider attacks (disgruntled employees)
- Untrained employees (open unknown email attachments)
- Overworked system administrators (don't notice attacks or misconfigure system)
- Social engineering (sweet talk)

Summary

- Threats determine policy;
- a policy is specified;
- a design is created from the specification;
- mechanisms are implemented from the design;
- the system is installed and must be operated and maintained on a daily basis.
- Feedback occurs in the opposite direction.

Main Point

3. A security system, no matter how well designed, will not protect a system if it is incorrectly configured or misused. The people factor is non-technical. The unit of world peace is the individual. TM increases the awareness of a practitioner bringing it to cosmic status.

Terminology

1. Computer security services (C, I, A)
2. Attack, attacker, threat, vulnerability, asset, risk
3. Security policies
4. Security mechanisms
5. Assurance

CONNECTING THE PARTS OF KNOWLEDGE WITH THE WHOLENESS OF KNOWLEDGE

1. Credit card information should only be sent if the padlock icon is visible in the browser.

2. Trust plays a critical role in computer security.

3. Transcendental Consciousness is an invincible, non-changing field.

4. Wholeness moving within itself : in Unity Consciousness one lives in a state of non-duality, free from all the fear that arises from duality; one feels intimately associated with all other things in creation as a result of perceiving a common basis in pure consciousness.