CS 5950/6030: Network Security - Fall 2005

L. Lilien
Department of Computer Science
Western Michigan University

Midterm Topics
*** Midterm is scheduled for F, 10/28/05 ***

Notes:

- Answers to all questions are available on the class slides.
- The word "**Given**" (in boldface) means that this information will be provided to you as a part of the question. For example, "**Given** the list of requirements for crypto protocols, discuss them." Means that I will provide you with the list of requirements (no need to memorize them).

List of questions

Section 1. Introduction

…

1.3. Introduction to Security

1.3.1. Examples-Security in Practice— Give a few examples of cyberterrorism / threats to personal privacy.

1.3.2. What is "Security"?— Give at least five types of security threats.

1.3.3. Pillars of Security: C-I-A— Define confidentiality / integrity / availability. Explain why C-I-A need be balanced, illustrate with an example. Which of C-I-A is binary (all-or-none) in nature?

1.3.4. Vulnerabilities, Threats, and Controls— Define vulnerability / threat / controls. Define attack. Illustrate each def with an example. List 4 kinds of threats to assets. List 4 levels of vulnerabilities and threats. Give examples of vulnerabilities and threats: for hardware / for software / for data / for other assets. How to prevent identity theft?

1.3.5. Attackers— Attackers' MOM. Types of attackers.

1.3.6. How to React to an Exploit?— Discuss the problem "To Report or Not To Report" showing arguments on both sides. What is computer forensics?

1.3.7. Methods of Defense— List five basic approaches to defense of computing systems. 5 types of controls. Which controls of the 5 is considered primary / secondary? How encryption protects C-I-A? Give a few examples of h/w controls. What are benefits of policy/procedure controls? Give a few examples of physical controls.

Discuss how each of the following affects effectiveness of controls: awareness of problem, likelihood of use, overlapping controls, periodic reviews.

1.3.8. Principles of Computer Security— Define in your own words Principle of Easiest Penetration / Principle of Adequate Protection / Principle of Effectiveness / Principle of Weakest Link.

Section 2. Introduction to Cryptology

2A. Terminology and Background

2A.1. Threats to Messages— Explain the following threats to messages: Interception / Interruption / Modification / Fabrication.

2A.2. Basic Terminology and Notation— Define cryptology, cryptography, cryptanalysis. Describe the basic cryptographic scheme both without and with keys, use important terms and show formal notation. Compare keyless, symmetric, and asymmetric cryptography.

---PART 2 of 2A.2 (Lecture 5)--- Define goals of cryptanalysis. Define information and tools used by cryptanalysts.

2A.3. Requirements for Crypto Protocols— **Given** the list of requirements for crypto protocols, discuss them.

2A.4. Representing Characters— Give example showing representation of letters and operations on letters.

2B. Basic Types of Ciphers

2B.1. Substitution ciphers

a. Caesar cipher— Give an example of using Caesar cipher, include formal notation. Describe attacking a substitution cipher using an example.

b. Other substitution ciphers— What are benefits of polyalphabetic substitution ciphers? Give an example of using 2-key substitution cipher. **Given** Vigenere Tableaux, show an example of using it for encryption/decryption.

c. One-time pads (OPTs)— What is the main reason for using OPTs? Give an example of using OPT; make sure to show its use by both sender and receiver. Why perfect synchronization is required in OPTs? Discuss key distribution for OPTs, is it easy or difficult?

2B.2. Transposition ciphers— Give an example of columnar transposition cipher, indicate key value. Describe attacking transposition ciphers, using an example.

2B.3. Product (combination) ciphers— Define product ciphers. Is it guaranteed to be stronger than any of its components?

2C. Making Good Ciphers

2C.1. Criteria for "Good Ciphers"

Given Shannon's criteria for "good ciphers," discuss these criteria. Define confusion and diffusion. Discuss confusion and diffusion as criteria for good ciphers.

2C.2. Stream and block ciphers

a. Stream ciphers— Define stream cipher. Describe polyalphabetic cipher using Vigenere Tableaux as a stream cipher.

b. Problems with stream ciphers— Illustrate with example problem with stream ciphers.

c. Block ciphers— Define block cipher. What are benefits of block ciphers?

d. Pros / cons for stream and block ciphers— Compare pros and cons for stream and block ciphers based on delay, error propagation, diffusion, and susceptibility to malicious insertion.

2C.3. Cryptanalysis

Define each one of 5 possible cryptanalyst attack approaches: ciphertext-only attack, known plaintext attack, probable plaintext attack, chosen plaintext attack, chosen ciphertext attack.

2C.4. Symmetric and asymmetric cryptosystems

Compare symmetric and asymmetric cryptography (in terms of: nr of keys, key secrecy, E/D secrecy, distribution of key or E/D, analogy to a safe or safe with a deposit slot).

2D. The DES Algorithm

2D.1. Background and History of DES— Motivation for DES.

2D.2. Overview of DES— **Given** figures showing basic encryption structure and key encryption structure, describe DES. Key length vs. effective key length. Problems with DES.

2D.3. Double and Triple DES— Describe double DES (show formal notation). Problem with double DES. Describe triple DES (show formal notation). What is the benefit of using triple DES.

2D.4. Security of DES— Is DES insecure?

[**SKIP**—2E. The Clipper Story]

2F. The AES Algorithm

2F.1. The AES Contest— **Given** list of NIST criteria for new encryption algorithm, discuss each criterion.

2F.2. Overview of Rijndael— Describe AES: specify key lengths and basic operations.

2F.3. Strength of AES— What are strengths of AES?

2F.4. Comparison of DES and AES— Compare DES and AES w.r.t. block size, key length, design rationale, selection process, and source.

2G. Public Key Encryption (PKE)

2G.1. Motivation for PKE— Problems with symmetric key encryption. Inventors of the first asymmetric cryptosystem. Describe the principles of asymmetric cryptosystems. Nr of keys for n users? (compare with symmetric encryption)

2G.2. Characteristics of PKE— List PKE requirements. Compare speed of symmetric and asymmetric encryption—what are the consequences?

2G.3. RSA (Rivest-Shamir-Adelman) Encryption— Give formulas for RSA encryption and decryption.

2H. The Uses of Encryption

2H.1. Cryptographic Hash Functions

strong hash fcn; collisions and attacks on message integrity; file checksum; keyed crypto checksums (using DES, AES) vs. keyless crypto checksums (remember names MD5/MD4, SHA/SHS)

2H.2. Key Exchange

motivation; deriving symmetric key via PKE – remember the "good" solution and problems with bad solutions;

2H.3. Digital Signatures (DSs)

motivation; roles for signatures; required and desirable properties for DSs; scenario of sending and receiving a digitally signed msg (msg encrypted or not);

2H.4. Certificates

a. Introduction

basic means of building trust; basic means of verifying trust; what is the role of trusted third party (TTP) in interactions?

b. Trust Through a Common Respected Individual

give examples of Common Respected Individual in an organization; problems with using a Common Respected Individual; solutions to the problems with using a Common Respected Individual

c. Certificates for Identity Authentication

certificate structure; scenario of creating certificates within a company; problems with certificates in multilevel certification hierarchy, and solution to the problem; requirements for a certification scheme

d. Trust Without a Single Hierarchy

how trust established if there is no "natural" trust hierarchy (or trust authority) to rely on?

Section 3. Program Security

What is the role of "program security"?

3.1. Secure Programs— Defining and Testing

a. Introduction

what is pgm security for user, for programmer, and for manager? give a single scenario in which error, fault, and failure can be illustrated

b. Judging S/w Security by Fixing Faults

explain principle of <u>penetrate and patch</u> approach to judging pgm security; what are shortcomings of the penetrate and patch approach?

c. Judging S/w Security by Testing Pgm Behavior

explain principle of <u>Testing Pgm Behavior</u> approach to judging pgm security;
why detecting a failure is not suffuicient?
should we be concerned with inadvertent faults and failures? why?
what are the shortcomings of the <u>Testing Pgm Behavior</u> approach?

d. Judging S/w Security by Pgm Security Analysis

explain principle of judging s/w security by <u>Pgm Security Analysis</u>

e. Types of Pgm Flaws

remember that flaws can be intentional/inadvertent
remember that intentional flaws can be malicious or nonmalicious

3.2. Nonmalicious Program Errors

a. Buffer overflows

what is buffer overflow flaw? why is it dangerous?
write a detailed scenario of a buffer overflow affecting a call stack area (emphasize/explain how attacker achieves her goals)
what is web server (overflow) attack?

b. Incomplete mediation

what is incomplete mediation?
write a detailed scenario showing incomplete mediation (emphasize/explain how attacker achieves her goals)
what are possible solutions for a server?

c. Time-to-check to time-to-use errors (TOCTTOU)

what is TOCTTOU?
give example of TOCTTOU in computing (emphasize/explain how attacker achieves her goals)

d. Combinations of nonmalicious program flaws

why nonmalicious flaws are security threats?

3.3. Malicious Code

define what is “malicious code”

3.3.1. General purpose malicious code

a. Introduction

are all viruses harmful? can viruses be transferred from trusted sources?

b. Kinds of Malicious Code (malware):

Define each of the following kinds of malware: Trojan horse, Virus, Worm, Logic bomb (incl. time bombs), Rabbit, Trapdoor / backdoor. Be able to indicate which are independent programs and which are sections (parts) of programs. Which are self-replicating?
Give an example of malware that is a combination of two(or more) of the basic kinds.

c. How Viruses Work

explain how viruses work. remember 2 phases of virus activities (infect and spread)
explain how document viruses work.
remember 4 kinds of viruses w.r.t. the way of attaching to infected pgms
what are characteristics of a perfect virus?

where viruses can hide and which of these hiding places are best for viruses? why?

d. Virus Signatures

what is Virus Signature? What patterns define Virus Signatures?
what are polymorphic viruses? Why are they more difficult to detect?
examples of polymorphic virus mutations

e. Preventing Virus Infections

remember recommended ways of Preventing Virus Infections

f. Seven Truths About Viruses

be able to confirm or deny each of the seven truths or their negations (e.g., you should be able to qualify as "false" the following statement: "Viruses can infect only some OSs")

g. Case Studies

what was the impact of Internet worm (1988)?
what kind of attack and on what target was performed by Code Red (2001)?
what are web bugs? how they work? How can they be neutralized (located/blocked/deleted)?

h. virus removal and system recovery after infection

describe virus removal and system recovery after infection
when infected file can't be recovered?

3.3.2. Targeted malicious code

a. Trapdoors

what is a trapdoor? who and why can insert it into code? give examples of benevolent trapdoors

b. Salami attack

what is salami attack: define and give example. what types of computations are easy targets for salami attacks?

c. Covert channels (CCs)

i. What is a covert channel – define and give examples. How programmers create CCs?
give examples how leaked data can be hidden to create CCs?
ii. List types of Covert Channels
iii. Describe in a sufficient detail an example of a storage Covert Channel
describe how CC can be created by using: file locks / disk storage quota / existence of a file
why storage CCs require synchronization?
iv. How timing CCs convey information? Describe in a sufficient detail an example of a Timing Covert Channel.
v. Name two techniques for identifying potential Covert Channels. What is the basic idea behind the Shared Resource Matrix method? Use an example to describe the Shared Resource Matrix method. What is the basic idea behind the Information Flow Method? Use an example (e.g, a 3-statement program with different kinds of statements) to describe the Information Flow Method.
vi. Can covert channels be prevented in open systems? Why?

3.4. Controls for Security

a. Introduction

List three types of security controls.

b. Developmental controls for security

How modularity / encapsulation / info hiding improves security?
Be able to discuss each of the following techniques for building solid software:

1) Peer reviews

   How reviews / walkthroughs / inspections differ from each other? How can they improve security?

2) Hazard analysis

   Remember two major components: hazard lists and what-if scenarios. How can it improve security?

3) Testing

   Define Module/component/unit testing, integration testing, function testing, performance testing, acceptance testing, installation testing, regression testing.

   Types of testing w.r.t. code availability.

   How can testing improve security?

4) Good design

   i. Modularity / encapsulation / info hiding (covered above)

   ii. Fault tolerance – What are principles of fault-tolerant approach? Explain majority voting / recovery block. How can it improve security?

   iii. Consistent failure handling policies— Name three ways of handling failures

   iv. Design rationale and history— How can it improve security?

   v. Design patterns— How can it improve security?

5) Risk prediction & management— Why is it important for security?

6) Static analysis— Define it. What program elements are checked by static analysis? How can it improve security?

7) Configuration management (CM)— Define CM. Define corrective changes, adaptive changes, perfective changes, preventive changes.

   What is a baseline and how is it used for CM? What are deltas, and how are they used for CM? What is conditional compilation and how is it used for CM?

   What is the purpose of configuration auditing?

   What is the purpose of status accounting?

   What is Configuration Control Board (CCB)?

   What are security benefits of configuration management?

8) Additional developmental controls

   How learning from pgm development an maintenance mistakes can improve pgm security?

   How (formal) proofs of program correctness can improve security? What are their practical limitations?

c. OS controls for security

   What is “trusted software”?

   What are ways of of increasing system security if untrusted pgms are present? Explain notions of mutual suspicion / confinement / access log, and tell how mutual suspicion / confinement / access log improve s security.

d. Administrative controls for security

   Explain how standards / security audits / separation of duties in pgm development contribute to better security.

Section 4. Protection in General-Purpose OSs

4.1. Protected Objects, Methods, and Levels of Protection

a. History of protection in OSs—How and why multiprogramming affected protection of objects in OSs?

b. Protected objects in OSs— List examples of objects that need protection.

c. Security methods in OSs— What is “separation” in OSs? List 4 types of separation. Which of three types of separation —logical, temporal, physical— is strongest/weakest? Which is least/most complex?

d. Levels of protection in OSs— List at least four (of six) levels of protection that can be provided by OS. Define: No protection / Isolation / Full sharing or no sharing / Sharing via access limitation / Sharing by *capabilities* / Limited object use. Order them by complexity of implementation and fineness of protection.

[e. Three dimensions of protection in OSs]

f. Granularity of *data* protection— List at least 4 levels of data granularity in the order from lowest to highest granularity. What are disadvantages of higher data granularity in access control?

4.2. Memory and Address Protection

a. Fence— Explain this approach and its security benefits. What is fence register and how is it used for memory protection?

b. Relocation— Explain this approach and its security benefits. What is relocation register and how is it used for memory protection?

c. Base/Bounds Registers— Explain this approach and its security benefits. What are base and bounds registersand how are they used for memory protection? How two pairs of Base/Bounds Registers can be used for security?

d. Tagged Architecture— Explain this approach and its security benefits. What is the main benefit of this approach? What are problems with the tagged architecture?

e. Segmentation— Explain how segmentation works, draw a figure illustrating this approach. What are security benefits of segmentation? What are its problems?

f. Paging— Explain how paging works, draw a figure illustrating this approach. What are problems with paging?

g. Combined Paging with Segmentation— Explain the principles of this approach and its security benefits.

4.3. Control of Access to General Objects

a. Introduction to access control for general objects— Give examples of subjects and objects for access control. Explain why access control should: check *every* access / enforce least privilege / verify acceptability of use. Explain how object homogeneity / number of points of access / existence of central access authority / kind of access affects complexity of access control.

List five mechanisms for access control for general objects.

b. Directory-like mechanism for access control— Explain this approach (include a figure). Explain problem for large nr of shared files. Explain problems for deletion of shared objects or revocation of access rights.

c. Access control lists— Explain this approach (show an example ACL). Why having default access rights (ARs) is advantageous?

d. Access control matrices— Explain this approach (show an example ACM).

e. Capabilities for access control— Explain this approach. What is a capability? How can capability be protected? Why capability revocation can be complicated?

f. Procedure-oriented access control— Explain this approach. How is it related to encapsulation?

4.4. File Protection Mechanisms

a. Basic forms of protection— Explain the group protection approach. What are its advantages and disadvantages?

b. Single file permissions— Explain the file password approach. What are its advantages and disadvantages? Explain the temporary acquired permission approach. What are its advantages and disadvantages?

c. Per-object and per-user protection— Explain this approach. What are its advantages and disadvantages?

4.5. User Authentication – **QUESTIONS TO BE ADDED SOON**

a. Introduction

b. Use of passwords

c. Attacks on passwords

i. Try *all* possible pwds

ii. Try many *probable* pwds

iii. Try *likely* passwords pwds

iv. Search system list of pwds

v. Find pwds by exploiting indiscreet users

d. Passwords selection criteria

e. One-time passwords (challenge-response systems)

f. The authentication process

g. Authentication other than passwords

h. Conclusions