CSCI 362 Computer and Network Security
Midterm Exam 2 Study Guide

- Review your assigned readings from the textbook.
- Review the material for the previous exam. All exams in this class are cumulative.
- Go through the activities. Make sure that you have a solid understanding of the topics they address.
- Go through the self-check list below and make sure that you are able to answer these questions.
- Note that this document doesn’t mean to give an exhaustive coverage of what might appear in the exam!

1. Goals in Security
   - Confidentiality
   - Authentication
   - Integrity
   - Availability
   - Non-repudiation
   - Freshness

   (a) Understand the security goals in the list above so that you are able to identify when a security technology or product is capable of meeting each goal.
   (b) Given a scenario or product, identify which of these goals is desirable.
   (c) Construct a scenario where one or more of these goals is desirable.

2. Network Security

   (a) What does an attacker gain by port scanning?
   (b) What does an attacker gain by learning what operating system runs on a computer?
   (c) What does an attacker gain by learning what version of a certain service is mapped to a network port number?
   (d) What is the function of a firewall?
   (e) Describe the basic principles upon which an intrusion detection system (IDS) is built. Discuss the compromises in adjusting an IDS to deal with false positive and false negatives.
3. Secure programming

(a) Given a Unix system or library call, identify whether it creates a security exposure. Think back to what makes strcpy and realpath insecure what kind of exposure is created by each of these calls? Try to extrapolate from your conclusions to derive the concepts that will allow you to identify an insecure (or risky) system or library call when you first see it.

(b) What causes buffer overflow in a C program? Given a C program, can you tell if it is vulnerable to a buffer overflow?

(c) Are there risks in using the strncpy C library call? Construct a scenario when this call does not produce a real C string into the buffer for the copy.

(d) What are the basic principles behind a stack smashing exploit? What may be the consequences of a stack smashing attack?

(e) Open the manual pages for the following system calls and explain what makes them insecure:
   - char *realpath(const char *path, char *resolved_path);
   - struct passwd *getpwuid(uid_t uid);
   - int execlp(const char *file, const char *arg, );
   - int stat(const char *file_name, struct stat *buf);

Given a Unix system or library call, identify whether it creates a security exposure. Think back to what makes strcpy, stat, and realpath insecure – what kind of exposure is created by each of these calls? Try to extrapolate from your conclusions to derive the concepts that will allow you to identify an insecure (or risky) system or library call when you first see it.

(g) What is the point of not allowing a program that crashes produce a core dump?

(h) If your program uses dynamically allocated memory to store a piece of secret information, is there any danger in freeing this memory when the program is done with it?

(i) If your program uses a disk file to store secret information, is there any danger of information spill when one simply deletes the secret file with rm?

(j) Create an example scenario (a C program fragment) to explain what a race-condition is. Why may security problems arise from race-conditions?

(k) We discussed a number of principles of security architecture and of software engineering. Can you justify the importance of these principles? Can you provide a scenario that explains what each principle means?
4. Malware

(a) What characterizes a “virus”? How does a virus infect a computer? How does one avoid infection?

(b) What is the difference between a “virus” and a “worm”? How do viruses and worms infect a computer? How can one prevent infection by viruses?

(c) Describe the components of a system that can ascertain that a certain piece of code is a virus.

(d) What characterizes a “Trojan”? How does one avoid exploits from Trojans?

(e) What is a “backdoor” and how is it created? What can one do to avoid exposures through backdoors?

(f) What is a “rootkit”? What assumptions have to hold for a rootkit attack to work?

(g) What is a “logic bomb?”

(h) What are “spyware”, “scareware”, and “adware”? What kinds of dangers do each of these represent for the various security properties of a system?

5. Cryptology

(a) What kinds of security properties of a system can be implemented with the help of cryptography?

(b) Define what a cipher is. If you are given an algorithm, can you tell whether what it implements is indeed a cipher?

(c) Understand the concepts below; be able to identify them in a given situation or algorithm.
   
   i) Substitution
   
   ii) Transposition
   
   iii) Confusion
   
   iv) Diffusion

(d) What characterizes a symmetric key cipher? List common, general characteristics of symmetric key ciphers.

(e) What role do random numbers play in cryptography?
(e) What are the characteristics of an ideal cipher? What makes the one-time pad an unbreakable cipher? What are the difficulties in implementing a one-time pad? Why isn’t a one-time pad widely used?

(f) Contrast a stream cipher to a block cipher. What are the keys to each cipher like? How does each kind of algorithm process plaintext?