For the test, you may bring one 8.5x11 sheet of paper wait, rates midterm it. The exam will consist of took multiple choice, short answer, and small problems.

1. Week 1
   (a) Lecture 1 Introduction to software security
      i. Comprehend the magnitude of the security problem.
      ii. Compare and contrast bugs and flaws
      iii. Compare and contrast secure software development, network security, and data security
      iv. Explain the trinity of trouble
      v. Justify why more code leads to more security bugs
   (b) Lecture 2
      i. Define Confidentiality, Integrity, and Availability
      ii. Define authentication, authorization, and accountability
      iii. Explain the concept of a risk management framework

2. Week 2
   (a) Lecture 1 Software Security Touchpoints
      i. Explain the concept of a Touchpoint
      ii. List the Software Security Touchpoints
      iii. Identify the most effective security practices to have within software development
      iv. Define the acronym OWASP
      v. Explain the basic premise for CLASP
      vi. List the 7 best practices for security according to CLASP
      vii. Explain the difference between the reactive and proactive approaches to software security.
      viii. Recognize an example of a bug and a flaw within a software system.
      ix. Explain the economic impact of various security activities.
   (b) Lecture 2 Requirements Part 1
      i. Differentiate between security goals and security functions
      ii. Explain the concept of a security requirement
      iii. List the three characteristics of secure software
      iv. Explain the concept of a security profile
      v. Explain confidentiality requirements
      vi. Explain integrity requirements
      vii. Explain authentication requirements
      viii. Compare and contrast simple authentication, two factor authentication, and multifactor authentication.

3. Week 3
   (a) Lecture 1 and 2 Abuse Cases
      i. Differentiate between security goals and security functions.
      ii. Explain the concept of a security requirement
      iii. Compare and contrast
      iv. Differentiate between security goals and security functions
v. Explain the concept of a security requirement
vi. Compare and contrast requirements and anti-requirements
vii. Explain the concept of an abuse case and explain how one would create an abuse case
viii. Explain what you need to do to think securely.
ix. Explain the concept of a use case diagram
x. Explain the concept of abuse cases as shown on a use case diagram
xi. Construct examples of abuse cases

4. Week 4

(a) Lecture 1 No Lecture Due to Christmas Break
(b) Lecture 2 Design Principles
   i. Explain why data and code commingling can cause problems
   ii. List and explain the Secure Design Principles
      A. Principle of Least Privilege
      B. Separation of Duties
      C. Defense in Depth
      D. Fail Secure
      E. Economy of Mechanisms
      F. Complete Mediation
      G. Open design
      H. Least Common Mechanisms
      I. Psychological Acceptability
      J. Leveraging Existing Components
   iii. Explain how Unix sudo enhances computer security for unix systems.

5. Week 5

(a) Lecture 1
   i. Explain the concept of a trust relationship
   ii. Define the concept of a trust boundary
   iii. Given an architecture, construct a diagram showing trust boundaries and trust relationships.
   iv. Explain the concept of threat modeling
   v. Define associated terms related to threat modeling
(b) Lecture 2 A Taxonomy of Coding Errors
   i. Recognize through code review simple programming mistakes.
   ii. Define the concept of a taxonomy
   iii. Explain the relationship between kingdoms and phyla
   iv. Explain the kingdoms for security vulnerabilities
      A. Input Validation and Representation
      B. API Abuse
      C. Security Features
      D. Time and State
      E. Errors
      F. Code Quality
      G. Encapsulation
      H. Environment
   v. Critique source code for examples of coding errors
   vi. Explain how simple coding errors might be exploited by an adversary

6. Week 6

(a) Lecture 1 Catchup
(b) Lecture 2 Midterm Exam
7. Week 7
   (a) Lecture 1 - Class Canceled
   (b) Lecture 2 Code Review with a Tool
       i. Explain why code is the single artifact that all software projects must have.
       ii. List disadvantages of code review
       iii. Explain the concept of static analysis
       iv. Define soundness and completeness
       v. Explain the risks of false positives and false negatives
       vi. Explain the goals of a static analysis tool developed for security analysis.

8. Week 8
   (a) Lecture 1 - More Mistakes
       i. Explain the concept of an SQL injection flaw.
       ii. Analyze an SQL query and explain how SQL injection could attack the query.
       iii. Explain the concept of OS command injection
       iv. Critique C++ implementations and uses of string libraries
       v. Explain the concept of scrubbing memory.
       vi. Interpret and explain a UML sequence diagram showing cross site scripting.
       vii. Draw a diagram showing how a man in the middle attack would occur.
       viii. Explain how code comments may reveal security vulnerabilities.
       ix. List assorted security misconfigurations which may impact systemic security.

9. Week 9
   (a) Lecture 1 Fuzz Testing
       i. Define Reliability, Resiliency, and Recoverability as it pertains to Security
       ii. Explain load testing and stress testing
       iii. Define means, motive, and opportunity as it pertains to security
       iv. Explain the steps to fuzz testing
       v. Explain the difference between dumb fuzzing and smart fuzzing
       vi. List the advantages and disadvantages of fuzz testing.
       vii. Explain the three possible outcomes for a fuzz test iteration.
   (b) Lecture 2 Penetration Testing
       i. Compare and contrast penetration testing techniques versus standard testing techniques used in software development. Define Reliability, Resiliency, and Recoverability as it pertains to Security
       ii. Define white box and black box testing, and explain the ramifications of security testing using each technique.
       iii. Define vulnerability, exploiting a vulnerability, penetration study, and penetration testing.
       iv. Explain the goals of penetration testing.
       v. Explain the steps in hacking a system.

10. Week 9
   (a) Lecture 1 Fuzz Testing
       i. Explain the security related problems of software installation
       ii. Define hardening
       iii. Understand the importance of continuous monitoring
       iv. Explain the concept of a Bastion Host
       v. Define the terms event, alert, and incident.
       vi. Draw the incident response lifecycle.
       vii. Explain the risks at the end of software life.
   (b) Lecture 2 Penetration Testing
       i. Define sunsetting.
       ii. Explain the proper handling for information and media sanitization
       iii. Explain the concept of the Common Vulnerability Dictionary
       iv. Explain the current trends in software security. security related problems of software installation