CSC 424 – Software Security

1. CSC 424 – Software Security

2. credit units 4  contact hours 6

3. Course Coordinator: Zachary Peterson

4. Textbook (or other required material): Building Secure Software by Viega and McGraw. Select readings from the research literature.

5. a. Course Description: Principles behind secure software design including threat models, trust management, common vulnerabilities and mitigation techniques, robust software development, isolation of untrusted code, auditability, and testing. 3 lectures, 1 laboratory.

   b. Prerequisite: CSC321; CSC 307 or CSC 309

   c. Required/Elective/Selective Elective for CPE, CSC, EE, SE

      |                | CSC | CPE | SE |
      |----------------|-----|-----|----|
      | Required       |     |     |    |
      | Elective       | X   | X   | X  |
      | Selective Elective |   |     |    |

6. a. Course Learning Objectives

   The student will be able to:
   - Evaluate the security risks inherent in the development of a software system
   - Identify assumptions of trust and vulnerabilities within a software system
   - Identify and apply standards for secure software development.
   - Select and apply appropriate techniques for mitigating vulnerabilities
   - Generate and clearly communicate appropriate security requirements for a software system
   - Evaluate the security status of a given software system

   b. Level at which Student Outcomes are addressed
   (“B” = Basic level, “I” = Intermediate level, “A” = Advanced level)

      | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
      |---|---|---|---|---|---|---|
      | A | A | I | I | A | A | N/A |
      | A | I | I | I | A | I | A  |

7. Major Topics Covered: (number of lecture hours per)
The process execution environment (process model, libraries, etc.), Programs as data (3)
Principles of secure software design (3)
Canonical software failures: control flow hijacking, race conditions, time of check time of use violations, pseudorandomness, side channels and other information leakage (6)
Mitigation techniques: input sanitization, authentication, propose use of cryptography (3)
Code correctness: Code reviews and audits, static and dynamic code analysis tools, formal methods and proofs of correctness (4)
Secure software engineering practices and managing risk (3)
Evaluation, Certification and Standards: Common Criteria, SOX, NIST (2)