Security and Secure Coding 2023-2024 Catal

[ARCHIVED CATALOG]

SDEV 245 - Security and Secure Coding

PREREQUISITES/COREQUISITE: SDEV 200 - Software Development using Jav OR SDEV 210 - Software Development using Visual Basic in the .NET Framework OR SDEV 220 - Software Development Using Python OR SDEV 230 - Software Development using C++ OR SDEV 240 - Software Development Using C# OR CSCI 102 - Computer Science II OR CSCI 201 - Computer Science II AND MATH 136 - College Algebra.

PROGRAM: Software Development
CREDIT HOURS MIN: 3
LECTURE HOURS MIN: 3
DATE OF LAST REVISION: Fall, 2017

The course introduces the secure software development process including designing secure applications, writing secure code designed to withstand various types of attacks, and security testing and auditing. It focuses on the security issues a developer faces, common security vulnerabilities and flaws, and security threats. The course explains security principles, strategies, coding techniques, and tools that can help make software fault tolerant and resistant to attacks. Students will write and analyze code that demonstrates specific security development techniques. Students will also learn about cryptography as an indispensable resource for implementing security in real-world applications. Students will learn foundations of cryptography using simple mathematical probability. Information theory, computational complexity, number theory, and algebraic approaches will be covered.

MAJOR COURSE LEARNING OBJECTIVES: Upon successful completion of this course the student will be expected to:

1. Describe and discuss key concepts in security, including confidentiality, integrity and availability, authentication, and access control.
2. Describe and discuss key concepts in cybersecurity, including cryptology, cryptography, cryptanalysis, cipher, cryptographic algorithm, private and public key encryption, public key infrastructure, and trust/trustworthiness.
3. Discuss the basic concepts of probability, random variables and probability distributions as they apply to information theory and cryptography.
4. Demonstrate the techniques to transform plaintext into ciphertext, the use of hash functions for authentication and data integrity, and the use of private and public key encryption.
5. Investigate security vulnerabilities in various data structures, such as out-of-bounds arrays and buffer overflows.
6. Discuss various types of cyberattacks on software and software systems along with possible countermeasures and security controls that minimize risk and exposure
7. Discuss current industry standards, tools, and security practices in software development, including use of multiple layers of defenses, wireless security, and risks in 3rd party applications and libraries.
8. Explain the tradeoffs of developing a program in a type safe language
9. Implement secure coding and testing techniques including input validation, data sanitization, and exception handling.
10. Examine the need to update software to fix security vulnerabilities.
11. Discuss the role of software security in a company-wide security policy.

COURSE CONTENT: Topical areas of study include -

- Encryption techniques
- Ciphers and ciphertext
- Authentication and Access Control

- Public and Private Key encryption
- Cybersecurity attacks
- Secure Software Development Lifecycle
- Ethics in Software Development
- Information Security
- Input/Data Validation and sanitation
- Open source and closed source software
- Secure and quality coding principles
- Random/Pseudo Random Numbers
- Design diagrams and artifacts
- Software testing techniques
- Exception/Error handling
- Software Security Audits
- Software maintenance
- Countermeasures
- Kerberos, SSH, Radius and TACACS+

Course Addendum - Syllabus (Click to expand)
