

COURSE DESCRIPTION

Department and Course Number: CSC 250

Course Coordinator: Isaac Ghansah

Course Title: Computer Security

Total Credits: 3

Catalog Description:

Principles and technologies behind computer security. Introduction to encryption and decryption; security mechanisms in computer programs, operating systems, databases, and networks; administration of computer security, and legal and ethical issues. **Prerequisite:** Fully classified graduate status in Computer Science, Software Engineering, or Computer Engineering

Textbook:

- Charles P. Pfleeger and Shari Lawrence Pfleeger, Security in Computing, Prentice Hall, 3rd edition, 2003.

References:

- Matt Bishop, Introduction to Computer Security, Addison-Wesley, 2004.
- Recent articles from technical literature and the Internet Engineering Task Force (IETF) Requests for Comments (RFC).

Course Goals:

- To develop knowledge of contemporary vulnerabilities, threats and attacks in computing systems.
- To provide understanding of how cryptography is used in secure communication.
- To develop knowledge of security mechanisms in computer programs, operating systems, databases, and networks, administration of computer security, and legal/ethical issues in computer security.
- To develop proficiency in the use of various software tools for computer security.

Prerequisites by Topic:

Thorough understanding of:

1. Storage representation and management of data and programs in a computer system (Runtime stack, Heap, Memory management hierarchy, etc.).
2. Internet (TCP/IP) standards and protocol stack.

Basic understanding of:

UNIX or Windows operating systems.

Major Topics:

- Principles of computer security (3 hours).
Basic security concepts.
Vulnerabilities, threats, attacks, security systems and control.
Confidentiality, Integrity, and Availability.
- Elementary cryptography (6 hours).
Substitution ciphers.
Transposition ciphers.

- Symmetric encryption.
- Public key encryption.
- Public Key Infrastructure (PKI).
- Authentication .
- Digital signature.
- Digital certificate.
- Program Security (6 hours).
 - Writing secure programs and analyzing malicious codes.
 - Buffer overflow.
 - Malicious codes (e.g., virus, Trojan horse, worms, and covert channel).
 - Finding security flaws.
- Protection in General-Purpose Operating Systems (4.5 hours).
 - Identifying vulnerabilities in operating systems.
 - Memory protection.
 - Access control.
- Designing Trusted Services and Systems (4.5 hours).
 - Security services in an operating systems
 - Models of security.
 - Designing secure operating systems.
 - Assurance in trusted operating systems.
- Database Security (4.5 hours).
 - Database security requirements.
 - Reliability and integrity.
 - Sensitive data.
 - Inference.
 - Multilevel data bases.
- Network Security (6 hours).
 - Understanding computer networks and their vulnerabilities.
 - Threats in networks.
 - Network security controls (IPsec, VPN).
 - Firewalls and intrusion detection systems.
 - Secure E-Mail (PGP).
 - Wireless network attacks and defenses.
- Administering Security (3 hours).
 - Personal computer security management.
 - UNIX security management.
 - Network security management.
 - Risk analysis.
 - Security planning.
 - Organizational security policies.
 - Disaster recovery.
- Legal, Privacy, and Ethical Issues in Computer Security (6 hours).
 - Protecting programs and data.

Information and the law.
Rights of employees and employers.
Computer crime.
Ethical issues in computer security.
Electronic privacy.

- Exams, reviews and evaluations (1.5 hours).

Laboratory Projects:

The following items describe suggested projects that students should complete outside of class periods.

1. Implementing basic encryption algorithms and encrypt message. RC-4, a widely used stream cipher algorithm, is easy to program and understand with less than 50 lines of code, and has a very good performance. Students are guided to program the algorithm and encrypt/decrypt short messages.
2. Learning the principles of Steganography by encoding and decoding secret messages using a common Steganography freeware tool such as S-Tools.
3. Evaluating the complexity of passwords by cracking the personal passwords on their own computers using common password cracking software, such as John The Ripper.
4. Configuring personal firewalls, such as ZoneAlarm, on personal computers and evaluating the protection capability by downloading non-destructive malicious software.
5. Learning the mechanisms of secure email and operational procedures for exchanging emails securely using popular software such as PGP mail.
6. Assessing the vulnerability of students' computers by downloading and running vulnerability assessment tools such as Nessus.
7. Observing insecure communications over insecure media such as Ethernet by sniffing packets using packet sniffing tools such as Ethereal.
8. File deleting and recovery. Students delete files using standard OS commands, and attempt to recover the deleted files. Students then exercise wiping the deleted files completely using widely available wiping tools.
9. Students create a sample database and construct several inference attacks, both direct and indirect, and modify the database or query to avoid the attack.
10. Downloading the Common Criteria certification documents for popular OS (Windows XP, Linux, etc.) and finding whether the security concepts covered in the textbook (e.g., reference monitors, security kernel, etc) are implemented, and how.
11. Analyzing source code of open source programs to find security flaws (e.g., buffer overflow, pointer errors, type checking, memory leak, etc.) with source code scanning tools, such as RAT Analytical aspect of security protocols.
12. Finding a big prime number is critical in RSA algorithm. Several mathematical methods have been developed to test primality for a given number. Generate a 10-decimal digit number randomly, and test primality using Fermat's little theorem with 10 different base numbers. Check if the number is a Carmichael number to avoid a false result.

Estimated Curriculum Category Content (Semester hours)

<i>Area</i>	<i>Core</i>	<i>Advanced</i>	<i>Area</i>	<i>Core</i>	<i>Advanced</i>
Algorithms			Data Structures		
Software Design			Prog. Languages		
Comp. Arch.					

Oral and Written Communications

Students will be required to research computer security topics given by the instructor and give oral presentations of approximately 15 minutes.

Social and Ethical Issues

It will be made clear that the students should not use the knowledge and skills with any malicious intent against any computer system. Students will be required to sign an agreement to observe a set of legal and ethical guidelines.

Theoretical Content

The course uses cryptographic algorithms applied to secure communication.

Problem Analysis

Each attack method will be analyzed in a rigorous manner.

Solution Design

Students will learn how to discover the vulnerabilities and how to develop techniques to protect computer systems.

Original by J. Jo

Updated 4-23-08 by sj