Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS3I01. Computer Security (Mandatory)
2. **Credits:** 3
3. **Hours of theory and Lab:** 1 HT; 4 HL; (15 weeks)
4. **Professor(s)**

   Meetings after coordination with the professor
5. **Bibliography**

[WL14] Stallings. W and Brown. L. *Computer Security: Principles and Practice*. Pearson Education, Limited, 2014. ISBN: 9780133773927.

6. **Information about the course**

   (a) **Brief description about the course** Nowadays, information is one of the most valuable assets in any organization. This course is oriented to be able to provide the student with the security elements oriented to protect the Information of the organization and mainly to be able to foresee the possible problems related to this heading. This subject involves the development of a preventive attitude on the part of the student in all areas related to software development.

   (b) **Prerrequisites:** CS2301. Networking and Communication. (6^{th} Sem)

   (c) **Type of Course:** Mandatory

   (d) **Modality:** Face to face

7. **Specific goals of the Course**

   - Discuss at an intermediate intermediate level the fundamentals of Computer Security.
   - Provide different aspects of the malicious code.
   - That the student knows the concepts of cryptography and security in computer networks.
   - Discuss and analyze together with the student the aspects of Internet Security.

8. **Contribution to Outcomes**

   a) An ability to apply knowledge of mathematics, science. (**Usage**)

   b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

   e) Understand correctly the professional, ethical, legal, security and social implications of the profession. (**Usage**)

   g) The broad education necessary to understand the impact of computing solutions in a global, economic, environmental, and societal context. (**Assessment**)

   h) A recognition of the need for, and an ability to engage in life-long learning. (**Usage**)

   i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

   a) An ability to apply knowledge of mathematics, science. (**Usage**)

   b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

e) Understand correctly the professional, ethical, legal, security and social implications of the profession. (**Usage**)

g) The broad education necessary to understand the impact of computing solutions in a global, economic, environmental, and societal context. (**Assessment**)

h) A recognition of the need for, and an ability to engage in life-long learning. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

9. Competences (IEEE)

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome a**

C8. Understanding of what current technologies can and cannot accomplish. ⇒ **Outcome g**

C9. Understanding of computing's limitations, including the difference between what computing is inherently incapable of doing vs. what may be accomplished via future science and technology.⇒ **Outcome g,a**

C21. Understanding the professional, legal, security, political, humanistic, environmental, cultural and ethical issues. ⇒ **Outcome e**

C22. Ability to demonstrate attitudes and priorities that honor, protect, and enhance the profession's ethical stature and standing..⇒ **Outcome e,h**

CS7. Apply the principles of effective information management, information organization, and information-retrieval skills to information of various kinds, including text, images, sound, and video. This must include managing any security issues.⇒ **Outcome i,h**

CS9. Identify any risks (and this includes any safety or security aspects) that may be involved in the operation of computing equipment within a given context. ⇒ **Outcome b**

CS11. Be aware of the existence of publicly available software and understanding the potential of opensource projects.⇒ **Outcome b,g**

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome a**

C8. Understanding of what current technologies can and cannot accomplish. ⇒ **Outcome g**

C9. Understanding of computing's limitations, including the difference between what computing is inherently incapable of doing vs. what may be accomplished via future science and technology.⇒ **Outcome g,a**

C21. Understanding the professional, legal, security, political, humanistic, environmental, cultural and ethical issues. ⇒ **Outcome e**

C22. Ability to demonstrate attitudes and priorities that honor, protect, and enhance the profession's ethical stature and standing..⇒ **Outcome e,h**

CS7. Apply the principles of effective information management, information organization, and information-retrieval skills to information of various kinds, including text, images, sound, and video. This must include managing any security issues.⇒ **Outcome i,h**

CS9. Identify any risks (and this includes any safety or security aspects) that may be involved in the operation of computing equipment within a given context. ⇒ **Outcome b**

CS11. Be aware of the existence of publicly available software and understanding the potential of opensource projects.⇒ **Outcome b,g**

10. List of topics

1. Foundational Concepts in Security
2. Principles of Secure Design
3. Defensive Programming

4. Threats and Attacks
5. Network Security
6. Cryptography
7. Web Security
8. Platform Security
9. Digital Forensics
10. Secure Software Engineering

11. Methodology and Evaluation
Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.
Evaluation System:

12. Content

<table>
<tr><td colspan="2">Unit 1: Foundational Concepts in Security (25)</td></tr>
<tr><td colspan="2">Competences Expected: C2,C8</td></tr>
<tr><td>Learning Outcomes</td><td>Topics</td></tr>
<tr><td>• Analyze the tradeoffs of balancing key security properties (Confidentiality, Integrity, Availability) [Familiarity]
• Describe the concepts of risk, threats, vulnerabilities and attack vectors (including the fact that there is no such thing as perfect security) [Familiarity]
• Explain the concepts of authentication, authorization, access control [Familiarity]
• Explain the concept of trust and trustworthiness [Familiarity]
• Recognize that there are important ethical issues to consider in computer security, including ethical issues associated with fixing or not fixing vulnerabilities and disclosing or not disclosing vulnerabilities [Familiarity]</td><td>• CIA (Confidentiality, Integrity, Availability)
• Concepts of risk, threats, vulnerabilities, and attack vectors
• Authentication and authorization, access control (mandatory vs. discretionary)
• Concept of trust and trustworthiness
• Ethics (responsible disclosure)</td></tr>
<tr><td colspan="2">Readings : [WL14]</td></tr>
</table>

Unit 2: Principles of Secure Design (25)

Competences Expected: C,9C21,C22

Learning Outcomes	Topics
• Describe the principle of least privilege and isolation as applied to system design [Familiarity] • Summarize the principle of fail-safe and deny-by-default [Familiarity] • Discuss the implications of relying on open design or the secrecy of design for security. [Familiarity] • Explain the goals of end-to-end data security [Familiarity] • Discuss the benefits of having multiple layers of defenses [Familiarity] • For each stage in the lifecycle of a product, describe what security considerations should be evaluated. [Familiarity] • Describe the cost and tradeoffs associated with designing security into a product [Familiarity] • Describe the concept of mediation and the principle of complete mediation [Familiarity] • Be aware of standard components for security operations, instead of re-inventing fundamentals operations [Familiarity] • Explain the concept of trusted computing including trusted computing base and attack surface and the principle of minimizing trusted computing base [Familiarity] • Discuss the importance of usability in security mechanism design [Familiarity] • Describe security issues that arise at boundaries between multiple components. [Familiarity] • Identify the different roles of prevention mechanisms and detection/deterrence mechanisms [Familiarity]	• Least privilege and isolation • Fail-safe defaults • Open design • End-to-end security • Defense in depth (e.g., defensive programming, layered defense) • Security by design • Tensions between security and other design goals • Complete mediation • Use of vetted security components • Economy of mechanism (reducing trusted computing base, minimize attack surface) • Usable security • Security composability • Prevention, detection, and deterrence

Readings : [WL14]

Unit 3: Defensive Programming (25)

Competences Expected: CS6,CS7,CS9

Learning Outcomes	Topics
• Explain why input validation and data sanitization is necessary in the face of adversarial control of the input channel. [Usage] • Explain why you might choose to develop a program in a type-safe language like Java, in contrast to an unsafe programming language like C/C++ [Usage] • Classify common input validation errors, and write correct input validation code [Usage] • Demonstrate using a high-level programming language how to prevent a race condition from occurring and how to handle an exception [Usage] • Demonstrate the identification and graceful handling of error conditions [Familiarity] • Explain the risks with misusing interfaces with third-party code and how to correctly use third-party code [Familiarity] • Discuss the need to update software to fix security vulnerabilities and the lifecycle management of the fix [Familiarity]	• Input validation and data sanitization • Choice of programming language and type-safe languages • Examples of input validation and data sanitization errors – Buffer overflows – Integer errors – SQL injection – XSS vulnerability • Race conditions • Correct handling of exceptions and unexpected behaviors • Correct usage of third-party components • Effectively deploying security updates • Information flow control • Correctly generating randomness for security purposes • Mechanisms for detecting and mitigating input and data sanitization errors • Fuzzing • Static analysis and dynamic analysis • Program verification • Operating system support (e.g., address space randomization, canaries) • Hardware support (e.g, DEP, TPM)

Readings : [WL14]

Unit 4: Threats and Attacks (25)	
Competences Expected: CS6,CS7,CS9	
Learning Outcomes	**Topics**
• Describe likely attacker types against a particular system [Familiarity] • Discuss the limitations of malware countermeasures (eg, signature-based detection, behavioral detection) [Familiarity] • Identify instances of social engineering attacks and Denial of Service attacks [Familiarity] • Discuss how Denial of Service attacks can be identified and mitigated [Familiarity] • Describe risks to privacy and anonymity in commonly used applications [Familiarity] • Discuss the concepts of covert channels and other data leakage procedures [Familiarity]	• Attacker goals, capabilities, and motivations (such as underground economy, digital espionage, cyberwarfare, insider threats, hacktivism, advanced persistent threats) • Examples of malware (e.g., viruses, worms, spyware, botnets, Trojan horses or rootkits) • Denial of Service (DoS) and Distributed Denial of Service (DDoS) • Social engineering (e.g., phishing) • Attacks on privacy and anonymity • Malware/unwanted communication such as covert channels and steganography
Readings : [WL14]	

Unit 5: Network Security (25)	
Competences Expected: CS6,CS7,CS9	
Learning Outcomes	**Topics**
• Describe the different categories of network threats and attacks [Familiarity] • Describe the architecture for public and private key cryptography and how PKI supports network security [Familiarity] • Describe virtues and limitations of security technologies at each layer of the network stack [Familiarity] • Identify the appropriate defense mechanism(s) and its limitations given a network threat [Usage]	• Network specific threats and attack types (e.g., denial of service, spoofing, sniffing and traffic redirection, man-in-the-middle, message integrity attacks, routing attacks, and traffic analysis) • Use of cryptography for data and network security • Architectures for secure networks (e.g., secure channels, secure routing protocols, secure DNS, VPNs, anonymous communication protocols, isolation) • Defense mechanisms and countermeasures (e.g., network monitoring, intrusion detection, firewalls, spoofing and DoS protection, honeypots, tracebacks) • Security for wireless, cellular networks • Other non-wired networks (e.g., ad hoc, sensor, and vehicular networks) • Censorship resistance • Operational network security management (e.g., configure network access control)
Readings : [WL14]	

Unit 6: Cryptography (25)

Competences Expected: CS6,CS7,CS9

Learning Outcomes	Topics
• Describe the purpose of Cryptography and list ways it is used in data communications [Familiarity] • Define the following terms: Cipher, Cryptanalysis, Cryptographic Algorithm, and Cryptology and describe the two basic methods (ciphers) for transforming plain text in cipher text [Familiarity] • Discuss the importance of prime numbers in cryptography and explain their use in cryptographic algorithms [Familiarity] • Illustrate how to measure entropy and how to generate cryptographic randomness [Usage] • Use public-key primitives and their applications [Usage] • Explain how key exchange protocols work and how they fail [Familiarity] • Discuss cryptographic protocols and their properties [Familiarity]	• Basic Cryptography Terminology covering notions pertaining to the different (communication) partners, secure/unsecure channel, attackers and their capabilities, encryption, decryption, keys and their characteristics, signatures • Cipher types (e.g., Caesar cipher, affine cipher) together with typical attack methods such as frequency analysis • Public Key Infrastructure support for digital signature and encryption and its challenges • Symmetric key cryptography – Perfect secrecy and the one time pad – Modes of operation for semantic security and authenticated encryption (e.g., encrypt-then-MAC, OCB, GCM) – Message integrity (e.g., CMAC, HMAC) • Public key cryptography: – Trapdoor permutation, e.g., RSA – Public key encryption, e.g., RSA encryption, EI Gamal encryption – Digital signatures – Public-key infrastructure (PKI) and certificates – Hardness assumptions, e.g., Diffie-Hellman, integer factoring • Authenticated key exchange protocols, e.g., TLS • Cryptographic primitives: – pseudo-random generators and stream ciphers – block ciphers (pseudo-random permutations), e.g., AES – pseudo-random functions – hash functions, e.g., SHA2, collision resistance – message authentication codes – key derivations functions

Readings : [WL14]

Unit 7: Web Security (25)

Competences Expected: C8,C9

Learning Outcomes	Topics
• Describe the browser security model including same-origin policy and threat models in web security [Familiarity] • Discuss the concept of web sessions, secure communication channels such as TLS and importance of secure certificates, authentication including single sign-on such as OAuth and SAML [Familiarity] • Investigate common types of vulnerabilities and attacks in web applications, and defenses against them [Familiarity] • Use client-side security capabilities [Usage]	• Web security model – Browser security model including same-origin policy – Client-server trust boundaries, e.g., cannot rely on secure execution in the client • Session management, authentication – Single sign-on – HTTPS and certificates • Application vulnerabilities and defenses – SQL injection – XSS – CSRF • Client-side security – Cookies security policy – HTTP security extensions, e.g. HSTS – Plugins, extensions, and web apps – Web user tracking • Server-side security tools, e.g. Web Application Firewalls (WAFs) and fuzzers

Readings : [WL14]

Unit 8: Platform Security (25)	
Competences Expected: CS6,CS7,CS9	
Learning Outcomes	**Topics**
• Explain the concept of code integrity and code signing and the scope it applies to [Familiarity] • Discuss the concept of root of trust and the process of secure boot and secure loading [Familiarity] • Describe the mechanism of remote attestation of system integrity [Familiarity] • Summarize the goals and key primitives of TPM [Familiarity] • Identify the threats of plugging peripherals into a device [Familiarity] • Identify physical attacks and countermeasures [Familiarity] • Identify attacks on non-PC hardware platforms [Familiarity] • Discuss the concept and importance of trusted path [Familiarity]	• Code integrity and code signing • Secure boot, measured boot, and root of trust • Attestation • TPM and secure co-processors • Security threats from peripherals, e.g., DMA, IOMMU • Physical attacks: hardware Trojans, memory probes, cold boot attacks • Security of embedded devices, e.g., medical devices, cars • Trusted path
Readings : [WL14]	

Unit 9: Digital Forensics (25)

Competences Expected: C8,C9

Learning Outcomes	Topics
• Describe what is a Digital Investigation is, the sources of digital evidence, and the limitations of forensics [Familiarity] • Explain how to design software to support forensics [Familiarity] • Describe the legal requirements for use of seized data [Familiarity] • Describe the process of evidence seizure from the time when the requirement was identified to the disposition of the data [Familiarity] • Describe how data collection is accomplished and the proper storage of the original and forensics copy [Familiarity] • Conduct data collection on a hard drive [Usage] • Describe a person's responsibility and liability while testifying as a forensics examiner [Familiarity] • Recover data based on a given search term from an imaged system [Usage] • Reconstruct application history from application artifacts [Familiarity] • Reconstruct web browsing history from web artifacts [Familiarity] • Capture and interpret network traffic [Familiarity] • Discuss the challenges associated with mobile device forensics [Familiarity]	• Basic Principles and methodologies for digital forensics • Design systems with forensic needs in mind • Rules of Evidence - general concepts and differences between jurisdictions and Chain of Custody • Search and Seizure of evidence: legal and procedural requirements • Digital Evidence methods and standards • Techniques and standards for Preservation of Data • Legal and Reporting Issues including working as an expert witness • OS/File System Forensics • Application Forensics • Web Forensics • Network Forensics • Mobile Device Forensics • Computer/network/system attacks • Attack detection and investigation • Anti-forensics

Readings : [WL14]

Unit 10: Secure Software Engineering (25)

Competences Expected: C21,C22

Learning Outcomes	Topics
• Describe the requirements for integrating security into the SDL [Familiarity] • Apply the concepts of the Design Principles for Protection Mechanisms, the Principles for Software Security (Viega and McGraw), and the Principles for Secure Design (Morrie Gasser) on a software development project [Familiarity] • Develop specifications for a software development effort that fully specify functional requirements and identifies the expected execution paths [Familiarity]	• Building security into the software development lifecycle • Secure design principles and patterns • Secure software specifications and requirements • Secure software development practices • Secure testing- the process of testing that security requirements are met (including static and dynamic analysis).

Readings : [WL14]