NATIONAL UNIVERSITY OF ENGINEERING

COLLEGE OF SCIENCES

COMPUTER SCIENCE PROGRAM

CC067 – COMPUTING SECURITY

I. GENERAL INFORMATION

CODE	: CC067 Computing Security
SEMESTER	: 8-10
CREDITS	: 3
HOURS PER WEEK	: 4 (Theory, Practice)
PREREQUISITES	: NONE
CONDITION	: Elective

II. COURSE DESCRIPTION

The course prepares students for understanding and applying the concepts, methods and techniques of computer, information and network security in their different levels of implementation and deployment, pointing to protect them from theft, damage, disruption or misdirection of the services they provide. Students apply the methods of security by design, security architecture, cryptography, security measures, secure coding, vulnerability management, vulnerabilities reduction, hardware protection mechanisms, secure operating systems, capabilities and access control. Along the course, students design, develop and implement a software application whose functionality is presented and defended.

III. COURSE OUTCOMES

At the end of the course students:

1. Understand and appraise the importance of data confidentiality, data integrity, authentication, and non-repudiation.
2. Understand and apply the principles of end-to-end secure design in computer and information systems.
3. Identify potential threats and attacks and apply defensive programming with data validation and computing process verification.
4. Understand the principles of cryptography for constructing and analyzing protocols for preventing non-authorized use of computer systems and technologies.

IV. LEARNING UNITS

1. SECURITY CONCEPTS AND FUNDAMENTALS

Risks and threats. Vulnerabilities. Attack vectors. Authentication and authorization. Mandatory and discretional access control. Trust and trustworthiness. Ethics and responsible disclosure. Vulnerabilities. CIA: Confidentiality, integrity, availability.

2. SECURE DESIGN PRINCIPLES

Minimum privileges and isolation. Fai-safe defaults. Open design. End-to-end security. Deep defense. Design security. Trade-off between security and other design goals. Complete mediation. Vetted security components. Economics of security mechanisms. Reduced trusted computing base. Minimized attack surface. Security composability. Prevention, detection and deterrence.

3. DEFENSIVE PROGRAMMING
Input validation and data cleaning. Selection of programming language and secure languages. Examples of flaws in input validation and data cleaning. Buffer overflows. Integer errors. SQL injection. XSS vulnerability. Cross site scripting. Race conditions. Exceptions management and unexpected behaviors. Correct randomness generation for security purposes. Use and verification of third-parties components. Security updates. Control of flow information. Input errors mitigation and detection. Fuzzing: tests using invalid, unexpected or random data. Static and dynamic analysis. Program verification. Programs verification. Operating systems support. Hardware support DEP, TPM.

4. THREATS AND ATTACKS
Attacker motivations, motivations and capacities. Malware: virus, worms, spyware, bots, trojans, rootkits. Social engineering: phishing, pharming, email spam, link spam. Side channels. Covered channels. Service denial (DOS). Distributed DOS. Privacy attacks. Anonymity.

5. SECURITY PLATFORMS
Code integrity and code signature. Secure boot, measured boot, root of trust. TPM and security co-processors. Security threats from peripheral devices DMA, IOMMU. Physical attacks: hardware Trojans, memory probes, boot attacks. Security of embedded devices. Trusted paths. Statistical disclosure limitations. Backup policies. Password updating policies. Breach disclosure policies. Data gathering and retention policies. Supply chain policies. Cloud security: advantages and disadvantages.

6. SECURE SOFTWARE ENGINEERING
Security within software life-cycle SDLC. Principles of secure design and patterns (Saltzer, Schroeder, etc.). Secure software specification and requirements. Secure coding techniques for vulnerability minimization. Security testing. Quality assurance and benchmarking metrics.

V. METHODOLOGY

The course takes place in theory and practice sessions. In theory sessions, the instructor presents the concepts, methodologies, processes and models for computing security. In practice session, instructor and students work together for analyzing and proposing solutions to different problems related to security of computing systems in their different levels of application and deployment. At the end of the course, each student group present and defend the final project report

VI. EVALUATION FORMULA

The final grade PF is calculated as follows:

$$PF = 0.20\ EP + 0.3\ EF + 0.20\ PR + 0.30\ TF$$

where:

PF: Final grade	EP: Mid-term exam	EF: Final exam
PR: Practice work:	TF: Final project report and defense	

VII. BIBLIOGRAPHY

1. Computer Security: Art and Science
   Matt Bishop
   Addison Wesley Editions, 2016.

2. Information Security Management Handbook
   Auerbach Publications. V Edition. 2016