SYLLABUS

1. Data about the program of study

1.1	Institution	Technical University of Cluj-Napoca
1.2	Faculty	Automation and Computer Science
1.3	Department	Computer Science
1.4	Field of study	Computer Science and Information Technology
1.5	Cycle of study	Master of Science
1.6	Program of study / Qualification	Cybersecurity Engineering / Master
1.7	Form of education	Full time
1.8	Subject code	8.

2. Data about the subject

2.1	Subject name				***Web Security***	
2.2	Course responsible/lecturer				Conf.dr.ing. Teodor ȘTEFĂNUȚ - teodor.stefanut@cs.utcluj.ro	
2.3	Teachers in charge of seminars				Conf.dr.ing. Teodor ȘTEFĂNUȚ - teodor.stefanut@cs.utcluj.ro	
2.4 Year of study		I	2.5 Semester	2	2.6 Type of assessment (E - exam, C - colloquium, V - verification)	E
2.7 Subject category		Formative category: DA – advanced, DS – speciality, DC – complementary				DA
		Optionality: DI – imposed, DO – optional (alternative), DF – optional (free choice)				DI

3. Estimated total time

3.1 Number of hours per week	3	of which	3.2 Course	2	3.3 Seminar	0	3.3 Laboratory	1	3.3 Project	0
3.4 Total hours in the curriculum	42	of which	3.5 Course	28	3.6 Seminar	0	3.6 Laboratory	14	3.6 Project	0

3.7 Individual study:	
(a) Manual, lecture material and notes, bibliography	16
(b) Supplementary study in the library, online and in the field	16
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays	49
(d) Tutoring	0
(e) Exams and tests	2
(f) Other activities	0

3.8 Total hours of individual study (sum (3.7(a)...3.7(f)))	83
3.9 Total hours per semester (3.4+3.8)	125
3.10 Number of credit points	5

4. Pre-requisites (where appropriate)

4.1	Curriculum	Security issues at the source code level
4.2	Competence	Web programming, Databases, Computer networks

5. Requirements (where appropriate)

5.1	For the course	blackboard, beamer, computers
5.2	For the applications	blackboard, beamer, computers

6. Specific competences

Professional competences	**C1. Identify and understand the security issues specific to the different contexts of computing system usage. Appropriately apply the basic elements of security management and methods of evaluation and management of information security risks.** • **C1.1.** Knowledge of advanced theoretical and practical terminology, concepts, and principles specific to cybersecurity field. Knowledge of concepts about cybersecurity risk evaluation, and management. • **C1.2.** Understanding cybersecurity risks specific to new situations and their relationship with previously experienced situations and risks. Be able to predict possible threat scenarios when using cybersecurity solutions in new fields or situations. • **C1.3.** Capability to identify and model new types of cybersecurity risks affecting end users, computing systems, and software applications, and identify and evaluate possible solutions against such risks. • **C1.4.** Capability to identify and assess the limitations of existing cybersecurity solutions and their security risks, relative to well-known classifications. • **C1.5.** Capability to develop new theoretical models and methods to analyze and assess the cybersecurity properties and effectiveness of existing solutions. **C3. Analyze and evaluate the security characteristics of computing system. Identify the misconfigurations and software vulnerabilities.** • **C3.1.** Theoretical and practical knowledge of different cases of computing system misconfiguration and misusage that expose them to cybersecurity attacks, and of different types of software vulnerabilities and possible cybersecurity attacks. • **C3.2.** Be able to analyze and understand new kinds of software and communication protocols, in order to identify new possible cybersecurity threats, vulnerabilities, and risks. Be able to use commonly used databases of reported vulnerabilities and attacks in the process of assessing the cybersecurity of a new computing system. • **C3.3.** Capability to make cybersecurity assessments and identify possible attack surface of unknown computing systems, networks, or software applications. • **C3.4.** Capability to identify and assess theoretical and practical limitations of existing automatic vulnerability detection tools and propose possible combinations of such tools for improved results, where and if possible. • **C3.5.** Capability to propose new vulnerability identification, analysis, and classification, methods. Capability to propose solutions against exploitation techniques of such vulnerabilities. **C4. Design and develop highly secure software, security solutions and tools.** • **C4.1.** Knowledge of basic concepts and principles of secure software development and evaluation. Knowledge of common types of security software and tools. Knowledge of different operating system architectures, hardware and software infrastructures and frameworks needed to develop effective security solutions. • **C4.3.** Capability to develop complex secure software, complying with recommended good practices of built-in security and secure coding. Capability to develop software tools used for cybersecurity pentesting and assessment. • **C4.4.** Capability to assess complex software projects and identify their cybersecurity vulnerabilities and flaws, regarding their design, implementation, or testing, and propose improved development methods from the cybersecurity perspective.
Cross competences	N/A

7. Discipline objectives (as results from the *key competences gained*)

7.1	General objective	Understanding of common vulnerabilities of Web applications and how they can be leveraged with malicious intentions. Learn best practice techniques for secure Web applications development, deployment, and configuration.
7.2	Specific objectives	1. Understand how Web applications work 2. Develop abilities for identifying vulnerabilities in the implementation of Web applications 3. Learn techniques to leverage vulnerabilities of Web applications (XSS, SQL injection, etc.) 4. Develop necessary skills to write secure code for Web applications

		5. Learn how to correctly configure Web applications, from security perspective

8. Contents

8.1. Lecture (syllabus)	Number of hours	Teaching methods	Notes
Overview of Web technologies (1): general concepts (client/server, web 2.0, DOM, etc.), architecture of a web application (frontend/middleware/backend)	2	Blackboard illustrations and explanations, beamer presentations, discussions, short challenges	
Overview of Web technologies (2): protocols (ISO-OSI, HTTP, FTO, TCP, SOAP etc.) and programming/description languages (HTML, CSS, SVG, JS, XML, JSON, PHP, Python, Ruby etc.)	2		
Web Security (1): authentication (identity), authorization, encryption, and applicable legislation	2		
Web Security (2): confidentiality, integrity and availability, network level (firewall, IPS)	2		
Servers' security (1): vulnerabilities and attacks (OWASP, SQL injection / session hijacking / SSL / direct objects referencing / etc.)	2		
Servers' security (2): availability assurance ((D)DoS attacks) and correct configuration	2		
Clients' security (1): common vulnerabilities (browsers, plugins, cookies, DNS), clickjacking	2		
Clients' security (2): configuration, sandboxing, user scripting, malware/spyware	2		
Web cryptography: general aspects, public/private keys, certificates, message integrity, protocols (SSL, HTTPS, etc.)	2		
Proactive security measures: detecting intrusions in Web applications, security incidents management, honeytokens	2		
Security on Web 2.0: AJAX paradigm, cloud computing, etc.	2		
Secure Web programming (1): input validation, sanitizing error messages, identity, access control, sessions management	2		
Secure Web programming (2): management of sensitive personal/financial data, best practices in secure programming of Web applications	2		
Synthetic overview of entire course, highlight of important conclusions, discuss subjects chosen by students	2		

Bibliography

1. 24 Deadly Sins of Software Security (Howard, Michael – 2010 – McGraw-Hill)
2. Web Penetration Testing with Kali Linux (Muniz, Joseph – 2013 – Packt Publishing)
3. Hacking Exposed: Web Application (Scambray, Joel – 2010 – McGraw-Hill) (3rd ed)
4. The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws (Stuttard, Dafydd – 2011 – Wiley) (2nd ed)
5. The Basics of Hacking and Penetration Testing, Second Edition: Ethical Hacking and Penetration Testing Made Easy (Engebretson, Patrick – 2013 – Sygress)
6. Web Security Testing Cookbook (Hope, Paco – 2008 – O'Reilly Media)
7. Online articles and web sites

8.2. Seminar / Laboratory / Project	Number of hours	Teaching methods	Notes
Implementation of a minimal Web application	1		

(frontend/middleware/backend)			
Study of network packages in Web protocols, implementation / configuration of a firewall	1		
Forensics of Web attacks: OWASP, session related vulnerabilities, SQL injection	1		
Forensics of Web attacks: XSS, CSRF, direct unsecured references, SSL	1		
Analysis and exploitation of vulnerabilities in Web browsers: JavaScript, path traversal, browsers' plugins (Unity, Java, etc.)	1		
Secure programming: input validation, error messages sanitization, sensitive data management, best practices in Web security	1		
Use of validation instruments for websites: fuzzers and vulnerabilities scanners	1		

Bibliography

1. 24 Deadly Sins of Software Security (Howard, Michael – 2010 – McGraw-Hill)
2. Web Penetration Testing with Kali Linux (Muniz, Joseph – 2013 – Packt Publishing)
3. Hacking Exposed: Web Application (Scambray, Joel – 2010 – McGraw-Hill) (3rd ed)
4. The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws (Stuttard, Dafydd – 2011 – Wiley) (2nd ed)
5. The Basics of Hacking and Penetration Testing, Second Edition: Ethical Hacking and Penetration Testing Made Easy (Engebretson, Patrick – 2013 – Sygress)
6. Web Security Testing Cookbook (Hope, Paco – 2008 – O'Reilly Media)
7. Online articles and web sites

9. Bridging course contents with the expectations of the representatives of the community, professional associations and employers in the field

Achieved through periodic discussions with the representatives of significant employers, mainly companies that have projects in information security.

Web security disciplines are present in many similar master programs in computers and information security, like:

- XACS241 - Web Security 2.0 (Stanford) – http://scpd.stanford.edu/search/publicCourseSearchDetails.do?method=load&courseId=1284858
- 06-20009 Network Security (University of Birmingham) – http://www.cs.bham.ac.uk/internal/modules/2010/20009/
- Internet and Security (Nottingham University) – http://targetpostgrad.com/course/31312-internet-and-security
- Master of Science in Cybersecurity (University of Maryland) – http://www.umuc.edu/academic-programs/masters-degrees/cybersecurity.cfm
- Applied Cyber Security (MIT) – http://web.mit.edu/professional/short-programs/courses/applied_cyber_security.html

10. Evaluation

Activity type	10.1 Assessment criteria	10.2 Assessment methods	10.3 Weight in the final grade
10.4 Course	Ability to address and solve problems specific to Web security Attendance, active participation to the activities	Written exam and/or multiple-choice questions and/or oral presentation and/or research presentation on topics from discipline. Examination will be face-	40%

	during lectures	to-face or online.	
		Exercises on identification and exploitation of specific vulnerabilities of web applications, organized during lectures	20%
10.5 Laboratory	Ability to solve problems that are specific to Web security Attendance, active participation to the activities during classes	Completion of practical activities, on-time submission of homework and/or solving specific problems in a practical exam. Multiple-choice exam for testing knowledge of important concepts in Web security, on paper or electronic support, organized face-to-face or online.	40%

Minimum standard of performance

Lecture. Attendance to **minimum 50%** of lecture in order to be admitted to the final exam. Solving the exercises from the lectures and submitting solution on time. These exercises cannot be recovered. Capability of defining and explaining basic concepts of Web applications' security (SQL injection, XSS, CSRF, etc.) and of identifying the main risks involved in data management and public Web applications.

Lab. Attendance to **100%** of classes (1 class can be recovered during the semester and a second one during the re-examination interval) in order to be admitted to the final exam. Activity from the laboratory classes is validated only after all the required exercises from each class are solved and submitted to the teacher. The submission deadline is three weeks from the laboratory class and they cannot be recovered later. The ability to identify basic/most common vulnerabilities (SQL injection, CSS, CSRF, configuration, etc.) in source code. The ability to write secure code for small Web applications.

Final grade for discipline: 40% laboratory + 40% final exam + 20% lectures exercises

Acceptance to final exam: minimum 50% attendance to lectures, 100% attendance to laboratory classes, ≥ 5 laboratory grade

Graduate requirements: ≥ 5 final exam

Date of filling in:		**Title Surname Name**	**Signature**
	Lecturer	Conf.dr.ing. Teodor ȘTEFĂNUȚ	
	Teachers in charge of application	Conf.dr.ing. Teodor ȘTEFĂNUȚ	

Date of approval in the department 20.02.2024	Head of department Prof.dr.ing. Rodica Potolea
Date of approval in the faculty 22.02.2024	Dean Prof.dr.ing. Mihaela Dinsoreanu