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	9.2

2. Data about the subject

2.1	Subject name			***Virtualization and Hardware-Based Security***	
2.2	Course responsible/lecturer			Conf. dr. ing. Adrian COLEȘA - adrian.colesa@cs.utcluj.ro	
2.3	Teachers in charge of seminars			Conf.dr.ing. Adrian COLEȘA - adrian.colesa@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				DS
	Optionality: DI – imposed, DO – optional (alternative), DF – optional (free choice)				DO

3. Estimated total time

3.1 Number of hours per week	4	of which	3.2 Course	2	3.3 Seminar	0	3.3 Laboratory	2	3.3 Project	0
3.4 Total hours in the curriculum	56	of which	3.5 Course	28	3.6 Seminar	0	3.6 Laboratory	28	3.6 Project	0

3.7 Individual study:	
(a) Manual, lecture material and notes, bibliography	24
(b) Supplementary study in the library, online and in the field	12
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays	56
(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)))	94
3.9 Total hours per semester (3.4+3.8)	150
3.10 Number of credit points	6

4. Pre-requisites (where appropriate)

4.1	Curriculum	computer programming, data structure and algorithms, operating systems
4.2	Competence	C programming, basic knowledge of (x86) computer architecture, basic Web programming

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	**C2. Investigate and analyze cyber-criminality actions and malware using advanced methods such as reverse engineering and behavior monitoring.** • **C2.1.** Advanced knowledge of classifications and characteristics of different cybersecurity attacks and malware. • **C2.2.** Be able to analyze and understand new kinds of malware, the new techniques they use to attack, gain persistence, escalate privileges etc., and be able to compare them with known attack techniques. • **C2.4.** Capability to identify and assess theoretical and practical limitations of existing automatic malware analysis tools and propose improvements, where and if possible. • **C2.5.** Capability to derive new classes of attacks and exploitation techniques, supposed to be used by new malware, and propose the appropriate methods to identify and classify them correctly. **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. • **C4.2.** Be able to identify new situations and scenarios when it is needed to develop a new cybersecurity solution or use an existing one. Be able to analyze proposed cybersecurity solutions and compare them with existing ones. • **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. • **C4.5.** Capability to develop software modules and tools that could provide a high degree of cybersecurity. Capability to propose new methods to assess the cybersecurity of computing systems and devices and ways to improve it. **C5. Develop rigorous and efficient security solutions to complex real-life problems and situations. Be able to use security mathematical tools and models, engineering approaches and technologies specific and appropriate for the information and computing system security field.** • **C5.1.** Knowledge of complex relationship between cybersecurity and real-life aspects. Knowledge of mathematical theory some cybersecurity mechanisms and solutions are based on. • **C5.2.** Be able to analyze and understand new complex real-life scenarios from the cybersecurity perspective. Be able to identify needed cybersecurity solutions and derive new ones for new particular cases. • **C5.4.** Capability to identify and assess limitations of existing cybersecurity solutions and tools used in real-life situations, their residual cybersecurity risks, and their criticality. Capability to identify and research new cybersecurity fields and methods that could be used to reduce the limitations of existing cybersecurity solutions.
Cross competences	N/A

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

7.1	General objective	Have knowledge regarding the ways different modern hardware mechanisms, in particular hardware virtualization support, could be used for an improved security of computers and their software.
7.2	Specific objectives	1. Understand the main, security aimed, hardware mechanisms provided by the x86_64 architecture. 2. Understand the hardware virtualization support provided by the x86_64 architecture. 3. Be able to develop a mini-hypervisor using hardware virtualization support provided by the x86_64 architecture. 4. Have detailed knowledge regarding the different ways virtualization mechanisms could be used to improve the security of computers and their software. 5. Be able to asses and implement in a hypervisor different virtualization-based

		security mechanisms.

8. Contents

8.1. Lecture (syllabus)	Number of hours	Teaching methods	Notes
Course Introduction. Context and Virtualization Fundamental Aspects	2	Blackboard illustrations and explanations, beamer presentations, discussions, short challenges	
Different Types of Virtualization. Hardware Support for Virtualization	2		
Memory Virtualization	2		
Virtual Machine Introspection (VMI). VMI Overview and Formalization	2		
Reducing the Semantic Gap of VMI by Using Guest OS Semantic Knowledge	2		
Reducing the Semantic Gap of VMI by Using Architectural Semantic Knowledge	2		
Reducing the Semantic Gap of VMI by Using Hardware Performance Counters and Virtualization Events	2		
Reducing the Semantic Gap of VMI by Using Guest Assisted Methods	2		
I/O Virtualization Mechanisms. Security Challenges and Solutions	2		
Trusted Computing. Protection of Security-Sensitive Applications by Using Separated Red-Green VMs	2		
Trusted Computing. Protection of Trusted (Parts of) Applications inside Untrusted OS	2		
Trusted Computing. System Integrity Checking and Attestation	2		
Trusted Computing. Providing Trusted I/O Paths	2		
Cloud Security	2		

Bibliography

1. Intel, „*Intel 64 and IA-32 Architectures Software Developer's Manual*”, Volume 1-3, 2014, http://www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-manual-325462.pdf
2. B. Parno, J. McCune, A. Perrig, „*Bootstapping Trust in Modern Computers*”, Springer, 2011, http://research.microsoft.com/pubs/154093/bootstrappingtrustbook.pdf
3. Intel, „*Intel Trusted Execution Technology (TXT). Software Development Guide*”, 2014, http://www.intel.com/content/www/us/en/software-developers/intel-txt-software-development-guide.html
4. D. Weinstein, „*Advanced x86: Virtualization with Intel VT-x*”, 2012, online: http://opensecuritytraining.info/AdvancedX86-VTX.html
5. A. Segall, „*Introduction To Trusted Computing*”, 2013, online: http://opensecuritytraining.info/IntroToTrustedComputing.html
6. Articole indicate pe parcurs. Vezi http://www.citeulike.org/group/18034 cu etichete precum: *virtualization*, *introspection*, *light-virtualization*, *trusted*, *hvs-course (to be added)*

8.2. Seminar / Laboratory / Project	Number of hours	Teaching methods	Notes
Introduction. Administrivia. Understand the architecture of the mini-hypervisor used during lab-related activity. Build the development environment. Configure, build, run and update the mini-HV development project.	2	Brief reviews, blackboard illustrations and explanations, tutorials,	
Get familiar to tge mini-HV architecture and code structure.	2		

Logging mechanisms, dynamic memory allocation, linked lists, synchronization.		roadmaps, short live demos and guidance of code development, discussions, homework	
Memory virtualization. EPT structure	2		
Memory virtualization. EPT-configured permission rights	2		
Booting a simplified virtual machine (VM). VM enters and exists (1)	2		
Booting a simplified virtual machine (VM). VM enters and exists (2)	2		
Handling a VMCALL from a VM	2		
Booting a Windows VM	2		
VM-HV inter-communication mechanisms	2		
Implement an in-guest agent to communicate to the HV and execute given commands. Get the running process list	2		
Get the in-guest running process list from the HV	2		
Hidden process detection	2		
Attack prevention using EPT-based protection	2		
Subject review, demos, discussions. Lab evaluation	2		

Bibliography

1. Intel, „*Intel 64 and IA-32 Architectures Software Developer's Manual*", Volume 1-3, 2014, http://www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-manual-325462.pdf
2. D. Weinstein, „*Advanced x86: Virtualization with Intel VT-x*", 2012, online: http://opensecuritytraining.info/AdvancedX86-VTX.html

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

It is performed by periodic talks with important cybsersecurity industry representatives.
We also take a look and keep in touch with good ideas and proposals of other academic institutions in our country and abroad that run cybersecurity related study programs or/and research projects.

10. Evaluation

Activity type	10.1 Assessment criteria	10.2 Assessment methods	10.3 Weight in the final grade
10.4 Course	Ability to define and explain concepts and methods specific to hardware-based security field. Attendance frequency, interest, and interactivity during lecture classes.	Written exam, including online quiz tests (e.g. on Moodle platform) and presentation(s) of different subjects / paper in the course's field during semester time.	50%
10.5 Laboratory	Capability and ability to give correct and functional solutions to problems specific to hardware-based security field. Attendance frequency, interest, and interactivity during lecture classes.	Evaluate lab activity. Evaluate lab assignments (homework). Evaluate solutions of problems given in a final lab exam.	50%

10.6 Minimum standard of performance

***Lecture*.** Attending **minimum 50%** of lecture classes, to be allowed to take the final examination. Students must be able to define and explain the main concepts related to hardware virtualization support on Intel x86_64 architecture. Minimum final grade must be 5 for the exam to be considered passed.

***Lab*.** Attending **all lab classes** (one lab could be recovered during the semester, and one more during re-

examination sessions). Students must be able to extend the given min-HV functionality such as booting a Windows VM and get the list of running processes from the VM. This kind of assessment could happen in relation to assignments given during semester or subjects given during the final lab evaluation.
Minimum lab grade must be 5 for being allowed at final exam.

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

Date of approval in the Computer Science Department 20.02.2024	Head of department Prof.dr.ing. Rodica Potolea
Date of approval in the faculty of Automation and Computer Science 22.02.2024	Dean Prof.dr.ing. Liviu Miclea