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	6.

2. Data about the subject

2.1 Subject Name			***Windows Internals and Kernel Driver Development***		
2.2 Course responsible/ Lecturer			Drd. Ing. Radu-Marian PORTASE - rportase@bitdefender.com		
2.3 Teachers in charg of semiaries			Drd. Ing. Radu-Marian PORTASE - rportase@bitdefender.com		
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	4	of which	3.2 Course	1	3.3 Seminar	0	3.3 Laboratory	3	3.3 Project	0
3.4 Total hours in the curriculum	56	of which	3.5 Course	14	3.6 Seminar	0	3.6 Laboratory	42	3.6 Project	0
3.7 Individual study										
(a) Manual, lecture material and notes, bibliography										12
(b) Supplementary study in the library, online and in the field										12
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays										43
(d) Tutoring										0
(e) Exams and tests										2
(f) Other activities										0

3.8 Total hours of individual study (summ (3.7(a)…3.7(f)))	69
3.8 Total hours per semester (3.4+3.8)	125
3.10 Number of credit points	5

4. Pre-requisites (where appropriate)

4.1 Curriculum	Operating systems
4.2 Competence	Computer architecture, knowledge about operating systems concepts and internals, programming in C and x86 or amd64 assembler, basic understanding of computer networks and protocols

5. Requirements (where appropriate)

5.1. For the course	Blackboard, Video Projector
5.2. For the laboratory	Blackboard, Video Projector, Laptop with internet access

6. Specific competences

Profesional competences	**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. • **C5.5.** Capability to run research activities and projects aimed to derive applicable cybersecurity solutions, implement their hardware and/or software prototype.
Cross competences	N/A

7. Obiectivele disciplinei

7.1 General objective	Understand Windows internals and how Windows drivers may be created. Gain practical experience with various Windows kernel technologies. Understand how Windows kernel drivers may be used for security.
7.2 Specific objectives	1. Understand Windows architecture and various executive subsystems 2. Know about the various types of kernel drivers that can be built 3. Learn how to build and debug kernel drivers 4. Learn how user mode applications interact with kernel drivers 5. Understand the security implications of Windows kernel drivers 6. Learn how kernel drivers can be used to increase security of an information system

8. Content

8.1 Lecture	Hours	Teaching Methods	Notes
Windows architecture; Using the kernel debugger to learn about various Windows structures and to debug the kernel	1	Blackboard illustrations and explanations, beamer presentations, discussions, short challenges	
Kernel programming fundamentals	2		
Windows I/O subsystem (I/O Manager, IRP processing); Memory management	2		
Monitoring application behavior with Windows Drivers	2		
Unscheduled execution (Interrupts, DPC queue and APCs)	1		
Creating device drivers created. Building drivers with KMDF and UMDF	1		
Programming drivers for USB devices	1		

Windows network filters. The Windows Filtering Platform	1		
Security of kernel modules. Specific vulnerabilities and exploitation techniques.	1		
Analysis of other Winows Internal Structures	1		
Concepts review	1		

Bibligraphy

1. Windows Kernel Programming (Yosifovich, Pavel – 2019 - CreateSpace Independent Publishing Platform)
2. Windows Internals (Russinovich, Mark – 2012 – Microsoft Press) (6th ed)
3. Windows NT File System Internals (Nagar, Rajeev – 2006 – OSR Reprint)
4. Windows Driver Kit (WDK) (Microsoft – 2010-2014 – electronic)
5. Windows Operating System Internals Curriculum Resource Kit (CRK) (Microsoft – 2006 – electronic)
6. Windows Research Kernel 1.2 (Microsoft – 2006 – electronic)

8.2 Applications (seminary/laboratoyr/project)	Hours	Teaching Methods	Notes
Learning about the development environment and debugging tools.	6	Brief reviews, blackboard illustrations and explanations, tutorials, roadmaps, short live demos and guidance of code development, discussions, homework	
Development of a NT legacy driver and a console application that is connected to it.	9		
Development of an anti-malware oriented driver. Understand the basic structure and architecture for a minisystem driver.	3		
Antimalware Driver: Interception of Windows filesystem activity	3		
Antimalware Driver:Interception of registry activity	3		
Antimalware Driver:Interception of process, threads, modules and object activity	3		
Antimalware Driver: Interception of network activity	6		
Device drivers created with KMDF. Keyboard interception	6		
Concepts review	3		

Bibliography

1. Windows Internals (Russinovich, Mark – 2012 – Microsoft Press) (6th ed)
2. Windows NT File System Internals (Nagar, Rajeev – 2006 – OSR Reprint)
3. Windows Driver Kit (WDK) (Microsoft – 2010-2014 – electronic)
4. Windows Operating System Internals Curriculum Resource Kit (CRK) (Microsoft – 2006 – electronic)
5. Windows Research Kernel 1.2 (Microsoft – 2006 – electronic)

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

Course was designed together with security profesionals from companies that have relevant activity in the field (e.g. Bitdefender) and is alligned with subjects evaluated for various pentesting certifications.

Related courses from other universities:

- *ECE 446 – Device Driver Development*, George Mason University, Fairfax, USA http://catalog.gmu.edu/preview_course_nopop.php?catoid=19&coid=226124
- *COP 5641 – Linux Kernel & Device Driver Programming,* Florida State University, USA http://www.cs.uni.edu/~diesburg/courses/dd/syllabus.html

Part of the details from the lecture sylabus are also part of various operating systems courses.

10. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course	Ability to solve domain-specific problems Presence, (inter)activity during class hours	Written and/or oral exam, onsite or online (depends on medical conditions) Tools used: MS Teams, Moodle	50%
Laboratory	Ability to solve domain-specific problems Presence, (inter)activity during class hours	We will evaluate the ability to create various security oriented drivers by reviewing code submissions. Discussions are on site if medical conditions allow it	50%

Minimum standard of performance

***Lecture*.** Attending **minimum 50%** of lecture classes, to be allowed to take the final examination.

Minimum grade for exam: 5

Desired competences: Understand Windows Internals at an intermediate level. Know how to develop Windows kernel drivers.

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

Minimum grade on all laboratory submissions: 5

Desired competences: Demonstrate ability write kernel level code. Understand basic security of kernel drivers. Understand architectures for anti-malware drivers.

Date of filling in:		**Title Surname Name**	**Signature**
	Lecturer	Drd. Ing. Radu Portase	
	Teachers in charge of application	Drd. Ing. Radu Portase	

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