Universidad
Zaragoza

Academic Year/course: 2022/23

62240 - Exploiting Software Vulnerabilities

Syllabus Information

Academic Year: 2022/23
Subject: 62240 - Exploiting Software Vulnerabilities
Faculty / School: 110 - Escuela de Ingeniería y Arquitectura
Degree: 534 - Master's Degree in Informatics Engineering
ECTS: 3.0
Year:
Semester: First semester
Subject Type: Optional
Module:

1. General information

1.1. Aims of the course

By taking this course, students will be able to analyze code and software systems to identify and solve the most common vulnerabilities and security problems. Thus, they will be able to apply various techniques to analyze security and compromise vulnerable software systems, reliably demonstrating the existing problems and proposing appropriate improvement solutions.

These approaches and objectives are aligned with the following Sustainable Development Goals (SDGs) of the United Nations 2030 Agenda (https://www.un.org/sustainabledevelopment), in such a way that the acquisition of the learning results of the course provides training and competence to contribute to a certain extent to its achievement. Specifically, they are aligned with the following objectives:

- Goal 9.1: Develop quality, reliable, sustainable and resilient infrastructure, including regional and transborder infrastructure, to support economic development and human well-being, with a focus on affordable and equitable access for all.
- Goal 11.2: By 2030, provide access to safe, affordable, accessible and sustainable transport systems for all, improving road safety, notably by expanding public transport, with special attention to the needs of those in vulnerable situations, women, children, persons with disabilities and older persons.

1.2. Context and importance of this course in the degree

Software development is one of the fundamental pillars of the development of any industrial product because it is either part of the product, or it is part of the product development, or both. In this development, more and more people are working in multidisciplinary teams, with people from different disciplines and with different capacities. However, many of these developments have flaws in their form, both at the design level and at the implementation level. These defects, which can lead to vulnerabilities, can be exploited, thus compromising the security of the system (undermining some of its confidentiality, integrity and availability properties). The publication of vulnerabilities in widely used software is increasingly common, such as OpenSSL (POODLE and Heartbleed vulnerabilities, among others) or the Unix Bash console (Shellshock vulnerability, among others).

To minimize these potentially exploitable defects in software systems, it is important to know the most frequent vulnerabilities introduced during code development, as well as the mechanisms to avoid them. This course will also introduce the underlying concepts behind vulnerabilities, possible attacks and defenses to prevent their exploitation, as well as some advanced techniques for automatic software analysis and exploitation. Finally, a methodology for the construction of vulnerability exploitation codes will be introduced.

1.3. Recommendations to take this course

Knowledge of computer programming and architecture at the level of a graduate in Computer Science.

2. Learning goals

2.1. Competences

- CB-06 - Possess and understand knowledge that provides a basis or opportunity to be original in the development and / or application of ideas, often in a research context

- CB-09 - That students know how to communicate their conclusions and the knowledge and ultimate reasons that support them to specialized and non-specialized audiences in a clear and unambiguous way
- CB-10 - That students possess the learning skills that allow them to continue studying in a way that will be largely self-directed or autonomous.
- CG-09 - Ability to understand and apply ethical responsibility, legislation and professional deontology of the activity of the profession of Computer Engineer
- CG-11 - Ability to acquire advanced and demonstrated knowledge, in a context of scientific and technological or highly specialized research, a detailed and well-founded understanding of the theoretical and practical aspects and of technological or highly specialized, a detailed and well-founded understanding of the aspects theoretical and practical and the work methodology in one or more fields of study.
- CG-13 - Ability to evaluate and select the appropriate scientific theory and the precise methodology of its fields of study to formulate judgments based on incomplete or limited information including, when necessary and pertinent, a reflection on the social or ethical responsibility linked to the solution proposed in each case

Achieve the following specific skills:

- CTI-01 - Ability to model, design, define the architecture, implement, manage, operate, administer and maintain applications,
computer networks, systems, services and content.
- CTI-02 - Ability to understand and know how to apply the operation and organization of the Internet, the technologies and protocols of Next-generation networks, component models, middleware, and services.
- CTI-03 - Ability to ensure, manage, audit and certify the quality of developments, processes, systems, services, computer applications and products.
- CTI-04 - Ability to design, develop, manage and evaluate certification mechanisms and security guarantee in the treatment and access to information in a local or distributed processing system.

2.2. Learning goals

Each student must be able to:

- RA1: Recognize the most common vulnerabilities in software systems.
- RA2: Evaluate the security of a software system.
- RA3: Mastering different software systems analysis techniques.
- RA4: Create proofs of concept that allow compromising the security of vulnerable software systems.

2.3. Importance of learning goals

Graduates will be able to recognize the most common vulnerabilities in software systems, in addition to proposing improvement solutions to avoid them or even developing proofs of concept that allow the exploitation of the vulnerability.

Today, the detection and exploitation of vulnerabilities is a booming field in the software development ecosystem, with many companies offering rewards of various kinds to those who improve the security of their products. Additionally, the profile of expert code analysts and vulnerability search is in high demand by technology companies, both in the defensive field (blue team) and in the offensive field (red team). We understand that knowing these vulnerabilities and their underlying principles, as well as defense techniques, enables graduates to design and implement systems in a more secure manner. All this also increases their employability in the labor market.

3. Assessment (1st and 2nd call)

3.1. Assessment tasks (description of tasks, marking system and assessment criteria)

The student must demonstrate that they have achieved the expected learning outcomes through the following assessment activities:

Practical statements for the analysis of vulnerable programs will be proposed that must be resolved in the laboratory. These works will be graded with a quantitative grade from 0 to 10. A correct explanation and development of the analysis carried out, based on the concepts studied in the subject, will be especially valued.

The presentation and defense of practical programming works will be assessed with a practical grade that will be weighted with 70% of the final grade for the subject. With this test the learning outcomes RA1, RA2, RA4 will be evaluated.

Finally, there will be a final evaluation test, consisting of a presentation of group work, which will serve to demonstrate that the required learning outcomes in the subject have been achieved. In this test, problems of a similar nature to those raised in class (code analysis and proof of concept) will be solved. The grade obtained in this test will weigh 30% of the final grade for the course. With this test the learning outcomes RA1, RA2, RA3, and RA4 will be evaluated.

The students who do not opt for the evaluation procedure described above, or do not pass these tests during the teaching period, or want to improve their grade, will have the right to take a global test that will be scheduled within the examination period corresponding to the first or second call.

4. Methodology, learning tasks, syllabus and resources

4.1. Methodological overview

The methodology followed in this course is oriented towards achievement of the learning objectives. A wide range of teaching and learning tasks are implemented, such as:

- Lectures. The instructor presents and explains the class contents, including illustrative examples.
- Laboratory sessions. Activities with specialized equipment (in the laboratory, computer room).
- Oral presentations. Preassigned problems will be presented on the classroom
- Assignments. Preparation of seminars, readings, small research projects, documents to be presented on the classroom or handed in to the teacher.

4.2. Learning tasks

The course (75 hours) includes the following learning tasks:

- 26 hours, approximately, of classroom activities: lectures, laboratory sessions, and problem-solving tasks.
- 30 hours, approximately, of assignments and research projects.
- 5 hours, approximately, of tutorials.
- 10 hours, approximately, of autonomous work and study.
- 4 hours, approximately, of the exam and defense of the course project.

4.3. Syllabus

- Introduction: vulnerability management, types of vulnerabilities, tools and analysis lab
- Program binary analysis: static analysis, dynamic analysis. Ethical concerns
- Software vulnerabilities and exploitation techniques: memory errors (in heap, in stack), integers, format strings, concurrency issues
- Software defenses
- Advanced exploitation techniques: ROP attacks, custom shellcode design

4.4. Course planning and calendar

The teaching planning of this course is organized as follows:

- Lectures and problem-solving tasks
- Laboratory sessions

The exact hours of lectures and laboratory sessions will be announced beforehand in the Center's and course's websites.

Further details concerning the timetable, classroom, office hours, assessment dates and other details regarding this course, will be provided on the first day of class and announced beforehand in the Center's and course's websites.

4.5. Bibliography and recommended resources

The recommended bibliography for this course can be consulted at this link of the University of Zaragoza Library.