Recep Tayyip Erdoğan University

Faculty of Engineering and Architecture, Computer Engineering

CEN429 - Secure Programming Course Syllabus

Fall Semester, 2024-2025

Download

- PDF
- DOC
- SLIDE
- PPTX

Instructor:	Asst. Prof. Dr. Uğur CORUH
Contact Information:	ugur.coruh@erdogan.edu.tr
Office No:	F-301
Google Classroom Code	?????????
Microsoft Teams Code	?????????
Lecture Hours and Days	Thursday, 13:00 - 15:00 IBBF 402, 4th Floor

Lecture Classroom	IBBF 402 4th Floor or Online via Google Meet / Microsoft Teams
Office Hours	Meetings will be scheduled via Google Meet or Microsoft Teams using your university account and email. Email requests for meetings are required. To receive a faster response, ensure your email subject begins with *[CEN429]*, and write clear, concise, formal emails.

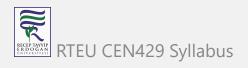| Lecture and Communication Language	English
Theory Course Hour Per Week	3 Hours
Credit	?????
Prerequisite	CE103- Algorithms and Programming I
Corequisite	TBD
Requirement	TBD

*TBD: To Be Defined.

A. Course Description

This course provides a comprehensive approach to understanding secure software development techniques. Students will learn software protection methods to address common security vulnerabilities such as buffer overflows, memory leaks, and injection attacks. The course focuses on real-world applications and best practices in secure software development.

B. Course Learning Outcomes

After successfully completing this course, a student will be able to:

1. Identify common software security vulnerabilities.

2. Apply encryption methods to secure sensitive data.

3. Implement code hardening techniques (such as RASP and code obfuscation).

4. Establish secure communication channels using encryption and authentication mechanisms.

5. Plan secure software design by employing defense strategies.

6. Perform security reviews and vulnerability assessments.

C. Course Topics

- **Introduction to Secure Programming and Application Protection Plan**

  - Overview of application protection

  - Buffer overflows and prevention techniques

  - Memory management and security

  - Protected code partitioning and secure processing with encryption

- **Computer Viruses and Security Models**

  - Types of viruses (program, macro, boot sector)

  - Attack trees and security models

  - Virus countermeasures

- **Data Security: In Transit, At Rest, and In Use**

  ○ SSL/TLS usage

  ○ Whitebox cryptography applications

  ○ Dynamic key management

  ○ Data masking techniques

- **Code Hardening Techniques (C/C++)**

  - Control flow flattening

  - Function name hiding

  - Memory allocation obfuscation

  - Dynamic encryption techniques

- **Code Hardening Techniques (Java/Interpreted Languages)**

  - Code obfuscation with ProGuard

  - Dynamic method obfuscation

  - Static string obfuscation and protection

  - Advanced ProGuard rules

- **Runtime Application Self-Protection (RASP) Techniques (C/C++)**

  - Root detection and APK signature verification

  - Dynamic memory protection

  - Debugger and memory monitoring detection

  - Protection against dynamic analysis tools

- **Advanced Code Obfuscation and Diversification Techniques**

  - Dynamic control flow obfuscation

  - Adding fake control flows

  - Program obfuscation techniques

  - Dynamic function calls and runtime optimizations

- **Quiz 1 + Midterm Project Report Submission**

  - Quiz covering the first 7 weeks of content
  - Submission of midterm project report

- **Certificates and Cryptographic Methods (AES, RSA, PKI)**

  ○ AES and RSA encryption

  ○ Data integrity with HMAC

  ○ Creating digital signatures and certificates

- **Whitebox Cryptography**

  - Whitebox AES and DES applications

  - Key protection with whitebox cryptography

  - Software-based security solutions

- **Security Certifications and Penetration Testing Planning**

  - ETSI and EMV security standards

  - Penetration testing planning

  - PCI DSS and ISO 27001 security testing

- **Midterm Project Report Submission**

  ○ Project progress and report presentation

- **Real-World Case Studies**

  - Mobile and desktop application security

  - Multi-layered defense implementations

  - Applied penetration testing scenarios

- **Quiz 2**
  - The second quiz covers topics from the previous weeks

- **Final Project Submission and Presentation**

  - Final project report and presentation

D. Textbooks and Required Hardware or Equipment

There is no required textbook for this course. However, the following sources are recommended:

- **C How to Program, 7/E. Deitel & Deitel. 2013, Prentice-Hall.**

- **Introduction to Algorithms, Third Edition By Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.**

- **Problem Solving and Program Design in C, J.R. Hanly, and E.B. Koffman.**

You are expected to have a personal laptop for class activities and exams.

E. Grading

You will complete one project and two written quizzes throughout the semester. You are expected to submit your Midterm Project Report at the midterm, demonstrating progress and outputs aligned with your project plan. In the 15th week, you will present and submit your Final Project Report. You will take a written quiz in the 8th week and another in the 14th week.

Assessment	Code	Weight	Scope
Midterm Project Report	RAP1	60%	Midterm
Quiz-1	QUIZ1	40%	Midterm
Final Project Report	RAP2	70%	Final
Quiz-2	QUIZ2	30%	Final

$$Grade_{Midterm} = 0.6RAP1 + 0.4QUIZ1$$

$$Grade_{Final} = 0.7RAP2 + 0.3QUIZ2$$

$$PassingGrade = (40 * Grade_{Midterm} + 60 * Grade_{Final})/100$$

F. Course Execution

Classes will be conducted in person, with supplemental materials and announcements shared via Google Classroom or Microsoft Teams. In the event of a pandemic-related shift to remote learning, the course will transition to synchronous and asynchronous online education methods.

G. Late Submissions

Assignments and projects must be submitted by the specified deadlines. Late submissions will not be accepted. Any unexpected situations must be communicated to the instructor as soon as possible.

H. Communication and Course Platform

Google Classroom and Microsoft Teams will serve as the platforms for this course. All resources and announcements will be shared through these channels. To ensure smooth communication, you are expected to check these platforms regularly.

I. Academic Integrity, Plagiarism, and Cheating

Academic integrity is one of the most important principles at RTEÜ University. Anyone who violates academic honesty will face serious consequences.

Collaborating with classmates or others to "study together" is normal. A student may seek help from someone else, paid or unpaid, to better understand a difficult topic or course. However, where is the line between "studying together" and "academic dishonesty"? When does it become plagiarism, and when is it cheating?

It is clear that looking at another student's paper or any unauthorized sources during the exam is cheating and will be punished. However, many students enter university with little experience regarding what is acceptable in assignments, particularly when it comes to "copying."

The following guidelines for the Faculty of Engineering and Architecture students highlight the philosophy of academic honesty for graded assignments. Should any situation arise that is not described here, students are advised to consult with the course instructor or assistant to determine whether their actions fall within the bounds of academic honesty.

a. What is acceptable when preparing an assignment?

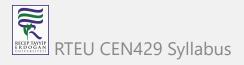- Communicating with classmates to better understand the assignment.

- Including ideas, quotes, paragraphs, or small code snippets found online or elsewhere in your assignment, provided that:
  - These do not represent the entire solution to the assignment.
  - The sources are properly cited.

- Asking for guidance in improving the English content of your assignment.

- Sharing small portions of your assignment in class for discussion.

- Using external sources for technical instructions, references, or troubleshooting, but not for direct answers to the assignment.

- Discussing solutions using diagrams or summarized statements, but not actual text or code.

- Working with (and even paying) a tutor to help with the course, as long as the tutor does not complete the assignment for you.

b. What is not acceptable?

- Asking to see a classmate's solution to a problem before submitting your own.

- Failing to cite the source of any text or code used from outside the course.

- Giving or showing your solution to a classmate struggling to solve the problem.

J. Expectations

You are expected to attend classes on time and complete weekly course requirements (readings and assignments) throughout the semester. The primary communication channel between the instructor and students will be email. Please send your questions to the instructor's university-provided email address. ***Be sure to include the course name in the subject line and your name in the body of the email.*** The instructor will also contact you via email when necessary, so it is crucial to check your email regularly for communication.

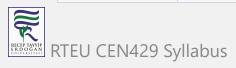## K. Course Content and Schedule Updates

The course content and schedule may be updated as needed. Any changes will be communicated to students by the instructor.

C. Weekly Lesson Plan

Week	Date	Topic	Other Tasks
1. Week	26 September 2024	Introduction to Secure Programming and Application Protection Plan	
2. Week	3 October 2024	Computer Viruses and Security Models	
3. Week	10 October 2024	Data Security: In Transit, At Rest, and In Use	

C. Weekly Lesson Plan

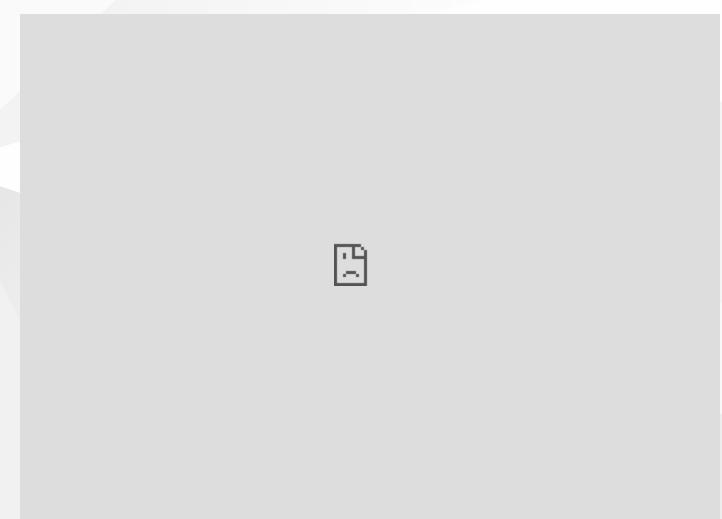| Week	Date	Topic	Other Tasks
4. Week	17 October 2024	Code Hardening Techniques (C/C++)	
5. Week	24 October 2024	Code Hardening Techniques (Java/Interpreted Languages)	
6. Week	31 October 2024	Runtime Application Self-Protection (RASP) Techniques (C/C++)	
7. Week	7 November 2024	Advanced Code Obfuscation and Diversification Techniques	

C. Weekly Lesson Plan

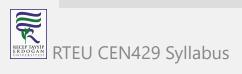| Week	Date	Topic	Other Tasks
8. Week	14 November 2024	**Quiz 1 + Midterm Project Report Submission**	Quiz and Midterm Report
9. Week	21 November 2024	Certificates and Cryptographic Methods (AES, RSA, PKI)	
10. Week	28 November 2024	Whitebox Cryptography	
11. Week	5 December 2024	Security Certifications and Penetration Testing Planning	

C. Weekly Lesson Plan

Week	Date	Topic	Other Tasks
12. Week	12 December 2024	Midterm Project Report Submission	Midterm Project Report
13. Week	19 December 2024	Real-World Case Studies	
14. Week	26 December 2024	**Quiz 2**	Quiz 2
15. Week	2 January 2025	**Final Project Submission and Presentation**	Final Project and Presentation

Bologna Information

$$End - Of - Syllabus$$