# CEN 5079: Software Vulnerabilities and Security

## Catalog Description

Development of applications that are free from common security vulnerabilities, such as buffer overflow, SQL injection, and cross-site scripting attacks. Emphasis is on distributed web applications.

Prerequisites: Graduate standing. (3 credits)

## Course Description and Purpose

Internet security has become part of everyday life, where security problems impact practical aspects of our lives. Even though there is a considerable corpus of knowledge about tools and techniques to protect systems, information about the actual vulnerabilities and how they are exploited is not generally available. This situation hampers the effectiveness of security research and practice. Understanding the details of attacks is a prerequisite for designing and implementing secure systems.

This course deals with common programming, configuration, design mistakes, and ways to detect and avoid them. Examples highlight general error classes, such as stack and heap overflows. Possible protection and detection techniques are examined. The course includes several practical lab assignments where participants must apply their knowledge and discuss the current research in the field. Students will learn how the security of systems can be violated and how such attacks can be detected and prevented.

The course aims to make the students "**security aware**" and gain an in-depth understanding of security issues.

## Course Outcomes

Upon completing this course, students will be able to:

1. Understand foundations of software security design patterns **[Understanding]**
2. Distinguish common systems security flaws and problems and recognize those problems in real-world applications **[Analyzing]**
3. Assess the attack surfaces in different software ecosystems (Web, OS, Systems) **[Evaluating]**

4. Decompose modern malware attacks (e.g., Remote Access Trojan, Cryptojacking, Ransomware) by investigating common evasion mechanisms **[Analyzing]**
5. Develop Proof of Concept (PoC) exploitation code to take control vulnerable software systems (e.g., buffer overflow, SQL injection) in a lab setting **[Creating]**
6. Evaluate and test systems for intrusion and data breach **[Evaluating]**

## Schedule and Timeline

| Dates | Slides | Reading | Extra Reading and Labs |
|---|---|---|---|
| Jan 10-12 | Introduction to Software Security and Vulnerabilities<br><br>History | | |
| Jan 17- 19 | Threat Modeling,<br>Secure Architecture | Setuid Demystified | Lab 0: Unix Basics |
| Jan 24- 26 | Reconnaissance<br>Password Security | Understanding password choices | Chal. 1 on Reconnaissance |
| Jan.31<br>Feb 02 | Honeywords,<br>Password Cracking<br>Access Control | Honeywords:making password cracking detectable | Chal. 2 on Unix Security |
| Feb. 07-09 | Authentication-Part 1<br>Authentication-Part 2 | | |
| Feb. 14-16 | Web Security Fundamentals<br>Command Injection | You are What You Include | Chal. 3 on Password |
| Feb. 21-23 | SQL Injection<br>Cross Site Scripting | Toss a Fault to Your Witcher | |
| Feb. 27- Mar 03 | | Spring break | |
| Mar07-09 | Into Systems Security | | Chal. 4 on SQL Injection<br>Introduction to ElF<br>X86 AsseIntroduction to ELFmbly |

|  |  |  |  |
|---|---|---|---|
| Mar14-16 | Process Interactions<br>Memory Registers |  |  |
| Mar21-23 | Stack Overflow<br>Smashing the Stack | Stackguard:Automatic Adaptive Detection and Prevention of Buffer-Overflow Attacks<br>ROP is still dangerous<br>On the Effectiveness of Address Space Randomization | Chal. 5 on Parameter Injection<br>Shellcode<br>Using GDB to Develop Exploits<br>Smashing the Stack<br>Return to libc |
| Mar28-30 | Format String Vulnerability |  | Chal6 on Buffer OverFlow<br>Exploit 101 |
| Apr04-06 | Reverse Engineering<br>Forensics | Does Every Second Count?<br>Unveil: A automated approach to detecting ransomware<br>Evading Malware Analysis Systems via Wear-and-Tear Artifacts |  |
| Apr. 11-13 | DDoS | SoK: detecting Amplification DDOS Attacks<br>SPIFFY: Inducing Cost-Detectability Tradeoffs for Persistent Link-Flooding Attacks | Chal. 7 on Reverse Engineering |
| Apr. 18 | Zoom Day |  | Chal. 8 on Forensics Analysis |
| Apr. 25 | Final Exam |  |  |

## Course Prerequisites

The topics discussed in this course require some prior exposure to software systems specifically operating systems. Students always ask what courses would be good to take before taking this course. Courses such as COP 4610 —Operating Systems Principles or CTS 4348 —Unix Sys Admin will cover the background for this course.

**Please note that these courses are not official requirements. This means if you know UNIX OS, the course will be easier for you. It doesn't mean you should**

**not take the course. However, if one doesn't have system skills, they may probably need to dedicate more time for challenges. Hope that makes sense :).**

**Software background:**

CEN 5079 requires significant programming experience. If you are a beginner, this course is not for you. For instance, constructing SQL queries, writing code in C/C++ should not be very difficult for you. Also, knowledge of the Unix/Linux command line is essential. You should know how to write code using emacs/vim, write a makefile, compile code using makefiles, use SSH and SCP, write very simple shell scripts, work with gdb, check for running processes, kill runaway processes, and create compressed archives.

## Textbook and Course Materials

## The book does not have any specific textbook. When needed, we refer you to open access journals and papers.

This is a known reference for learning most of low-level attacks such as format string vulnerabilities, buffer overflow

- **Hacking: the Art of Exploitation (Optional)**
  - Publisher: No Starch Press, 2nd Edition
  - ISBN-13: 978-1593271442

There are tons of resources online on learning how to work with a linux-based environment. If you haven't had any exposure to linux before, these resources are good for absolute beginners.

- **Linux Journey** -- a free way to learn linux
- **Interactive Wargames to practice command line**

**Expectations of this Course**

In this course, students are expected to:

- **review the getting started page** located in the course modules;
- **introduce yourself to the class** during the first week by posting a self-introduction in the appropriate discussion;
- **interact** with instructor and peers;
- **review** and follow the course calendar and weekly outlines;
- **log in** to the course 3 **times** per week;
- **No late work will be accepted;**
- **respond** to **emails** within 2 **days;**
- **submit** assignments by the corresponding deadline.
- **communicate** effectively (at least 24 hours) for exam accommodation in case getting sick

The instructor will:

- log in to the course four times each week;
- respond to **emails** within 48 **hours**;
- grade assignments within two **days** of the assignment deadline.

### Needed Software

The reference operating system for this course is ubuntu 22 LTS. The lab environment machines are also Ubuntu OS. You can rely on SSH or PuTTY

to get a remote command line on the lab environment, but you run the risk of Wifi connection issues leaving you unable to work. macOS users should be able to use the default Mac command line and Homebrew; Windows users can install Linux in a virtual machine, or, if you have a recent version of Windows 10, you can install the Windows Subsystem for Linux (WSL) and then download a copy of Ubuntu right from the Windows Store.

.

### Course Communication

Communication in this course will take place via the Canvas Inbox and Piazza forum. Check out the Canvas Conversations Tutorial or Canvas Guide to learn how to communicate with your instructor and peers using Announcements, Discussions, and the Inbox. I will respond to all correspondences within **24 hours.**

## Discussion Forums

Keep in mind that your discussion forum postings will likely be seen by other members of the course. Care should be taken when determining what to post.

### Discussion Forum Expectations:

- If you have specific questions about challenges, you need to send private email.
- Class home works are solo challenges so do not disclose answers in the class discussions.
- Include screenshots when having issues. This helps us to quickly identify and understand the issue.

## Quizzes

In order to mitigate any issues with your computer and online assessments, it is very important that you take the Practice Quiz from each computer you will be using to take your graded quizzes and exams. Assessments in this course are not compatible with mobile devices and should not be taken through a mobile phone or a tablet.

- There will be four online quizzes.
- Quizzes are closed-book
- Quizzes will be posted on canvas on Mondays
- The duration of quizzes are 15 to 20 minutes.
- Quizzes will be posted on Tuesdays at 6:00 PM
- List all assessments (i.e. graded or practice)
- Quizzes will be graded in two weeks.
- Students will be able to see the grades on canvas as well as the course's lab environment.
- Students will receive feedback in form of comments or exam keys.
- **Makeups will be provided on a needs-driven basis. *We cannot accommodate requests after the test date.***

## Accommodations for Students with Disabilities

If you have a disability-related need for reasonable academic accommodations in this course and have not yet met with a Disability Specialist, please visit FIU's DRC.

and follow the outlined procedure to request services. If the Disability Resource Center has formally approved you for an academic accommodation in this class, please present the instructor with your "Professor Notification Letter" at your earliest convenience, so that we can address your specific needs as early as possible.

## Exams

- Midterm and final exams are closed book
- Final exam is commutative.

## Assignments

- You will have 9 software security challenges.
- **Assignments are due at 11:59:59pm on the specified date.**
- The due date for each assignment is two weeks from the time we release the assignment.
- You will use a turn-in script to create a compressed archive of the necessary files for the assignments, timestamp them, and submit them for grading. I highly recommend that students start assignments early!
- You should complete all parts of this *assignment* entirely on your own without help from any other person
- You will be provided detailed guideline and demos on how to submit your responses
- Your responses will be automatically graded and you will see the grade in almost real-time.
- Most projects can be programmed in a language of your choice. The only universal requirement is that your projects must compile and run on an *unmodified* Linux machine that we give you access to.
- **Notice the stress on unmodified: if you're relying on libraries or tools that are only available in your home directory, then we will not be able to run your code and you will fail the assignment. You are welcome to develop and test code on your home machines, but in the end everything needs to work on the course Linux machines. If you have any questions about the use of particular languages or libraries, post them to Piazza.**

For more information, please review the important information about the assignments page.

## Zoom Meetings will be held on the following dates/time:

- Office Hours: Thursday at 11 AM to 11:45 AM
- Live Sessions (for quizzes and and exams, as well as live admin updates)
- Zoom Link

**Piazza Link:**

- Class Forum

Zoom meetings can be accessed via the Zoom link in the course navigation menu. Once you click on the Zoom link, it will route you to join the meeting for the respective class session. You will also be able to view upcoming meetings, previous meetings that you have already joined, and meeting recordings. Before joining an actual class session:

- Reference the Zoom Student Tutorials

If you encounter any technical difficulties, please contact the FIU Canvas Help Team. Please ensure you contact support immediately upon the issue occurring.

## Ethics

In this class, you will learn about security techniques and tools that can potentially be used for offensive purposes, "hacking" in other words. It is imperative that students only use these tools and techniques on systems they own (your personal computers) or systems that are sanctioned by the instructor. NEVER perform attacks against public systems that you do not control. As we will discuss in class, it is ethically problematic to attack systems you do not own and may violate the law.

## Cheating Policy

It's ok to ask your peers about the concepts, algorithms, or approaches needed to do the assignments. We encourage you to do so; both giving and taking advice will help you to learn. **However, what you turn in must be your own**. **Looking at or copying code or homework solutions from other people or the Web is strictly prohibited**.

In particular, looking at other solutions (e.g., from other students who previously took the course) is a direct violation. Projects and exams must be *entirely* the work of the students turning them in. **Needless to say, using ChatGPT's responses to answer quizzes, midterm, and final exams are strongly prohibited and will be considered as cheating.**

If you have any questions about using a particular resource, ask the course staff or post a question to the class forum. **Please take a moment and read the academic misconduct.**

All students are subject to the FIU's Academic Integrity Policy. Per College policy, all cases of suspected plagiarism or other academic dishonesty *must* be referred to the Office of Student Conduct and Conflict Resolution (OSCCR). This may result is deferred suspension, suspension, or expulsion from the university.

## Title IX

Title IX makes it clear that violence and harassment based on sex and gender are Civil Rights offenses subject to the same kinds of accountability and the same kinds of support applied to offenses against other protected categories such as race, national origin, etc. If you or someone you know has been harassed or assaulted, you can find the appropriate resources here.

.