CSCI 8245 Secure Programming

With Rust Language

Spring 2022

Instructor: Mustakimur Rahman Khandaker.
Office: 547 Boyd Graduate Studies Research Center.
Email: mrkhandaker@uga.edu

Office Hours: Monday and Thursday 02:00 PM - 03:00 PM or by appointment.

Lecture time and location:
N/A

Monday	N/A
Tuesday/Thursday	N/A

Teaching Assistant:
N/A.

Textbooks:

- The Rust Programming Language

Prerequisite:
Good understanding of system application programming (C/C++).

Course Objective:
Our core software stacks are built upon unsafe language i.e. C/C++. With the advancement of memory-safe languages such as Rust and Go, industries are showing interest in rewriting those legacy system applications. Primarily because they are focused on providing control and performance similar to C/C++ language. And, they also guarantee memory, thread, and type safety. Most of them also support advanced features such as functional programming, object-oriented programming, and unsafe code separation (useful to support inline assembly code).

Particularly, Rust is well-known for its intimidating focus on system application development. To guarantee memory safety, unlike Go, Rust adopts an ownership model (fast and effective) that avoids garbage collector (GC). Rust also introduce a packaging system called Cargo that enhances the collaboration experience. Most recently, top industries (e.g. Microsoft, Google, Amazon, Facebook) partners together to found Rust foundation. Very soon, those industries will have a high demand for Rust developers.

Learning Rust is not like learning any other language. It is not about syntax, it is about understanding the underlying reason of the language features. So, a good understanding of software vulnerabilities is a top requirement. In this course, students will first learn about security threats of unsafe language and program analysis methods to identify software vulnerabilities. Later, the class will continue based on the fastest-growing memory-safe language (i.e. Rust) and explore how it guarantees memory, thread, and type safety besides some of the advanced features. The course will equally inspire students to explore state-of-the-art systems developed in memory-safe language e.g. operating systems and the web framework.

Course Description:
The course materials are designed to teach students how vulnerable codes are introduced in unsafe languages and how they can be avoided with a memory-safe language. Besides regular lectures on security threats of unsafe language, program analysis methods, and exploring memory-safe language, students will be assigned a program analysis homework (10%), multiple small coding (Rust) assignments (30%), one group project (30%) in Rust, and two papers (20%). There will be multiple quizzes that will cover the remaining 10% of the grading.

The course syllabus will be as following:
Section I: Software Security.

I. Software vulnerabilities.
    A. Memory corruption.
    B. Synchronization bugs.
    C. Exploitation.
II. Software Analysis.
    A. Program analysis.
    B. Sanitizers.
    C. Fuzzing.

Section II: Memory-safe Language.

III. Basics.
    A. Cargo.
    B. Data types.
    C. Control-flow.
IV. Memory safety.
    A. Ownership model.
    B. Lifetimes.
    C. Error handling.
    D. Smart pointers.
V. Complex types.
    A. Structs, enums, collectors.
    B. Generics.

C. Traits.

VI. Thread safety.
   A. Concurrency.
   B. Shared memory.
   C. Channel.

Section III: Advanced Language Features.

VII. Functional programming.
   A. Iterators.
   B. Closures.
   C. Pattern matching.

VIII. Engineering.
   A. Modules and privacy.
   B. Nonblocking I/O.
   C. Automated testing.

IX. Rust special:
   A. Unsafe Rust.
   B. Macros.
   C. Crates.

X. Applications.
   A. Operating system.
   B. Web framework.
   C. Blockchain.

Evaluation and Grading Policy:

- Quizzes: 10%
  - There will be 5 quizzes.
  - Each worth 2 pts of overall grading.
  - Will be held on eLC on Monday class.
- Program Analysis: 10% (C/C++ with AFL Fuzzing)
  - Students will assign open-source projects to run AFL fuzzer.
  - Report the procedures followed.
  - Report vulnerabilities discovered.
- Individual Assignments: 30% (Rust)
  - There will be a total of 5 assignments.
  - Each worth 6 pts of overall grading.
  - Should avoid unrecoverable error in every aspect possible.
  - Should meet the goal.
- Group Project: 30% (Rust) [3 members team]
  - The project should be maintained in GitHub.
  - Every member should have an equal contribution.
  - Code quality should maintain SE standard (e.g. code format).
  - Should avoid unrecoverable error in every respect possible.

- Presentation: 20% (2 presentations).
    - Papers will be assigned to students from a shortlist of top-quality research.
    - Every paper will be shared by 2 students and presented together.
    - Equal contribution to presentation is a must.

Letter Grade Assignment:

Letter Grade	**Percentage**
A	91 - 100%
A-	86 - 90%
B+	81 - 85%
B	76 - 80%
B-	71 - 75%
C+	66 - 70%
C	61 - 65%
C-	56 - 60%
D	51 - 55%
F	0 - 50%

Instruction Policy:

- Course instructions will be (hopefully) in person.
    - Instruction policy depends on the UGA plan.
- Monday class will be mostly used for presentations and quizzes.
    - The quiz date will be declared a week before.
    - Will showcase lab assignments (coding and analysis).
    - Also, general discussion on project assignments.

Group Projects:

- A shortlist of good projects will be published.
- Details of expectations will be included in the handout.
- The project final deadline will be the last day of the course lecture.
    - There will be 3 intermediate release deadlines.
- A late release will receive a 2% penalty for each day.

Homework/Assignment Policy:

- A report including required data should be submitted for the program analysis homework.
    - Submit an archive on the eLC course page.
- The assignments will due on at Sunday 11:59 PM on the specific days.
    - Submit code on the eLC course page.
- Late submission will receive a 3% penalty for each day (maximum 3 days allowed).

Quiz Policy:

- The quiz will be online at class time (@ Monday class).
- The quiz will be announced at least a week in advance.
    - There will be no midterm or final exam.

Academic Honesty:

- Each student is expected to do his/her own work.
    - Teamwork is not allowed unless explicitly specified.
- Acknowledge all sources of information you have used/referred to in your assignments outside the textbook.
- Students are expected to familiarize themselves with the academic honesty policy of the University of Georgia: https://ovpi.uga.edu/academic-honesty.

Accommodations due to a disability:

If you plan to request accommodations for a disability, please register with the Disability Resource Center (DRC). The DRC can be reached by visiting Clark Howell Hall, by calling 706-542-8719 (voice) or 706-542-8778 (TTY), or by visiting http://drc.uga.edu.