Software Engineering 421
Software Analysis and Verification for Safety and Security

Instructor: Ben Holland

Fall 2018, Revision 1

E-mail: bholland@iastate.edu
Private Meetings: Available upon request
Open Help Hours: 1:00pm-2:00pm
Open Help Location: Gilman 2354

Web: **https://se421.github.io**
Class: M/W/F 12:10-1:00pm
Location: Gilman 2354

This course syllabus should be interpreted as a contract of understanding between students, teaching assistants, and the instructor. By participating in the course you are agreeing to this understanding. Please review this document carefully and inform the instructor of any concerns. Revisions to this document will be made as needed then posted and announced in class.

Instructor Statement

As a participant in this course it is my hope that you find the course materials not only engaging but also challenging. Above all else this course aims to provide opportunities for creative and critical thinking. Answers will not always be served up to you on a silver platter. The path forward to practical solutions will not be obvious and in some cases there may be more than one solution.

In an effort to produce a practical skill set valuable in the current workforce, almost every assignment will have a hands on component that involves building a software analysis tool. Because of the nature of the materials, it is highly probable that you will face a variety of technical challenges. Use any and all resources that are available to you and allow yourself the time to use those resources. There are no stupid questions. Your time is valuable to me and I do not wish to waste it. Please do the teaching assistants and myself the same favor by planning ahead to ask well researched questions during the allotted class and help hours. There will be very little tolerance for lazy students or those that wait until the last minute to complete assignments.

It is my hope that all students rightfully earn high marks in this course, but it is impossible to teach a person that does not want to learn. Please come to this class with an enthusiasm to learn and a willingness to think critically. I look forward to an exciting semester with each and everyone one of you!

Course Description

Software is pervasive, and for better or worse it now controls our daily lives. Almost every recent computer security issue has been rooted in software, so it is critical to develop and maintain secure software. To make matters worse the size of modern software is enormous and as it approaches mind boggling proportions, we must develop tools to assist in program comprehension.

Specific course topics will include: binary exploitation, program control and data flow, path analysis, pointer analysis, call graph construction, program slicing, projected control graphs, fuzzing, algorithmic complexity and side channel attacks, web security, Android security, threat modeling, among other topics. Guest speakers from a variety of backgrounds will provide an up to date accounting of modern practices.

Other topics include: significance of software safety and security; various facets of security in cyber-physical and computer systems; threat modeling for software safety and security; and categorization of software vulnerabilities. Software analysis and verification: mathematical foundations, data structures and algorithms, program comprehension, analysis, and verification tools; automated vs. human-on-the-loop approach to analysis and verification; and practical considerations of efficiency, accuracy, robustness, and scalability of analysis and verification. Cases studies with application and systems software; evolving landscape of software security threats and mitigation techniques. Understanding large software, implementing software analysis and verification algorithms.

In this course we will explore many facets of security from bug hunting to exploitation. **There is a heavy focus on learning by doing (many lectures will contain group coding activities).** Throughout the course, we will build a suite of program analysis tools to systematically analyze software for a variety of tasks. We will analyze applications and libraries written in C, C++, Java, and Java bytecode. In place of a final exam, a comprehensive course project to penetration test a vulnerable YouTube style web service will provide students the opportunity to participant in a purple team capture the flag style exercise to detect, patch, and exploit vulnerabilities (a live engagement where participants simultaneously act as attackers and defenders). A potential collaboration with a University abroad and domestic partners will provide realistic and challenging adversaries.

This course provides students with a unique experience to hone their software development and software security skills in a hands-on and fast paced environment. Course materials are designed to prepare students for the workforce with practical skills while also providing a solid foundation to continue learning and research beyond the course.

Prerequisites

Prerequisites: COM S 309 and either CPR E 310 or COM S 230

Required Materials

- There are no required books for this course. We will reference provided lecture notes and paper publications during the course.
- Due to the heavy focus on hands on learning students will often need to bring a laptop with a minimum of 4GB ram memory to class in order to run analysis software and virtual machines during group activities. Participants that cannot meet this requirement should inform the instructor in order to make practical arrangements throughout the course.

Teaching Assistants

This course will include multiple teaching assistants, whose contact information and office hours will be announced in a later revision of this document. Teaching assistants will support help hours, support in class activities, assist in the development of assignments, grade assignments, and occasionally give lectures.

Course Structure

Classroom Meetings

Generally class periods are arrange into learning modules that span one to two weeks. Learning modules include introductory materials, group work, advanced materials, an assignment, and optionally relevant readings and/or guest speakers. This course follows a partially flipped classroom model (see https://en.wikipedia.org/wiki/Flipped_classroom). Some classroom meetings will contain a standard lecture with interactive discussion. In other meetings, readings or warm up activities may be assigned before class and in place of a lecture a group activity (usually group discussions or peer programming) is scheduled. The group work is intended to foster learning from peers and serve as an instructor guided time period to begin the homework assignments during class.

Assignments

There will be roughly one assignment per learning module, not counting any readings or warm up activities assigned before group work activities. Almost every assignment will require developing a program analysis solution to a challenging software analysis problem. Conceptual questions reinforce important concepts and may only be practically achievable with a working software implementation. Partial credit is always available so do your best to complete whatever you can in each assignment.

Assignment reports must be professionally typed and submitted digitally as PDF files along with any required source code. Photos of hand written solutions embedded in a digital submission are not allowed and will not be graded. Reports must use complete sentences and proper grammar. The instructor and TA's reserve the right to deduct points for unprofessionalism found in reports. You are encouraged to develop your written reports using LaTeX, but PDFs generated from tools such as Microsoft Word are also acceptable. Read every assignment carefully and

follow all instructions.

Almost every assignment will also include an optional set of tasks for extra credit weighted as assignment points. These are the only extra credit points available in the course. Each extra credit task will require significant thinking and exploration of the subject. Partial credit is also available for extra credit tasks. Attempting to complete the extra credit tasks is highly encouraged for students considering pursing a graduate education or to secure an A grade in the course.

Midterm Exams

Two traditional written midterm exams will be given that cover concepts covered in the course. Midterm exams will take place during regular class meetings and are closed note.

Final Project

In place of a final exam, a comprehensive course project to penetration test a vulnerable YouTube style web service will provide students the opportunity to participant in a purple team capture the flag style exercise to detect, patch, and exploit vulnerabilities (a live engagement where participants simultaneously act as attackers and defenders). Students are encouraged to use the tools built in previous assignments to complete the software audit.

Final Exam Activity

A required activity, that will account for a portion of the class participation points, will take place during the regularly scheduled final exam period. The activity will be a red team exercise that will test your creative and critical thinking skills with respect to a subset of the topics in this course. The precise details of the activity will be announced the week before dead week. Extensive preparation is not required to perform well in the activity.

Grading Policy

The standard grading scale ($0-59=F$, $60-69=D$, $70-79=C$, $80-89=B$, $90-100=A$) will be used. At the end of the course, I reserve the right to curve the grading scale dependent on overall class scores and performance *after* extra credit has been counted. Any curve will only ever make it easier to obtain a certain letter grade. The curve is intended to correct any deficiencies in the course delivery while extra credit is meant to allow students to correct any deficiencies in their performance. It is possible that extra credit can affect the final curve.

The final grade will be computed using the following weights:

- 40% Assignments
- 30% Midterm Exams
- 25% Project
- 5% Participation

If you disagree with a grade given on an assignment, exam, or project see the grader and then the instructor if the problem is not resolved. If there is a disagreement the student should inform the grader or instructor within two weeks of the returned grade. Old grades will not be revisited after the two week time period.

Course Policies

During Class

Please refrain from using computers for anything but note taking or other activities related to the class. Please mute computers and other electronic devices so as not to distract others. Eating and drinking are allowed in class but please refrain from it affecting the course. Try not to eat your lunch in class as the classes are typically active.

Attendance Policy

If you plan to sleep or be distracted during class it is better that you do not attend. That being said your active attendance is required to enable the learning of others through in class discussions and activities. A portion of your overall grade is based on attendance and in class participation. On days with group activities your attendance is required unless excused in advance by the instructor. If you have a conflict during an announced in class activity, please inform the instructor as soon as possible.

Late Policy

Late assignments will be accepted for no penalty if a valid excuse is communicated to the instructor *before* the deadline. It is left to the discretion of the instructor to determine if an excuse is valid. All students are allotted a one time, no questions asked, no penalty, two day extension of exactly 48 hours from the original due date time. If a student wishes to use the one time extension the intention must be communicated to the instructor within the 48 hour window before or after the original due date time. Aside from the aforementioned exceptions, late assignments will not be graded and will result in a zero score.

Academic Integrity and Honesty

- Don't cheat. Please don't be that person.
- Students are required to comply with the university policy on academic integrity found in the Code of Student Conduct (https://www.studentconduct.dso.iastate.edu/). Suspected academic misconduct will be reported to the dean of students office http://www.dso.iastate.edu/ja/academic/misconduct.html.
- During the course you are encourage to seek additional resources online, through similar courses, open source projects, etc. If you reference a source (directly or indirectly) you must cite it! For example, this syllabus was based off of https://www.overleaf.com/latex/templates/syllabus/bdqfmfnncffs. Failure to cite a source will be considered plagiarism.

At the same time you must do your own thinking. Simply copying and citing large portions of a resource will earn little to no points.

- Group work is encouraged during and outside of the course. You will learn from working with your peers and discussing solutions, however every student is responsible for and must do their own work! If you work with other students document the group efforts in your report and code. For example, if you learn of a particularly elegant code snippet it is OK to include that solution as a part of work if you *1) credit the student or source in a code comment and 2) explain how and why the code works*.

Accommodations for Disabilities

Reasonable accommodations will be made for students with verifiable disabilities. In order to take advantage of available accommodations, students must register with the Disability Services Office at https://sas.dso.iastate.edu and inform the instructor of any required accommodations.

Discrimination

Discrimination based on race, color, religion, creed, sex, national origin, age, disability, veteran status, or sexual orientation is a violation of state and federal law and/or Iowa State University policy will not be tolerated. Any person who feels that he or she has been the subject of prohibited discrimination, harassment, or retaliation should contact the instructor and/or the Office for Equal Opportunity at https://www.eoc.iastate.edu.

Course Schedule

The schedule is tentative and subject to change. An up to date version of the schedule will be posted on the course website with additional materials.

Week 01, 08/20 - 08/24: Course Introduction

Week 02, 08/27 - 08/31: Binary Exploitation Part 1

Week 03, 09/03 - 09/07: Binary Exploitation Part 2

- *Monday, September 3, 2018:* Labor Day (No Class)

Week 04, 09/10 - 09/14: Control Flow Graphs + Path Counting

Week 05, 09/17 - 09/21: Data Flow Graphs + Points-to Analysis

- *Monday, September 17, 2018:* Midterm Exam 1

Week 06, 09/24 - 09/28: Call Graph Construction

Week 07, 10/01 - 10/05: Program Slicing + Taint Analysis + Projected Control Graphs

Week 08, 10/08 - 10/12: Dynamic Analysis

Week 09, 10/15 - 10/19: Algorithmic Complexity and Side Channel Attacks

- *Monday, October 15, 2018:* Midterm Exam 2

Week 10, 10/22 - 10/26: Web Security

Week 11, 10/29 - 11/02: Android Security

Week 12, 11/05 - 11/09: Threat Modeling + Human-in-the-loop Analysis

Week 13, 11/12 - 11/16: Secure Software Development + Penetration Testing

Week 14, 11/19 - 11/23: Thanksgiving (No Classes)

Week 15, 11/26 - 11/30: Final Project Engagement (Attack Phase)

Week 16, 12/03 - 12/07: Dead Week, Modern Trends in Program Analysis

- Final Project Report Due

Week 17, 12/10 - 12/14: Final Exams

- **Required Final Exam Activity** during two hour period arranged by *First Contact Hour* (see https://www.registrar.iastate.edu/students/exams/fallexams#stand)

Additional Resources

GitHub

The course assignments, project, lecture notes, and other materials will be posted on GitHub. Please keep materials distributed via private repositories private. The instructor and teaching assistants will have access to your private git repositories for each assignment in order to better assist you during the course. Developing or releasing assignment solutions in the public impedes the learning of current and future students.

GitHub Organization: https://github.com/se421

Canvas

Canvas will be used to assign due dates for assignments and collect assignment report submissions. Assignment reports should have a reference to the GitHub commit containing the relevant source code submission. Canvas will also be used to distribute any necessary course announcements.

Atlas

Atlas (https://www.ensoftcorp.com/atlas) is a program analysis framework that will be used heavily during the course. Atlas is developed by a local Ames, Iowa company called EnSoft Corporation (http://www.ensoftcorp.com). For support with general Atlas troubleshooting issues email support@ensoftcorp.com.

The Atlas program analysis framework was chosen because:

- The instructor and TAs have familiarity with the framework
- The framework is actively maintained and supported by a commercial business
- The framework is highly extensible
- The framework is completely free for students (request a license with your *.edu* email at https://www.ensoftcorp.com/atlas/academic-license)
- The framework APIs are written in Java (a familiar programming language in software engineering curriculum)
- The framework can be used to create auto graders in the form of unit tests that fairly and efficiently grade student solutions
- The instructor believes Atlas is objectively one of the best program analysis frameworks for teaching program analysis concepts

Slack Chat

A course Slack chat has been setup to allow you to seek help from your peers, teaching assistants, and the instructor. You may discuss approaches to an assignment and provide technical help to peers, but do not directly provide solutions. The chat may also be used to provide class specific feedback to the instructor and teaching assistants.

Slack Chat: https://se421.slack.com

Stack Overflow

For Atlas framework programming questions that have not already been answered by the community you can post a well researched question on Stack Overflow (see https://stackoverflow.com/help/how-to-ask). When you post the question tag it with the tag "`atlas-pa`" so that users following the tag are alerted to your question. This will not only help you but it will also help your peers having similar problems as well as reduce the overhead of the instructor and teaching assistants having to answer duplicate questions. You can follow recently tagged questions by updating your account preferences at: https://stackexchange.com/filters. Finally, don't be afraid to post a link to your question in the Slack chat to get the attention of other students, the teaching assistants, and the instructor.

Remember: Do not ask homework or course specific questions on Stack Overflow! Course specific questions are better asked during class or in the Slack chat.

Google

Of course you can always ask peers, a teaching assistant, or the instructor for help but don't make us send you a https://lmgtfy.com link. Make sure to google around first and be sure that your question has been well researched before asking for additional help.