CSCI 699: Testing in WebAssembly Ecosystem: From Development to Deployment

Units: 4

Fall 2023—Friday—1-4:20pm

Location: TBA

Instructor: Weihang Wang

Office: SAL 342
Office Hours: Friday 4:30-6:30pm
Contact Info: weihangw@usc.edu

Course Description

The goal of this course is to help the students learn fundamental software engineering (SE) techniques and how to use them in software testing, debugging, maintenance, and security. To achieve this goal, this course contains two components: (1) familiarizing students with fundamental SE techniques and recent advancements, and (2) applying or improving these techniques to a new web standard, WebAssembly, to improve the reliability of WebAssembly applications.

The first component covers fundamental SE topics, such as program representation, tracing, profiling, slicing, information flow tracking, static program analysis, symbolic execution, fuzzing, and so on. The second component uses WebAassembly as the subject program to put learned techniques into practice. WebAssembly is a new type of client-side web programming language serving as a universal compilation target for the web. It aims to support high-performance computations on the web, such as games, virtual reality, and image recognition. WebAssembly is supported on all major browsers (i.e., Chrome, Firefox, Safari, and Edge) and compiles from several programming languages, including C, C++, C#, Rust, Go, and more.

Learning Objectives

The objective of this course is to get students acquainted with SE techniques and their application in the domain of WebAssembly. Each student will work on a group project in a team of 3~5 students. The project topics will be program analysis of WebAssembly programs, and a list of concrete problems and project scopes will be provided. Each student will choose a topic from the list and then work in a team to accomplish the desired goals. Students will then present their project findings and write up their work as a 10-page two-column research paper, which will be submitted to a SE conference if possible. Students are also welcome to come up with their own ideas.

Recommended Preparation

The course assumes that students have prior experience with system programming. While there are no prerequisite courses, this course will include a team project which requires some amount of dedication, critical thinking, and/or substantial implementation effort.

Course Notes

The course will be letter graded. The lecture slides and notes will be posted on Blackboard.

Technological Proficiency and Hardware/Software Required

Students in this course will utilize a personal laptop or desktop and be familiar with the usage of the ACM Digital Library and Google Search.

Required Readings and Supplementary Materials

Papers, slides, and chapters handed out by the instructor.

Description and Assessment of Assignments

The workload of this course includes project progress presentations and a written research paper.

Each student is expected to give four brief presentations on the project's progress and outcome:

- *Project Proposal:* Each team will discuss the problem statement, project goals, and related work/the state-of-the-art. Each student will focus on a different aspect of the project, e.g., system design, user study, implementation, paper writing, etc.
- *Project Update 1:* Students will have a clear plan for the approach's design, as well as at least one or more working examples demonstrating that the approach is feasible. Students will submit a paper draft that includes the high-level idea and working examples.

- *Project Update 2:* Students will complete the approach implementation and prepare the datasets for evaluation. The experiments will include evaluation metrics and well-defined research questions. The selection of the datasets should be justified.
- *Project Demo:* Students will complete the experiments, analyze the results and findings, and demonstrate their system and results.

At the end of the course, each team will develop a research paper.

Grading Breakdown

We plan to determine grades as follows:

- The project presentations (*Project Proposal*, *Project Update 1*, *Project Update 2*, and *Project Demo*) will be graded: 15 points for each presentation, for a total of 60 points.
- The research paper will be graded for a total of 40 points based on its quality as well as each student's effort.

Table 1 Grading Breakdown

Assessment Tool (assignments)	**Points**
Project Proposal	15
Project Update 1	15
Project Update 1	15
Project Demo	15
A Research Paper	40
TOTAL	100

Grading Scale

Course final grades will be determined using the following scale:

Table 2 Course Grading Scale

Letter grade	Corresponding numerical point range
A	95-100
A-	90-94
B+	87-89
B	83-86
B-	80-82
C+	77-79
C	73-76
C-	70-72
D+	67-69
D	63-66
D-	60-62
F	59 and below

Course Schedule

Table 3 Course schedule

	Topics/Daily Activities	**Readings**
Week 1	Course Logistics WebAssembly Introduction	WebAssembly
Week 2	Course Project Introduction	WebAssembly
Week 3	Program Representation Program Tracing	Program Tracing
Week 4	*Project Proposal*	
Week 5	WebAssembly Abstractions WebAssembly Program Instrumentation	
Week 6	Program Slicing Information Flow Tracking	Program Slicing Program Profiling
Week 7	Static Program Analysis	Static Program Analysis
Week 8	*Project Update 1*	
Week 9	Test Programs and User Study Browser Automation	
Week 10	N-version Programming & Diversifying Programs Delta Debugging	Delta Debugging
Week 11	*Project Update 2*	
Week 12	Compiler Optimization Techniques Symbolic Execution	Symbolic Execution
Week 13	Fuzzing Test Generation	Fuzzing Test Generation
Week 14	*Project Demo*	
Week 15	Concolic Testing	Concolic Testing
FINAL	*Project Final*	*Due on university-scheduled date of the final exam*

Course Topics and Reading Lists:

WebAssembly

- Daniel Lehmann, Michael Pradel: Wasabi: A Framework for Dynamically Analyzing WebAssembly. ASPLOS 2019.

https://software-lab.org/publications/asplos2019_Wasabi.pdf

- Aaron Hilbig, Daniel Lehmann, Michael Pradel: An Empirical Study of Real-World WebAssembly Binaries: Security, Languages, Use Cases. WWW 2021.

https://software-lab.org/publications/www2021.pdf

- Daniel Lehmann, Johannes Kinder, Michael Pradel: Everything Old is New Again: Binary Security of WebAssembly. USENIX Security Symposium 2020.

https://www.usenix.org/system/files/sec20-lehmann.pdf

- Evan Johnson, David Thien, Yousef Alhessi, Shravan Narayan, Fraser Brown, Sorin Lerner, Tyler McMullen, Stefan Savage, Deian Stefan: Доверя'й, но проверя'й: SFI safety for native-compiled Wasm. NDSS 2021

https://ndss-symposium.org/wp-content/uploads/ndss2021_5B-3_24078_paper.pdf

- Daniel Lehmann, Michael Pradel: Finding the dwarf: recovering precise types from WebAssembly binaries. PLDI 2022.

https://software-lab.org/publications/pldi2022.pdf

- Alan Romano, Daniel Lehmann, Michael Pradel, Weihang Wang: Wobfuscator: Obfuscating JavaScript Malware via Opportunistic Translation to WebAssembly. IEEE Symposium on Security and Privacy 2022.

https://weihang-wang.github.io/papers/Wobfuscator-sp22.pdf

- Alan Romano, Yunhui Zheng, and Weihang Wang: MinerRay: Semantics-Aware Analysis for Ever-Evolving Cryptojacking Detection. ACM/IEEE International Conference on Automated Software Engineering 2020.

https://weihang-wang.github.io/papers/ASE2020-MinerRay.pdf

Program Tracing

- Nicholas Nethercote and Julian Seward, Valgrind: A Framework for Heavyweight Dynamic Binary Instrumentation, PLDI'07

https://dl.acm.org/citation.cfm?id=1250746

- Xiangyu Zhang and Rajiv Gupta, Whole Execution Traces and Their Applications, MICRO'04

https://dl.acm.org/citation.cfm?id=1089012

Program Profiling

- Thomas Ball and James R. Larus, Efficient Path Profiling, MICRO'96

https://dl.acm.org/citation.cfm?id=243857

- Darko Marinov and Robert O'Callahan, Object Equality Profiling, OOPSLA'03

https://dl.acm.org/citation.cfm?id=949333

Program Slicing

- Mark Weiser, Program Slicing, ICSE'81

https://dl.acm.org/citation.cfm?id=802557

- Ben Xin and Xiangyu Zhang, Efficient Online Detection of Dynamic Control Dependence, ISSTA'07

https://dl.acm.org/citation.cfm?id=1273489

Delta Debugging

- Andreas Zeller, Yesterday, my Program Worked. Today, it Does Not. Why? FSE'99

https://link.springer.com/chapter/10.1007%2F3-540-48166-4_16

- Andreas Zeller and Ralf Hildebrandt, Simplifying and Isolating Failure-Inducing Input, IEEE Transactions on Software Engineering, 2002

https://dl.acm.org/citation.cfm?id=506206

Static Program Analysis

- Al Bessey, Ken Block, Ben Chelf, Andy Chou, Bryan Fulton, Seth Hallem, Charles Henri-Gros, Asya Kamsky, Scott McPeak, and Dawson Engler, A Few Billion Lines of Code Later: Using Static Analysis to Find Bugs in the Real World, CACM'10

https://cacm.acm.org/magazines/2010/2/69354-a-few-billion-lines-of-code-later/fulltext

Symbolic Execution

- Cristian Cadar, Daniel Dunbar, and Dawson Engler, KLEE: Unassisted and Automatic Generation of High-Coverage Tests for Complex Systems Programs, OSDI'08

http://www.doc.ic.ac.uk/~cristic/papers/klee-osdi-08.pdf

- Cristian Cadar, Vijay Ganesh, Peter M. Pawlowski, David L. Dill, and Dawson R. Engler: EXE: Automatically Generating Inputs of Death. CCS'06.

https://www.doc.ic.ac.uk/~cristic/papers/exe-ccs-06.pdf

Fuzzing

- Christian Holler, Kim Herzig, and Andreas Zeller, Fuzzing with Code Fragments. USENIX Security 2012.

https://www.usenix.org/system/files/conference/usenixsecurity12/sec12-final73.pdf

- Rebert, Alexandre; Cha, Sang Kil; Avgerinos, Thanassis; Foote, Jonathan; Warren, David; Grieco, Gustavo; Brumley, David (2014), Optimizing Seed Selection for Fuzzing, USENIX Security'14

https://www.usenix.org/system/files/conference/usenixsecurity14/sec14-paper-rebert.pdf

Test Generation

- C Pacheco, SK Lahiri, MD Ernst, and T Ball, Feedback-Directed Random Test Generation, ICSE'07

https://dl.acm.org/citation.cfm?id=1248841

- SaswatbAnand, Edmund K.bBurke, Tsong YuehbChen, JohnbClark, Myra B. Cohen, Wolfgang Grieskamp, Mark Harman, Mary Jean Harrold, Phil McMinn, An orchestrated survey of methodologies for automated software test case generation. Journal of Systems and Software, August 2013.

https://www.sciencedirect.com/science/article/pii/S0164121213000563?casa_token=_H2-sGRiO5kAAAAA:gbA7COmOlf7bJu2SI2EpBKwg9AwpheC99RmJn8PbIuk7gLys6OesLrwJrKlZdzLd-5btA41d

Concolic Testing

- Koushik Sen, Darko Marinov, and Gul Agha, CUTE: a concolic unit testing engine for C, FSE'05

https://dl.acm.org/citation.cfm?id=1081750

Statement on Academic Conduct and Support Systems

Academic Integrity:

The University of Southern California is foremost a learning community committed to fostering successful scholars and researchers dedicated to the pursuit of knowledge and the transmission of ideas. Academic misconduct is in contrast to the university's mission to educate students through a broad array of first-rank academic, professional, and extracurricular programs and includes any act of dishonesty in the submission of academic work (either in draft or final form).

This course will follow the expectations for academic integrity as stated in the USC Student Handbook. All students are expected to submit assignments that are original work and prepared specifically for the course/section in this academic term. You may not submit work written by others or "recycle" work prepared for other courses without obtaining written permission from the instructor(s). Students suspected of engaging in academic misconduct will be reported to the Office of Academic Integrity.

Other violations of academic misconduct include, but are not limited to, cheating, plagiarism, fabrication (e.g., falsifying data), knowingly assisting others in acts of academic dishonesty, and any act that gains or is intended to gain an unfair academic advantage.

The impact of academic dishonesty is far-reaching and is considered a serious offense against the university and could result in outcomes such as failure on the assignment, failure in the course, suspension, or even expulsion from the university.

For more information about academic integrity see the student handbook or the Office of Academic Integrity's website, and university policies on Research and Scholarship Misconduct.

Course Content Distribution and Synchronous Session Recordings Policies:

USC has policies that prohibit recording and distribution of any synchronous and asynchronous course content outside of the learning environment.

Recording a university class without the express permission of the instructor and announcement to the class, or unless conducted pursuant to an Office of Student Accessibility Services (OSAS) accommodation. Recording can inhibit free discussion in the future, and thus infringe on the academic freedom of other students as well as the instructor. (Living our Unifying Values: The USC Student Handbook, page 13).

Distribution or use of notes, recordings, exams, or other intellectual property, based on university classes or lectures without the express permission of the instructor for purposes other than individual or group study. This includes but is not limited to providing materials for distribution by services publishing course materials. This restriction on unauthorized use also applies to all information, which had been distributed to students or in any way had been displayed for use in relationship to the class, whether obtained in class, via email, on the internet, or via any other media. (Living our Unifying Values: The USC Student Handbook, page 13).

Students and Disability Accommodations:

USC welcomes students with disabilities into all of the University's educational programs. The Office of Student Accessibility Services (OSAS) is responsible for the determination of appropriate accommodations for students who encounter disability-related barriers. Once a student has completed the OSAS process (registration, initial appointment, and submitted documentation) and accommodations are determined to be reasonable and appropriate, a Letter of Accommodation (LOA) will be available to generate for each course. The LOA must be given to each course instructor by the student and followed up with a discussion. This should be done as early in the semester as possible as accommodations are not retroactive. More information can be found at osas.usc.edu. You may contact OSAS at (213) 740-0776 or via email at osasfrontdesk@usc.edu.

Support Systems:

Counseling and Mental Health - (213) 740-9355 – 24/7 on call
Free and confidential mental health treatment for students, including short-term psychotherapy, group counseling, stress fitness workshops, and crisis intervention.

988 Suicide and Crisis Lifeline - 988 for both calls and text messages – 24/7 on call
The 988 Suicide and Crisis Lifeline (formerly known as the National Suicide Prevention Lifeline) provides free and confidential emotional support to people in suicidal crisis or emotional distress 24 hours a day, 7 days a week, across the United States. The Lifeline is comprised of a national network of over 200 local crisis centers, combining custom local care and resources with national standards and best practices. The new, shorter phone number makes it easier for people to remember and access mental health crisis services (though the previous 1 (800) 273-8255 number will continue to function indefinitely) and represents a continued commitment to those in crisis.

Relationship and Sexual Violence Prevention Services (RSVP) - (213) 740-9355(WELL) – 24/7 on call
Free and confidential therapy services, workshops, and training for situations related to gender- and power-based harm (including sexual assault, intimate partner violence, and stalking).

Office for Equity, Equal Opportunity, and Title IX (EEO-TIX) - (213) 740-5086
Information about how to get help or help someone affected by harassment or discrimination, rights of protected classes, reporting options, and additional resources for students, faculty, staff, visitors, and applicants.

Reporting Incidents of Bias or Harassment - (213) 740-5086 or (213) 821-8298
Avenue to report incidents of bias, hate crimes, and microaggressions to the Office for Equity, Equal Opportunity, and Title for appropriate investigation, supportive measures, and response.

The Office of Student Accessibility Services (OSAS) - (213) 740-0776

OSAS ensures equal access for students with disabilities through providing academic accommodations and auxiliary aids in accordance with federal laws and university policy.

USC Campus Support and Intervention - (213) 740-0411
Assists students and families in resolving complex personal, financial, and academic issues adversely affecting their success as a student.

Diversity, Equity and Inclusion - (213) 740-2101
Information on events, programs and training, the Provost's Diversity and Inclusion Council, Diversity Liaisons for each academic school, chronology, participation, and various resources for students.

USC Emergency - UPC: (213) 740-4321, HSC: (323) 442-1000 – 24/7 on call
Emergency assistance and avenue to report a crime. Latest updates regarding safety, including ways in which instruction will be continued if an officially declared emergency makes travel to campus infeasible.

USC Department of Public Safety - UPC: (213) 740-6000, HSC: (323) 442-1200 – 24/7 on call
Non-emergency assistance or information.

Office of the Ombuds - (213) 821-9556 (UPC) / (323-442-0382 (HSC)
A safe and confidential place to share your USC-related issues with a University Ombuds who will work with you to explore options or paths to manage your concern.

Occupational Therapy Faculty Practice - (323) 442-2850 or otfp@med.usc.edu
Confidential Lifestyle Redesign services for USC students to support health promoting habits and routines that enhance quality of life and academic performance.