Konrad Biegaj & Vineet Patel
CS: 568 | Fall 2020
Project Update II:

Completed Milestones:

- (10/16) We plan to deploy a functional web app using Web assembly, originally written in a language like C.

- (10/23) We will identify an open source library in C or C++ and incorporate them into example applications or find some open source programs in C or C++.

Status Update:

We have reviewed common and practical examples of open source C libraries from this common source library. Redis, an in-memory database, was an interesting example for review because being able to manipulate data through the stack when making calls has real world implications if this library was deployed as a WASM.

We have completed the two above milestones and implemented some automation to allow us to cover more native libraries. We wrote a script that takes the initial findings from the WASM crash course and manual analysis and scripted the steps needed to build a WASM site hosted locally and log all instructions and their inputs/results with the use of wasabi.

Ideally we will use this automation to increase our data collection and build a pattern for instruction sets that are prone to breakage. Being able to show a vulnerability in open source C is cool but not novel. We should be able to use our findings to identify common no-no's when porting native code with WASM.

We have started writing the components of the final paper such as defining the abstract and establishing the current limitation of using open source which is not known for its security. Additionally, obtaining access to live C source code is its own endeavour.

Upcoming Milestones:

(10/30) After identifying an app or app group we will start working more in-depth with Wasabi to try to detect the vulnerabilities. We will attempt to find a way to programmatically probe or scan the attack surface on any given C library when compiled into WASM.

(11/06) Continue writing the final paper based on updated findings.