Hmmmmm

Nicolai Nebel Jørgensen - dqz439

January 2, 2019

Project description

WebAssembly is a new, binary instruction format for a virtual machine, intended to be used on the web for both client and server applications. Because it is a binary format resembling normal assembly, it is well suited as a compiler target for other languages. WebAssembly is also interesting in a theoretical context: It already has a well-specified formal semantics, making it well-suited for reasoning about in a mechanical manner.

Since WebAssembly does not, in its core, specify any web-specific behaviour it is not limited to just being supported inside of web browsers. Nebulet[**?**] is a Google Summer of Code project for creating a simulated WebAssembly usermode running in ring 0 in the Linux kernel.

Proof-carrying code is a technique for allowing safe execution of untrusted code. A piece of proof-carrying code consists of two parts: The program itself and a proof of its safety. Safety in this context means that the program obeys a set of safety rules defined as a prior. Then, a user who wants to run a program will first verify its safety according to the safety rules using a trusted, mechanical *proof validator*. When the program is proof-checked, the untrusted program can now be considered safe to execute.

In particular, proof-checking a program prior to execution allows greater freedom when executing it safely. For instance, if a program can be proved to only access memory within a certain range it is not also necessary to check that this happens at *runtime*. Using this technique, untrusted code can be executed more efficiently than otherwise safe.

I wish to explore applications of proof-carrying code in relation to WebAssembly. To name a couple, this project will involve exploring the possible gains of proof-carrying techniques, the formulation of useful security policies code as well as the implementation of said techniques. I wish to evaluate the usefulness and applicability of proof-carrying code techniques to WebAssembly in different domains. This assessment will be informed both by theoretic exploration and practical implementation of the technique.

Learning Objectives

1. Explain the principles of Proof-carrying code techniques.
2. Explain the semantics of WebAssembly and the structure of its virtual machine.
3. Exmaine applications of proof-carrying code in WebAssembly.
4. Analyze security policies and identify the guarantees they provide to a run-time system.
5. Evaluate advantages and drawbacks of applying proof-carrying code techniques to WebAssembly in different domains.

6. Implement proof validator and proof-generating compiler for a small, functional language