Stages OCamlPro 2023

Fuzzing d'interprète Webassembly

Niveau requis : M1 ou M2

Durée du stage : 3 à 6 mois

Contexte du stage

WebAssembly (ou WASM) est un langage se voulant être une cible de compilation sûre, rapide et portable, tout en étant doté d'une sémantique clairement définie et permettant une interopérabilité avec les navigateurs webs (Bringing the Web up to Speed with WebAssembly).

Afin de garantir la portabilité, le langage a été formalisé, sa spécification est précise et complète. En conséquence, elle ne laisse place à aucun comportement indéfini. Chaque cas problématique met donc fin à la validation ou à l'exécution du programme. C'est une propriété importante car il est attendu que les programmes compilés vers WASM aient à s'exécuter dans un contexte où les utilisateurs ne peuvent pas faire confiance au programme exécuté - notamment des pages web.

Cet objectif de sûreté est en tension avec l'autre objectif de WASM : être un langage bas niveau, efficace et ayant des performances aussi proches que possible de code natif, par exemple en étant une cible de compilation du C.

Dans sa version actuelle, le langage est relativement minimal et simple à appréhender. Il existe de nombreuses implémentations d'interprètes (interprète de référence, interprète développé à OCamlPro, interprètes de Chrome et Firefox...). Il est aussi attendu que les navigateurs utilisent des compilateurs *Just In Time* (JIT) optimisants, qui sont des logiciels complexes.

Pour s'assurer de la correction et de la compatibilité entre ces interprètes, il faudrait développer des outils permettant de les tester intensivement.

Sujet du stage

Une technique permettant de mener des tests intensifs est le *fuzzing*. Cela consiste en l'implémentation d'un *fuzzer*, à savoir un programme qui va :

- générer des programmes WASM (valides et invalides),
- les exécuter avec deux interpréteurs,
- comparer les résultats,
- signaler les cas où les résultats diffèrent.

Lorsqu'une différence est trouvée, le programme à l'origine de celle-ci est souvent très gros. Il peut alors être intéressant de chercher à le minimiser automatiquement tout en conservant son comportement observé pour faciliter la recherche du bug dans les interprètes - on parle parfois de *delta debugging*.

Un autre aspect qu'il est possible d'étudier est la comparaison des performances des interprètes. En effet, il est attendu que l'exécution d'un même programme WASM sur différentes implémentations ait un rapport de performance de l'ordre de la constante - et qu'il ne dépende que très peu du programme.

Stages OCamlPro 2023

Compétences requises

- Notions de compilation
- Programmation en OCaml
- Techniques de tests automatiques

Compétences acquises

- Sémantique de WASM
- Techniques de fuzzing

Lieu du stage

Le stage se déroulera dans les locaux d'OCamlPro : 21, rue de Châtillon 75014 Paris.

Contact

Léo Andrès, leo.andres@ocamlpro.com / Pierre Chambart, pierre.chambart@ocamlpro.com

À propos d'OCamlPro

OCamlPro SAS est une société issue de l'INRIA, créée en avril 2011, pour promouvoir l'utilisation de langages de programmation à l'état de l'art tels qu'OCaml et Rust dans le milieu industriel. Elle participe activement à des programmes de recherche et de développement visant à améliorer la sûreté et la sécurité des applications informatiques en général. Plus d'informations sur notre site web : https://www.ocamlpro.com/