**An Automated Analysis of Safety and Security in the Drake System**

This document presents the results of running Petablox and other state-of-the-art automated static analysis tools on the Drake codebase to uncover safety and security issues. We present analytics results for various Drake modules, describe potential issues we discovered, provide pointers to raw output of the tools, and outline future directions to improve the tools.

**The Checkers**

We ran various checkers in four different automated static analysis tools on the Drake codebase obtained from <http://drake.mit.edu/from_source.html>

1. Petablox: API misuse checkers (<http://www.seas.upenn.edu/~mhnaik/pubs/sec16.pdf>)

2. FB Infer: Memory safety checkers ([http://FBinfer.com](http://fbinfer.com))

3. MIT Kint: Integer overflow checker ([http://css.csail.mit.edu/kint](http://css.csail.mit.edu/kint/))

4. Coverity: bug pattern checkers ([https://scan.coverity.com](https://scan.coverity.com/))

All of the above tools except Coverity are freely available. Coverity provides a free service for github hosted projects (such as Drake); its results of analyzing Drake will be available in 48 hours.

**Checker Analytics**

|  |  |  |  |
| --- | --- | --- | --- |
| Modules | Petablox | FBInfer | Kint |
| drake/automotive | 0 | 22 |  |
| drake/multibody | 559 | 1 |  |
| drake/systems | 1203 | 13 |  |
| drake/common | 1008 | 2 |  |
| externals/libbot | 38 | 108 |  |
| externals/ipopt | 13 | 70 |  |
| others |  |  |  |

**Analysis Results**

Our analysis of potential safety/security issues discovered in Drake by the tools is available in files under directory ‘analysis\_results’.

**Detailed Output**

The raw output of running the tools on Drake is available in folders under directory ‘raw\_logs’.