LJGS: Gradual Security Types for Object-Oriented Languages (Artifact)

Luminous Fennell[1] and Peter Thiemann[2]

1 **University of Freiburg**
Georeges-Köhler Allee 79, 79110 Freiburg, Germany
fennell@informatik.uni-freiburg.de

2 **University of Freiburg**
Georeges-Köhler Allee 79, 79110 Freiburg, Germany
thiemann@informatik.uni-freiburg.de

Abstract

JGS-check is the accompanying artifact to "LJGS: Gradual Security Types for Object-Oriented Languages". LJGS is a Java-like language with gradual security typing. It features a constraint based information flow type system that includes a type dynamic and type casts. Dynamically typed fragments are liberally accepted by the type checker and rely on run-time enforcement for security. JGS-check is a type checker for the subset of Java that corresponds to the calculus presented in the paper and that implements the constraint generation and satisfiability checks of LJGS' type system. It's purpose is to illustrate and substantiate the behavior of our gradual security type system. It takes a directory of Java source code as input and reports methods that violate the typing rules. JGS-check is merely a type checker and does not implement code generation.

The submission archive includes the compiled type checker, the code of the example section (Section 2) as well additional examples and testcases that did not fit into the paper. The user should also be able to check custom code as long as it corresponds the subset of Java that is covered by LJGS.

1998 ACM Subject Classification D4.6. Security and Protection

Keywords and phrases gradual typing, security typing, Java, hybrid information flow control

Digital Object Identifier 10.4230/DARTS.2.1.4

Related Article Luminous Fennell and Peter Thiemann, "LJGS: Gradual Security Types for Object-Oriented Languages", in Proceedings of the 30th European Conference on Object-Oriented Programming (ECOOP 2016), LIPIcs, Vol. 56, pp. 9:1–9:26, 2016.
http://dx.doi.org/10.4230/LIPIcs.ECOOP.2016.9

Related Conference 30th European Conference on Object-Oriented Programming (ECOOP 2016), July 18–22, 2016, Rome, Italy

1 Scope

JGS-check implements the constraint generation and satisfiability checks of LJGS' type system. It supports the subset of Java that corresponds to the calculus presented in the paper. JGS-check is merely a type checker and does *not* implement code generation. It's purpose is to illustrate and substantiate the behavior of our gradual security type system and allow readers to experiment with the type system and to get a feeling for the possibilities and restrictions of gradually type-checking code.

2 Content

The artifact package includes:

- a VirtualBox disk image (`JGS-check-ecoop2016`) including
  - a jar-file with class- and source-files of JGS-check (`~/Desktop/ecoop2016-ae-submission/GradualConstraints.jar`)
  - support classes required to annotated programs for JGS-check (`~/Desktop/ecoop2016-ae-submission/JGSSupport`)
  - annotated example programs (including the example code from the paper) that JGS-check can check. (`~/Desktop/ecoop2016-ae-submission/JGSTestclasses`)
  - a Makefile for running JGS-check on the examples (`~/Desktop/ecoop2016-ae-submission/Makefile`)
- detailed instructions for using the artifact, provided as an `index.html` file.

3 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS). The latest version of our code is available on github: `https://github.com/luminousfennell/gradual-java/tree/ecoop2016-ae/GradualConstraints` .

4 Tested platforms

The artifact is known to work on Oracle VirtualBox version 5 (`https://www.virtualbox.org/`) on an Intel Core i7 Mac Book Pro with 8 GB of RAM running OS X Yosemite. It should also work on other x64 systems that are able to run VirtualBox. The size of the VM image is 1GB.

5 License

BSD-3 (`https://opensource.org/licenses/BSD-3-Clause`)

6 MD5 sum of the artifact

8a2c4fd720bafe998aa000ed65e29d81

7 Size of the artifact

580 MB