

Master / Bachelor

Statische Sicherheitsanalyse von Programmen mit Java-Reflection Static Security-Analysis of Programs using Java-relection

This thesis can be supervised and written in German or English.

Motivation — Deutsch

92,6% der top 202 Android-Apps verwenden Jingling Xue zufolge Java-Reflection. Java-Reflection ist eine sehr mächtige API, die es Entwicklern ermöglicht Typen von Objekten zur Laufzeit zu untersuchen, zusätzliche Klassen zur Laufzeit nachzuladen sowie den Aufruf und Zugriff auf als privat deklarierte Klassenmember. In den meisten Fällen basieren diese Funktionalitäten auf Laufzeitinformationen und deren Auswirkungen sind in statischen Analysen unentscheidbar. Da die Verwendung von Java-Reflection ein inkrementaler Bestandteil von sicherheitskritischer aber auch schädlicher Software ist, ist es wichtig Java-Reflection bei der statischen Analyse von Software zu berücksichtigen.

Statische Programmanalysen werden von uns mittels des in Abbildung 1 dargestellten GRaViTY-Frameworks durchgeführt. Das GRaViTY-Framework kann über eine Transformation aus Java-Sourcecode ein leicht zu analysierendes Programmmodell erstellen und darauf verschiedene Analysen durchführen.

Das Ziel dieser Arbeit ist es das GRaViTY-Framework um eine Sicherheitsanalyse, welche Java-Reflection berücksichtigt zu erweitern.

Mögliche Teilaufgaben sind dabei:

- Erarbeitung eines Ansatzes zur Auflösung von möglichen Auswirkungen der Verwendung von Java-Reflection.
- Erweiterung der Transformation von Java-Sourcecode zu dem Programmmodell um die Java-reflection Analyse
- Spezifikation und Implementierung einer Sicherheitsanalyse, welche die zusätzlichen Informationen aus der Java-Reflection-Analyse verwendet.

Motivation — English

According to Jingling Xue, 92.6% of the top 202 Google Play Android apps are using Java-Reflection. Java-Reflection is a very powerful API and enables developers to inspect the types of objects, to load new classes at runtime as well as the invocation or access of private class members. In most cases these functionalities are based on runtime information and their effects are not statically decidable. As the use of Java-Reflection is an incremental part of security critical as well as malicious software it is important to take the use of Java-Reflection at static analysis of software into account.

We perform static program analyses with the GRaViTY framework shown in figure 1. The GRaViTY framework is able to create an easy to analyze program model from Java source code by using a transformation and to perform multiple analyses on it.

- Development of an approach to resolve effects of the usages of Java-reflection.
 - Extension of the transformation from Java source code into the program model with this Java-Reflection analysis
 - Specification and implementation of a security analysis using the new information of the Java-Reflection analysis.
-

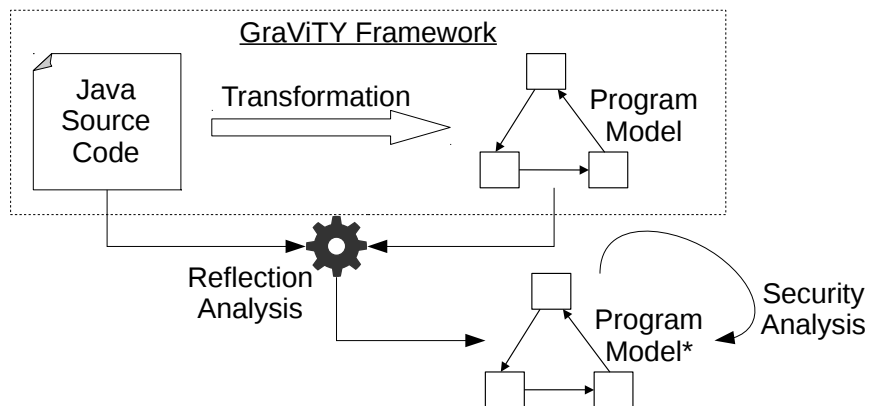


Abbildung 1: Integration der Java-Reflection Analyse in das GRaViTY-Framework / Integration of the Java-Reflection analysis into the GRaViTY framework

Organisatorisches

Kontakt:

M.Sc. Sven Peldszus (speldszus@uni-koblenz.de)