Testen von Informationsflusseigenschaften

Es ist wünschenswert, dass Programme sensible Informationen wie Kontodaten oder Passwörter nicht veröffentlichen. Um dies sicherzustellen, kann man mit Hilfe von Verifikationswerkzeugen die Datenflüsse eines Programms analysieren und zeigen, dass keine sicherheitsrelevante Informationen geleckt werden. Die Eingaben und Ausgaben des Programms werden in zwei Kategorien eingeteilt – vertraulich (high) und öffentlich (low). Man muss dann zeigen, dass die öffentlichen Ausgaben unabhängig von den vertraulichen Eingaben sind, also bei gleichen öffentlichen Eingaben müssen gleiche öffentliche Ausgaben herauskommen, unabhängig von den vertraulichen Eingaben. Diese Eigenschaft (non-interference) spricht also über zwei beliebige Programmabläufe und ist deswegen schwieriger zu beweisen als funktionale Eigenschaften, die nur einen Programmablauf betrachten.
Da das Beweisen dieser Eigenschaften oft zu kompliziert ist, möchten wir sie testen. Dafür müssen ausführbare Testfälle aus einem Teilbeweis generiert werden. Das Testen erfolgt automatisch und ist somit einfacher als das Beweisen, wo interaktive Schritte nötig sein können. Allerdings kann ein Test die gewünschte Eigenschaft nur für die beim Testen verwendete Eingaben zeigen. Daher müssen die Testeingaben so ausgewählt werden, dass sie das Programmverhalten so weit wie möglich abdecken.

Mögliche Aufgaben im Rahmen dieses Themas sind:

- Definieren des Konzepts eines Informationsfluss-Testfalls
- Definieren von Abdeckungskriterien für Informationsfluss-Testfälle - wie aussagekräftig ist eine Testsuite?
- Erweiterung des Testfallgenerierung-Frameworks in KeY, so dass neben funktionalen jetzt auch Informationsflusseigenschaften getestet werden können
- Exploit-Generation - statt eine Testsuite zu generieren, die möglichst viel abdeckt, gezielt nach Inputs suchen, die die Eigenschaft verletzen und Testfälle nur für diese Inputs erzeugen. Die generierte Testsuite soll einen Angreifer simulieren, der die öffentliche Eingabe beliebig setzen kann, um Informationen über die vertrauliche Eingabe zu gewinnen.

Kontakt / Betreuer: Mihai Herda (ITI Beckert) herda@kit.edu