Secure Contracts

David Poxon

October 17, 2017

Abstract

A program is sound with respect to a policy of noninterference if any secret inputs do not influence public outputs. The field has a long history of study that has focused predominantly on the analysis of whole programs with a defined entry point. However, modern software engineering practice regularly incorporates the use of components, which are partial programs with no strict entry point. We present Secure Contracts, a new programming paradigm and model of analysis. Taking design cues from 'design by contract' and software interfaces, Secure Contracts allows imperative code to be used to define the 'legal' data flow behaviours of a component. The analysis model has two main functions with respect to this: 1) the model is able to determine if a component implementation is valid with respect to the Secure Contract it implements; 2) taking as input a program, made up of a Secure Contract and a client with an entry point, the model can determine if a coupling of the client with an implementation of the Secure Contract will be sound with respect to noninterference. We discuss the initial work on Secure Contracts, along with the beginnings of a formal proof, and provide a demonstration.