Статический масштабируемый межпроцедурный анализ помеченных данных

Кошелев Владимир Константинович

Аспирант

Факультет ВМК МГУ имени М. В. Ломоносова, Москва, Россия

E-mail: vedun@ispras.ru

В настоящее время пользователи мобильных устройств зачастую используют их для хранения персональных и конфиденциальных данных. В связи с этим злонамеренные разработчики размещают в магазинах приложений вредоносные программы с целью получить пользовательские данные. Таким образом, перед владельцами магазина приложений стоит задача проверки загруженной программы на вредоносность.

Задача отслеживания утечки пользовательских данных как правило решается с помощью анализа помеченных данных. Перед началом анализа выделяются два множества системных функций, называемых "истоками"и "стоками". Изначально помеченными считаются данные, полученные посредством вызова функции из множества "истоков". Далее помеченные данные определяются как зависящие по данным (и в некоторых случаях по управлению) от уже помеченных. Задачей анализа является выявление того факта, что помеченные данные были переданы в функции из множества "стоки".

Стоит отметить, что данная задача может решаться как статически[1], так и динамически[2]. Динамический анализ отличается от статического большей точностью результата, однако он способен проанализировать лишь конечный набор входных данных. Статический анализ способен проанализировать сразу все пути выполнения программы, однако при таком анализе как правило игнорируются условия на переходы.

В данной работе будет рассмотрен статический масштабируемый потоково-, контекстно-, полечувствительный анализ помеченных данных, использующий в качестве входных данных LLVM биткод. Мотивацией к использованию LLVM биткода послужил тот факт, что в новой мобильной платформе Tizen в качестве представления для нативных приложений используется именно он.

Предлагаемый подход схож с подходом разработчиков системы поиска утечек данных в приложениях для Android — FlowDroid[3]. Авторы FlowDroid сводят задачу анализа помеченных данных к

потоково-,контекстно- чувствительной межпроцедурной задаче анализа потока данных. При этом фактами для анализа потока данных являются помеченные поля Java-объектов. Для того, чтобы потоковая и контекстная чувствительность исходного анализа имела смысл, анализ псевдонимов также должен быть потоково и конекстно чувствительным. Поэтому для анализа псевдонимов в системе FlowDroid используется тот же потоково-,контекстно- чувствительный межпроцедурный анализ потока данных.

По сравнению с результатами разработчиков FlowDroid в алгоритм были внесены следующие изменения и улучшения:

- Переработана концепция фактов анализа потоков данных с учётом адресной арифметики в LLVM биткоде.
- Разработана концепция, использующая def-use цепочки, позволяющая достичь линейного роста времени работы алгоритма относительно размера анализируемой программы.
- Переработан алгоритм решения межпроцедурной задачи потока данных для возможности одновременного анализа независимых функций.

На основе предложенного алгоритма был разработан и реализован прототип на базе LLVM. Данный прототип в настоящий момент проходит стадию тестирования на реальных приложениях. В будующем планируется улучшение поддержки массивов и рекурсивных типов данных.

Литература

1. Myers A. C. JFlow: Practical Mostly-Static Information Flow Control// In Proceedings of the 16th ACM Symposium on Principles of Programming Languages, pages 228–241, 1999.
2. Clause J., Li W., Orso A. Dytan: a generic dynamic taint analysis framework// Proceedings of the 2007 international symposium on Software testing and analysis. Pages 196 - 206 New York, NY, USA 2007
3. Fritz C., Arzt S., Rasthofer S., Bodden E., Bartel A., Klein J., Traon Y., Octeau D., McDaniel P. Highly Precise Taint Analysis for Android Applications// EC SPRIDE Technical Report TUD-CS-2013-0113, May 2013