И. В. Арзамарцев, А. В. Моисеев

СТАТИЧЕСКИЙ АНАЛИЗ СЛОЖНЫХ ЦИКЛИЧЕСКИХ КОНСТРУКЦИЙ

В настоящее время постоянно увеличивается количество программного обеспечения, выполняющего различные прикладные задачи, которое интегрируется в сложные вычислительные системы. С увеличением сложности данных систем повышается число разнообразных уязвимостей, которые в нем содержатся, что в ряде случаев позволяет нарушителю получить полный контроль над системой.

Для сокращения потенциального числа уязвимостей необходимы методы анализа как исходного, так и двоичного кода ПО, которые позволят снизить временные затраты на нахождение и последующее устранение данных свойств ПО.

Одним из распространенных подходов к проведению такого типа исследования ПО является статический анализ [1], который основан на изучении исполнимого кода без его реального исполнения. Весьма распространенными методом данного анализа является символьное исполнение.

Символьное исполнение (от англ. symbolic execution) — метод анализа потоков данных, основанный на построении алгебраических уравнений, описывающих входные и выходные параметры некоторого функционального блока исследуемого ПО путем подстановки вместо реальных значений входных данных переменных величин и исполнения программы [2].

Такой метод, несмотря на обеспечение обобщенной возможности анализа ПО, не позволяет эффективно осуществить анализ сложных циклических конструкций с большим либо неопределенным числом итераций.

Для преодоления данного ограничения предлагается использовать:

- функции ветвления;
- обобщенное представление данных;
- выявление зависимостей между условиями и операциями над данными;
- склеивание потока данных в итерациях цикла.

Использование функций ветвления позволяет обобщить поток данных при использовании программных инструкций переходов в зависимости от данных, которые должны удовлетворять определенным условиям, и от операций, которые над данными осуществляются в зависимости от перехода на ту или иную ветвь исполнения.

Обобщенное представление данных предназначено для описания сложных структур данных, таких как списки различной связности, деревья и т. п. Анализ использования зависимостей между элементами данных даёт возможность выделить операции обращения к одним элементам данных на основе информации о других.

Выявление зависимостей между условиями и операциями над данными иногда позволяет выявить ограничения на результат исполнения. Например, в случае некоторых алгоритмов сортировки применяется сравнение отдельных элементов массива данных, после чего в зависимости от выполнения или невыполнения этого условия осуществляется перестановка значений данных элементов таким образом, что они оказываются в некотором упорядоченном состоянии. Такую взаимосвязь с помощью склеивания потока данных в итерациях цикла можно применить и к циклическим конструкциям.

Склеивание потока данных в итерациях цикла предназначено для выявления взаимосвязей между соседними итерациями циклических конструкций. Такая операция направлена на выявление изменений, которые могут быть осуществлены над данными внутри цикла, после чего производится продление изменений с учетом следующей итерации в цикле. Это позволит выявить возможность монотонного изменения данных и диапазоны их допустимых значений, а также нахождение множества дискретных значений, которые могут принимать данные в результате работы указанного цикла.

Предложенные методы анализа алгоритмов способствуют улучшению существующих методов анализа программного обеспечения, таких как символьное исполнение и использование помеченных данных [3], но не позволяют провести полноценный анализ сложных программных выражений, например анализ вычисления однонаправленных функций. Дальнейшая работа будет направлена на исследование потока данных в сложных циклических конструкциях и на формальное доказательство возможности или невозможности исполнения определенной ветви кода программы.

СПИСОК ЛИТЕРАТУРЫ:

1. *Ахо А. В., Моника С., Джефри Д.* Компиляторы: принципы, технологии и инструментарий. 2-е изд. Пер. с англ. М.: ООО «И.Д. Вильямс», 2008.
2. *King J. C.* Symbolic Execution and Program Testing // Communications of the ACM. 1976. V. 19. № 7. P. 385–394
3. *Moser A., Kruegel C., Kirda E.* Exploring Multiple Execution Paths for Malware Analysis // IEEE Symposium on Security and Privacy May 2007. P. 231–245

В. М. Барбашов, А. В. Зубаков, В. Г. Иваненко, Н. С. Трушкин

ИСПОЛЬЗОВАНИЕ ВЕРОЯТНОСТНЫХ И НЕЧЕТКИХ МОДЕЛЕЙ ДЛЯ ПРОГНОЗИРОВАНИЯ БЕЗОПАСНОЙ РАБОТЫ ЦИФРОВЫХ СИСТЕМ ПРИ ВОЗДЕЙСТВИИ РАДИАЦИИ

Создание сложных систем на основе больших интегральных схем (БИС), устойчивых к воздействию радиационных дестабилизирующих факторов, на сегодняшний день невозможно без активного использования логического моделирования, обеспечивающего необходимую адекватность описания и точность расчетов. Анализ радиационного поведения БИС показывает, что при нарушении работоспособности как по функциональным, так и по электрическим параметрам в ряде случаев характерен значительный статистический разброс пороговых уровней отказа для однотипных образцов. При этом реальный характер радиационного поведения сложной электронной системы определяется конкретным соотношением радиационно-чувствительных параметров ее элементов и учетом влияния их статистического разброса. Соотношение между функцией распределения плотности вероятности разброса и критериальной функцией принадлежности (КФП) определяет, в конечном итоге, целесообразность использования функционально-логических моделей радиационного поведения БИС применительно к каждому конкретному случаю [1]. Так, например, уменьшение дисперсии разброса порога отказа при облучении наблюдалось при объемных эффектах смещения в биполярных структурах, радиационная чувствительность которых определяется изменением времени жизни (рис. 1).