Статическая верификация модулей ядра Linux с учетом межмодульных связей

Полушкин Алексей Юрьевич

Студент

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

E-mail: polushkin@ispras.ru

Ядро операционной системы Linux является крупнейшим проектом с открытым исходным кодом. Исходный код модулей составляет большую часть исходного кода ядра. Модули должны отвечать высоким требованиям к надежности, поскольку работают в том же адресном пространстве и с тем же уровнем привилегий, что и остальное ядро [4]. Одним из способов проверки соответствия кода программы заявленным требованиям является статическая верификация. Инструменты статической верификации реализуют такие методы, как BMC [2], CEGAR [3].

Система Klever [5] предназначена для статической верификации модулей ядра Linux при помощи инструмента статической верификации CPAchecker [1]. Поскольку ядро ОС Linux является сложным для инструментов статической верификации, в системе Klever используется моделирование окружения [6] для обеспечения возможности анализа исходного кода модуля отдельно от исходного кода остального ядра. Модули могут реализовывать интерфейс ядра или экспортируемые функции для их использования в других модулях. Метод моделирования окружения в системе Klever не позволяет верифицировать модули вызывающие экспортируемые функции вместе с их реализациями, в результате чего возникают ложные сообщения об ошибках. Кроме того, не верифицируется код экспортируемых функций.

Для статической верификации модулей, реализующих или вызывающих экспортируемые функции, предлагается метод объединения модулей в группы для совместной статической верификации. Модули объединяются в группы в соответствии со связями по экспортируемым функциям. Пользователь может настроить максимальное число модулей в группе, выбрать модуль, все экспортируемые функции которого должны быть проверифицированы, или задать другие параметры. Для моделирования окружения для группы модулей потребовалось модифицировать алгоритм генерации моделей окружения, чтобы реализовать моделирование загрузки и выгрузки модулей из памяти.

Метод был реализован в системе статической верификации Klever. Для апробации данного метода было выбрано 66 модулей, которые верифицировались на соответствие 32 требованиям корректности. Было получено 3 ложных сообщения об ошибках из-за отсутствия исходного кода экспортируемых функций. Совместная верификация модулей позволила избавиться от соответствующих сообщений об ошибках и получить корректный вердикт.

Литература

1. Beyer D. Keremoglu M. E. CPACHECKER: a tool for configurable software verification // Proceedings of the 23rd International Conference on Computer Aided Verification, Berlin, Heidelberg, 2011, P. 184–190.
2. Clarke E. Biere A. Raimi R. Zhu Y. Bounded Model Checking Using Satisfiability Solving // Formal Methods in System Design, 2001, V. 19 № 1 P. 7–34
3. Clarke E. Grumberg O. Jha S. Lu Y. Veith H. Counterexample-Guided Abstraction Refinement // J. ACM, 2003, V. 50, № 5 P. 752–794.
4. Palix N. Thomas G. Saha S. Calvès C. Lawall J. Muller G. Faults in linux: ten years later // Proceedings of the 16th International Conference on Architectural Support for Programming Languages and Operating Systems, Newport Beach, California, USA, 2011, P. 305–318.
5. Zakharov I. S. Mandrykin M. U. Mutilin V. S. Novikov E M. Petrenko A K. Khoroshilov A. V. Configurable Toolset for Static Verification of Operating Systems Kernel Modules // Programming and Computer Software, 2015, V. 41 № 1 P. 49–64.
6. Zakharov I. S. Mutilin V. S. Khoroshilov A. V. Pattern-based environment modeling for static verification of Linux kernel modules // Programming and Computer Software, 2015, V. 41, № 3, P. 183—195.