УДК 004.056.53

М.А. ГРИШИН[1], И.Ю. КОРКИН[2]

[1]*Национальный исследовательский ядерный университет «МИФИ», Москва*

[2]*АО «Позитив Текнолоджиз», Москва*

ГИБРИДНАЯ СХЕМА НАПРАВЛЕННОГО ФАЗЗИНГ-ТЕСТИРОВАНИЯ ЯДРА LINUX С ИСПОЛЬЗОВАНИЕМ ИНСТРУМЕНТА SYZKALLER

Предложена новая гибридная схема фаззинг-тестирования, сочетающая статический анализ, профилирование и реализованных инструментов автоматизации. Данный подход в отличие от существующих обеспечил возможность реализации направленного фаззинга ядра Linux без необходимости многократного запуска syzkaller. Экспериментальная оценка показала увеличение покрытия кода и обнаружение большего числа уязвимостей по сравнению с классическим подходом за одинаковое время тестирования.

Существует множество подходов к фаззингу ядра операционной системы [1, 3]. Комбинированные методы или гибридные схемы [2–4, 8] сочетают элементы статического и динамического анализа. В данной работе предлагается способ улучшения схемы направленного фаззинга с использованием статического анализатора [3], а в качестве базового инструмента выбрано средство syzkaller [6], реализующее технологии мутационного и генерационного фаззинга [5].

Предлагаемая схема заключается в односеансовом использовании syzkaller с предварительной обработкой результатов статического анализа для определения целей и восстановления методом профилирования [7] последовательности системных вызовов. Данная последовательность составляет корпус начальных входных данных.

Авторами реализован программный комплекс (фаззинг-система) из трех модулей: FuzzController, SvFilter и SysFinder. FuzzController управляет процессом, инициируя сборку ядра и статический анализ, результаты которого сохраняются в csv-файл. SvFilter обрабатывает этот файл, выделяя функции ядра с критическим уровнем опасности, а SysFinder использует ftrace для записи трасс выполнения и определения последовательности системных вызовов. Утилита FuzzController инициирует сборку ядра с заранее подготовленной конфигурацией и запуск статического анализа.

В рамках исследования проводилось сравнение разработанной фаззинг-системы с популярной реализацией Syzkaller. Время тестирования составило 46 часов для каждого фаззера. По результатам работы статического анализатора были выделены потенциально опасные функциональные объекты, разделенные на группы по типу потенциальной уязвимости.

Итоги тестирования показали, что построенная фаззинг-схема позволила верифицировать большее число ошибок в исходных текстах по сравнению с классической реализацией, что отражено в табл. 1: было обнаружено 21 уязвимость, из которых 9 являются уникальными.

Таблица 1. Сравнение результатов верификации двумя фаззерами

Класс ошибок	**Количество ФО**	**Syzkaller (direct + FuzzController)**	**Syz-kaller**
Несоответствие размеров буфера при выделении памяти	28	6	4
Переполнение буфера на стеке	23	5	2
Разыменование нулевого указателя	20	2	0
Повторное освобождение памяти	30	3	2
Переполнение буфера на куче	5	4	3
Прочие ошибки при работе с памятью	19	1	0

Список литературы

1. Обзор различных средств фаззинга как инструментов динамического анализа программного обеспечения [Электронный ресурс] – 2018 – Режим доступа: https://moluch.ru/archive/186/47575/.

2. No Grammar, No Problem: Towards Fuzzing the Linux Kernel without System-Call Descriptions /Bulekov Alexander, Das Bandan, Hajnoczi Stefan. – 2023. – p. 16.

3. Егорова В.В., Панов А.С., Тележников В. Ю., Подходы, направленные на повышение эффективности фаззинг-тестирования компонентов защищенной ОС [Текст]//Труды ИСП РАН: Т.34, вып. 4 – 2022 г. – 21 – 34 с.

4. Dan Li, Hua Chen. KLEE: FastSyzkaller: Imporving Fuzz Efficiency for Linux Kernel Fuzzing // IOP Conference Series: Journal of Physics. – 2020 – P.30 – 38.

5. Бегаев А.Н., Кашин С.В. Выявление уязвимостей и недекларированных возможностей в программном обеспечении [Текст] / Университет ИТМО, 2020 – 38 с.

6. Syzkaller – kernel fuzzer. [Электронный ресурс] – Режим доступа: https://github.com/google/syzkaller.

7. Введение в ptrace [электронный ресурс] – 2018 – Режим доступа: https://habr.com/ru/post/430302/.

8. Maxim Grishin, Igor Korkin: Human-Controlled Fuzzing With AFL // Proceedings of the 15th Annual ADFSL 2022 Conference on Digital Forensics, Security and Law, Florida, USA, July 25, 2022, https://commons.erau.edu/adfsl/2022/presentations/3/