Статический анализ исходного кода библиотек для поиска незадекларированных возможностей

Харис Лаврентий Александрович

Студент

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

E-mail: harisL@yandex.ru

Научный руководитель *— Самосадный Кирилл Алексеевич*

Задача поиска незадекларированных возможностей в проектах с открытым исходным кодом в репозиториях библиотек является одной из важнейших в сфере информационной безопасности. По данным из отчёта Sonatype, за 2025 год было найдено 454648 проектов, содержащих вредоносный код. При этом для 35% пакетов потребовалось более 3 месяцев для обнаружения [1].

Существующие решения задачи поиска незадекларированных возможностей используют методы машинного обучения, поиск по метаданным и сигнатурный анализ. В работах [2-4] указаны следующие недостатки существующих решений:

- Методы, основанные на машинном обучении, требуют большой тренировочной выборки для поиска вредоносных проектов, которую может быть невозможно собрать в силу природы предметной области поиска вредоносного ПО [2];
- Методы, основанные на сигнатурном анализе, используют чёрные списки конкретных сигнатур, из-за чего оказываются слепы к копированию и мутациям библиотек [3];
- Методы, основанные на поиске аномалий в метаданных проектов, имеют большую долю ложноположительных срабатываний [4].

Для решения описанной проблемы методов, основанных на сигнатурном анализе, было решено проверить гипотезу, согласно которой в репозиториях программных компонент существуют библиотеки, имитирующие базовые методы языка, часто использующиеся для совершения атак. Поскольку в сигнатурах указываются конкретные методы и структуры кода, такие клоны библиотек будут невидимы для существующих методов сигнатурного анализа.

Для поиска специфических данных используется анализ помеченных данных. Метод заключается в том, что составляется набор пар из чувствительных данных и соответствующих им функций. При анализе кода библиотеки рассчитывается, могут ли чувствительные данные попасть в поставленные им в пару методы или функции в ходе исполнения программы. Если есть путь исполнения программы, при котором чувствительные данные попадают в парные им функции, то фрагмент кода считается подходящим под искомые параметры. При нахождении таким методом клонов библиотек они используются для обогащения пар для поиска вредоносного кода.

В ходе тестирования были найдены искомые библиотеки, которые в данный момент используются для поиска вредоносных компонент с открытым исходным кодом.

Литература

1. Отчёт компании Sonatype «2026 State of the Software Supply Chain»: www.sonatype.com/state-of-the-software-supply-chain/
2. Zahan N. et al. Leveraging large language models to detect npm malicious packages //2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE). – IEEE, 2025. – С. 2625-2637.
3. Gibert D. et al. Assessing the impact of packing on static machine learning-based malware detection and classification systems //Computers & Security. – 2025. – Т. 156. – С. 104495.
4. Mehedi S. T. et al. DySec: a machine learning-based dynamic analysis for detecting malicious packages in PyPI ecosystem //IEEE Transactions on Information Forensics and Security. – 2026.