рывной оценки «здоровья» оборудования и прогнозирования его будущего состояния. Это ключевой элемент концепции «Цифрового двойника» (Digital Twin) и Индустрии 4.0.

Таким образом, инженер-механик будущего, специалист по неразрушающему контролю – это не только человек с датчиками и приборами, но и оператор, который работает с AI-системами, интерпретирует их выводы и принимает финальные ответственные решения.

УДК 004.051

КРИТЕРИИ ЭФФЕКТИВНОСТИ АЛГОРИТМОВ ОБФУСКАЦИИ И ДЕОБФУСКАЦИИ ПРОГРАММНОГО КОДА И ИХ СРАВНИТЕЛЬНЫЙ АНАЛИЗ НА ОСНОВЕ ОНЛАЙН-ОБФУСКАТОРОВ

В.С. КАНТАРОВИЧ, П.П. УРБАНОВИЧ
Белорусский государственный технологический университет
Минск, Беларусь

Растёт количество случаев несанкционированного копирования и использования интеллектуальной собственности, что создает необходимость в эффективных механизмах защиты уникальных алгоритмов и логики работы программ. Одним из таких важных инструментов в сфере информационной безопасности является обфускация, которая направлена на защиту интеллектуальной собственности и предотвращение несанкционированного доступа к исходному коду. Процесс обфускации особенно важен в контексте программного обеспечения и безопасности беспилотных аппаратов, поскольку он не только защищает внутреннюю логику и структуру кода, затрудняя анализ и реверс-инжиниринг, но и служит дополнительным слоем защиты от кибератак. В условиях растущих угроз обфускация минимизирует вероятность успешного обнаружения и эксплуатации уязвимостей, что является важным для надежности и устойчивости беспилотных систем. Таким образом, применение обфускации является необходимым шагом для повышения безопасности программного обеспечения и защиты от несанкционированного использования кода [1].

Обфускация – это процесс преднамеренного усложнения исходного кода программы и внесение в него модификаций с целью затруднения его анализа и понимания. Оригинальные алгоритмы обфускации

могут значительно усложнить анализ и декомпиляцию, так как современные деобфускаторы эффективно справляются с устаревшими методами.

Эффективность алгоритмов обфускации и их уровень защиты программного кода может зависеть от нескольких факторов, таких как: сложность исходного кода (определяется количеством строк, структурой, количеством условий, циклов и вложенных конструкций); язык программирования, так как разные языки имеют свои особенности и средства для реализации методов обфускации; методы обфускации; среда выполнения программного кода (операционная система, аппаратное обеспечение и контекст выполнения, также влияют на эффективность обфускации и производительность обфусцированного исходного кода). Исходя из этих факторов можно выбрать наиболее подходящие алгоритмы обфускации [2, 3].

Онлайн-обфускаторы кода поддерживают широкий спектр языков программирования, включая как интерпретируемые (JavaScript, PHP, Python), так и компилируемые (Java, C#). Особую популярность имеют онлайн-обфускаторы для JavaScript, поскольку выполнение этого кода на стороне клиента делает исходный код доступным пользователю в незащищенном виде, что подчеркивает необходимость его защиты от несанкционированного доступа и реверсивной инженерии.

Оценка эффективности обфускаторов кода может включать несколько критериев:

– устойчивость (Resistance) – используется для описания уровня сложности реализации реверсивной инженерии над программой после обфускации кода;

– читабельность (Readability) – характеризует степень возможности анализа и понимания кода для злоумышленников;

– сохранение функциональности (Functionality Preservation) – характеризует влияние обфускации на работоспособность программы;

– сложность декомпиляции (Decompilation Difficulty) – характеризует сложность восстановления исходного кода после обфускации;

– производительность (Performance) – характеризует влияние-обфускации на скорость выполнения программы;

– цена «запутывания» (Cost of Obfuscation) – отражает объем ресурсов устройства, необходимого для запуска обфусцированного кода и необфусцированного кода;

– размер кода (Size of Code) – отражает влияние обфускации на общий размер программного обеспечения;

– технические возможности (Technical Capabilities) – включает в себя поддержку различных языков программирования, а также наличие инструментов для анализа и отладки защищённого кода.

В рамках исследования эффективности онлайн-обфускации был проведен анализ соответствующих инструментов для языков программирования PHP и JavaScript. Существует множество онлайн-обфускаторов для JavaScript, которые используют схожие методы обфускации и не сильно отличаются по техническим возможностям. Для анализа выбраны следующие обфускаторы: JavaScript Obfuscator Tool [4] (2 режима), EvalPacker Obfuscator [5] (4 режима), JavaScript Obfuscator [6] (4 режима). Для PHP-кода были выбраны: PHP obfuscator [7] (2 режима) и WB PHP Obfuscator [8] (2 режима).

Для оценки эффективности обфускаторов использовался анализ их работы с двумя тестовыми файлами. Оценивались время выполнения и размер выходных данных для различных режимов обфускаторов, с усреднением значений для точности. Тестирование проводилось на MacBook Pro 2019 года с 2,4 GHz 4-ядерным процессором Intel Core i5 и 8 ГБ оперативной памяти, что обеспечивало стандартные условия для оценки производительности обфускаторов.

Первый файл «test_1» представлял собой простой код размером 98 байт, среднее время выполнения которого составляло 0,008 миллисекунд. Второй файл «test_2» содержал более сложный код размером 6833 байта, среднее время выполнения которого для JavaScript составило 32 миллисекунды, а для PHP – 112,38 миллисекунд. Использование этих файлов позволило оценить устойчивость обфускатора к различным типам кода и влияние методов обфускации на производительность.

Установлено, что все обфускаторы предлагают несколько режимов обфускации. Однако в EvalPacker Obfuscator и JavaScript Obfuscator некоторые режимы не сохраняли функциональность кода после обфускации, что подтвердилось тестированием на обоих файлах. Кроме того, режимы обфускации этих инструментов и PHP Obfuscator не показали устойчивости к декомпиляции: обфусцированный код можно было успешно восстановить с помощью онлайн-деобфускаторов.

Для оценки читабельности обфусцированного кода была разработана шкала с тремя уровнями: низкая (код непонятен), умеренная (код частично читаем) и высокая (код легко читаем).

Сравнения обфускаторов по таким критериям, как читабельность кода, средняя производительность, средний размер файла для разных режимов рассматриваемых онлайн-обфускаторов приведены в таблице, представленной ниже.

Результаты сравнения эффективности онлайн-обфускаторов:

Язык	Обфускатор	Читабельность кода	Средняя производительность, мс		Средний размер файлов, байт	
			Файл «test_1»	Файл «test_2»	Файл «test_1»	Файл «test_2»
JavaScript	JavaScript Obfuscator Tool	низкая	0.012	98.68	1473	6243
JavaScript	EvalPacker Obfuscator	низкая	0.011	34	4015	72048
JavaScript	JavaScript Obfuscator	низкая	0.014	77.858	650	4779
PHP	PHP Obfuscator	низкая	0.00095	106.697	740	4466
PHP	WB PHP Obfuscator	низкая	0.0015	399.78	610	6043

Анализ данных показывает, что некоторые режимы обфускации значительно увеличивают размер файла, что негативно сказывается на скорости загрузки и производительности приложения. Кроме того, в некоторых случаях наблюдается существенное увеличение времени выполнения кода, что делает программу практически непригодной для использования. Наиболее эффективным оказался JavaScript Obfuscator Tool с множеством дополнительных возможностей.

Исследование подчеркивает необходимость разработки надежных методов обфускации и деобфускации, так как существующие инструменты часто компрометируют размер файла, скорость выполнения или устойчивость к декомпиляции.

ЛИТЕРАТУРА

1. Урбанович П. П. Защита информации методами криптографии, стеганографии и обфускации: учеб.-метод. пособие. – Минск: БГТУ, 2016. – 220 с.

2. Assessment of Source Code Obfuscation Techniques / Viticchie A. [et. al]. 2016 IEEE 16th International Working Conference on Source Code Analysis and Manipulation (SCAM), Raleigh, NC, USA, 2016, p. 11-20. DOI: 10.1109/SCAM.2016.17.

3. Sutter B. D. A New Framework of Software Obfuscation Evaluation Criteria. arXiv:2502.14093v, 2025. DOI: 10.48550/ arXiv.2502.14093

4. JavaScript Obfuscator Tool [Электронный ресурс]. – Бесплатный и эффективный обфускатор для JavaScript (поддержка ES2022). – https://obfuscator.io/.

5. EvalPacker Obfuscator [Электронный ресурс]. – Инструмент для обфускации кода с возможностью защиты от копирования. – https://tools.100zona.com/evalpacker.html.

6. JavaScript Obfuscator [Электронный ресурс]. – Обфускатор JavaScript с функциями минификации и защиты кода. – https://beautifytools.com/javascript-obfuscator.php.

7. PHP Obfuscator [Электронный ресурс]. – Онлайн-обфускатор для защиты PHP-кода. – https://php-minify.com/php-obfuscator/.

8. WB PHP Obfuscator [Электронный ресурс]. – Инструмент для обфускации PHP-кода с множеством функций. – http://wb0.ru/phpobf.php.

УДК 004.03.26+004.56

АНАЛИЗ НАПРАВЛЕНИЙ ПРИМЕНЕНИЯ В БЕСПИЛОТНЫХ СИСТЕМАХ МОДЕЛЕЙ СТЕГАНОГРАФИЧЕСКОГО ПРЕОБРАЗОВАНИЯ ИНФОРМАЦИИ НА ОСНОВЕ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ

Д.В. САЗОНОВА
Белорусский государственный технологический университет
Минск, Беларусь

Беспилотные системы (БПС) в последние годы прочно вошли в сферу гражданских и военных технологий, охватывая широкий спектр задач – от мониторинга и картографирования до логистики и обеспечения безопасности [1]. Их развитие невозможно без комплексного применения цифровых технологий, специализированного программного обеспечения и искусственного интеллекта (ИИ).

Однако широкое внедрение автономных платформ неизбежно сопровождается ростом киберугроз. БПС постоянно обмениваются данными — как между отдельными устройствами, так и с наземными пунктами управления. Передача телеметрии, видеопотоков и управляющих команд делает беспилотники уязвимыми для перехвата, подмены и атак «человек посередине». В связи с этим одной из актуальных задач становится разработка и внедрение инновационных методов защиты информации.