Преимуществами использования вейвлет-преобразования является то, что информация внедряется в наименее заметные области на основе различных поддиапазонов, что обеспечивает возможность встраивания, предлагая баланс надежности и незаметности, а также данный метод более устойчив к сжатию с потерями и другим модификациям изображения, так как данные внедряются именно в частотные характеристики.

ЛИТЕРАТУРА

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

2. Сейеди, С.А. Сравнение методов стеганографии в изображениях / С.А. Сейеди, Р.Х. Садыхов // Информатика. – 2013. – № 1 (37). – С. 66–75.

3. Mallat, S. A theory for multiresolutional signal decomposition: the wavelet representation // IEEE Trans. Pattern Analysis and Machine Intelligence. – 1989. – №. 7. – P.674–693.

4. Sara, U. Image Quality Assessment through FSIM, SSIM, MSE and PSNR. A Comparative Study / U. Sara, M. Akter, M. Uddin // Journal of Computer and Communications. – 2019. – Vol 7, № 3. – P. 8–18.

5. Хартанович, А.А. Комбинирование каскадной модели и стеганографического метода для размещения информации в файлах изображений / А.А. Хартанович // Технические средства защиты информации: тезисы докладов XXII Белорусско-российской науч.-технич. конференции, Минск, 12 июня 2024 г. – Минск: БГУИР, 2024. – С. 94–95.

УДК 004.051

В.С. Кантарович, ассист.; П.П. Урбанович, проф.
(БГТУ, г. Минск)

КРИТЕРИИ ЭФФЕКТИВНОСТИ АЛГОРИТМОВ ОБФУСКАЦИИ И ДЕОБФУСКАЦИИ ПРОГРАММНОГО КОДА

В современном мире программные продукты всё чаще становятся объектом атак, важно защищать код от обратного инжиниринга и анализа злоумышленниками. Таким образом обфускация программного кода играет ключевую роль в современном мире информационной безопасности и защиты интеллектуальной собственности [1].

Обфускация программного кода – это процесс изменения и запутывания кода таким образом, чтобы его стало труднее понимать и ана-

лизировать. Основная цель обфускации – это затруднить процесс обратного инжиниринга, защитить интеллектуальную собственность и повысить безопасность программных продуктов.

Процесс обфускации применим для защиты авторского контента путём затруднения анализа кода, уменьшения размера программного кода и представления вредоносного программного обеспечения в виде защищённого представления.

Обфускации кода осуществляются при помощи сторонних независимых программ. Они называются обфускаторами. Такое программное обеспечение изменяет код по определённым алгоритмам и методам. Для обфускации программного кода существуют алгоритмы общего типа, которые рассчитаны на разные языки программирования и используют общие методы и подходы к реализации «запутывания», а также алгоритмы специализированного типа, которые опираются на языки разработки, обладающие определенными свойствами.

Онлайн-обфускаторы кода обычно поддерживают несколько языков программирования, среди которых часто встречаются как интерпретируемые языки, такие как JavaScript, PHP, Python, так и компилируемые, такие как Java, C# [2]. Наиболее широко используются онлайн-обфускаторы JavaScript кода, так как JavaScript, в отличии от других языков выполняется на стороне клиента и исходных код передаётся пользователю в открытом виде.

Существует несколько видов обфускации, каждый из которых направлен на запутывание и затруднение понимания программного кода [3]: лексическая обфускация, обфускация структур данных, обфускация потока управления, превентивная обфускация.

Лексическая обфускация. Это самый простой и часто используемый вид обфускации, который применяется во многих обфускаторах. При лексической обфускации происходит изменение программного кода и приведение его в нечитабельный вид. Включает такие способы преобразования кода, как удаление комментариев или замена их на бессмысленные, минификация кода, замена имён переменных и функций на трудно воспринимаемые случайные сгенерированные последовательности символов, изменение структуры программного кода, добавление «мёртвого» кода.

Обфускация структур данных. Данный способ обфускации связан с преобразованием структур данных. Включает в себя три группы методов: обфускация хранения, обфускация соединения, обфускация переупорядочивания.

Обфускация хранения подразумевает трансформацию хранилищ

данных и включает разделение одной переменной на комбинацию переменных, замену существующих типов данных, создание и использование нестандартных типов данных для определённых задач, изменение области действия переменной.

Обфускация соединения направлена на усложнение представления используемых структур данных, путём объединения или разделении данных. Включает такие способы усложнения кода, как встраивание функций в места её вызова, объединение фрагментов кода воедино, дублирование функций и реструктуризация циклов.

Обфускация переупорядочивания заключается в изменении внутреннего порядка структур данных, а именно изменение последовательности объявления переменных или функций, переупорядочивание методов, свойств, полей.

Следующим видом обфускации является обфускация потока управления. Данный вид обфускации подразумевает запутывание потока управления, то есть изменение последовательности выполнения программного кода. Может быть выполнена путем изменение порядка операторов выполнения программы.

Обфускация потока управления включает такие методы, как выделение участка кода в отдельную функцию, распараллеливание кода, добавление недостижимого кода, клонирование и объединение функций, реструктуризация циклов, изменение и расширение области действия переменной, переупорядочивание циклов и функций, изменения порядка операторов выполнения программы, добавление избыточного кода и избыточных операций для запутывания логики программы.

Превентивная обфускация применяется заранее, еще до того, как код будет подвергнут анализу или атаке. Данный вид обфускации использует ошибки в работе деобфускаторов, включает в себя шифрование строк, искажение логики программы, методы лексической обфускации, обфускации потока управления и байт-кода.

Оценка эффективности обфускаторов кода может включать несколько критериев:

– устойчивость. Используется для описания уровня сложности реализации реверсивной инженерии над программой после обфускации кода;

– уровень защиты. Как сильно обфускация затрудняет анализ и понимание кода злоумышленниками. (сравнение читабельности кода до и после);

– сохранение функциональности. Как обфускация влияет на работоспособность программы;

– сложность декомпиляции. Как трудно восстановить исходный код после обфускации;

– производительность. Как обфускация влияет на скорость выполнения программы;

– цена «запутывания». Отражает объем ресурсов устройства, необходимого для запуска обфусцированного кода и необфусцированного кода;

– размер кода. Как обфускация влияет на общий размер программного обеспечения;

– технические возможности. Включает в себя поддержку различных языков программирования, а также наличие инструментов для анализа и отладки защищённого кода;

Для оценки эффективности онлайн-обфускаторов были проанализированы онлайн-обфускаторы для таких языков программирования как PHP и JavaScript.

Анализ эффективности онлайн-обфускаторов проводился при помощи написанного заранее одинакового кода для выбранных языков программирования, который подсчитывает сумму двух чисел и считает время выполнения программного кода. Размер файла с исходным программным кодом занимает на диске 98 байт и выполняется за время 0,008 мс.

Существует множество онлайн-обфускаторов для JavaScript-кода, которые использует одинаковые виды обфускации и которые особо не отличаются между собой по своим техническим возможностям. Для анализа выбраны следующие онлайн-обфускаторы для JavaScript-кода: JavaScript Obfuscator Tool (рассмотрено 2 режима), EvalPacker Obfuscator (рассмотрено 4 режима), JavaScript Obfuscator (рассмотрено 4 режима). Для PHP-кода для анализа выбраны следующие онлайн-обфускаторы: PHP obfuscator (рассмотрено 2 режима), WB PHP Obfuscator (рассмотрено 2 режима).

Рассматриваемые онлайн-обфускаторы используют следующие виды обфускации: лексическую обфускацию, обфускацию структур данных (соединения, переупорядочивания) и обфускацию потока управления.

Все рассматриваемые обфускаторы имеют несколько режимов обфускации. В таких обфускаторах как EvalPacker Obfuscator и JavaScript Obfuscator некоторые режимы не прошли устойчивость на сохранение функциональности программного кода. Режимы в обфускаторах EvalPacker Obfuscator, Javascript Obfuscator, PHP obfuscator не прошли проверку на устойчивость к декомпиляции и обфусцированный код был восстановлен при помощи онлайн-деобфускаторов.

Сравнения по другим выбранным критериям приведены в таблице, представленной ниже.

Таблица – Результаты сравнения эффективности онлайн-обфускаторов

Язык	Обфускатор	Читабельность кода	Средняя производительность	Средний размер файлов
JavaScript	JavaScript Obfuscator Tool	низкая	0.012 мс	1473 байт
JavaScript	EvalPacker Obfuscator	низкая	0.011 мс	4015 байт
JavaScript	JavaScript Obfuscator	низкая	0.014 мс	650 байт
PHP	PHP obfuscator	низкая	0.00095 мс	740 байт
PHP	WB PHP Obfuscator	низкая	0.0015 мс	610 байт

Результаты сравнения эффективности онлайн-обфускаторов позволяют сделать вывод, что большинство из них не являются надёжными и с легкостью могут быть деобфусцированы. Самым эффективным из рассмотреных онлайн-обфускаторов выявлен JavaScript Obfuscator Tool, который обладает множеством дополнительных технических возможностей и настроек.

Таким образом, можно сделать вывод о необходимости разработки надёжного метода для обфускации и деобфускации программного кода.

ЛИТЕРАТУРА

1. Пласковицкий В.А. Защита программного обеспечения от несанкционированного использования и модификации методами обфускации / В.А. Пласковицкий, П.П. Урбанович // Труды БГТУ. № 6. Физико-математические науки и информатика, 2011. – С. 173–176.

2. Урбанович П.П. Защита информации методами криптографии, стеганографии и обфускации: учеб.-метод. пособие для студентов специальности 1-98 01 03 «Программное обеспечение информационной безопасности мобильных систем», направления специальности 1-40 05 01-03 «Информационные системы и технологии (издательско-полиграфический комплекс)», специальности 1-40 01 01 «Программное обеспечение информационных технологий» специализации 1-40 01 01 10 «Программмирование Интернет-приложений» / П.П. Урбанович. – Минск : БГТУ, 2016. – 220 с.

3. Никольская К.Ю., Хлестов А.Д. Обфускация и методы защиты программных продуктов / Вестник УрФО. Безопаность в информационной сфере № 2(16). 2015. – С. 7–10.