



УДК 004

## МЕТОДЫ ОБФУСКАЦИИ КОДА

**Градский Дмитрий Юрьевич**

магистрант 2 курса ФГБОУ ВО «Пензенский государственный университет»  
г. Пенза, Россия

### Аннотация

В статье рассматриваются основные виды алгоритмов обфускации программного кода, а также приведено описание наиболее распространенных приемов, используемых этими алгоритмами.

**Ключевые слова:** информационная безопасность, программа, исходный код, интерпретируемые языки программирования, скрипты, обфускация.

## CODE OBFUSCATION METHODS

**Dmitry Yu. Gradsky**

2nd year master's student of the Penza State University  
Penza, Russia  
[gradsky2012@yandex.ru](mailto:gradsky2012@yandex.ru)

### ABSTRACT

The article discusses the main types of software obfuscation algorithms, as well as describes the most common techniques used by these algorithms...

**Key words:** information security, program, source code, interpreted programming languages, scripts, obfuscation.

В настоящее время высока потребность в средствах и методах защиты интеллектуальной собственности, в частности программных продуктов. Поэтому возникают задачи создания средств для защиты того или иного программного продукта.

В качестве одного из возможных методов защиты выступает обфускация. Обфускация, или запутывание кода – это приведение исходного или исполняемого кода программы к виду, при котором сохраняются функциональность и работоспособность программы, но затрудняется анализ, понимание работы алгоритмов и их модификация или декомпиляция [1].

Для интерпретируемых языков это особенно важно, так как скрипт, который содержит легко читаемый код, может стать объектом несанкционированной модификации. Именно поэтому необходимо сделать так, чтобы для возврата кода к

исходному виду до обфускации затрачивалось больше времени, чем для разработки аналога.

Основные виды алгоритмов обфускации подразделяются на [2]:

- 1) лексическую обфускацию (понижение читабельности кода);
- 2) обфускацию данных (различные манипуляции с данными);
- 3) обфускацию управления (запутывание последовательности выполнения программного кода).

Наиболее простыми являются методы лексической обфускации, которые основываются на изменении форматирования исходных текстов программ. Данный метод включает в себя следующие наиболее распространенные преобразования [3, 4]:

- 1) удаление символов форматирования, делающих код читаемым, таких как: пробельные символы, символы перехода на новую строку, символы табуляции и прочие подобные символы;
- 2) замена комментариев на бессмысленные или полное удаление;
- 3) переименование переменных, функций, классов.

Но следует так же учитывать, что существуют встроенные, ключевые и зарезервированные имена, которые нельзя изменять, поэтому методы обфускации должны сохранять такие имена без изменений.

Методы обфускации данных связаны с преобразованием структур данных. Данные методы подразделяются на [2, 3]:

- 1) обфускацию хранения;
- 2) обфускация соединения;
- 3) обфускация переупорядочивания.

Обфускация хранения направлена на создание и использование сложных типов данных, замену существующих типов данных и т. п. Включает следующие преобразования [3]:

- 1) разделение переменных;
- 2) замена статических данных на процедурные, например, замена строк на функцию, возвращающую требуемую строку в зависимости от входных параметров;
- 3) изменение области видимости переменных.

Обфускация соединения состоит в объединении или разделении данных, или структур данных. В качестве примера таких методов могут служить операции с массивами [5]:

- 1) разделение одного массива на множество массивов;
- 2) изменение размерности массивов;
- 3) соединение нескольких переменных в массив.

Обфускация переупорядочивания состоит в изменении порядка следования объектов. Включает следующие преобразования [5]:

- 1) изменение порядка объявления переменных;
- 2) изменение порядка следования функций;
- 3) изменения внутренней организации структур и классов.

Обфускация управления направлена на преобразования порядка исполнения функциональных блоков программы. Методы данной группы являются одними из наиболее сложными в проектировании и реализации и включают в себя следующие преобразования [5]:

- 1) распараллеливание исполнения независимых участков кода;
- 2) добавление «мертвого кода» – кода, который никогда не будет выполняться;

- 3) извлечение метода – вынос функционально самостоятельного и независимого участка кода в отдельную функцию;
- 4) вставка тела функции во все места вызова этой функции;
- 5) преобразование области действия переменных;
- 6) соединение нескольких функций в одну;
- 7) уменьшения числа библиотечных и системных вызовов по возможности;
- 8) преобразование графа потока управления;
- 9) преобразование циклов;
- 10) переупорядочивание вызовов функций выражений, циклов, операций условного и безусловного перехода.

Также при добавлении каких-либо участков кода необходимо, чтобы добавляемый код не отличался или был похож на искомый для неразличимости произведенных вставок.

Особым образом стоят методы превентивной обфускации, которые в первую очередь направлены на защиту от методов деобфускации и декомпиляции специальными программами, которые основаны на алгоритмах статического и динамического анализа, анализа неиспользуемого или недостижимого кода. Данный метод обфускации основан на использовании ошибок, недоработок и недостатков программ деобфускации.

Таким образом, существует большое число приемов проведения запутывающих преобразований, которые следует применять в некоторой комбинации и определенном порядке. Именно задача получения таких комбинации и порядка являются актуальной задачей для проектирования новых эффективных алгоритмов обфускации.

### Список литературы

1. Обфускация (программное обеспечение) // Википедия – свободная энциклопедия [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/Обфускация\\_\(программное\\_обеспечение\)](https://ru.wikipedia.org/wiki/Обфускация_(программное_обеспечение)) (дата обращения 20.05.2020).
2. Ивченкова Ю. С. Виды и способы обфускации / Ю. С. Ивченкова, М. К. Савкин // Электронный журнал: наука, техника и образование [Электронный ресурс]: 2016. – №2(6). – С. 60-66. – Режим доступа: <http://nto-journal.ru/catalog/informacionnye-texnologii/152> (дата обращения 20.05.2020).
3. Collberg С. Taxonomy of Obfuscating Transformations / С. Collberg, С. Thomborson, D. Low [Электронный ресурс]. – Режим доступа: [https://researchspace.auckland.ac.nz/handle/2292/3491\\_2292/3491](https://researchspace.auckland.ac.nz/handle/2292/3491_2292/3491) (дата обращения 20.05.2020).
4. Медгаус С. В. Обфускатор программного кода языка JavaScript / С. В. Медгаус, А. В. Чернышова // Информатика и кибернетика [Электронный ресурс]: Донецк: ДонНТУ, 2016. – №4(6). – С. 59-66. – Режим доступа: <http://masters.donntu.org/2017/fknt/medgaus/library/article4.htm> (дата обращения 20.05.2020).
5. Носов А. С. Анализ методов защиты программных кодов с помощью обфускации / А. С. Носов, Н. Е. Губенко [Электронный ресурс]: Донецк: ДонНТУ, 2013. – Режим доступа: <http://masters.donntu.org/2013/fknt/nosov/library/article2.htm> (дата обращения 20.05.2020).

### References

1. Obfuscation (software) // Wikipedia - the free encyclopedia [Electronic resource]. Access mode: [https://ru.wikipedia.org/wiki/Obfuscation\\_\(software\)](https://ru.wikipedia.org/wiki/Obfuscation_(software)) (accessed 05.20.2020) [in Russian].

2. Ivchenkova Yu. S. Types and methods of obfuscation / Yu. S. Ivchenkova, MK Savkin // Electronic journal: science, technology and education [Electronic resource]: 2016. - No. 2 (6). - S. 60-66. - Access mode: <http://nto-journal.ru/catalog/informacionnye-texnologii/152> (date of access 05.20.2020) [in Russian].
3. Collberg C. Taxonomy of Obfuscating Transformations / C. Collberg, C. Thomborson, D. Low [Electronic resource]. - Access mode: <https://researchspace.auckland.ac.nz/handle/2292/3491> (accessed 05.20.2020).
4. Medgaus S. V. Obfuscator of the program code of the JavaScript language / S. V. Medgaus, A. V. Chernyshova // Informatics and cybernetics [Electronic resource]: Donetsk: DonNTU, 2016. - No. 4 (6). - S. 59-66. - Access mode: <http://masters.donntu.org/2017/fknt/medgaus/library/article4.htm> (accessed 05.20.2020) [in Russian].
5. Nosov A. S. Analysis of methods for protecting program codes using obfuscation / A. S. Nosov, N. E. Gubenko [Electronic resource]: Donetsk: DonNTU, 2013. - Access mode: <http://masters.donntu.org/2013/fknt/nosov/library/article2.htm> (accessed 05.20.2020) [in Russian].