УДК 004.056.53

# РЕАЛИЗАЦИЯ ОБФУСКАТОРА ДЛЯ JAVASCRIPT-КОДА

***А. А. Горбачев***
Донской государственный технический университет (г. Ростов-на-Дону, Российская Федерация)

Рассмотрены современные методы обфускации языка JavaScript. Продемонстрирована работа онлайн-ресурсов, обеспечивающих запутывание кода. Выявлены их достоинства и недостатки. Разработана и инициализирована программа для обфускации JavaScript-кода.
***Ключевые слова:*** обфускатор, JavaScript, запутывание кода, алгоритм, защита от несанкционированного доступа.

UDC 004.056.53

# OBFUSCATOR IMPLEMENTATION FOR JAVASCRIPT CODE

***A. A. Gorbachev***
Don State Technical University (Rostov-on-Don, Russian Federation)

The article considers modern methods of JavaScript obfuscation. The work of online resources providing code obfuscation is demonstrated. Their advantages and disadvantages are revealed. A program for obfuscation of JavaScript code was developed and initialized.
***Keywords:*** obfuscator, JavaScript, code obfuscation, algorithm, protection against unauthorized access.

**Введение.** В современном мире существует множество угроз кражи программ и интеллектуальной собственности, нарушения авторских прав. Данные проблемы связаны с доступностью разработанных программных средств и их исполняемого кода. Существуют разные способы борьбы с киберугрозами. Например, можно хранить код программы на защищенном сервере, а результаты вычисления передавать по запросу от клиента, либо использовать компилируемый код. Но такой способ не подойдет, если код находится в открытом доступе. Чаще всего это десктопные программы или веб-приложения. Лучшим вариантом защиты таких программ является применение обфускации [1]. Цель данной статьи — описать современные методы обфускации языка JavaScript, рассмотреть онлайн-ресурсы, обеспечивающие запутывание кода, выявив их достоинства и недостатки.

**Теоретическая часть.** Обфускация — это алгоритм, который придает коду программы трудный для анализа вид, но при этом полностью сохраняет его функциональность. Также уменьшается размер файла, содержащего исходный код, и происходит сокрытие авторских прав. Программа, с помощью которой реализуется данный алгоритм, называется обфускатор. К минусам обфускации можно отнести затруднение дальнейшей отладки кода и возможное влияние на его функциональность.

**Применение онлайн обфускатора.** В запутывании кода особенно нуждаются веб-приложения, написанные на языке JavaScript. Этот язык выполняется на стороне браузера, следовательно, зная принципы работы браузера, можно получить код. Для JavaScript существует множество онлайн обфускаторов и библиотек. Для демонстрации работы онлайн-обфускатора функция нахождения факториала была помещена на два таких ресурса: javascriptobfuscator и daftlogic [2–3].

Исходная JavaScript-функция приведена на рис. 1.

```
function factorial(n) {
  return (n != 1) ? n * factorial(n - 1) : 1;
}
alert( factorial(5) );
```

Рис. 1. Функция до обфускации

Результат работы javascriptobfuscator показан на рис. 2.

```
var _0xdd84=[];function factorial(_0xef93x2){return (_0xef93x2!= 1)?
_0xef93x2* factorial(_0xef93x2- 1):1}alert(factorial(5))
```

Рис. 2. Функция после обфускации ресурсом javascriptobfuscator

Результат работы онлайн-обфускатора daftlogic показан на рис. 3.

```
eval(function(p,a,c,k,e,d){e=function(c){return
c};if(!''.replace(/^/,String)){while(c--){d[c]=k[c]||c}k=[function(e)
{return d[e]}];e=function(){return'\\w+'};c=1};while(c--){if(k[c])
{p=p.replace(new RegExp('\\b'+e(c)+'\\b','g'),k[c])}}return p}('3 2(0)
{6(0!=1)?0*2(0-
1):1}4(2(5));',7,7,'n||factorial|function|alert||return'.split('|'),0,{}))
```

Рис. 3. Функция после обфускации ресурсом daftlogic

Стоит заметить, что приведенные выше онлайн-обфускаторы делают исходный код достаточно неразборчивым. Такой вид способен отпугнуть множество злоумышленников, но возникают новые угрозы потери интеллектуальной собственности:

— владелец сайта может собирать код при его импорте на сетевой ресурс для обфускации, нарушая при этом авторские права;

— сайт обфускатора может быть уязвим для третьих лиц, имеющих доступ к коду;

— для большинства онлайн-обфускаторов имеется ресурс, который деобфусцирует код, поэтому целеустремленный злоумышленник может воспользоваться им.

Таким образом, лучший способ для запутывания JavaScript-кода и сохранения авторских прав — написание собственного обфускатора, так как доступ третьих лиц к такому алгоритму минимален.

**Реализация программы обфускатора.** Для создания обфускатора необходимо разработать функционал, который будет выполнять следующие действия:

— поддержание работы с файловой системой для нахождения обфусцируемого файла;

— применение модификатора, уменьшающего размер результирующего файла путём удаления ненужных символов без изменения его функциональности;

— обфусцирование кода.

В ходе данной работы реализован обфускатор, который удовлетворяет требования всех вышеприведенных пунктов. Чтобы соединить воедино минификатор, обфускатор и работу с файловой системой, используется программная платформа. Для реализации необходимого функционала была выбрана платформа Node.js. Она позволяет подключать внешние библиотеки, которые написаны на разных языках и направлены на работу с JavaScript-кодом.

В качестве минификатора подходит библиотека uglify-js: версии 3.6.9.

Главную роль в программе играет библиотека javascript-obfuscator, которая является мощным бесплатным обфускатором для JavaScript. Также она имеет широкий набор функций, обеспечивающих защиту исходного кода [4].

Алгоритм программы обфускатора довольно прост. В ней прописывается, с каким именно файлом будет происходить работа. Если файл найден, то проводятся минификация, обфускация и создание нового файла, в который записывается код, прошедший все операции. Если же нужный файл не найден, то программа выдает ошибку. Блок-схема данной программы представлена на рис. 4, а ее код — на рис. 5.

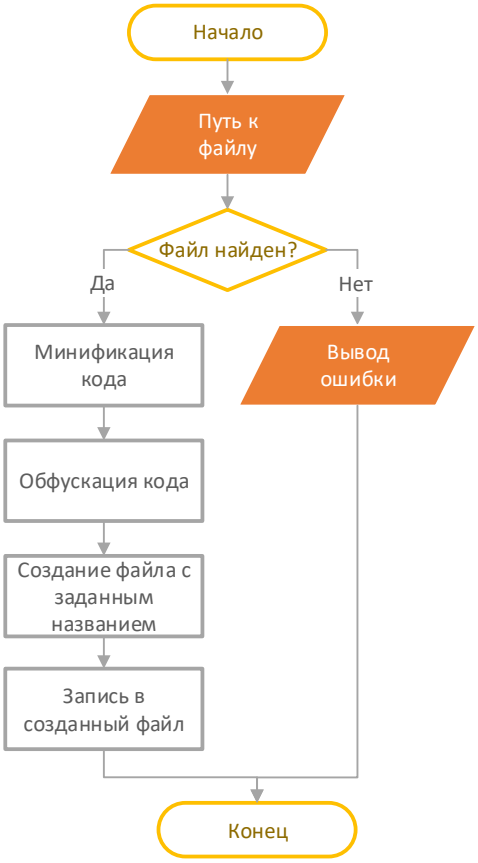


Рис. 4. Блок-схема программы обфускатора

```
const fs = require('fs');
const JSObfuscator = require('javascript-obfuscator');
const Minif = require('uglify-js');

fs.readFile('./mycode.js', 'UTF-8', function (err, data) {
    if(err) {
        throw err;
    }
    var min = Minif.minify(data);
    var obfuscationResult = JSObfuscator.obfuscate(min.code);
    fs.writeFile('./coder.js', obfuscationResult.getObfuscatedCode(), function (err) {
        if(err) {
            return console.log(err);
        }

        console.log("the file was saved!");
    })
});
```

Рис. 5. Код программы обфускатора

В результате работы программы был обфусцирован код, представленный на рис. 6. Во время работы программы автоматически создается файл coder.js, который добавляется в каталог, со-

держащий файл с исходным кодом. Созданный элемент содержит обфусцированный код, который показан на рис. 7.

```
function fun(a,b) {
    var c = 0;
    if(a > b) {
        c = a - b;
    } else if (a < b) {
        c = b - a;
    } else {
        c = b + a
    }
    return c;
}
console.log( fun( a: 5,  b: 6) );
```

Рис. 6. Код до обфускации программой

```
function fun(_0x21105c,_0x40994c)
{return _0x40994c<_0x21105c?_0x21105c-_0x40994c:_0x21105c
<_0x40994c?_0x40994c-_0x21105c:_0x40994c+_0x21105c;}
console['log'](fun( _0x21105c: 0x5, _0x40994c: 0x6));
```

Рис. 7. Код, прошедший программную обфускацию

**Заключение.** Описаны современные методы обфускации языка JavaScript. Рассмотрены два онлайн-ресурса, обеспечивающие запутывание кода: javascriptobfuscator и daftlogic. Продемонстрирована их работа, выявлены достоинства и недостатки. Разработан и инициализирован обфускатор JavaScript-кода.

Проведенный анализ показал, что онлайн обфускаторы делают код достаточно неразборчивым и могут быть применены, но существует угроза потери интеллектуальной собственности. Поэтому лучший способ и для запутывания JavaScript-кода, и для сохранения авторских прав — написание собственного обфускатора, так как доступ третьих лиц к такому алгоритму минимален.

**Библиографический список**

1. Родичев, Ю. А. Нормативная база и стандарты в области информационной безопасности : учебное пособие / Ю. А. Родичев. — Санкт-Петербург : Питер, 2017. — 256 с.

2. Javascript Obfuscator [Электронный ресурс] / Javascript Obfuscator. — Режим доступа: https://javascriptobfuscator.com/Javascript-Obfuscator.aspx/ (дата обращения: 30.11.2019).

3. Online Javascript Obfuscator [Электронный ресурс] / DaftLogic. — Режим доступа: https://www.daftlogic.com/projects-online-javascript-obfuscator.htm/ (дата обращения: 30.11.2019).

4. Библиотека javascript-obfuscator [Электронный ресурс] / GitHab. — Режим доступа: https://github.com/javascript-obfuscator/javascript-obfuscator/ (дата обращения: 24.11.2019).

*Об авторе:*

**Горбачев Андрей Александрович,** студент Донского государственного технического университета (344000, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), andrew.gorba4ev2018@yandex.ru