Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Информационные сети. Основы безопасности

ОТЧЁТ

к лабораторной работе №7

на тему

**ЗАЩИТА ПО ОТ НЕСАНКЦИОНИРОВАННОГО ИСПОЛЬЗОВАНИЯ**

Выполнил: студент гр.253504 Давыдовский Д.В.

Проверил: ассистент кафедры информатики Герчик А.В.

Минск 2025

**СОДЕРЖАНИЕ**

[1 Цель работы 3](#_Toc188986632)

[2 Ход работы 4](#_Toc188986633)

[Заключение 6](#_Toc188986634)

# 1 ЦЕЛЬ РАБОТЫ

Цель работы заключается в разработке программного продукта для обфускации исходного кода программ. В ходе работы будет разработан веб-сайт для удобства использования пользователем, а также сервер, который будет выполнять все необходимые задачи для обфускаци кода.

Перед началом работы стоит разобраться, что же такое обфускация. Обфускация — это способ запутать код программы, чтобы его было сложно читать и понимать. Представьте, что у вас есть понятный текст, например: "Привет, мир!", а после обфускации он превращается во что-то вроде "#$gH8&l@12". Смысл остаётся тем же, но разобрать, что тут написано, уже гораздо сложнее.

В программировании это делают, чтобы защитить код от хакеров и конкурентов. Например, если вы написали классную программу, но не хотите, чтобы кто-то взял её код, понял, как она работает, и использовал в своих целях, можно применить обфускацию. После неё код будет по-прежнему работать так же, но читать и анализировать его станет очень трудно.

Обычно обфускация меняет названия переменных и функций на случайные символы, убирает пробелы и делает структуру кода максимально запутанной. Человек, открывший такой код, увидит нагромождение непонятных символов, тогда как компьютер будет по-прежнему понимать его и выполнять без ошибок. Обфускацию часто используют для защиты программ, игр, мобильных приложений и даже вредоносных вирусов, чтобы антивирусы не могли легко их распознать.

Стоит упомянуть, что обфускация бывает разной степени сложности. Можно выделить три основных уровня: базовый, средний и высокий.

Базовый уровень — это простейшая обфускация, которая делает код менее читаемым, но не слишком усложняет его понимание. Например, переименование переменных и функций в случайные буквы и цифры, удаление комментариев и пробелов, замена некоторых операторов на более запутанные конструкции.

Средний уровень — это уже более сложная запутанность, которая включает не только переименование, но и изменение структуры кода. Например, вставка лишних строк, изменение порядка выполнения операций, использование запутанных условий и циклов.

Высокий уровень — это максимальная обфускация, при которой код становится практически неразбираемым. Здесь могут использоваться сложные алгоритмы шифрования, динамическое изменение кода во время выполнения, подмена инструкций, виртуализация команд и другие хитрые методы.

Чем выше уровень обфускации, тем сложнее анализировать код, но при этом программа может работать медленнее, так что важно находить баланс между защитой и производительностью.

# 2 ХОД РАБОТЫ

В ходе работы был дополнен веб-сайт из лабораторной работы №6, была добавлена страница для обфускации кода. Стоит упомянуть, что веб-сайт поддерживает обфускацию кода только на *JS* (или *ReactJS*). При переходе на страницу пользователю предлагается ввести код вручную или же выбрать файл. При выборе файл текст из него автоматически скопируется в поле ввода. На рисунке 1 представлен интерфейс пользоватя.

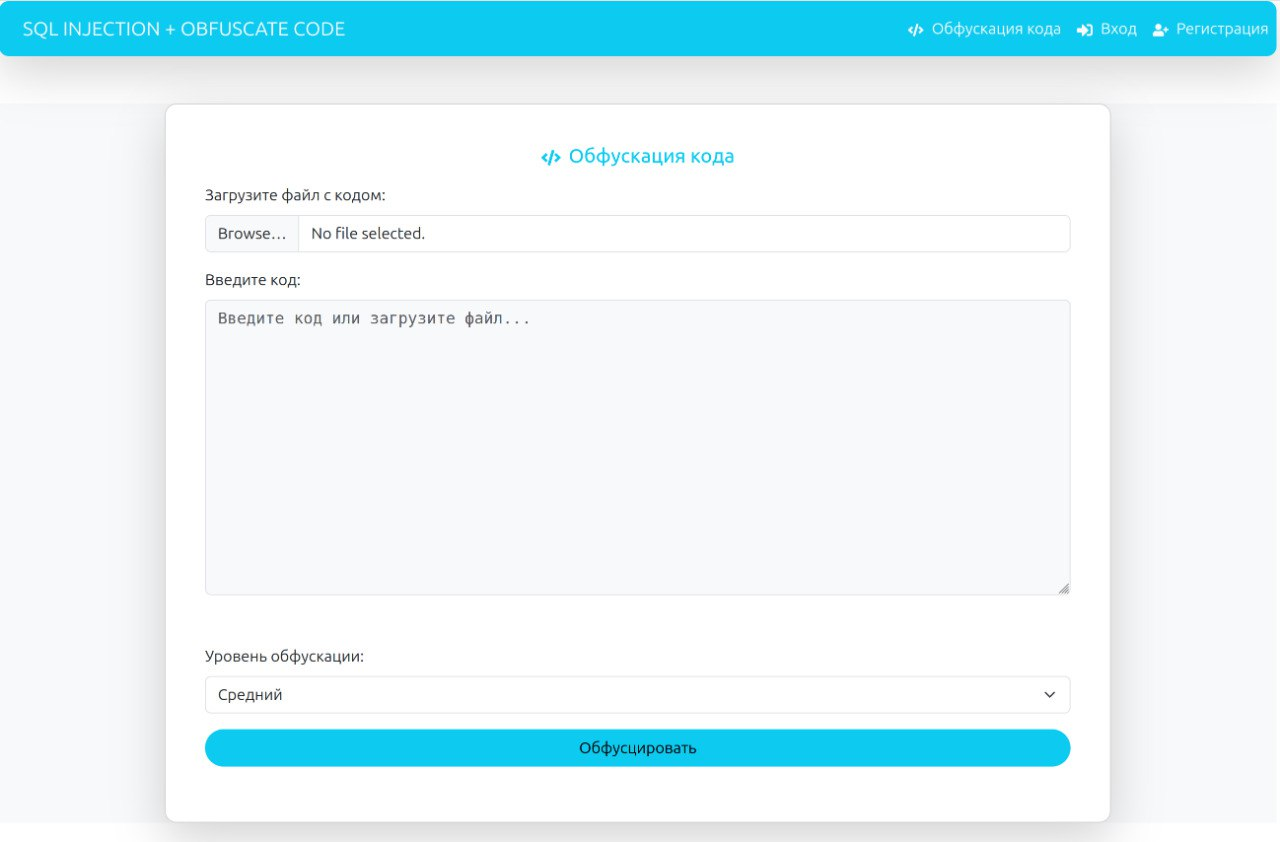


Рисунок 1 – Интерфейс пользователя

После ввода кода или загрузки файла, пользователю предоставляется возможность выбора уровня сложности обфускации. После этиъ несложных действий, единственной, что нужно сделать пользователю это нажать на кнопку обфускации, после чего под кнопкой появится поле с запутанным кодом, с возможность скачать файл с итоговым кодом. На рисунке 2 представлен результат работы приложения.

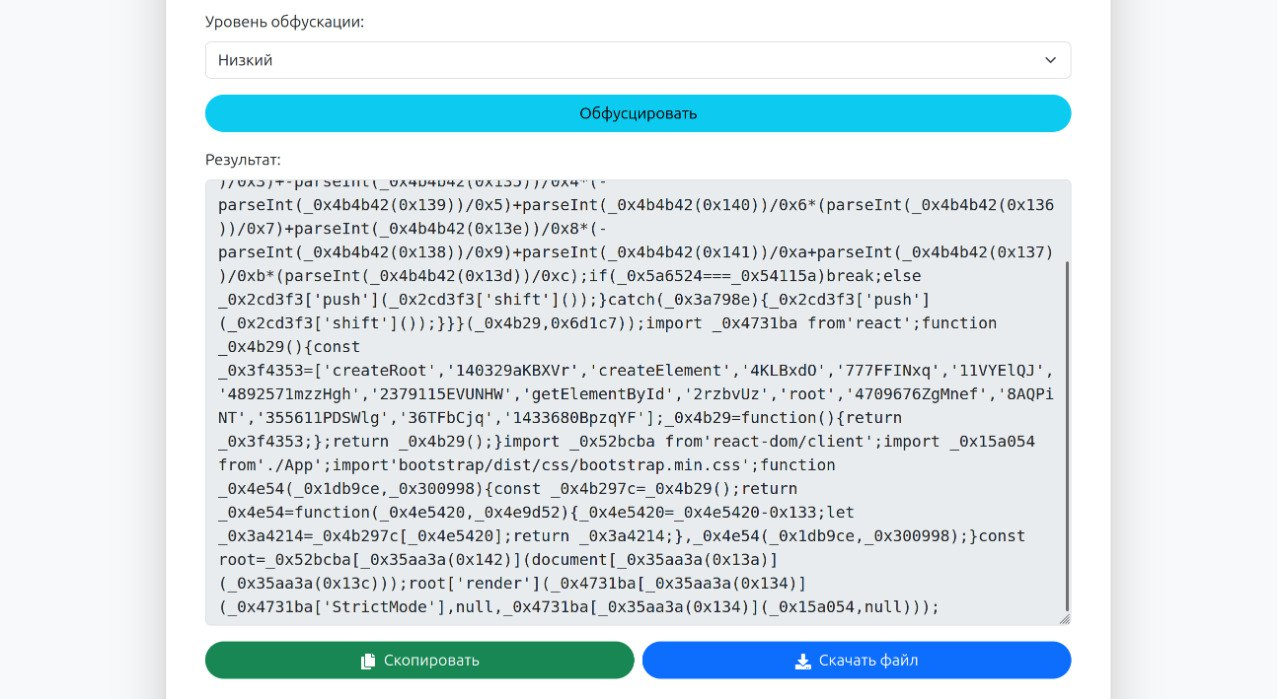


Рисунок 2 – Результат работы приложения

Для проверки того, что программа действительно ничего не испортила, можно использовать любого нейро-помощника, например *ChatGPT*. Стоит отметить, что чем выше уровень обфускации, тем труднее нейросети будет понять, что происходит в коде, поэтому проверку работоспособности лучше проверять на практическом примере.

Подробнее разберём работу сервера. Сервер выполняет всего одну функцию. Для обфускации кода используется библиотека *JavaScriptObfuscator,* которая выполняет запутывание автоматически. Для повышения уровня сложности запутывания в функцию запутывания передаются специальные параметры. Стоит отметить, что для поддержки кода на *React*, используется библиотека *babel*, которая переводит код написанный на *ReactJS* в обычный *JS* код. Основная часть исходного кода представлена на рисунке 3.



Рисунок 3 – Основная часть исходного кода сервера

# ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы была разработана и протестирована программа для обфускации исходного кода программ на языке *JS* (*ReactJS*) с поддержкой выбора сложности запутывания. Для удобства пользователя был разработан веб-сайт, который позволяет вводить код пользователю или загружать файлы, а также скопировать или скачать файл с запутанным кодом.