

# **Лабораторная работа №8**

## **Защита программного обеспечения на уровне кода**

### **1 Цель работы**

1.1 Познакомиться с методами защиты исходного кода от декомпиляции и анализа.

### **2 Литература**

2.1 Зверева В. П., Сопровождение и обслуживание программного обеспечения компьютерных систем : учебник для студ. учреждений сред. проф. Образования / В. П. Зверева, А. В. Назаров. – М. : Издательский центр «Академия», 2018. – 256 с.

### **3 Подготовка к работе**

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

### **4 Основное оборудование**

4.1 Персональный компьютер.

### **5 Задание**

5.1 Декомпиляция приложения

5.1.1 Разработать консольное приложение для авторизации пользователя в системе с использованием пароля хранимого в коде.

5.1.2 Выполнить сборку приложения

5.1.3 Открыть Расширения -> Управление расширениями и при помощи диспетчера расширений установите ILSpy.

5.1.4 С помощью ILSpy выберите и декомпилируйте DLL файл разработанного приложения и изучите его внутреннее устройство, найдите строку, в которой вы сохранили пароль приложения.

5.2 Использование Obfuscator

5.2.1 Используя консоль установите инструмент Obfuscator

```
dotnet tool install --global Obfuscator.GlobalTool
```

5.2.2 Создайте в корне решения конфигурационный файл obfuscator.xml со следующим содержимым (при необходимости скорректируйте пути файлов):

```
<?xml version="1.0" encoding="utf-8"?>
<Obfuscator>
  <Var name="InPath" value="bin\Release\net8.0" />
  <Var name="OutPath" value="bin\Release-Obfuscated" />
  <Var name="KeepPublicApi" value="false"/>
  <Module file="$(InPath)\ProjectName.dll"/>
</Obfuscator>
```

5.2.3 Выполнить команду:

```
obfuscator.console obfuscator.xml
```

5.2.4 С помощью ILSpy декомпилируйте DLL или EXE файл

обфусцированного приложения и изучите его внутреннее устройство, найдите строку, в которой вы сохранили пароль приложения.

5.2.5 По отдельности попробуйте выполнить обфускацию, добавив в конфигурацию следующие свойства:

```
<Var name="UseKoreanNames" value="true"/>
<Var name="UseUnicodeNames" value="true"/>
```

5.3 Исключение кода из обфускации.

5.3.1 Добавить в проект новое пространство имен, в котором создать публичный класс с некоторым публичным методом и приватным полем.

5.3.2 Изменить значение параметра `KeepPublicApi` на `true`, выполнить обфускацию, изучить полученный результат. Вернуть значение параметра к исходному состоянию.

5.3.3 Изменить параметр `Module`, добавив внутрь него строку

```
<Module file="$ (InPath) \ProjectName.dll">
  <SkipNamespace name="MyNamespace" />
</Module>
```

Выполнить обфускацию, изучить полученный результат.

5.3.4 Поочередно опробовать обфускацию со строками, сравнить результаты

```
<SkipType name="MyNamespace.MyClass" />
<SkipMethod type="MyNamespace.MyClass" name="SomeMethod" />
<SkipField type="MyNamespace.MyClass" name="Fieldname" />
```

5.4 Обфускация оконного приложения

5.4.1 Загрузить проект оконного приложения, например <https://github.com/ReyRom-Edu/GameStore> или любой свой

5.4.2 Создать в проекте файл `obfuscar.xml`, скопировать в него конфигурацию, изменить значение параметра `KeepPublicApi` на `true`, выполнить обфускацию, изучить полученный результат.

5.4.3 Скопировать в папку с обфусцированным `dll` остальные файлы проекта. Проверить запуск приложения при помощи `exe`-файла.

## 6 Порядок выполнения работы

6.1 Повторить теоретический материал п. 3.1;

6.2 Выполнить задания 5.1-5.8

6.3 Ответить на контрольные вопросы п. 8;

6.4 Заполнить отчет п. 7.

## 7 Содержание отчета

7.1 Титульный лист;

7.2 Цель работы;

7.3 Таблица п. 5.2

7.4 Ответы на контрольные вопросы п. 6.3;

7.5 Вывод по проделанной работе.

## **8 Контрольные вопросы**

- 8.1 Для чего применяется обфускация?
- 8.2 Как работает обфускация?