ISSN 2225-9309 (Print)
2024. № 1 (30). С. 88–93
Молодежный Вестник УГАТУ
http://journal.ugatu.su

УДК 004.4

Веб-приложение для защиты ПО от несанкционированного копирования: обфускация кода и управление лицензиями

Г. О. Орлов[1]

[1] orlovgleb99@mail.ru

ФГБОУ ВО «Уфимский университет науки и технологий» (УУНиТ)

Аннотация. В рамках защиты ПО от несанкционированного копирования и пиратства была описана структура системы защиты, реализованная в виде веб-приложения, состоящего из двух модулей – обфускатора кода и генератора лицензионных ключей. Представлен алгоритм обфускации исходного кода с применением ложных строк, имитирующих верификацию ключа. Алгоритм составлен таким образом, чтобы запутывать код без серьезных потерь в быстродействии. Описана структура модуля генерации лицензионных ключей, предложен способ получения уникальных идентификаторов оборудования пользователей для дальнейшего внедрения этих идентификаторов в лицензионные ключи с целью повышения надежности защиты.

Ключевые слова: веб-приложение; обфускация; защита от пиратства; онлайн-активация ПО; управление лицензиями; python; Django.

ВВЕДЕНИЕ

В данной статье описывается реализация системы защиты ПО от нелегального копирования в виде веб-приложения. Под несанкционированным копированием ПО в данной работе будет пониматься копирование, распространение, использование и (или) изменение программных продуктов без согласия правообладателя (обладателя авторских прав) [1]. Основной целью исследования является обеспечение защиты ПО от нелегального копирования, а основной задачей – разработка модулей веб-приложения, которые обеспечат эту защиту посредством шифрования кода и генерации лицензионных ключей.

Данная проблема является актуальной, поскольку с ростом технических возможностей ЭВМ злоумышленникам становится проще подобрать необходимые ключи шифрования на любом этапе алгоритма при использовании, например, реверс-инжиниринга. Кроме того, на сегодняшний день мы имеем крайне увеличенные киберриски, в том числе потому, что число пользователей в интернете с 2000 г. выросло почти на 1000% [2]. Следовательно, совершенствовать и усложнять для злоумышленника алгоритмы шифрования данных необходимо. Для этого в данной работе разработаны модуль шифрования программного кода с усложненным алгоритмом обфускации, а также концепция надежного модуля генерации и проверки лицензионных ключей.

ОБЩАЯ СТРУКТУРА ЗАЩИТНОГО ВЕБ-ПРИЛОЖЕНИЯ

Код защищаемой программы передается через текстовую форму методом POST, затем поступает как список строк и обрабатывается в соответствии с разработанным алгоритмом обфускации. После обработки код передается в шаблон, и разработчик, сделавший запрос на обфускацию, увидит результат обработки на странице сайта.

После того как владелец программы выполнит запрос на генерацию ключа, создается ключ с учетом данных о пользователе (это может быть имя пользователя или ID его оборудования), затем ключ добавляется в базу и выдается разработчику. Общая структура приложения представлена на схеме (рис. 1).

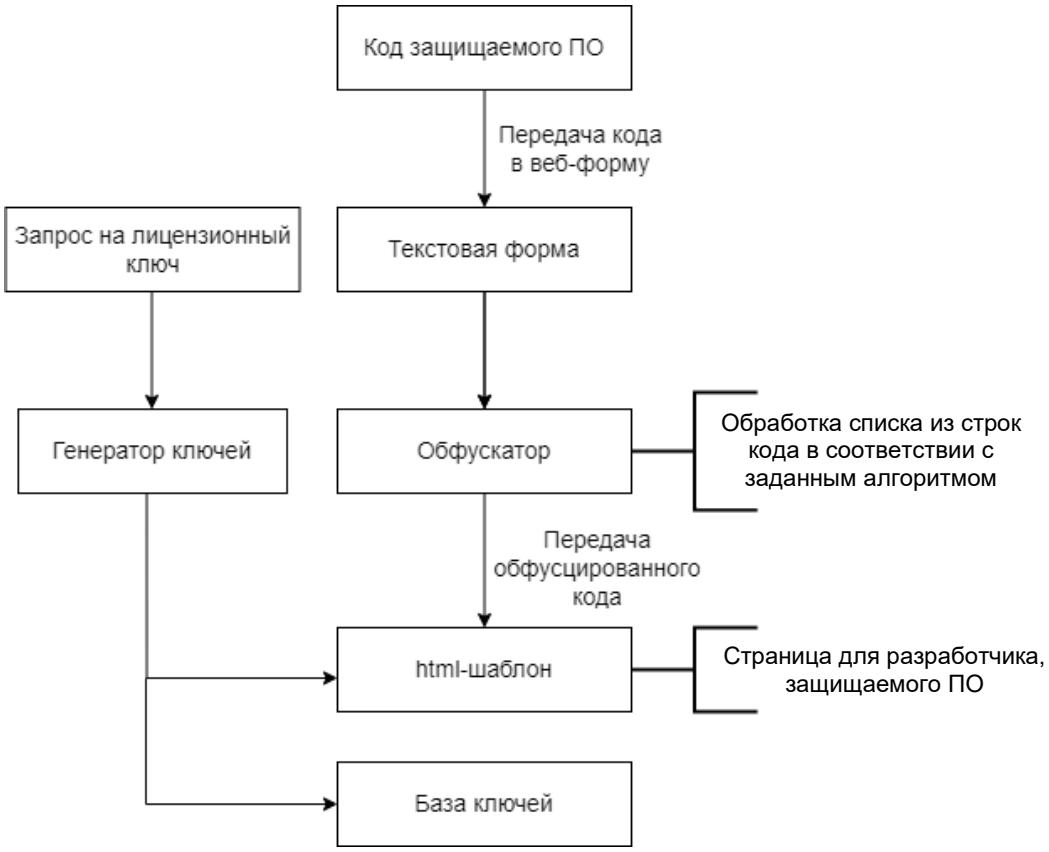

Рис. 1. Структура защитного приложения

АЛГОРИТМ РАБОТЫ МОДУЛЯ ОБФУСКАЦИИ КОДА

Работа предлагаемого обфускатора кода основана на внедрении ложных строк в код защищаемой программы. Ложные строки не внедряются в тех местах кода, где это может слишком сильно сказаться на быстродействии, например, в длинных циклах. Также они никак не влияют на логику программы. Однако они имитируют те секции кода, которые требуется изменить для взлома защитного модуля ПО, чем дополнительно запутывают злоумышленника.

Более детально алгоритм обфускации кода представлен на блок-схеме (рис. 2).

code[i] на блок-схеме – это переменные, в которые считываются строки исходного кода. В зависимости от значений code[i], переменная flag принимает значение 1 либо 0. Ложная строка будет добавлена только при flag, равном 1. Это нужно, чтобы не позволить обфускатору вставить лишние строки в тех местах, где это нарушит логику программы или внутри тела цикла.

Подфункция CreateStr() генерирует лишние строки в программе, имея заготовленный набор названий переменных, по смыслу похожих на те, которые использовались бы при верификации ключа. Каждый раз названия для них выбираются случайно.

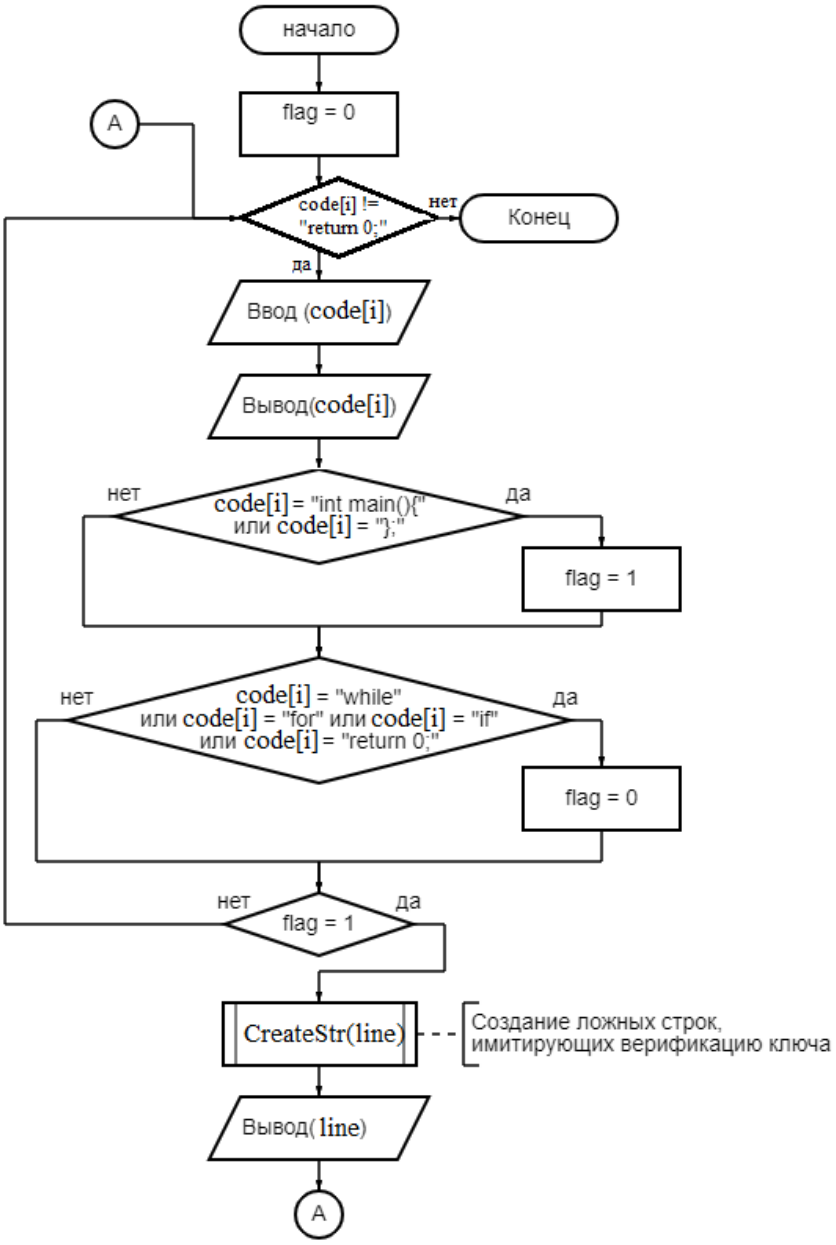

Рис. 2. Блок-схема алгоритма работы обфускатора

Ниже приведен фрагмент кода, реализующий данную подфункцию на языке Python в двух вариациях, обе из которых задействуются в алгоритме:

```
  def CreateStr1(var):
    trick = "int " + var + "=" + str(randint(1000, 5000)) + ";"
    return trick

def CreateStr2(var1, var2):
    trick = "if(" + var1 + " == " + var2 + "){bool valid = 1;}"
    return trick
```

Первая вариация имитирует строку инициализации ключа, а вторая – верификацию ключа. Переменная var, которая передается как аргумент, это имя переменной для ложной строки, которое генерируется отдельной подфункцией CreateVarName(). Ниже приведен фрагмент кода для данной подфункции:

```
  def CreateVarName():
case = randint(1, 3)
if (case == 1):
   trick = "user_key" + str(randint(1, 100))
if (case == 2):
   trick = "valid_key" + str(randint(1, 100))
if (case == 3):
   trick = "license_key" + str(randint(1, 100))
return trick
```

Данный алгоритм предназначен для обфускации кода, написанного на языке C++. Помимо предложенного обфускатора, обфускация через лишние строки также применялась в решении от компании StarForce [3].

АЛГОРИТМ РАБОТЫ МОДУЛЯ ГЕНЕРАЦИИ И ПРОВЕРКИ ЛИЦЕНЗИОННЫХ КЛЮЧЕЙ

При создании лицензионных ключей соответствующий модуль приложения должен использовать уникальный пользовательский ID – рекомендуется использовать ID машины пользователя, это значение может быть получено из реестра с применением специальных функций. К примеру, в случае Python можно использовать функцию uuid.getnode, ниже приведен фрагмент кода с примером использования данной функции, который представлен на электронном ресурсе stackoverflow.com [4]:

```
import uuid
uuid.UUID(int=uuid.getnode())
```

После того как ID пользователя получен, он добавляется к сгенерированному случайным образом лицензионному ключу и шифруется вместе с ним секретным ключом, таким образом формируя итоговый лицензионный ключ. Процесс формирования лицензионного ключа представлен на схеме (рис. 3).

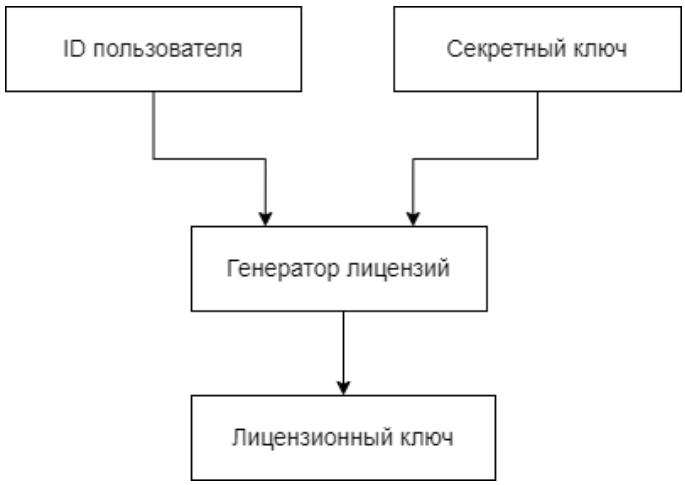

Рис. 3. Схема формирования лицензионного ключа

Затем лицензионный ключ заносится в базу ключей. Когда пользователь будет применять ключ, на сервере нужно проверить, соответствует ли ID пользователя заложенному в ключе и корректен ли сам ключ. Таким образом, повторно использовать один и тот же ключ будет намного сложнее.

ОПИСАНИЕ АРХИТЕКТУРЫ ВЕБ-ПРИЛОЖЕНИЯ

Если обобщить, есть два варианта реализации архитектуры в рамках поставленных задач: через микросервисную и монолитную архитектуры. Монолит может быть реализован через MVC, MVP или MVVM шаблон [5].

Для разработки защитного приложения была выбрана монолитная архитектура, а именно – MVC. Хранить требуется только сгенерированные лицензионные ключи, а обработка запросов на запутывание исходного кода и генерацию лицензионного ключа не являются ресурсоемкими операциями, поэтому масштабирование проекта не потребуется. Также разрабатываемое приложение состоит всего из двух модулей.

Для разработки веб-приложения использовался фреймворк Django. Модель MVC выражена в Django так, что Model – это созданные модели(Models), View(отображение) – это HTML шаблоны(Templates), а Controller, отвечающий за взаимодействие с пользователем реализован во view файле приложения.

То, как выбранная архитектура будет интегрирована, представлено на диаграмме развертывания (рис. 4).

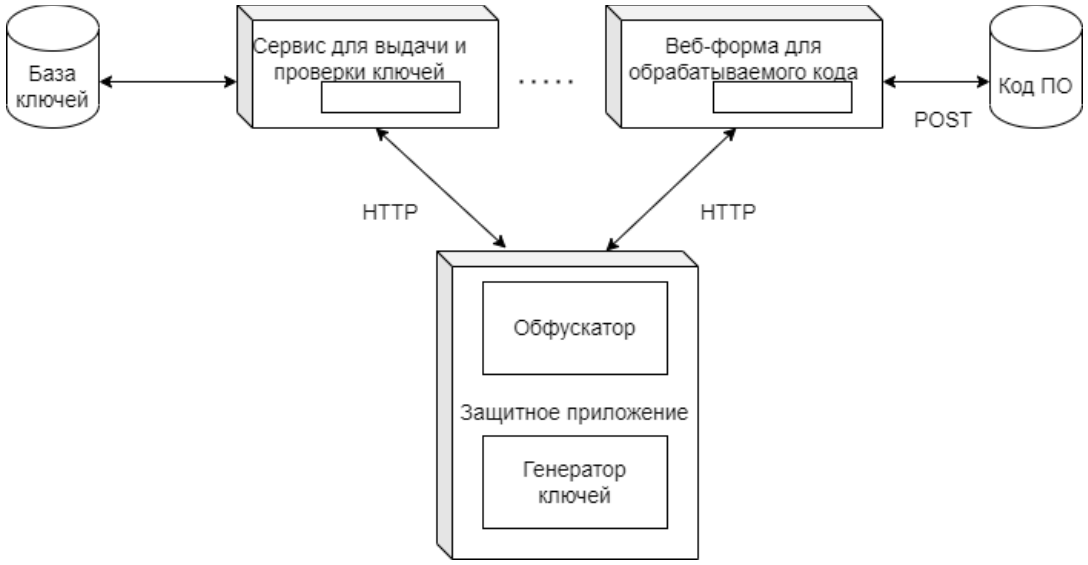

Рис. 4. Диаграмма развертывания

В качестве контроллера выступает файл view.py в проекте, в котором содержится описание поведения приложения при переходе на страницу обфускатора. В нем также прописаны логика самого обфускатора и метод render, который посылает результаты обработки защищаемого кода и работы генератора ключей в HTML – шаблон. При этом приложение для управления лицензиями обращается к БД ключей для записи нового сгенерированного ключа или для сверки пользовательского ключа с базой, если это пользовательский сценарий.

ЗАКЛЮЧЕНИЕ

В данной статье представлены результаты разработки системы защиты ПО от нелегального копирования в виде веб-приложения на базе Python Django. Приложение состоит из модуля обфускации кода и модуля управления лицензиями.

Представленный модуль обфускации кода использует лишние строки, имитирующие верификацию ключа, чтобы запутать злоумышленника. Также предусмотрены исключения для срабатывания алгоритма, чтобы не нарушать логику и не снижать производительность ПО слишком сильно.

Представленная концепция модуля управления лицензиями предполагает создание лицензионных ключей на основе уникального ID оборудования пользователя для предотвращения несанкционированного использования лицензионного ключа.

СПИСОК ЛИТЕРАТУРЫ

1. **Рогозин В. Ю.** Несанкционированное копирование программ как тип несанкционированного доступа [Электронный ресурс]. URL: https://bstudy.net/829574/pravo/nesanktsionirovannoe_kopirovanie_programm_nesanktsionirovannogo_dostupa (дата обращения 12.03.2023).

2. **Шваб К.** Технологии четвертой промышленной революции. М.:БОМБОРА, 2019, с. 138.

3. StarForce C++ Obfuscator [Электронный ресурс]. URL: https://www.star-force.ru/products/starforce-obfuscator/ (дата обращения 13.03.2012).

4. Get a unique computer ID in Python on windows and linux [Электронный ресурс]. URL: https://stackoverflow.com/questions/2461141/get-a-unique-computer-id-in-python-on-windows-and-linux (дата обращения 13.03.2023).

5. **Юзвик А. И.** Архитектура приложений и интеграции [Электронный ресурс]. URL: https://habr.com/ru/company/itq_group/blog/705598/ (дата обращения 14.03.2023).

.

ОБ АВТОРАХ

ОРЛОВ Глеб Олегович, магистрант каф. ВМиК. Дипл. специалист по защите информации (УГАТУ, 2021).

METADATA

Title: Web application for protection of software from unauthorized copying: code obfuscation and license management.

Author: G. O. Orlov [1]

Affiliation:

[1] Ufa University of Science and Technology (UUST), Russia.

Email: [1] orlovgleb99@mail.ru

Language: Russian.

Source: Molodezhnyj Vestnik UGATU (scientific journal of Ufa University of Science and Technology), no. 1 (30), pp. 88-93, 2024. ISSN 2225-9309 (Print).

Abstract: Within the framework of software protection from unauthorized copying and piracy, the structure of the protection system was described, implemented as a web application consisting of two modules – code obfuscator and license key generator. The algorithm of source code obfuscation with the use of false strings imitating key verification is presented. The algorithm is designed in such a way that it can obfuscate the code without serious losses in performance. The structure of the module of generation of license keys is described, the method of obtaining unique identifiers of users' equipment is proposed for further implementation of these identifiers in license keys in order to improve the reliability of protection.

Key words: web application; obfuscation; piracy protection; online software activation; license management; python; Django.

About authors:

ORLOV Gleb Olegovich, Master's student at the Department of Computational Mathematics and Cybernetics. Diploma of a specialist in information protection (Ufa State Aviation Technical University, 2021).