МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«**Вятский государственный университет**»

**(«ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Защита от копирования

Отчет по лабораторной работе №1

по дисциплине «Защита информации»

Выполнил студент группы ИВТб-41 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Жеребцов К. А./

Проверил старший преподаватель кафедры ЭВМ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Караваева О. В./

Киров 2024

1. Задание на лабораторную работу

Разработать программу, защищённую от копирования

1. Алгоритм

Сервер (server.py):

Сервер имеет 2 функции:

1. активация:

Эта функция используется активатором для создания лицензионного ключа на основе MAC-адреса компьютера, на котором проводится активация. Данная функция сохраняет ключ, который ей присылает активатор, для последующих проверок активации, выполняющихся при запуске основной программы.

1. проверка активации:

Данная функция вызывается основной программой при запуске. Она сравнивает ключ, который сгенерировала программа с ключами, которые записаны на сервере. Если такой ключ присутствует, то программа продолжает работу, а если нет, то выводит сообщение о необходимости провести активацию.

Активатор (activator.py):

Активатор получает MAC-адрес компьютера, на котором проводится активация, и хэширует его. Дальше он делает запрос типа POST на сервер и посылает ему ключ, то есть хэшированый MAC-адрес.

Основная программа (main.py):

Основная программа так же, как и активатор, создает свой ключ, то есть получает МАС-адрес и хэширует его. После чего она делает запрос на сервер, посылая ему ключ для проверки. Сервер в свою очередь проверяет наличие такого ключа у себя и отправляет ответ обратно с разрешением или запретом работы.

Таким образом, схема такова:

активатор создает ключ и отправляет его на сервер с целью сохранить его. Основная программа так же создает ключ, отправляет его на сервер, но не для сохранения, а для сравнения с уже имеющимися ключами. Следовательно, если запустить основную программу без активатора, то нужного ключа не будет на сервере, и сервер отправит ответ за запертом дальнейшей работы.

1. Код

Сервер (server.py):

from flask import Flask, request, jsonify

def print\_database():

    print("======================================================\n")

    print("Current database:")

    for key, value in activation\_status.items():

        print(f"License key: {key}")

    print("======================================================\n")

app = Flask(\_\_name\_\_)

activation\_status = {}

@app.route('/activate', methods=['POST'])

def activate():

    data = request.get\_json()

    license\_key = data.get('license\_key')

    if license\_key in activation\_status:

        return jsonify({"success": False, "message": "License key already activated"})

    activation\_status[license\_key] = license\_key

    print\_database()

    return jsonify({"success": True, "message": "Activation successful"})

@app.route('/check\_activation', methods=['POST'])

def check\_activation():

    data = request.get\_json()

    license\_key = data.get('license\_key')

    if license\_key in activation\_status:

        if activation\_status[license\_key] == license\_key:

            return jsonify({"activated": True})

        else:

            return jsonify({"activated": False, "message": "This license key is activated on another device"})

    else:

        return jsonify({"activated": False, "message": "License key is not activated"})

if \_\_name\_\_ == '\_\_main\_\_':

    app.run()

Активатор (activator.py):

import requests

import hashlib

import psutil

def get\_mac\_address():

    try:

        mac = psutil.net\_if\_addrs()['Wi-Fi'][0].address

    except KeyError:

        try:

            mac = psutil.net\_if\_addrs()['Ethernet'][0].address

        except KeyError:

            mac = 'unknown'

    return mac

def generate\_license\_key(mac\_address):

    key = mac\_address.encode('utf-8')

    hashed\_key = hashlib.sha256(key).hexdigest()

    return hashed\_key

def activate\_program():

    url = 'http://127.0.0.1:5000/activate'

    hardware\_id = get\_mac\_address()

    license\_key = generate\_license\_key(hardware\_id)

    data = {'license\_key': license\_key}

    response = requests.post(url, json=data)

    if response.status\_code == 200:

        activation\_result = response.json()

        if activation\_result['success']:

            print("Программа успешно активирована.")

        else:

            print("Не удалось активировать программу:", activation\_result['message'])

    else:

        print("Не удалось подключиться к серверу активации.")

if \_\_name\_\_ == "\_\_main\_\_":

    activate\_program()

Основная программа (main.py):

import requests

import hashlib

import psutil

def get\_mac\_address():

    try:

        mac = psutil.net\_if\_addrs()['Wi-Fi'][0].address

    except KeyError:

        try:

            mac = psutil.net\_if\_addrs()['Ethernet'][0].address

        except KeyError:

            mac = 'unknown'

    return mac

def generate\_license\_key(mac\_address):

    key = mac\_address.encode('utf-8')

    hashed\_key = hashlib.sha256(key).hexdigest()

    return hashed\_key

def check\_activation():

    url = 'http://127.0.0.1:5000/check\_activation'

    hardware\_id = get\_mac\_address()

    license\_key = generate\_license\_key(hardware\_id)

    data = {'license\_key': license\_key}

    response = requests.post(url, json=data)

    if response.status\_code == 200:

        activation\_result = response.json()

        if activation\_result['activated']:

            print("Программа активирована на данном компьютере.")

        else:

            print("Программа не активирована на данном компьютере. Обратитесь к администратору.")

    else:

        print("Не удалось подключиться к серверу активации.")

if \_\_name\_\_ == "\_\_main\_\_":

    check\_activation()

1. Вывод

В ходе выполнения лабораторной работы была разработана защищённая от копирования программа. Для реализации защиты программы использовались характеристики компьютера, а именно MAC-адрес, на котором устанавливалась программа и создавался лицензионный ключ. Также был создан сервер, который подтверждал активацию программы на определенном компьютере.