

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Вятский государственный университет»

Факультет автоматики и вычислительной техники

Кафедра электронно-вычислительных машин

Отчет по лабораторной работе №1 дисциплины
«Защита информации»

Выполнил студент группы ИВТ-42 _____ / Рзаев А.Э./
Проверил преподаватель _____ / Караева О.В./

Киров 2020

1 Задание

Реализовать защиту от копирования.

2 Ход работы

Для защиты программы от копирования используется привязка к аппаратуре. Вместе с основной программой поставляется программа-активатор. Также действует сервер активации. Программа-активатор с помощью средств ОС получает идентификатор процессора, вычисляет от него хэш по алгоритму SHA-256. Далее создается цифровая подпись от этого хэша, которая вместе с идентификатором процесса отправляется на сервер активации, где выполняется проверка цифровой подписи и, в случае если подпись верна, идентификатор записывается в базу данных.

Основная программа при очередном запуске получает идентификатор процессора и отправляет его на сервер активации. Если на сервере активации в базе данных этот идентификатор есть, то отправляется положительный ответ, в противном случае – отрицательный. При положительном ответе программа продолжает свое нормальное выполнение, иначе выводит сообщение о том, что программа не активирована.

Основная программа, программа-активатор и сервер активации были реализованы на языке Python. Листинг кода представлен в приложении А.

Экранные формы, демонстрирующие работу системы защиты, представлены в приложении Б.

3 Вывод

В ходе выполнения лабораторной работы были изучены методы защиты информации от копирования. Был изучен способ получения и использования информации об аппаратуре для реализации привязки программы к конкретному устройству. Был реализован способ защиты программы от копирования с использованием цифровой подписи и сервера активации.

Приложение А

Листинг кода

Основная программа

```
import requests

def pcid() -> str:
    import cpuid
    from functools import reduce

    regs = cpuid.CPUID()(1)

    return str(reduce(lambda x, y: x ^ y, [regs[0], regs[2], regs[3]]))

SERVER_URL = 'http://localhost:5000'

id_ = pcid.pcid()

try:
    response = requests.get(f'{SERVER_URL}/check', params={
        'id': id_
    }).text

    if response == 'TRUE':
        print('Program is activated')
    else:
        print('Program is not activated')
except:
    print('Can not check activation status')
```

Сервер активации

```
import pickle
import os.path as path
import base64
import rsa
import flask
import sqlite3
import functools

def transact(fun):
    @functools.wraps(fun)
    def wrapper():
        conn = sqlite3.connect('ids.db')
        cursor = conn.cursor()
        try:
            ret = fun(cursor=cursor)
            conn.commit()
            return ret
        except Exception as ex:
            conn.rollback()
            raise ex
        finally:
            conn.close()

    return wrapper

PUBKEY_PATH = 'public.pem'

with open(PUBKEY_PATH, 'rb') as keyfile:
    pubkey = rsa.PublicKey.load_pkcs1(keyfile.read())

# IDS_PATH = 'ids.pickle'

# ids = set()

# if path.exists(IDS_PATH) and path.isfile(IDS_PATH):
#     with open(IDS_PATH, 'rb') as file:
#         ids = pickle.load(file)

app = flask.Flask(__name__)

@app.route('/check', methods=['GET'])
@transact
def check(cursor):
    id_ = flask.request.args.get('id', None)

    if id_ is None:
        return 'FALSE'

    # if id_ not in ids:
    #     return 'FALSE'
    cursor.execute('SELECT id FROM ids WHERE id=?', (id_,))
    row = cursor.fetchone()
    if row is None:
        return 'FALSE'

    return 'TRUE'

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

```

@transact
def register(cursor):
    if not flask.request.is_json:
        return 'FAILED'

    try:
        data = flask.request.json

        id_ = data['id']
        signature = base64.b64decode(data['signature'])

        rsa.verify(id_.encode('utf-8'), signature, pubkey)

        # ids.add(id_)

        # with open(IDS_PATH, 'wb') as file:
        #     pickle.dump(ids, file)
        cursor.execute('INSERT OR IGNORE INTO ids VALUES (?)', (id_,))

        return 'OK'
    except Exception as ex:
        return 'FAILED'

@app.cli.command('init-db')
@transact
def initdb(cursor):
    cursor.execute('''CREATE TABLE IF NOT EXISTS ids (id TEXT PRIMARY
KEY)''')

```

Программа-активатор

```
import base64
import rsa
import requests

def pcid() -> str:
    import cpuid
    from functools import reduce

    regs = cpuid.CPUID()(1)

    return str(reduce(lambda x, y: x ^ y, [regs[0], regs[2], regs[3]]))

SERVER_URL = 'http://localhost:5000'

PRIVATEKEY_PATH = 'private.pem'

with open(PRIVATEKEY_PATH, 'rb') as keyfile:
    privatekey = rsa.PrivateKey.load_pkcs1(keyfile.read())

id_ = pcid.pcid()

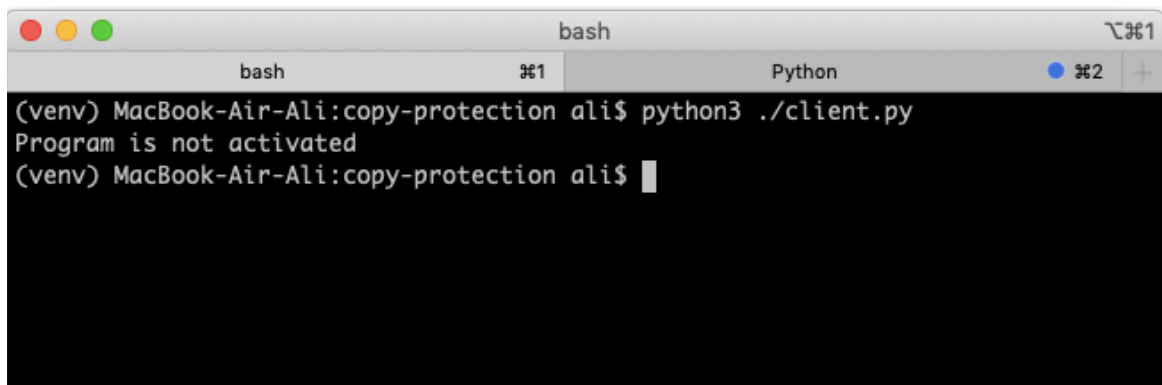
signature = rsa.sign(id_.encode('utf-8'), privatekey, 'SHA-256')

try:
    response = requests.post(f'{SERVER_URL}/register', json={
        'id': id_,
        'signature': base64.b64encode(signature).decode('utf-8')
    }).text

    if response == 'OK':
        print('Successfully activated')
    else:
        print('An error occurred during activation')
except:
    print('Can not establish connection with activation server')
```

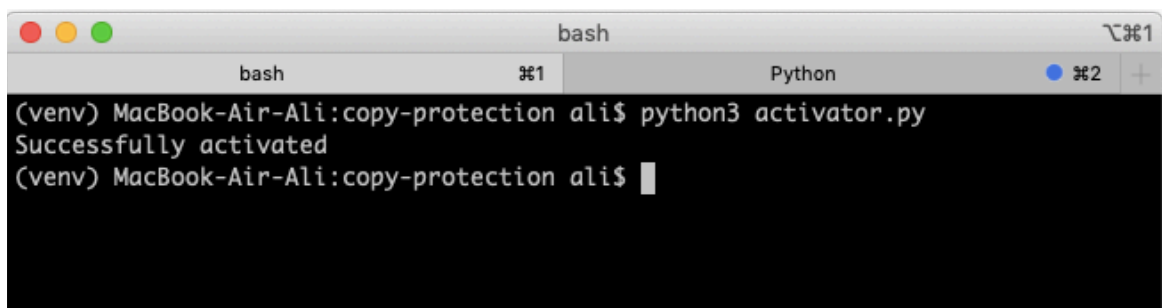
Приложение Б

Экранные формы



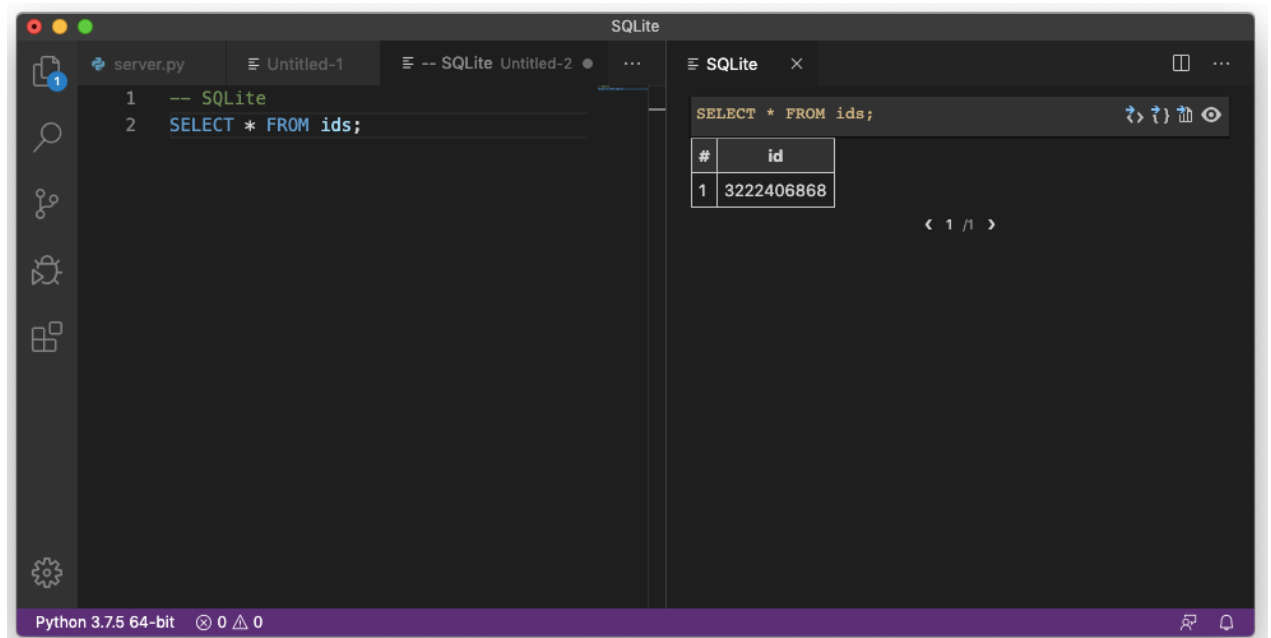
```
bash
Python
(venv) MacBook-Air-Ali:copy-protection ali$ python3 ./client.py
Program is not activated
(venv) MacBook-Air-Ali:copy-protection ali$
```

Рисунок 1 – Попытка запуска нелицензионной копии



```
bash
Python
(venv) MacBook-Air-Ali:copy-protection ali$ python3 activator.py
Successfully activated
(venv) MacBook-Air-Ali:copy-protection ali$
```

Рисунок 2 – Активация программы

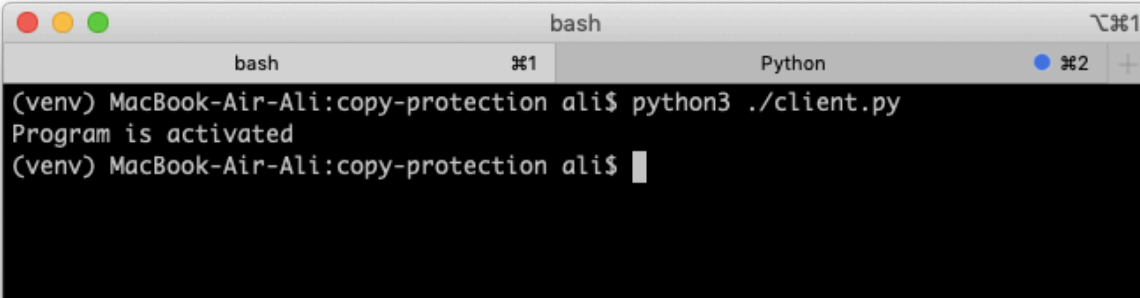


```
server.py
Untitled-1
-- SQLite
SELECT * FROM ids;
```

#	id
1	3222406868

Python 3.7.5 64-bit 0 0 0

Рисунок 3 – Идентификатор процессора в базе данных



A screenshot of a macOS terminal window. The window has a title bar with three colored buttons (red, yellow, green) on the left and a title 'bash' on the right. Below the title bar is a tab bar with two tabs: 'bash' (selected) and 'Python'. The terminal content shows a user prompt '(venv) MacBook-Air-Ali:copy-protection ali\$' followed by the command 'python3 ./client.py'. The output of the command is 'Program is activated'. The prompt is then shown again with a cursor: '(venv) MacBook-Air-Ali:copy-protection ali\$'.

```
(venv) MacBook-Air-Ali:copy-protection ali$ python3 ./client.py
Program is activated
(venv) MacBook-Air-Ali:copy-protection ali$
```

Рисунок 4 – Запуск приложения