

Міністерство освіти і науки України
Національний технічний університет «ХПІ»
Навчально-науковий інститут комп'ютерних наук та інформаційних
технологій
Кафедра комп'ютерної інженерії та програмування

ЗВІТ

з лабораторної роботи № 12
з дисципліни «Сучасні технології безпечного програмування»
**«ДОСЛІДЖЕННЯ ЗАСОБІВ ОБФУСКАЦІЇ ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ»**

Виконав:
студент гр. КН-Н9226
Кулик Д.І.

Перевірів:
Бульба С. С.

Мета роботи: Дослідити існуючі утиліти обфускації програмного забезпечення.

Індивідуальне завдання

Для обраної мови програмування слід обфускувати будь-який проект.

Слід використовувати декілька утиліт для обфускації. Мінімальна кількість обфускаторів - 4. У звіті навести порівняльну характеристику обраних обфускаторів, що складатиметься з:

- інформації щодо використовуваних методів обфускації
- плюсів та мінусів кожного з обфускаторів.

Навести опис використання кожного з обраних обфускаторів.

Хід роботи

Обфускація в широкому значенні - приведення вихідного тексту або виконуваного коду програми до виду, що зберігає її функціональність, але утруднює аналіз, розуміння алгоритмів роботи та модифікацію при декомпіляції.

Більшість методів обфускації перетворюють такі аспекти коду:

- Дані: роблять елементи коду схожими на те, чим вони не є
- Потік коду: виставляють логіку програми абсурдної або навіть недетермінованої.
- Структура формату: застосовують різне форматування даних, перейменування ідентифікаторів, видалення коментарів коду тощо.

Інструменти обфускації можуть працювати як із source або байт кодом, так і з бінарним, проте обфускація двійкових файлів складніша, і повинна змінюватись в залежності від архітектури системи.

При обфускації коду важливо правильно оцінити, які частини коли можна ефективно заплутати. Слід уникати обфускації коду критичного щодо продуктивності.

```

def generate_hash(message: bytearray) -> bytearray:
    """Повертає хеш SHA-256 із переданого повідомлення.
    Аргумент має бути об'єктом bytes, bytearray або string."""

    if isinstance(message, str):
        message = bytearray(message, 'ascii')
    elif isinstance(message, bytes):
        message = bytearray(message)
    elif not isinstance(message, bytearray):
        raise TypeError

    # Заповнювання
    length = len(message) * 8 # len(message) - це кількість байтів

    message.append(0x80)

    # перетворене повідомлення заповнити нулями доки дані не стануть кратними 512 без останніх 64 біт
    while (len(message) * 8 + 64) % 512 != 0:
        message.append(0x00)

    message += length.to_bytes(8, 'big') # доповнення до 8 байтів або 64 бітів

    assert (len(message) * 8) % 512 == 0, "Padding did not complete properly!"

```

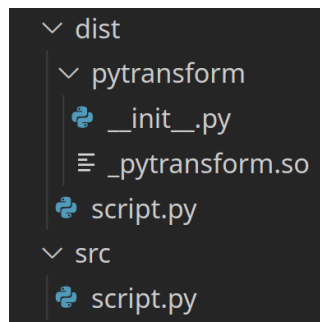
Рисунок 1 – Початковий код (лабораторна робота 01)

Проведемо обфускацію за допомогою утиліти pyarmor

Заплатувати сценарій Python за допомогою PyArmor так само просто, як виконати таку команду в консолі:

pyarmor obfuscate script.py

Це створить папку dist, яка містить обфускований файл Python з такою самою назвою, як і вихідний, разом із залежностями середовища виконання, як показано в структурі папок нижче:



- Обфускація рядкових літералів непридатна для конфіденційної інформації, оскільки її можна тривіально зламати.

```
(stbp) C:\Users\Daniil\PycharmProjects\stbp\LABS\kulyk12>python opy.py
Opy (TM) Configurable Multi Module Python Obfuscator Version 1.1.28
Copyright (C) Geatec Engineering. License: Apache 2.0 at http://www.apache.org/licenses/LICENSE-2.0
```

Obfuscated words: 180

```
(stbp) C:\Users\Daniil\PycharmProjects\stbp\LABS\kulyk12>
```

Рисунок 5 – Виконання команди для обфускації

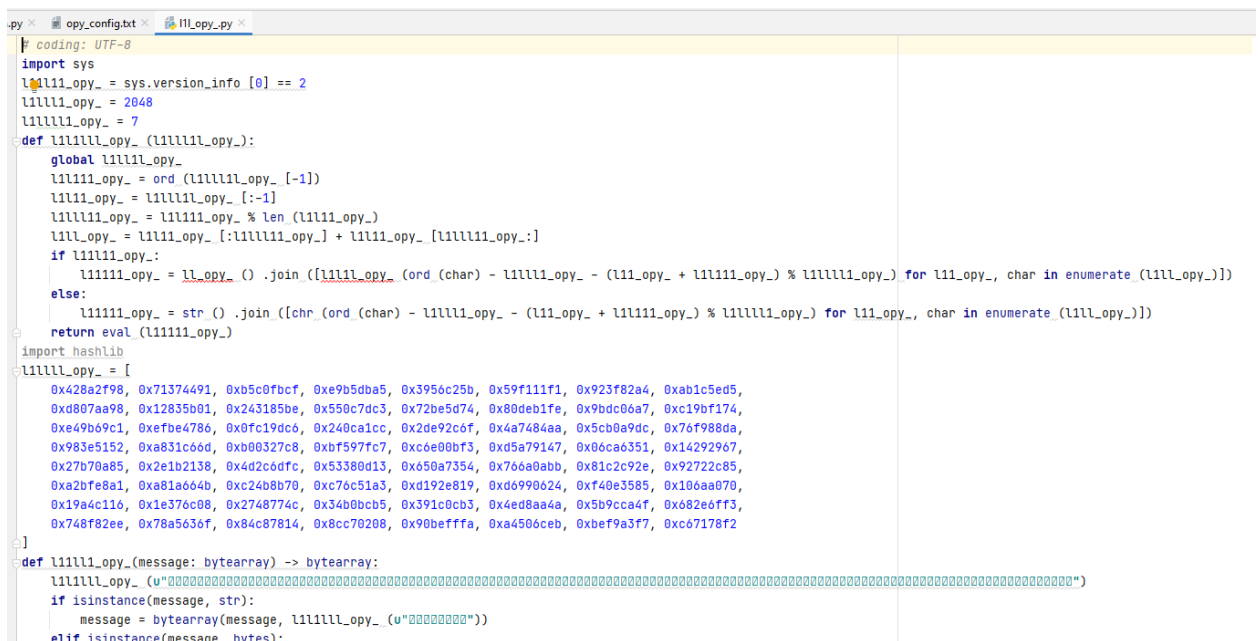


Рисунок 6 – Обфускація коду за допомогою ору

```
(stbp) C:\Users\Daniil\PycharmProjects\stbp>python LABS\kulyk12_opy\l1l1l_opy_.py
Введений текст: Daniil Kulyk
Розроблений SHA256 = 88d44f1bf7c5886b9f0eb28963037d7d9fff9e99a7c24a869124fede16bc46b8
Hashlib SHA256      = 88d44f1bf7c5886b9f0eb28963037d7d9fff9e99a7c24a869124fede16bc46b8
```

Рисунок 7 – Успішний запуск обфускованого файлу

Висновки: в результаті виконання лабораторної роботи було досліджено існуючі утиліти обфускації програмного забезпечення.