Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Тульский государственный университет»

КАФЕДРА ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

**ЗАЩИТА ОТ РАЗРУШАЮЩИХ ПРОГРАММНЫХ ВОЗДЕЙСТВИЙ, ЗАЩИТА ПРОГРАММ ОТ ИЗМЕНЕНИЯ И КОНТРОЛЬ ЦЕЛОСТНОСТИ**

отчет о   
лабораторной работе №1

по дисциплине

*ПРОГРАММНЫЕ СРЕДСТВА ЗАЩИТЫ ИНФОРМАЦИИ*

***ВАРИАНТ 13***

|  |  |  |
| --- | --- | --- |
| Выполнила: | ст. гр. 230711 | Павлова В.С. |
| Проверил: | асс. каф. ИБ | Греков М.М. |

Тула, 2023 г.

# **ЦЕЛЬ И ЗАДАЧА РАБОТЫ**

**Цель:** познакомиться с общими принципами защиты программного обеспечения и способами организации контроля целостности исполняемых модулей и важных программных данных.

**Задача:** в данной работе требуется написать программу, демонстрирующую использование изученных принципов.

# **ЗАДАНИЕ НА РАБОТУ**

Написать программу для контроля целостности используемой Windows библиотеки.

# **СХЕМА АЛГОРИТМА**

Схема алгоритма программы для контроля целостности библиотеки, представлена на рисунке 1.



Рисунок 1 – Схема алгоритма программы

# **ТЕКСТ ПРОГРАММЫ**

Текст программы на языке программирования С++ для контроля целостности библиотеки представлен в листинге 1.

## **Листинг 1. Текст программы**

#include <iostream>

#include <fstream>

#include <string>

#include <cstring>

#include <openssl/evp.h>

#include <openssl/md5.h>

using namespace std;

int main()

## **Листинг 1. Текст программы (продолжение)**

{

string path = "C:\\Windows\\RtlExUpd.dll";

ifstream file(path, ios::binary);

if (!file)

{

cout << "Error: Unable to open file " << path << "\n";

return 1;

}

EVP\_MD\_CTX\* md5\_context = EVP\_MD\_CTX\_new();

EVP\_DigestInit\_ex(md5\_context, EVP\_md5(), nullptr);

char buf[1024];

while (file.read(buf, sizeof(buf)).gcount() > 0)

{

EVP\_DigestUpdate(md5\_context, buf, file.gcount());

}

unsigned char hash[EVP\_MAX\_MD\_SIZE];

unsigned int hashLength;

EVP\_DigestFinal\_ex(md5\_context, hash, &hashLength);

cout << "The hash sum is: ";

for (int i = 0; i < hashLength; i++)

{

cout << hex << (int)hash[i];

} cout << "\n";

EVP\_MD\_CTX\_free(md5\_context);

file.close();

return 0;

}

# **ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЯ**

Данная программа предназначена для контроля целостности библиотеки. По заданному пути производится расчёт хэш-суммы соответствующего файла и выводится на экран в консоль.

# **ИНСТРУКЦИЯ ПРОГРАММИСТА**

Структуры данных, используемые в программе, приведены в таблице 1.

Таблица 1 – Структуры данных в программе

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип (класс)** | **Предназначение** |
| path | string | Полный путь к файлу |
| file | ifstream | Файловая переменная |
| md5\_context | EVP\_MD\_CTX\* | Контекст хеширования |

Таблица 1 – Структуры данных в программе (продолжение)

|  |  |  |
| --- | --- | --- |
| buf | char | Буфер для хранения данных |
| hash | unsigned char | Буфер для хранения хэш-суммы |
| hashLength | unsigned int | Длина хэш-суммы |

# **ДЕМОНСТРАЦИОННЫЙ ПРИМЕР**

Пусть имеется путь: C:\Windows\RtlExUpd.dll. Для сравнения сперва вычислим его хэш-сумму по алгоритму MD5 с помощью PowerShell. Как видно по рисунку 2, она должна быть равна D0B22CFA9BFB7635A5B5D4C46B515257.

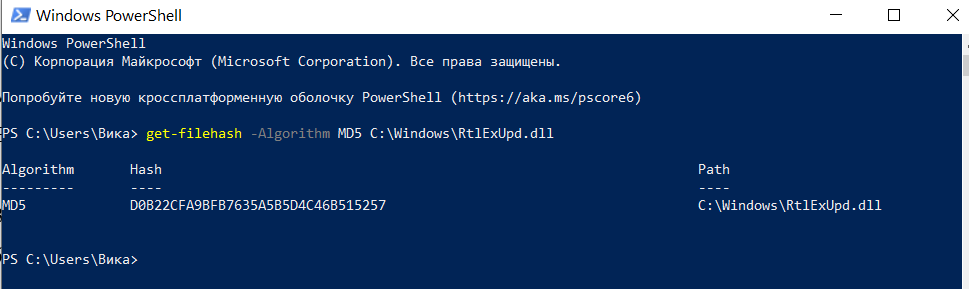


Рисунок 2 – Результат расчёта хэш-суммы с помощью PowerShell

Результат работы программы приведён на рисунке 3. Полученная хэш-сумма d0b22cfa9bfb7635a5b5d4c46b515257 соответствует расчётам с точностью до регистра.

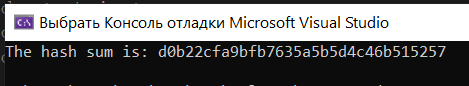


Рисунок 3 – Результат работы программы

# **ВЫВОДЫ**

В ходе данной лабораторной работы я ознакомилась с общими принципами защиты программного обеспечения и способами организации контроля целостности исполняемых модулей и важных программных данных.