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

КАФЕДРА ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

**МЕТОДЫ И СРЕДСТВА ПРИВЯЗКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ К АППАРАТНОМУ ОКРУЖЕНИЮ И ФИЗИЧЕСКИМ НОСИТЕЛЯМ**

отчет о   
лабораторной работе №2

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:** изучить способы взаимодействия программного обеспечения с операционной системой и аппаратурой для реализации привязки ПО к текущей аппаратно-программной конфигурации ЭВМ.

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

# **ЗАДАНИЕ НА РАБОТУ**

Осуществить привязку программы к аппаратной части ЭВМ, а именно: к конфигурации системы (жесткий диск).

# **ОПИСАНИЕ АЛГОРИТМА**

Реализация механизма привязки программы к жёсткому диску, который может быть использован для защиты программы от нелегального использования, выполнена в соответствии со следующим алгоритмом:

1. Программа считывает серийный номер жёсткого диска. Конкретно в данной программной реализации это происходит с помощью запросов к WMI (Windows Management Instrumentation).
2. Программа хэширует полученные данные с помощью алгоритма MD5.
3. Программа считывает данные о номере жёсткого диска, хранящиеся на сервере.
4. Программа хэширует полученные с сервера данные так же с помощью алгоритма MD5.
5. Происходит сравнение полученных хэш-сумм. При их совпадении устройство считается подтверждённым, и программе разрешено выполняться дальше.

# **ТЕКСТ ПРОГРАММЫ**

Текст программы на языке программирования С# для реализация механизма привязки программы к жёсткому диску представлен в листинге 1.

## **Листинг 1. Текст программы**

using System.Security.Cryptography;

using System.Text;

using System.Management;

namespace BindingToSoftware

{

class Program

{

static void Main(string[] args)

{

// Получаем данные о жестком диске и хэшируем их

string serialNumber = GetHardDriveSerialNumber();

byte[] hashedKey = GetHashedKey(serialNumber);

// Получаем ключевую информацию с сервера, хэшируем её

string key = GetKeyFromServer(serialNumber);

byte[] hashedKeyFromServer = GetHashedKey(key);

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

// Сравниваем полученные хэш-суммы

if (IsSerialNumberHashCorrect(hashedKey, hashedKeyFromServer))

{

Console.WriteLine("Correct hard drive serial number. Program was started successfully!");

}

else

{

Console.WriteLine("Wrong hard drive serial number. Program is not allow to start!");

return;

}

}

static bool IsSerialNumberHashCorrect(byte[] hashedKey, byte[] hashedKeyFromServer)

{

bool flag = true;

for (int i = 0; i < 16; i++)

{

flag = hashedKey[i] == hashedKeyFromServer[i];

}

return flag;

}

static byte[] GetHashedKey(string input)

{

using (MD5 md5Hash = MD5.Create())

{

byte[] data = md5Hash.ComputeHash(Encoding.UTF8.GetBytes(input));

return data;

}

}

static string GetHardDriveSerialNumber()

{

ManagementObjectSearcher searcher = new ManagementObjectSearcher("SELECT \* FROM Win32\_DiskDrive");

foreach (ManagementObject drive in searcher.Get())

{

string serialNumber = drive["SerialNumber"].ToString();

Console.WriteLine($"Current Hard Drive serial number is: {serialNumber}");

return serialNumber;

}

return null;

}

static string GetKeyFromServer(string data)

{

/\*

При использовании реального сервера:

string url = "http://example.com/get\_key.php?data=" + data;

WebClient client = new WebClient();

string key = client.DownloadString(url);

\*/

string key = File.ReadAllText($"D:\\WORK\\2 КУРС\\ПСЗИ\\ЛР2\\BindingToSoftwareApp\\server.txt");

return key;

}

}

}

# **ДЕМОНСТРАЦИОННЫЙ ПРИМЕР**

Пусть в качестве сервера для хранения ключевой информации о необходимом номере жёсткого диска будет использоваться локальный файл. Запишем в него номер текущего установленного жёсткого диска, полученный через командную строку – 0026\_B768\_439E\_EC35. Результат исполнения программы для данного номера будет следующим:

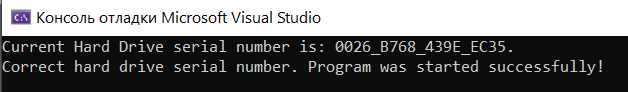


Рисунок 1 – Результат работы программы

Теперь запишем в файл какой-нибудь другой номер, отличный от установленного в системе, например – 0036\_AC68\_77F7\_2122. Из-за несовпадения номеров программа выдаст соответствующее сообщение и завершится.

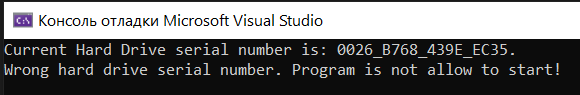


Рисунок 2 – Результат работы программы

# **ВЫВОДЫ**

В ходе данной лабораторной работы я изучила способы взаимодействия программного обеспечения с операционной системой и аппаратурой для реализации привязки ПО к текущей аппаратно-программной конфигурации ЭВМ. Это можно осуществить с помощью считывания данных из реестра, API или WMI.