ФЕДЕРАЛЬНОЕ АГЕНСТВО СВЯЗИ Федеральное государственное бюджетное образовательное учреждение высшего образования «МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ СВЯЗИ И ИНФОРМАТИКИ» Кафедра «Информационная безопасность»

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

по дисциплине

«Защита информации от вредоносного программного обеспечения»

Разработка модуля хранения антивирусных баз

Выполнила:

студентка группы БВТ1601 Фурман С.И.

Проверил:

ассистент кафедры ИБ

Барков В. В.

**Разработка модуля хранения антивирусных баз**

Антивирусные базы хранятся в двоичном файле в виде набора записей.

Файл содержит:

* последовательность байт для идентификации
* количество записей в файле
* последовательность записей

Запись включает в себя:

* название вируса
* диапазон смещения сигнатуры
* длина сигнатуры
* первые байты сигнатуры
* хеш сигнатуры
* тип файла, в котором может находиться сигнатура

***Задание***

Разработать класс, хранящий всю необходимую информацию о записи

Разработать алгоритм загрузки записей из файла антивирусных баз

Разработать алгоритм сохранения записей в файл антивирусных баз

**Текст модулей**

**Node\_Write** - класс, хранящий всю необходимую информацию о записи. **Node\_Write*.cs***

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Windows.Forms;

using System.Windows;

using System.IO;

using System.Runtime.InteropServices;

using System.Runtime.Serialization;

using System.Runtime.Serialization.Formatters.Binary;

namespace AntiVirus

{

public partial class Node\_Write

{

ScanEngine F2 = new ScanEngine();

// Вирусы, для базы вирусов

[Serializable]

public struct Virus

{

public int length; //Длина сигнатуры

public int lenght\_sec; //Длина исполняемой секции

public int first\_sec; //Начало исполняемой секции

public string hash; //Хеш

public string hash8; //Хеш первых 8 символов из файла

public string name; //Название вируса

public int detect; //Кол-во обнаружений при сканировании

public string byts\_header; //Байты идентификации заголовка

public string byts\_body; //Байты идентификации исполняемой облости

}Signatures\_List- функция, загружающая записи из файла антивирусной базы, вызывается при выборе пункта «Добавить антивирусную базу» в меню(рис. 1).

**Node\_Write*.cs***

//вывести список сигнатур

public void Signatures\_List(ListBox listbox, Label files)

{

string line;

List<Virus> sign = F2.getVirusDB();

files.Text = "Кол-во записей в файле: " + sign.Count;

listbox.Items.Clear();

for (int i = 0; i < sign.Count; i++)

{

line = "Имя: " + sign[i].name +

", хеш: " + sign[i].hash +

", хеш8: " + sign[i].hash8 +

", длина: " + sign[i].length +

", byte header: " + sign[i].byts\_header +

", byte body: " + sign[i].byts\_body +

", начало исполняемой секции: " + sign[i].first\_sec +

", длина исполняемой секции: " + sign[i].lenght\_sec +

", кол-во заражений: " + sign[i].detect;

listbox.Items.Add(line);

}

}

}

}

addSign\_Click – функция, которая вызывается при выборе пункта «Добавить сигнатуру» в меню, открывает форму, показанную на рис. 1.

private void addSign\_Click(object sender, EventArgs e)

{

addSign addSign = new addSign();

addSign.Show();

} }

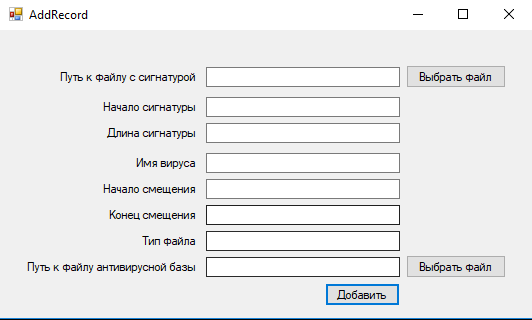


Рис. 1. Форма для добавления сигнатуры в антивирусную базу

***Node\_Write*** – класс формы для добавления сигнатуры в антивирусную базу.

***Node\_Write.cs***

private void add\_Click(object sender, EventArgs e)

{

if (path.Text == "")

{

System.Windows.MessageBox.Show("Путь до сигнатуры не указан.");

return;

}

if (name.Text != "" && offset.Text != "" && length.Text != "")

{

// заполняем структуру

Node\_Write.Virus sign;

// открываем файл, чтобы считать код и захешировать

FileStream file = new FileStream(path.Text, FileMode.Open, FileAccess.ReadWrite);

// узнаем длину исп секции и смещение исп секции от начала файла

int[] data = exeSect(file);

// смещаем файл на начало исп секции + смещение от нее

file.Seek(data[1] + Convert.ToInt32(offset.Text), SeekOrigin.Begin);

// считываем сигнатуру из файла от смещения до смещения + длина

string virusText = read(file, Convert.ToInt32(length.Text));

// хэшируем

sign.hash = hash(virusText);

// смещаем файл на начало исп секции + смещение от нее

file.Seek(data[1] + Convert.ToInt32(offset.Text), SeekOrigin.Begin);

// считываем сигнатуру из файла от смещения до смещения + длина

string virusText8 = read(file, 8);

// хэшируем

sign.hash8 = hash(virusText8);

// имя вируса

sign.name = name.Text;

// длина вируса

sign.length = Convert.ToInt32(length.Text);

// кол-во обнаружений

sign.detect = 0;

// Присвоение номера каждой сигнатуре

// sign.number = Sign\_number();

//Начало исполняемой секции

sign.first\_sec = data[1];

//Длина исполняемой секции

sign.lenght\_sec = data[0];

//Последовательность байт для идентификации

file.Close();

//8 байт идентификации расширения

sign.byts\_header = byte\_id\_header();

//8 байт идентификации тела

sign.byts\_body = byte\_id\_body();

// Путь к файлу базы сигнатур

string line = key\_create.GetValue("sign").ToString();

// сериализуем структуру (превращаем в бинарную форму)

FileStream VDB = new FileStream(line, FileMode.Append, FileAccess.Write);

BinaryFormatter binForm = new BinaryFormatter();

// добавляем в VDB (файл базы сигнатур) объект sign

// оно добавится туда в 2ой форме и сложно будет что-то понять

// чтобы что-то понять, надо будет десериализовывать

binForm.Serialize(VDB, sign);

VDB.Close();

this.Close();

}

else

System.Windows.MessageBox.Show("Вы что-то не заполнили.");

}