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

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5

по дисциплине

«Защита информации от вредоносного программного обеспечения»

Разработка модуля подготовки файлов к сканированию

Выполнила:

студентка группы БВТ1601 Фурман С.И.

Проверил:

ассистент кафедры ИБ

Барков В. В.

**Разработка модуля подготовки файлов к сканированию**

Модуль подготовки файлов для сканирования определяет тип файла и подготавливает объект или объекты для сканирования. Объект для сканирования является абстракцией, с которой работает модуль сканирования. Объект для сканирования предоставляет адрес памяти, с которого начинать сканирование, и длину объекта. При создании объекта сканирования из файла необходимо воспользоваться механизмом отображения файлов в память. При сканировании PE-файлов необходимо найти адрес начала сегмента кода и длину сегмента кода.

***Задание***

Разработать абстрактный класс ScanObject, предоставляющий доступ к адресу начала и длине объекта.

Разработать конкретные реализации ScanObject для форматов PE или ELF.

Разработать фабрику ScanObjectFactory, которая получает путь к файлу, отображает его в память и в зависимости от его формата создаёт необходимую реализацию ScanObject. В случае если файл не является исполняемым, создаётся пустой объект для сканирования.

**Текст модулей**

DirScan\_Click – класс объекта для сканирования.

***Form1.cs***

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

{

// Путь сканирования

String line = wFile.pathScan();

scanPath.Text = "Путь сканирования: " + line;

// Вывести все pe файлы директории в listbox

DirectoryInfo dir = new DirectoryInfo(line);

listBox1.Items.Clear();

files.Text = "PE файлы директории";

//foreach (string f in Directory.GetFiles(line, "\*.\*", SearchOption.AllDirectories).Union(Directory.GetFiles(line, "\*.dll", SearchOption.AllDirectories)))

//{

// string[] splitpath = f.Split('\\');

// string name = splitpath[splitpath.Length - 1];

// scan\_listBox.Items.Add(name);

//}

//scan\_label\_2.Text = "Кол-во файлов в данной директории: " + scan\_listBox.Items.Count;

}

/// <summary>

/// +Путь к файлу базы вирусов

/// </summary>

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

{

System.Windows.Forms.OpenFileDialog OBD = new System.Windows.Forms.OpenFileDialog();

if (OBD.ShowDialog() == DialogResult.OK)

{

try

{

key\_create.SetValue("sign", OBD.FileName);

}

catch (Exception err)

{

System.Windows.MessageBox.Show("Exception: " + err.Message);

}

signPath.Text = "Путь к файлу базы вирусов: " + OBD.FileName;

}

}

*backgroundWorker2\_DoWork* – класс, создающий объект для сканирования.

***Form1.cs***

public void Print\_scan\_listBox(String line, List<string> InfFile)

{

if (scan\_listBox.InvokeRequired)

{

//scan\_listBox.Invoke(new Action(() => scan\_listBox.Items.Clear()));

//for (int j = 0; j < InfFile.Count; j++)

//{

// line = InfFile[j];

//scan\_listBox.Invoke(new Action(() => scan\_listBox.Items.Add(line)));

// listbox\_scan\_listBox(line);

//}

listbox\_scan\_listBox("Кол-во вирусом: "+InfFile.Count.ToString());

//this.Invoke(new System.Threading.ThreadStart(delegate

//{

// progressBar1.Value = 0;

// bunifuCircleProgressbar1.Value = 0;

//}));

}

}

// void func\_timer()

private void backgroundWorker2\_DoWork(object sender, DoWorkEventArgs e)

{

DateTime time;

TimeSpan value;

string date;

while (btimer)

{

this.Invoke(new System.Threading.ThreadStart(delegate

{

time = DateTime.Now;

value = time.Subtract(timer\_nach);

date = value.ToString().Substring(0, 8);

scan\_label\_2.Text = String.Format("Время сканирования {0}", date);

}));

}

}