МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ КОМП’ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра ІОСУ

**КОМПЛЕКСНО-ПРАКТИЧНЕ**

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**

**з навчальної дисципліни “Конструювання ПЗ”**

**На тему: “** **Розробка системи антивірусного та антиспамерного захисту електронної пошти”**

**Виконав:** ст.гр. ПЗС-31

Файфура В.В.

**Перевірив:** Сирник О.Й.

ТНЕУ– 2015

**План**:

Вступ

1. Реалізація системи мовою програмування
2. Тестування програми
3. Висновок

Список використаних джерел

**Вступ:**

В час високих технологій широкого поширення набула, так звана електронна пошта. На сьогоднішній день електронну пошту використовують не тільки для обміну інформацією, але й для свого роду ідентифікації користувача в мережі Інтернет. Зазвичай для реєстрації на певному сервісі вимагається наявність власної електронної скриньки.

Так за статистикою Radicati Group станом на 2015 рік у світі зареєстровано близько 4,5 мільярди поштових скриньок, а частина спаму в загальному трафіку електронної форми становить понад 70%.

У силу надзвичайно високої прибутковості спамерського бізнесу технічні засоби боротьби зі спамом не завжди досягають своєї мети - спамери без кінця винаходять все нові й нові способи обходу фільтрів.

Метою цього проекту є розробка власної програмної системи антивірусного та антиспамерного захисту електронної пошти.

1. **Реалізація системи мовою програмування**

Завдяки наявній великій кількості готових бібліотек, зокрема бібліотеки «OpenPop», призначеної для роботи з поштовим офісним протоколом, для розробки системи було обрано мову програмування C#. Протокол POP3 використовується клієнтом для доступу до повідомлень електронної пошти на сервері. Хоча за своєю функціональністю цей протокол є значно бідніший ніж, IMAP протокол, його можливостей достатньо реалізації розроблюваної системи. Принцип роботи POP3 протоколу зображено на рис 1.1.

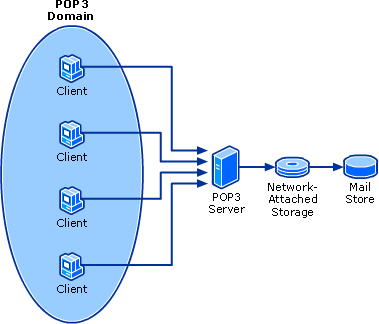


Рисунок 1.1 – Принцип роботи протоколу POP3

Програма містить один клас – Program (рис 1.2) та функції GetAttachments, VirusScan, MessageCheck.



Рисунок 1.2 – Клас Program

Далі подано лістинг коду головної функції програми:

static void Main(string[] args)

{

var handle = GetConsoleWindow();

ShowWindow(handle, SW\_HIDE);

Console.OutputEncoding = Encoding.UTF8;

Console.Title = "Spam Stop v 1.0";

var icon = new NotifyIcon();

icon.Icon = new Icon("icon.ico");

icon.Visible = true;

bool flag = true;

while (flag)

{

GetAttachmets();

MessageCheck();

Thread.Sleep(30000);

}

}

Повний код програми подано у додатку А.

Програма працює у фоновому режимі та згортається в трей під час роботи.

1. **Тестування програми**

Для перевірки працездатності програми було обрано один з найбільш популярних почтових сервісів – Gmail. До запуску програми потрібно ввімкнути підтримку POP3 в налаштування почтової скриньки та дозволити доступ для менш безпечних додатків.

Одразу після запуску програми консольне вікно зникне, натомість іконка програми буде відображатися в системному треї (рис 2.1).

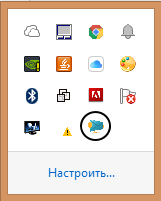


Рисунок 2.1 – Виконання програми в фоновому режимі

Для детального тестування розроблюваної системи функція мінімізації вікна програми було вимкнено. Також на тестову електронну скриньку надіслано декілька листів з вкладенням та лист спам характеру. Після цього програму знову запущено (рис 2.2).

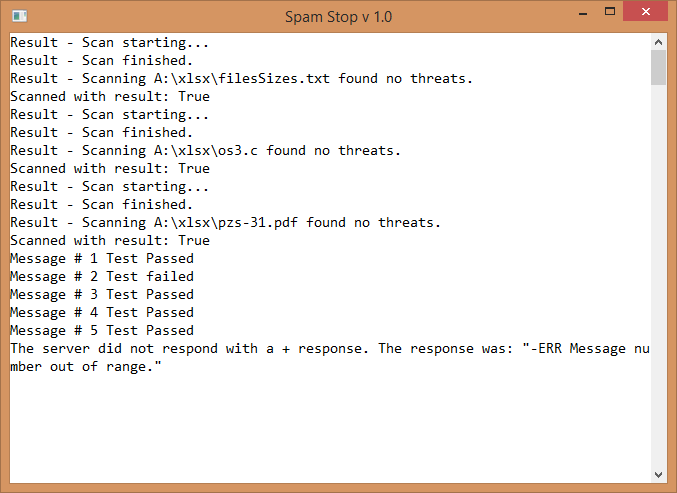


Рисунок 2.2 – Виконання програми

Як видно з рисунку 2.2 спочатку програма перевірила вкладення листів на наявність вірусів. Заражених файлів виявлено не було. Далі листи було перевірено на предмет спаму, згідно з результатами перевірки лист під номером два було позначено як спам та видалено. По завершенню перевірки програма виконує 30 секундну паузу, після чого знову перевіряє листи на предмет спаму та вкладень, що містять заражені файли.

1. **Висновок**

Збільшення долі спаму в інформаційному потоці та поява нових видів шкідливих програм спонукає шукати нові засоби боротьби з цими явищами. Розроблена система пропонує базовий захист від шкідливих програм та спам фільтр з власним словником. На завершення слід зауважити, що жодна антиспам та антивірусна система не може гарантувати стовідсоткового захисту від шкідливих програм чи спам листів. Адже ефективність роботи таким систем напряму залежить від актуальності бази шкідливих програм або словника характерних для спам листів слів, виразів або ж вкладених файлів.

**Список використаних джерел**:

1. [Електронний ресурс]. – Режим доступу: http://www.msdn.com.
2. [Електронний ресурс].– Режим доступу: http://www.codenet.ru
3. [Електронний ресурс]. – Режим достпу: http://professorweb.ru/
4. [Електронний ресурс]. – Режим достпу: http://uk.wikipedia.org/
5. С#: Пер. С англ./Дейтел Х.,Дейтел П., Листфилд Дж., Нието Т., Йегер Ш., Златкина М. – СПб.: БХВ-Петербург, 2006. – 1056 с.:ил.
6. Джеффри Рихтер - CLR via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке С#. Мастер-класс./Пер. с англ. — М.: Издательство «Русская Редакция»; СПб.: Питер, 2007. — 656 стр. : ил.
7. Кристиан Нейгел, Билл Ивьен, Джей Глинн, Карли Уотсон, Морган Скиннер - С# 2008 и платформа .Net 3.5 для профессионалов.

**Додаток А**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Net.Mail;

using System.IO;

using System.Threading;

using OpenPop;

using OpenPop.Mime;

using OpenPop.Pop3;

using OpenPop.Common;

using System.Drawing;

using System.Windows.Forms;

using System.Drawing.Design;

using System.Runtime.InteropServices;

using System.Diagnostics;

namespace EmailSpamChecker

{

class Program

{

[DllImport("kernel32.dll")]

static extern IntPtr GetConsoleWindow();

[DllImport("user32.dll")]

static extern bool ShowWindow(IntPtr hWnd, int nCmdShow);

const int SW\_HIDE = 0;

const int SW\_SHOW = 5;

static string attachmentPath;

static List<string> scannedFiles = new List<string>();

static void GetAttachmets()

{

using (OpenPop.Pop3.Pop3Client client = new Pop3Client())

{

client.Connect("pop.gmail.com", 995, true);

client.Authenticate("recent:wedesweb@gmail.com", "arca1313", AuthenticationMethod.UsernameAndPassword);

if (client.Connected)

{

int messageCount = client.GetMessageCount();

List<OpenPop.Mime.Message> allMessages = new List<OpenPop.Mime.Message>(messageCount);

for (int i = messageCount; i > 0; i--)

{

allMessages.Add(client.GetMessage(i));

}

foreach (OpenPop.Mime.Message msg in allMessages)

{

var att = msg.FindAllAttachments();

foreach (var ado in att)

{

ado.Save(new System.IO.FileInfo(System.IO.Path.Combine(@"A:\xlsx", ado.FileName)));

attachmentPath = System.IO.Path.Combine(@"A:\xlsx", ado.FileName);

if (!scannedFiles.Contains(ado.FileName))

{

VirusScaner();

scannedFiles.Add(ado.FileName);

}

}

}

}

}

}

static void VirusScaner()

{

try

{

System.Diagnostics.Process process = new System.Diagnostics.Process();

System.Diagnostics.ProcessStartInfo startInfo = new System.Diagnostics.ProcessStartInfo();

startInfo.WindowStyle = System.Diagnostics.ProcessWindowStyle.Hidden;

startInfo.FileName = @"C:\Program Files\Windows Defender\MpCmdRun.exe";

string args = @"-Scan -ScanType 3 -File " + attachmentPath;

startInfo.Arguments = args;

startInfo.RedirectStandardOutput = true;

startInfo.UseShellExecute = false;

process.StartInfo = startInfo;

var result = process.Start();

while (!process.StandardOutput.EndOfStream)

{

string line = process.StandardOutput.ReadLine();

Console.WriteLine("Result - " + line);

}

Console.WriteLine("Scanned with result: " + result.ToString());

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

static void MessageCheck()

{

string path = @"A:\Work\Third Step\First Term\Організація комп'ютерних мереж\IPT\EmailSpamChecker\EmailSpamChecker\bin\Debug\base.txt";

bool isDisonnected = false;

using (OpenPop.Pop3.Pop3Client client = new Pop3Client())

{

client.Connect("pop.gmail.com", 995, true);

client.Authenticate("recent:wedesweb@gmail.com", "arca1313");

if (client.Connected)

{

int messageCount = client.GetMessageCount();

List<OpenPop.Mime.Message> allMessages = new List<OpenPop.Mime.Message>(messageCount);

if (File.Exists(path))

{

string[] split = System.IO.File.OpenText(path)

.ReadToEnd().Replace("\r\n", " ").Split(new Char[] { ' ' });

try

{

for (int i = 1; i <= messageCount; i++)

{

var message = client.GetMessage(i);

string output = "Test Passed";

foreach (string word in split)

{

if (message.Headers.Subject.Contains(word))

{

if (i + 1 == messageCount)

Console.WriteLine("SHOULD DISCONNECT");

client.DeleteMessage(i);

output = "Test failed";

break;

}

}

output = "Message # " + i.ToString() + " " + output;

Console.WriteLine(output);

client.Disconnect();

isDisonnected = true;

if (isDisonnected)

{

isDisonnected = false;

client.Connect("pop.gmail.com", 995, true);

client.Authenticate("recent:wedesweb@gmail.com", "arca1313");

}

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

}

}

}

static void Main(string[] args)

{

var handle = GetConsoleWindow();

ShowWindow(handle, SW\_HIDE);

Console.OutputEncoding = Encoding.UTF8;

Console.Title = "Spam Stop v 1.0";

var icon = new NotifyIcon();

icon.Icon = new Icon("icon.ico");

icon.Visible = true;

bool flag = true;

while (flag)

{

GetAttachmets();

MessageCheck();

Thread.Sleep(30000);

}

}

}

}