

```
using System;
using System.Configuration;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
using System.Net;
using System.IO;
using System.Runtime.InteropServices;
using System.Net.Sockets;
using System.Threading;
using System.Text;

namespace WindowsApplication1
{
    public class Form1 : System.Windows.Forms.Form
    {
        // Send referrer
        public void spamHost(object i)
        {
            string [] input = (string []) i;
            string url = input[0];
            string userAgent = input[1];
            string referrer = input[2];

            try
            {
                if(!url.StartsWith("http://"))
                {
                    url = "http://" + url;
                }

                System.Uri u = new System.Uri(url);
                Socket s = ConnectSocket(u.Host, 80);

                string request = "GET " + u.AbsolutePath + u.Query + " HTTP/1.1" + "\r\n"
                    + "Host: " + u.Host + "\r\n"
                    + "Referer: " + referrer + "\r\n"
                    + "User-Agent: " + userAgent + "\r\n"
                    + "Connection: Close" + "\r\n" + "\r\n";

                s.Send(Encoding.ASCII.GetBytes(request));
                s.Close();
            }
            catch(Exception e){
                Console.WriteLine(e + " URL: "+url);
            }

            Interlocked.Decrement(ref threadCount);
        }

        private static Socket ConnectSocket(string server, int port)
        {
            Socket s = null;
            IPEndPoint hostEntry = null;

            // Get host related information.
            hostEntry = Dns.Resolve(server);

            foreach(IPAddress address in hostEntry.AddressList)
            {
                IPEndPoint ipe = new IPEndPoint(address, port);
                Socket tempSocket =
                    new Socket(ipe.AddressFamily, SocketType.Stream, ProtocolType.Tcp);
            }
        }
    }
}
```

```
        tempSocket.Connect(ipe);

        if(tempSocket.Connected)
        {
            s = tempSocket;
            break;
        }
        else
        {
            continue;
        }
    }
    return s;
}

// Beep
public enum MessageBeepType
{
    Default = -1,
    Ok = 0x00000000,
    Error = 0x00000010,
    Question = 0x00000020,
    Warning = 0x00000030,
    Information = 0x00000040,
}

[DllImport("user32.dll", SetLastError=true)]
public static extern bool MessageBeep(
    MessageBeepType type
);

private System.Windows.Forms.Label label1;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.Label label3;
private System.ComponentModel.IContainer components;
private System.Windows.Forms.TextBox textBoxUrlList;
private System.Windows.Forms.Button buttonUrlList;
private System.Windows.Forms.ListBox listBoxUrlList;
private System.Windows.Forms.Label label4;
private System.Windows.Forms.TextBox textBoxReferrer;
private System.Windows.Forms.Button buttonReferrerList;
private System.Windows.Forms.TextBox textBoxReferrerList;
private System.Windows.Forms.OpenFileDialog openFileDialogUrlList;
private System.Windows.Forms.OpenFileDialog openFileDialogReferrerList;
private System.Windows.Forms.ComboBox comboBoxUserAgent;
private System.Windows.Forms.ProgressBar progressBar;
private System.Windows.Forms.Button buttonStart;
private System.Windows.Forms.CheckBox checkBoxRandomizeUserAgent;
private System.Windows.Forms.CheckBox checkBoxRandomizeReferrer;
private System.Windows.Forms.Label labelUrlCount;

// REEF variables
private ArrayList urlList = new ArrayList();
private ArrayList referrerList = new ArrayList();
private ArrayList log = new ArrayList();
private static long threadCount;

private System.Windows.Forms.Label msg;
private System.Windows.Forms.ToolTip toolTip1;
private System.Windows.Forms.MainMenu mainMenu1;
private System.Windows.Forms.MenuItem menuItem1;
private System.Windows.Forms.MenuItem menuItem2;
private System.Windows.Forms.MenuItem menuItem3;
private System.Windows.Forms.MenuItem menuItem4;
private System.Windows.Forms.MenuItem menuItem5;
private System.Windows.Forms.MenuItem menuItem6;
```

```
private System.Windows.Forms.MenuItem menuItem9;
private System.Windows.Forms.MenuItem menuItem10;
private System.Windows.Forms.MenuItem menuItem12;
private System.Windows.Forms.MenuItem menuItem13;
private System.Windows.Forms.MenuItem menuItem14;
private System.Windows.Forms.MenuItem menuItem15;
private System.Windows.Forms.SaveFileDialog saveFileDialogLog;
private System.Windows.Forms.CheckBox checkBoxLoop;
private System.Windows.Forms.MenuItem menuItem16;

public Form1()
{
    InitializeComponent();
    comboBoxUserAgent.SelectedIndex = 0;
}

/// <summary>
/// Clean up any resources being used.
/// </summary>
protected override void Dispose( bool disposing )
{
    if( disposing )
    {
        if (components != null)
        {
            components.Dispose();
        }
        base.Dispose( disposing );
    }
}

Windows Form Designer generated code

/// <summary>
/// The main entry point for the application.
/// </summary>
[STAThread]
static void Main()
{
    Application.Run(new Form1());
}

// OPEN URL LIST
private void button1_Click(object sender, System.EventArgs e)
{
    if(openFileDialogUrlList.ShowDialog() == DialogResult.OK)
    {
        // Read in the list of URLs
        System.IO.StreamReader s = new System.IO.StreamReader
(openFileDialogUrlList.FileName);
        string temp = "";
        urlList.Clear();
        while((temp = s.ReadLine()) != null)
        {
            temp.Trim();
            urlList.Add(temp);
        }
        s.Close();

        // Write the FileName to the box
        textBoxUrlList.Text = openFileDialogUrlList.FileName;

        // Update the list of URLs
        listBoxUrlList.Items.Clear();
        listBoxUrlList.Items.AddRange(urlList.ToArray());
    }
}
```

```
        listBoxUrlList.Refresh();

        // Count them
        labelUrlCount.Text = urlList.Count + " urls.";
    }
}

private void textBoxReferrer_TextChanged(object sender, System.EventArgs e)
{
    if(((TextBox)sender).Text.Length > 0)
    {
        referrerList.Clear();
        referrerList.Add(((TextBox)sender).Text);
        textBoxReferrerList.Enabled = false;
        buttonReferrerList.Enabled = false;
    }
    else
    {
        textBoxReferrerList.Enabled = true;
        buttonReferrerList.Enabled = true;
    }
}

// OPEN REFERRER LIST
private void buttonReferrerList_Click(object sender, System.EventArgs e)
{
    if(openFileDialogReferrerList.ShowDialog() == DialogResult.OK)
    {
        // Read in the list of URLs
        System.IO.StreamReader s = new System.IO.StreamReader
(openFileDialogReferrerList.FileName);
        string temp = "";
        referrerList.Clear();
        while((temp = s.ReadLine()) != null)
        {
            temp.Trim();
            referrerList.Add(temp);
        }
        s.Close();

        // Write the FileName to the box
        textBoxReferrerList.Text = openFileDialogReferrerList.FileName;
    }
}

private void buttonStart_Click(object sender, System.EventArgs e)
{
    // Error Checking
    if(urlList.Count <= 0)
    {
        msg.Text = "There are no URLs. Please add URLs.";
        log.Add("ERROR: There are no URLs. Please add URLs.");
        return;
    }
    if(referrerList.Count <= 0)
    {
        msg.Text = "There are no referrers. Please add some.";
        log.Add("ERROR: There are no referrers. Please add some.");
        return;
    }

    // Turn off Start Button
    this.buttonStart.Enabled = false;

    // Initialize objects
    Random r = new Random();
}
```

```
ArrayList temp = (ArrayList) urlList.Clone();

// Adjust progress bar / log
progressBar.Minimum = 1;
progressBar.Maximum = temp.Count;
log.Add("Starting job on "+System.DateTime.Now+" with "+temp.Count+" hosts.")

;

// Start loop
while(temp.Count > 0)
{
    // Set URL
    int rand = r.Next(0, temp.Count);
    string url = (string) temp[rand];
    temp.RemoveAt(rand);

    // Set User-Agent
    string userAgent;
    if(checkBoxRandomizeUserAgent.Checked)
    {
        userAgent = (string) comboBoxUserAgent.Items[r.Next(0,
comboBoxUserAgent.Items.Count)];
    }
    else
    {
        userAgent = comboBoxUserAgent.Text;
    }

    // Set Referrer
    string referrer;
    if(textBoxReferrer.Text.Length > 0)
    {
        referrer = textBoxReferrer.Text;
    }
    else
    {
        if(checkBoxRandomizeReferrer.Checked)
        {
            referrer = (string) referrerList[r.Next(0, referrerList.Count)];
        }
        else
        {
            referrer = (string) referrerList[0];
        }
    }

    // Send the Referrer spam
    Interlocked.Increment(ref threadCount);
    ThreadPool.QueueUserWorkItem(new WaitCallback(this.spamHost), new string
[] {url, userAgent, referrer});

    // Update the log file
    log.Add("Host \""+url+"\" with User-Agent \""+userAgent+"\" and Referrer
\""+referrer+"\" done.");

    // Update Progress Bar
    progressBar.Increment(1);
    progressBar.Refresh();
}

progressBar.Value = 1;
progressBar.Maximum = 100;
while(threadCount > 0)
{
    if(progressBar.Value == 100)
    {
```

```
        progressBar.Value = 1;
    }
    else
    {
        progressBar.Value++;
    }
    progressBar.Refresh();
    Thread.Sleep(50);
    Application.DoEvents();
}

progressBar.Value = 1;

log.Add("Job done at "+System.DateTime.Now+".");

Thread.Sleep(500);
if(checkBoxLoop.Checked == true)
{
    buttonStart_Click(sender, e);
}
else
{
    MessageBeep(MessageBeepType.Ok);
    this.buttonStart.Enabled = true;
}
}

private void menuItem2_Click(object sender, System.EventArgs e)
{
    this.Dispose();
}

private void menuItem9_Click(object sender, System.EventArgs e)
{
    if(this.saveFileDialogLog.FileName.Length == 0)
    {
        if(this.saveFileDialogLog.ShowDialog() != DialogResult.OK)
        {
            return;
        }
    }

    FileStream file = new FileStream(this.saveFileDialogLog.FileName, System.IO.
FileMode.Create);
    StreamWriter s = new StreamWriter(file);
    foreach(string line in log)
    {
        s.WriteLine(line);
    }
    s.Flush();
    s.Close();
}

private void menuItem14_Click(object sender, System.EventArgs e)
{
    buttonStart_Click(sender, e);
}

private void menuItem16_Click(object sender, System.EventArgs e)
{
    System.Windows.Forms.Form about = new About();
    about.Show();
}

private void menuItem4_Click(object sender, System.EventArgs e)
{

```

```
        button1_Click(sender, e);
    }

    private void menuItem5_Click(object sender, System.EventArgs e)
    {
        buttonReferrerList_Click(sender, e);
    }
}
```