**Metasploit: Introduction**

**Giới thiệu**

Metasploit là thư viện khai thác được dùng phổ biến nhất. Metasploit là công cụ mạnh mẽ hỗ trợ cho tất cả giai đoạn kiểm thử thâm nhập, từ việc thu thập đến post-exploitation.

Metasploit có hai phiên bản chính:

* Metasploit pro: phiên bản thương mại cung cấp sự tự động và quản lí các tác vụ. Nó có giao diện đồ họa (GUI).
* Metasploit framework: phiên bản mã nguồn mở làm việc với giao diện dòng lệnh (CLI)

Thư viện Metasploit là 1 danh sách các công cụ cho phép thu thập thông tin, quét, khai thác, post-exploitation và hơn thế nữa. Trong khi chức năng chính của Metasploit tập trung vào kiểm thử thâm nhập, nó cũng hữu ích để nghiên cứu lỗ hổng và phát triển khai thác.

Các thành phần chính của thư viện Metasploit có thể được tổng hợp như sau:

* msfconsole: giao diện dòng lệnh chính
* Modules: hỗ trợ các modules như khai thác, scanners, payloads
* Tools: giúp nghiên cứu, đánh giá lỗ hổng hoặc kiểm thử thâm nhập. Một số tools như msfvenom, pattern\_create và pattern\_offset.

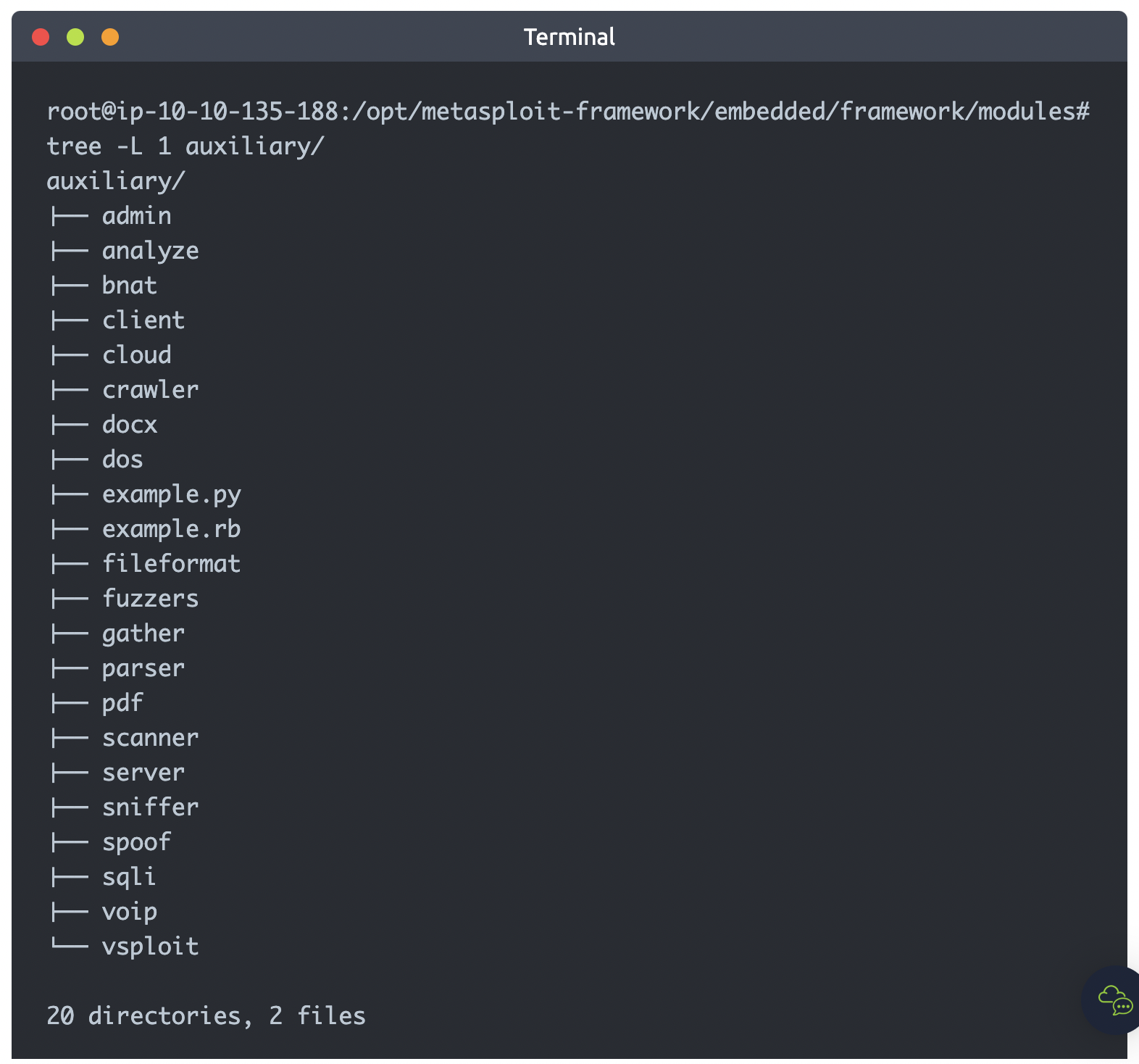
Phòng này sẽ cung cấp cho bạn 1 nền tảng vững chắc cách để tìm những khai thác có liên quan, đặt tham số và khai thác các dịch vụ có lỗ hổng trên hệ thống đích.

**Các thành phần chính của Metasploit**

Trong khi dùng thư viện Metasploit, bạn sẽ tương tác chính với msf console. Console sẽ là giao diện để tương tác với các modules khác nhau của thư viện. Modules là những thành phần được dựng để thực hiện 1 tác vụ cụ thể như khai thác lỗ hổng, quét mục tiêu hay tấn công dồn dập.

**Auxiliary**

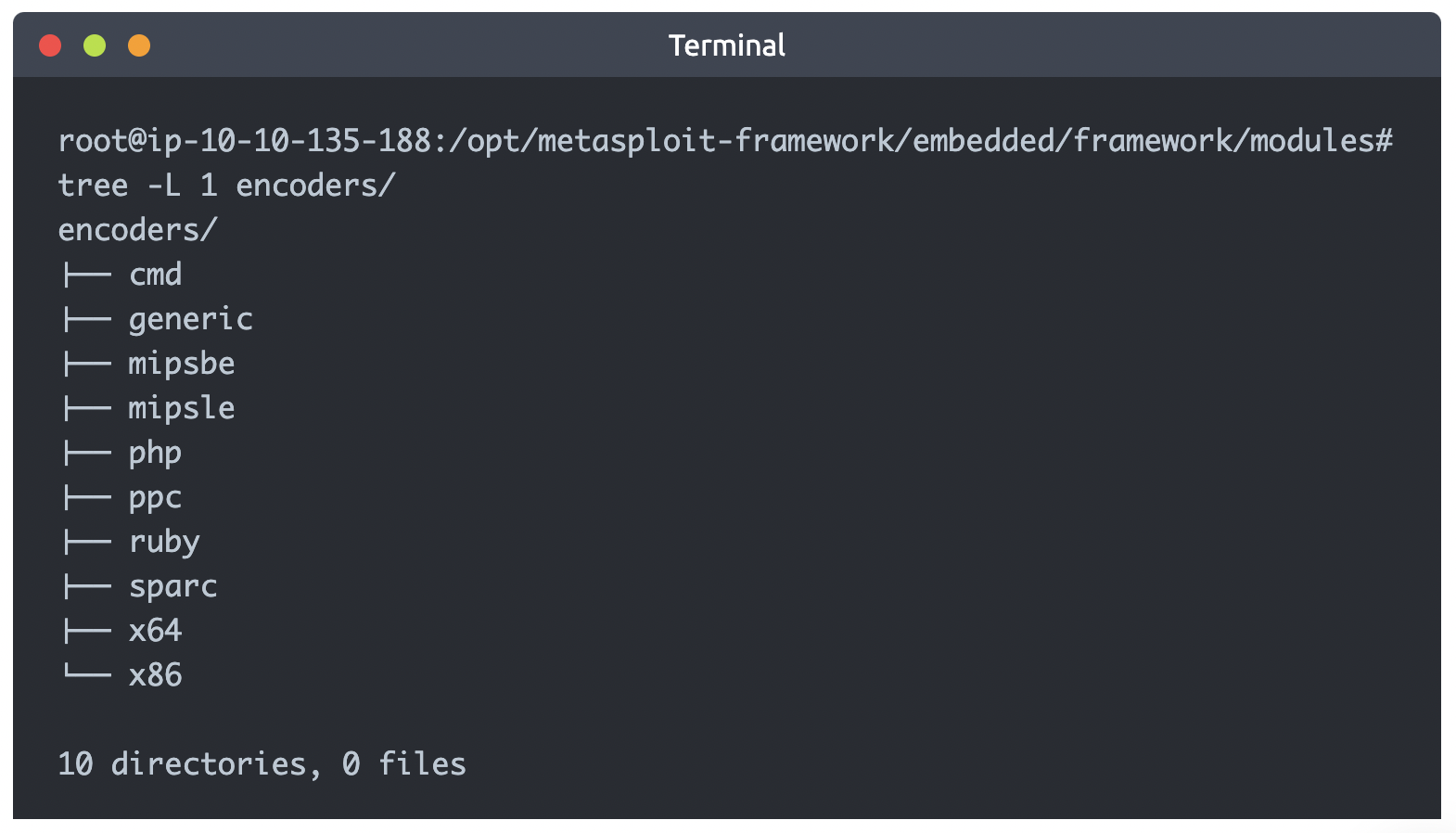
Bất cứ module nào hỗ trợ như scanners, crawlers và fuzzers có thể được tìm thấy ở đây.



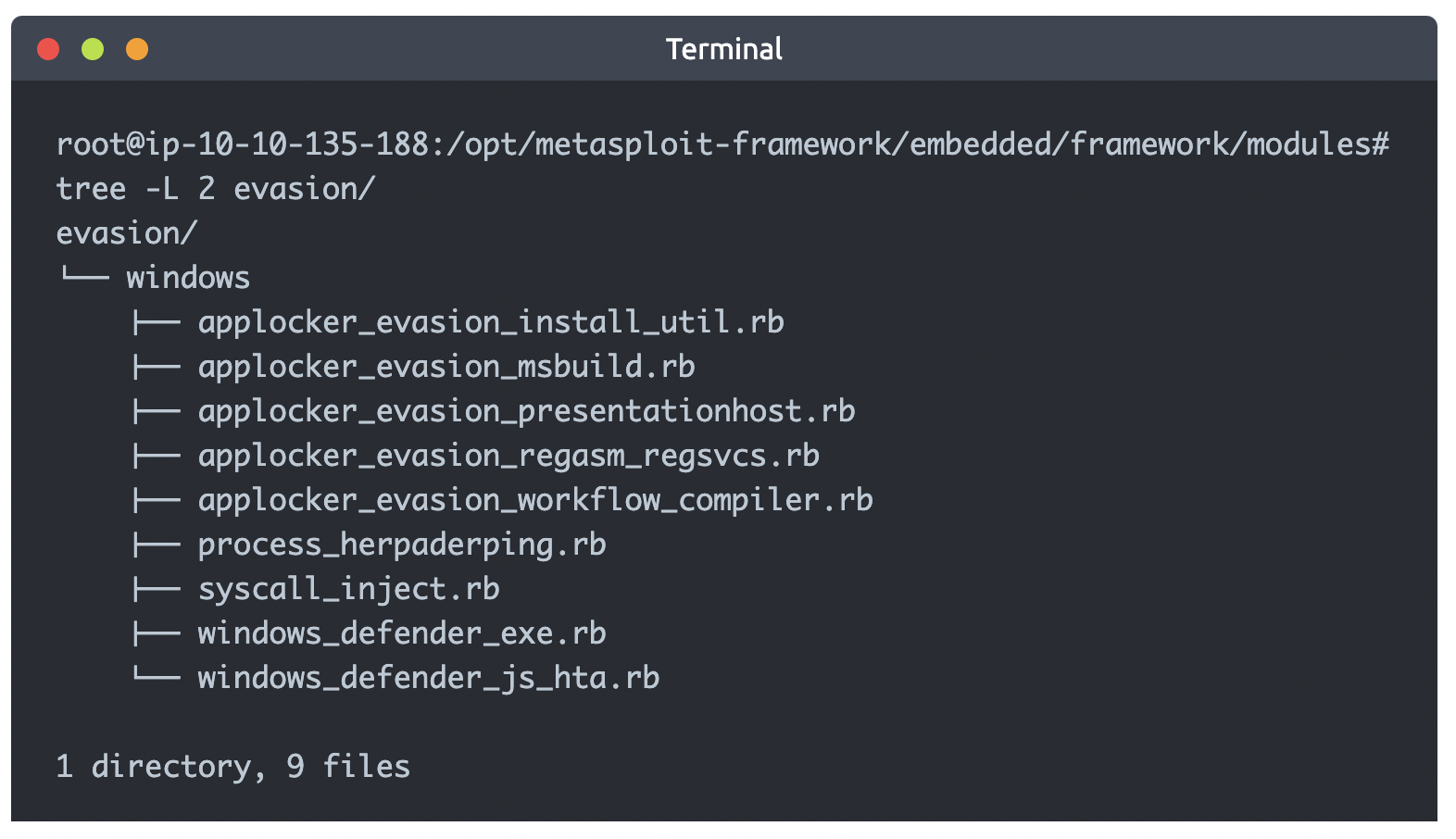
**Encoders**

Encoders cho phép bạn mã hóa khai thác và hi vọng phần mềm antivirus dựa signature có thể bỏ qua.

Antivirus dựa vào signature và các giải pháp bảo mật có cơ sở dữ liệu các mối đe dọa đã biết. Chúng phát hiện mối đe dọa bằng cách so sánh các tệp độc hại với csdl này và cảnh báo nếu khớp với nhau. Do đó encoders có thể giúp tăng tỉ lệ thành công khi antivirus kiểm tra.

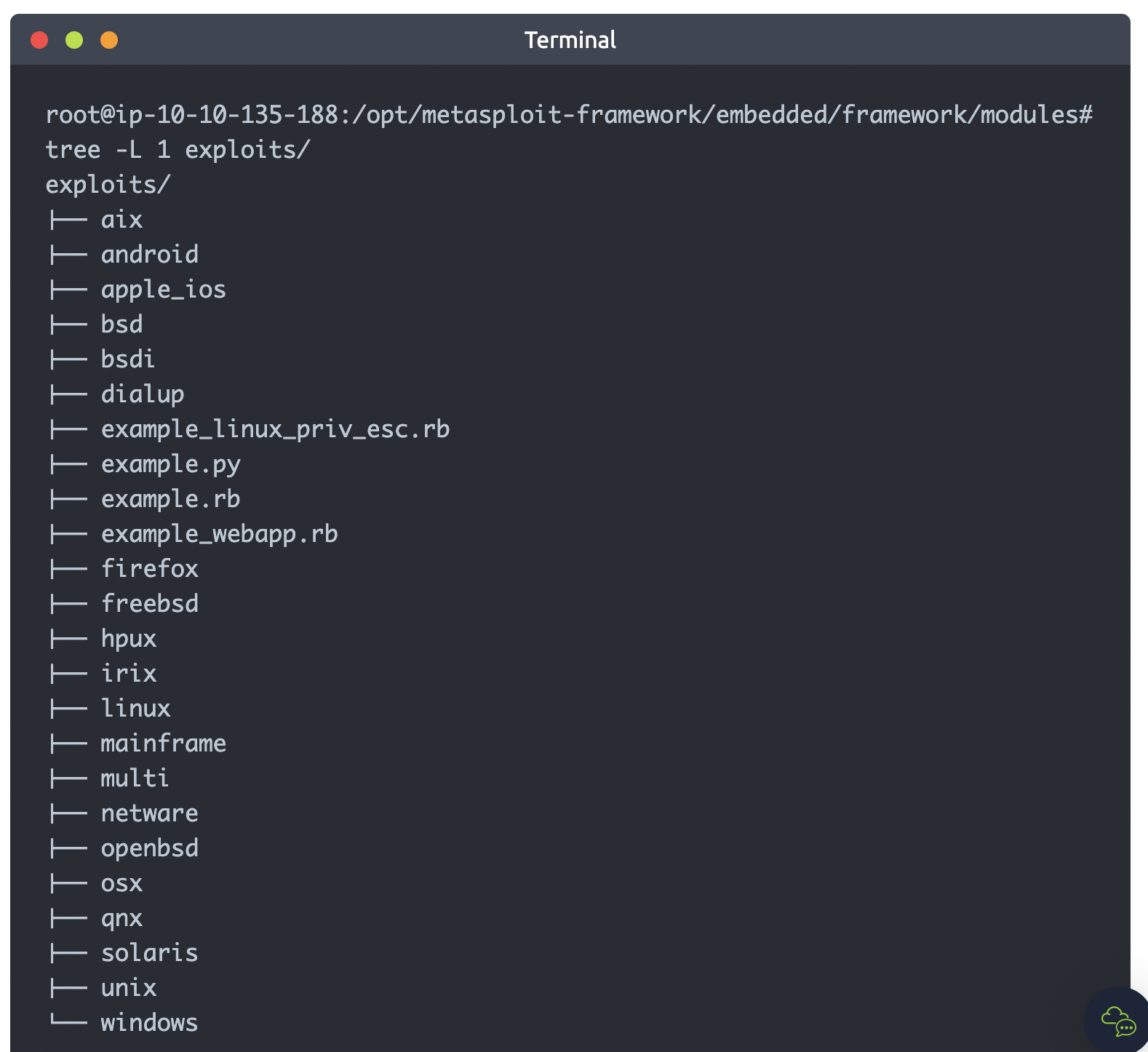


**Evasion**

Mặc dù các encoders sẽ mã hóa payload, nhưng đây không phải là cách trực tiếp để trốn tránh phần mềm chống virus. Nói cách khác, các modules evasion sẽ làm điều đó nhưng ít nhiều thành công.

**Exploits**

Các khai thác được tổ chức dựa trên hệ thống đích:

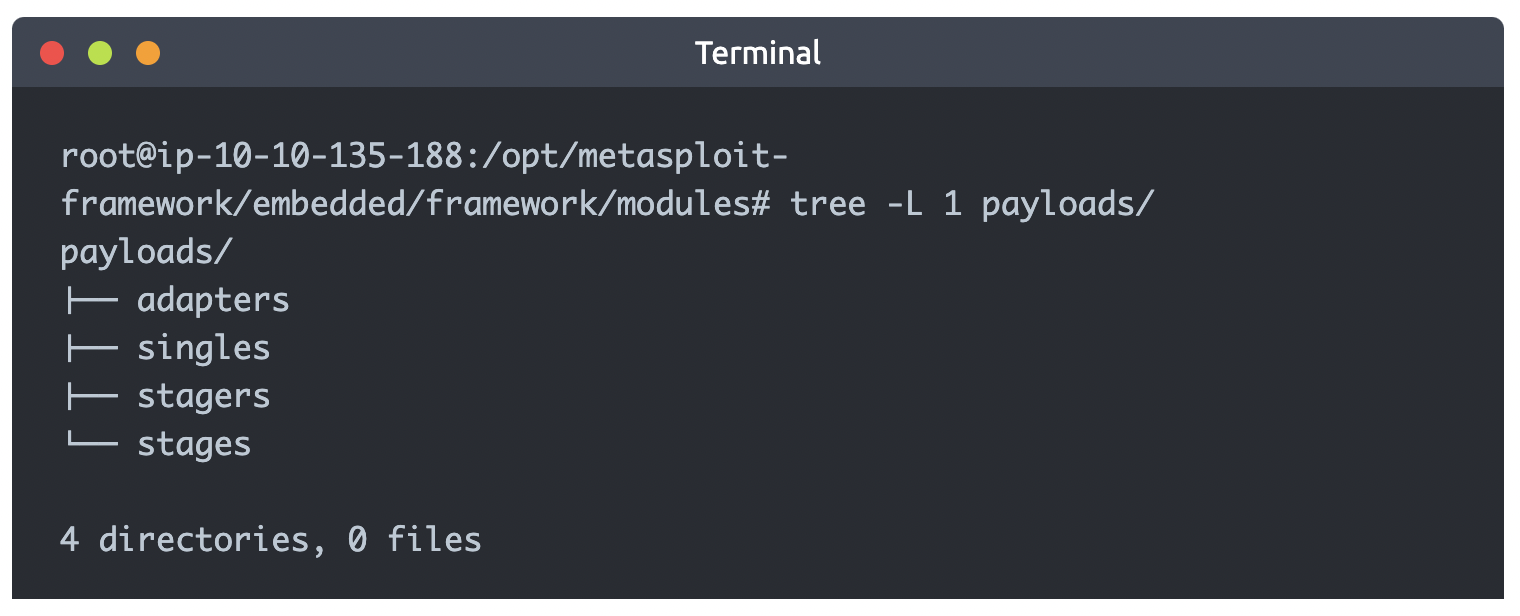


**Payloads**

Payloads là những đoạn mã sẽ chạy trên hệ thống đích.

Các khai thác sẽ tận dụng 1 lỗ hổng trên hệ thống đích, nhưng để có được kết quả mong muốn, chúng ta sẽ cần 1 payload. Ví dụ như: lấy shell, tải mã độc hoặc backdoor đến hệ thống đích, chạy 1 lệnh hoặc khởi tạo calc.exe giống như POC được đưa vào báo cáo khi kiểm thử thâm nhập . Khởi động máy tính trên hệ thống từ xa bằng cách chạy calc.exe là cách đơn giản để chạy lệnh.

Chạy lệnh trên hệ thống đích rất quan trọng nhưng có 1 interactive shell cho phép bạn nhập lệnh sẽ tốt hơn. Metasploit cho bạn tính năng để gửi các payload khác nhau mà có thể mở shell trên hệ thống đích.



Bạn sẽ thấy 4 thư mục: adapters, singles, stagers và stages.

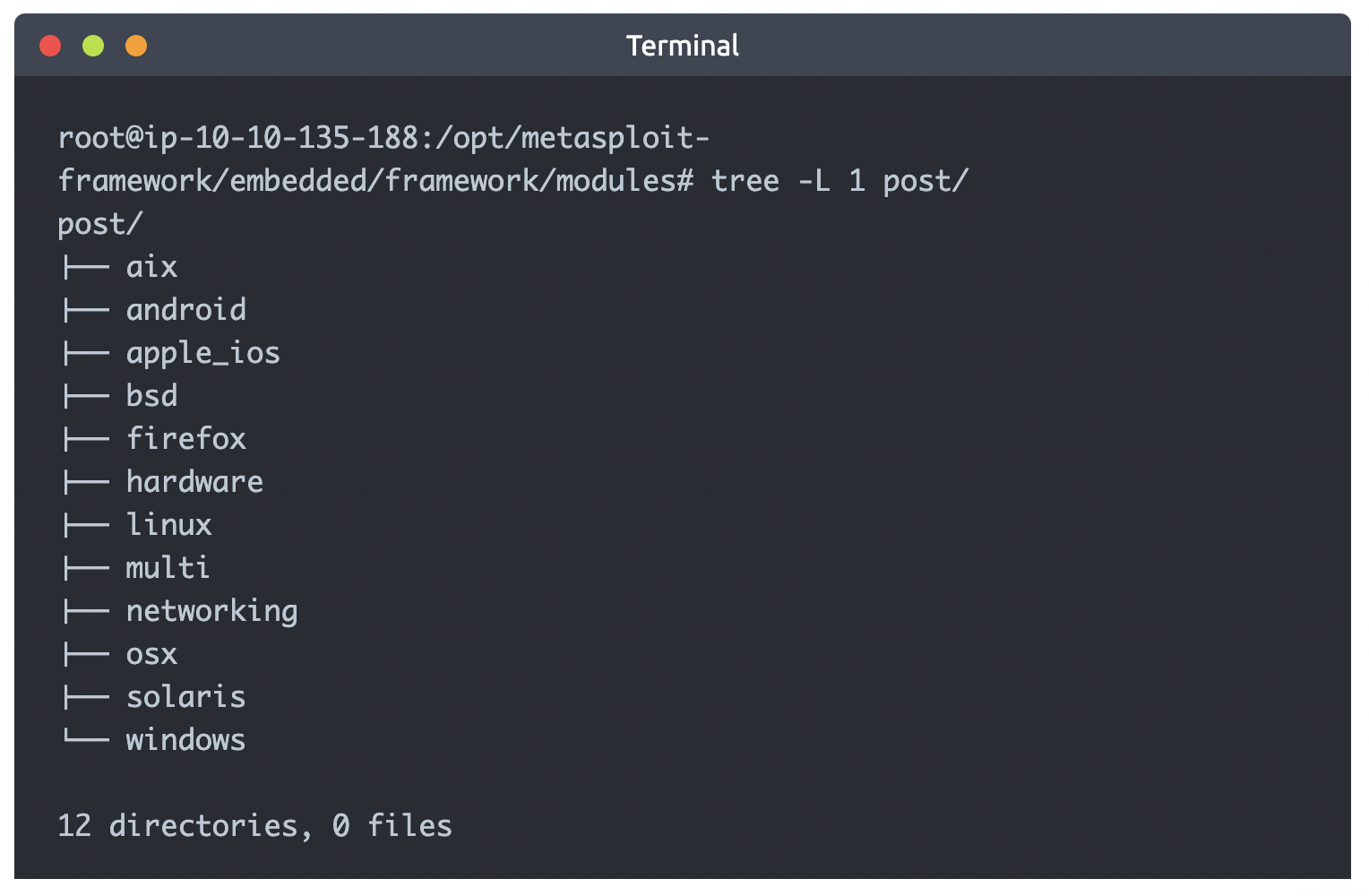
* Adapters: 1 bộ chuyển đổi bao bọc lấy payload để chuyển nó sang 1 định dạng khác. Ví dụ, 1 payload bình thường có thể được bao bọc trong Powershell adapter sẽ làm cho powershell thực thi payload.
* Singles: payload độc lập không cần tải thêm mã nguồn để chạy
* Stagers: chịu trách nhiệm kết nối giữa metasploit và hệ thống đích. Rất hữu ích khi làm việc với staged paylaod.
* Stages: được tải bởi stager. Điều này cho phép bạn dùng payload có kích thước lớn hơn.

Để xác định single và staged payload:

* generic/shell\_reverse\_tcp (stageless)
* windows/x64/shell/reverse\_tcp (staged)

**Post**

Post modules sẽ hữu ích ở giai đoạn cuối của quá trình kiểm thử là post-exploitation.



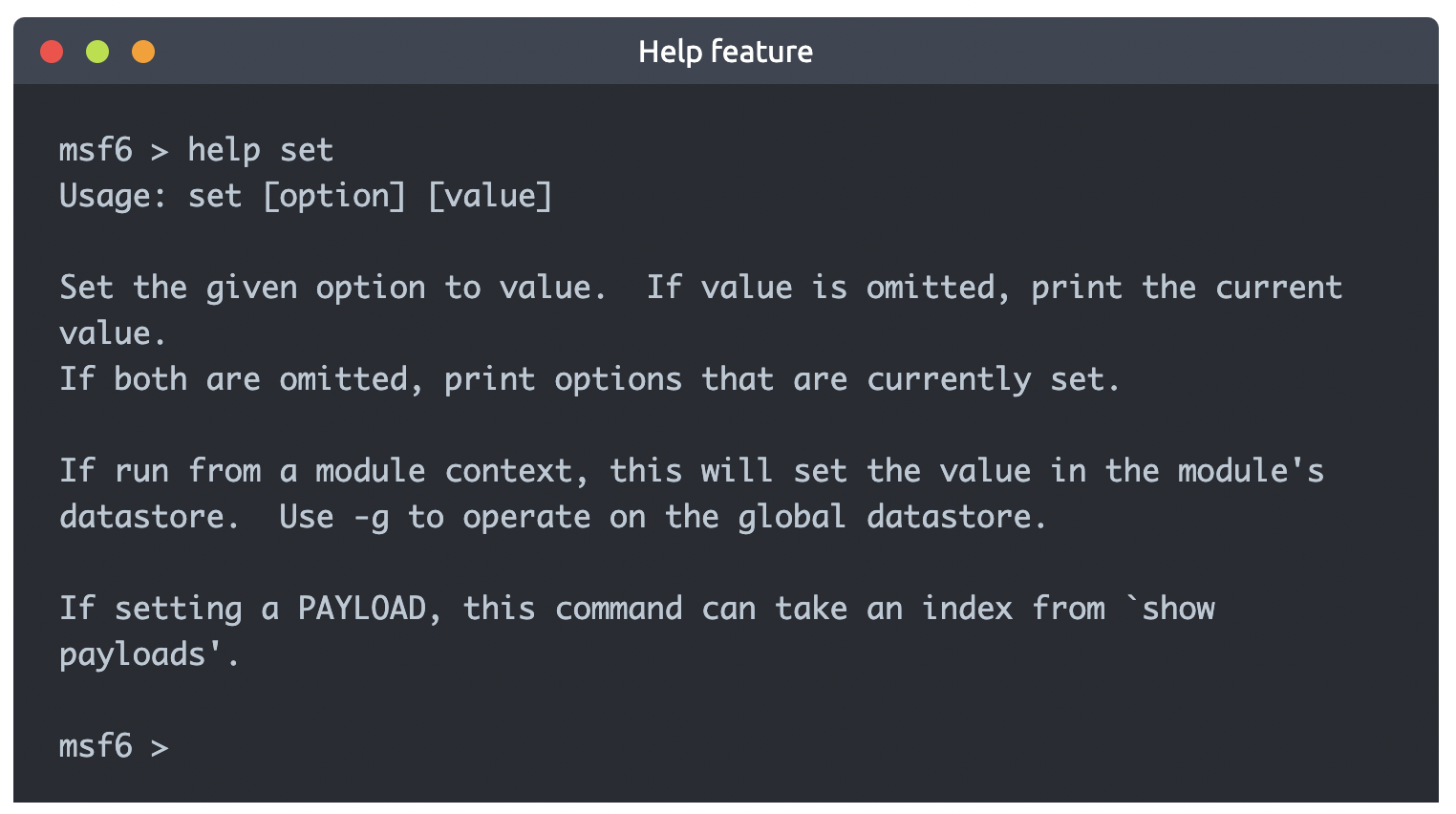
**Msfconsole**

Console là giao diện chính của thư viện Metasploit.



msfconsole có thể được dùng như 1 shell thông thường. Lệnh ls sẽ liệt kê nội dung từ thư mục Metasploit đang chạy.

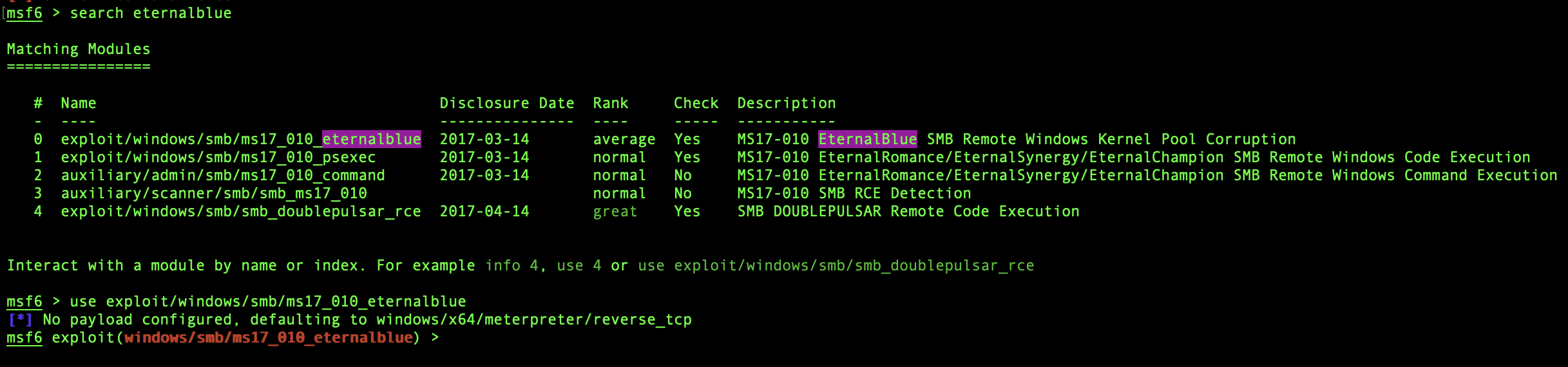
Lệnh help có thể được dùng chung với 1 lệnh bất kì.



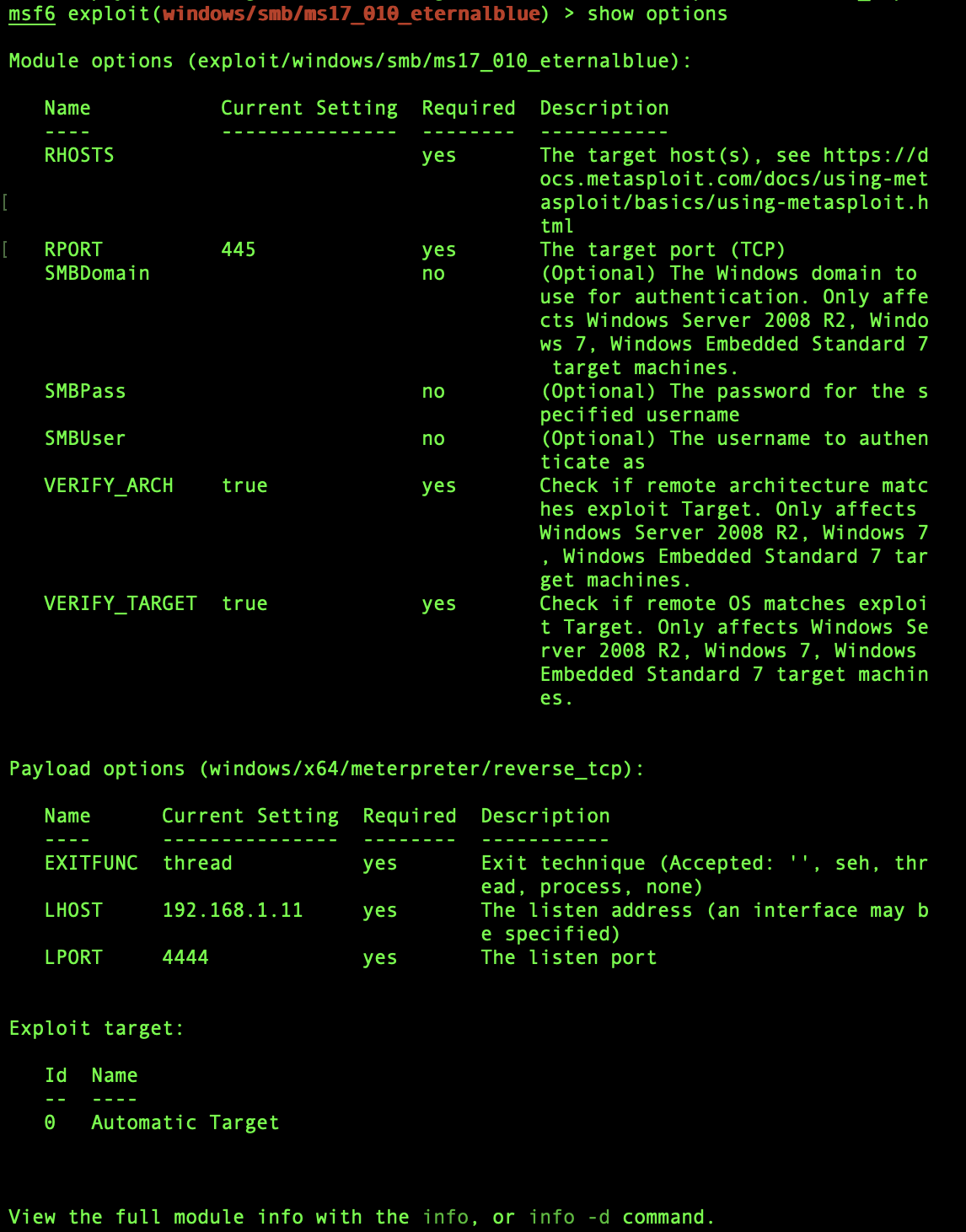
Một đặc điểm quan trọng nữa của msfconsole là hỗ trợ tab completion. Ví dụ, nếu bạn đánh he và nhấn Tab, bạn sẽ thấy help.

Nếu bạn không đặt tham số global, tất cả tham số bạn đặt sẽ bị mất nếu bạn thay đổi module. Ví dụ, chúng ta đang dùng khai thác ms17\_010\_eternalblue và đặt các tham số như RHOSTS. Nếu ta chuyển sang module khác (port scanner), chúng ta sẽ đặt lại RHOSTS lần nữa.

Giả sử ta muốn tìm khai thác MS17-010 “Eternalblue”

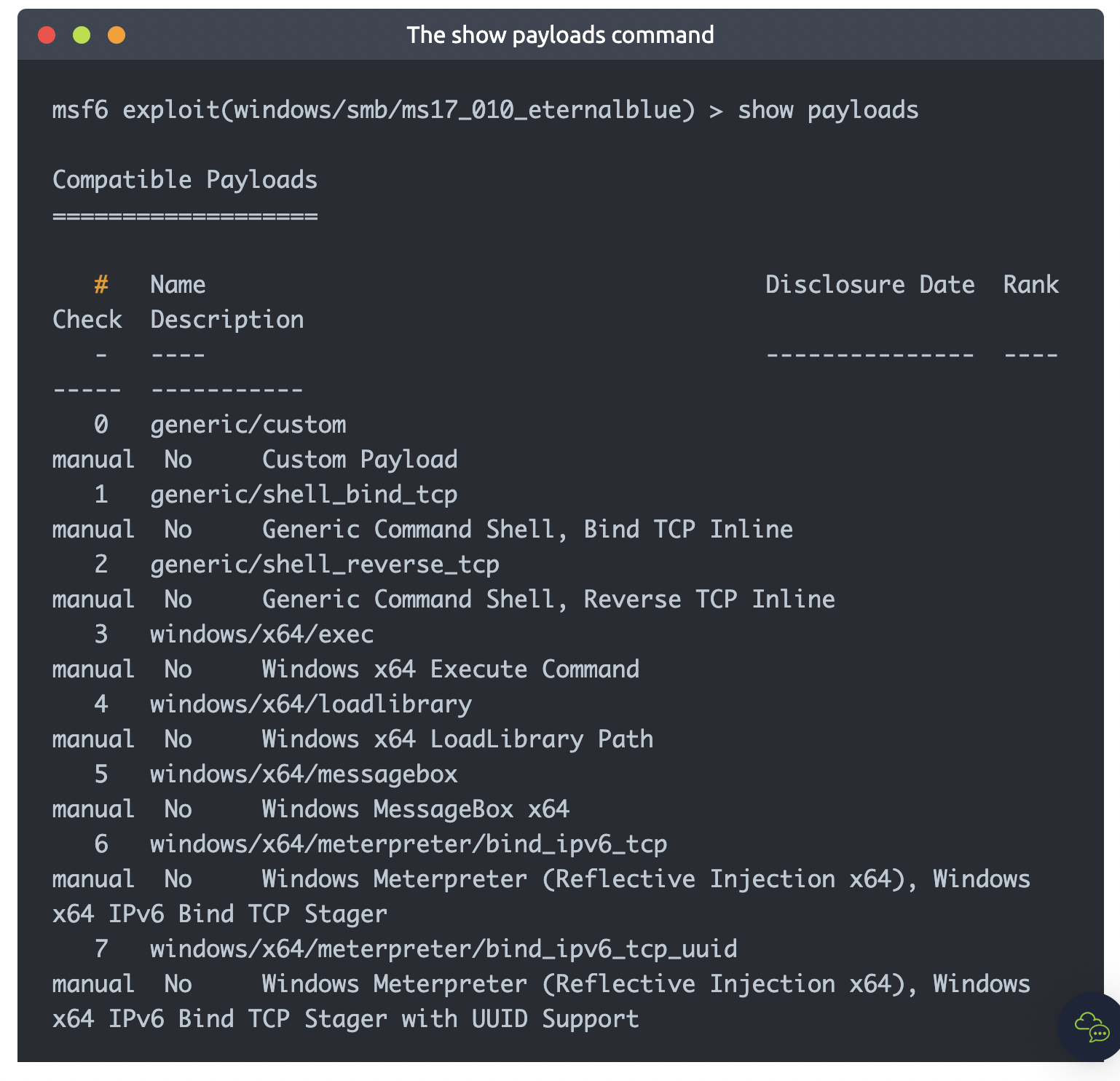


Lệnh use 0 tương tự như use exploit/windows/smb/ms17\_010\_eternalblue

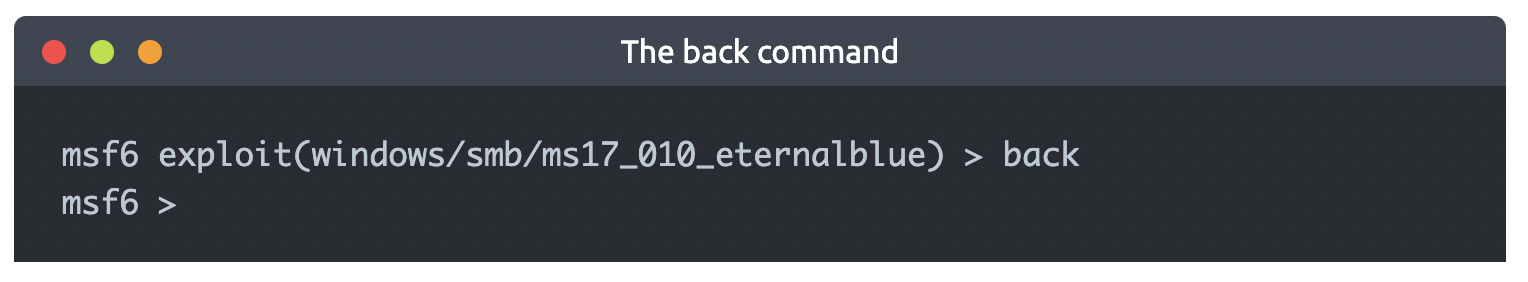


Ở đây ta cần đặt lại RHOSTS và RPORT.

Lệnh show có thể dùng với các module như auxiliary, payload, exploit để liệt kê.

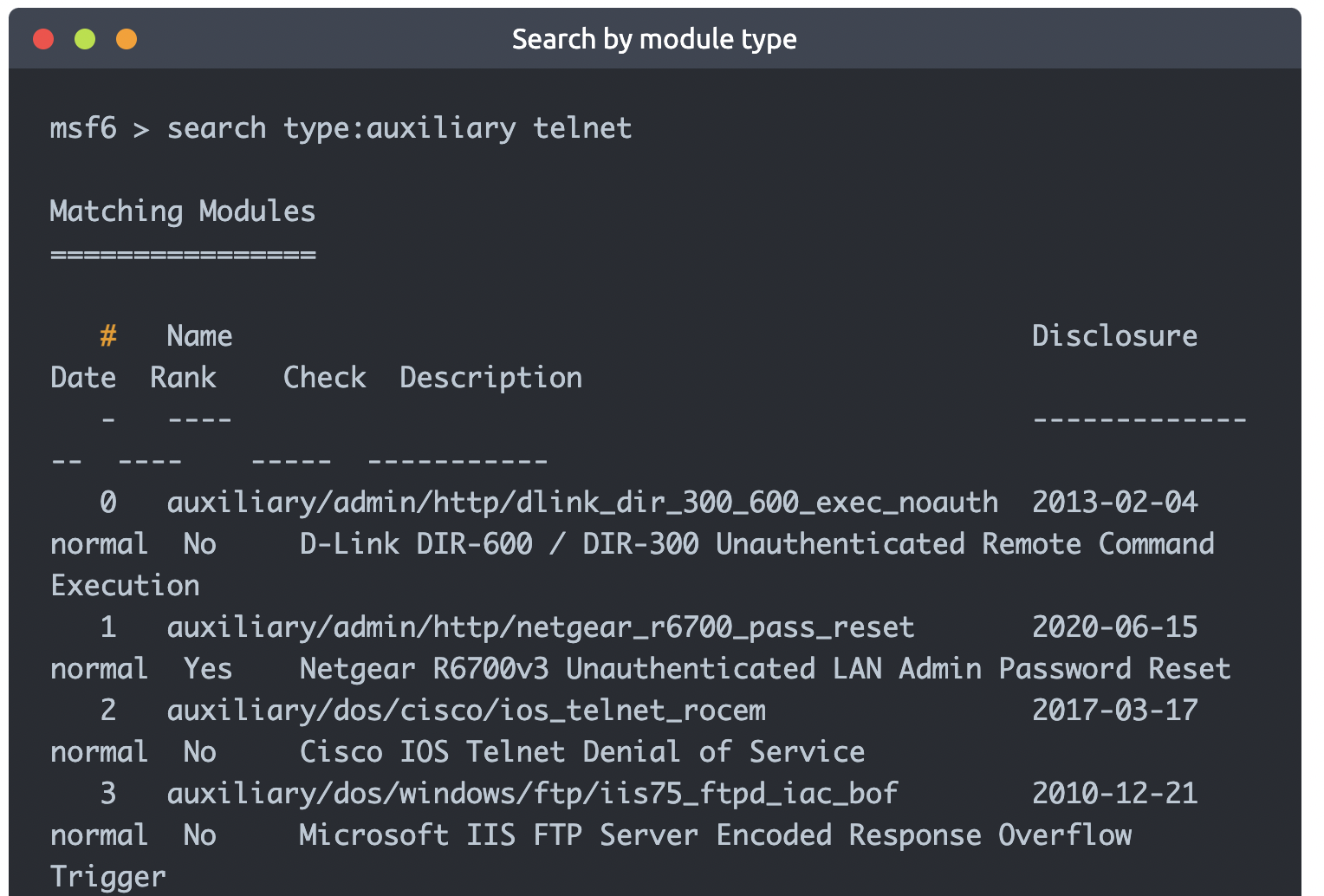


Bạn có thể quay lại bối cảnh trước bằng lệnh back.



Lệnh info để xem chi tiết module.

Bạn có thể tìm kiếm bằng keywords cũng như type và platform.



Tất cả tham số được set theo cú pháp sau:

set PARAMETER\_NAME VALUE

Các tham số bạn thường dùng là:

* RHOSTS
* RPORT
* PAYLOAD
* LHOST
* LPORT
* SESSION

Bạn có thể ghi đè bất cứ tham số nào bằng lệnh set. Bạn cũng có thể xóa 1 giá trị của tham số bằng unset hay unset all để xóa hết.

Bạn có thể dùng lệnh setg (set global) để đặt giá trị cho tất cả modules.

**Using modules**

Khi tất cả tham số của module được đặt lại, bạn có thể chạy module bằng lệnh exploit. Metasploit cũng hỗ trợ lệnh run, đây là 1 bí danh được tạo cho lệnh khai thác vì từ khai thác không hợp với các module không phải là exploits như port scanners, vulnerability scanners,…

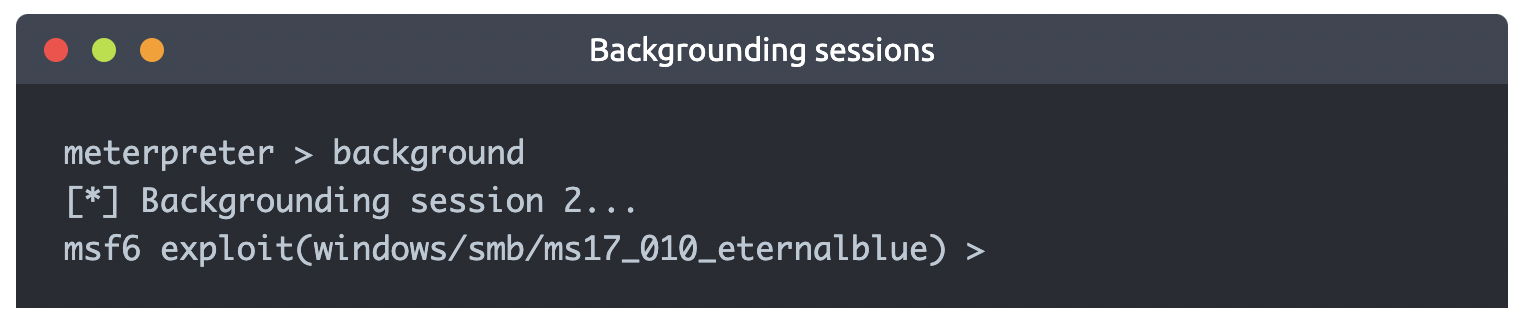
Lệnh exploit -z sẽ chạy khai thác và background session ngay khi phiên được mở.

Một vài modules hỗ trợ lệnh check. Nó sẽ kiểm tra hệ thống đích có dính lỗ hỗng để khai thác không.

**Sessions**

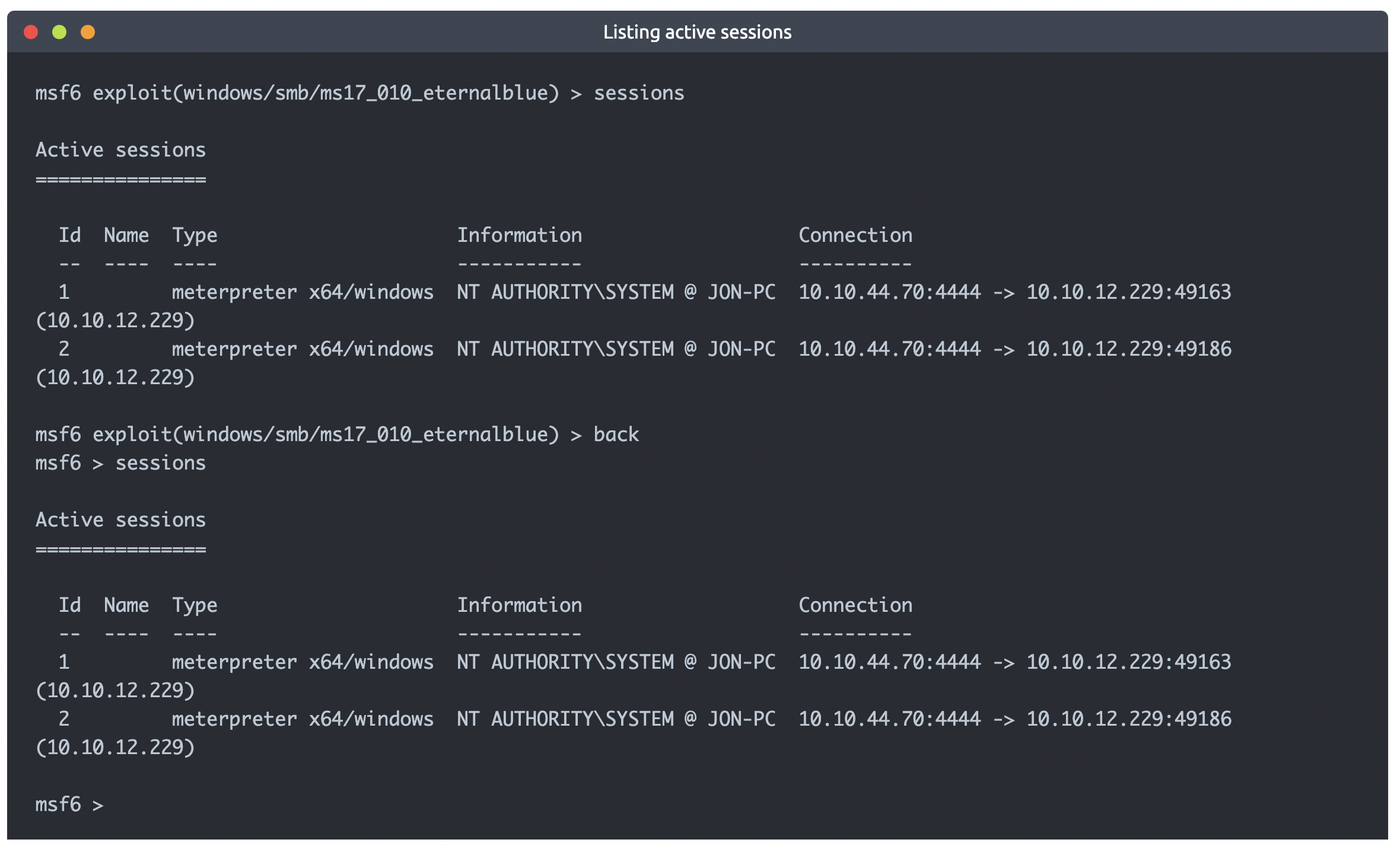
Khi một lỗ hổng được khai thác thành công, 1 phiên sẽ được tạo. Đây là 1 kênh giao tiếp được thiết lập giữa hệ thống đích và Metasploit.

Bạn có thể dùng lệnh background để background phiên session và quay lại msfconsole.



Ctrl + Z có thể được dùng để background sessions.

Lệnh sessions có thể được dùng để xem các session đang chạy:



Bạn có thể dùng sessions -i cùng với số thứ tự của phiên để kết nối:

