MSFCONSOLE COMMANDS

TABLE OF CONTENTS

MSFCONSOLE CORE COMMANDS TUTORIAL

The MSFconsole has many different command options to choose from. The following are a core set of Metasploit commands with reference to their output.

(https://www.offensive-security.com/wp-content/uploads/2015/04/ms-core-commands.png)

msfconsole core commands | Metasploit Unleashed

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
exit	Exit the console
get	Gets the value of a context-specific va
getg	Gets the value of a global variable
go_pro	Launch Metasploit web GUI

grep	Grep the output of another command
help	Help menu
info	Displays information about one or more
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 p
makerc	Save commands entered since start to a
popm	Pops the latest module off the stack ar

```
previous      Sets the previously loaded module as th
pushm         Pushes the active or list of modules or
quit          Exit the console
reload_all    Reloads all modules from all defined mo
rename_job    Rename a job
resource      Run the commands stored in a file
route         Route traffic through a session
save          Saves the active datastores
search        Searches module names and descriptions
sessions      Dump session listings and display infor
```

```
set           Sets a context-specific variable to a v
setg          Sets a global variable to a value
show          Displays modules of a given type, or al
sleep         Do nothing for the specified number of
spool         Write console output into a file as wel
threads       View and manipulate background threads
unload        Unload a framework plugin
unset         Unsets one or more context-specific var
unsetg        Unsets one or more global variables
use           Selects a module by name
version       Show the framework and console library
```

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. This, however is not required. Just as you can in commercial routers, you can switch modules from within other modules. As a reminder, variables will only carry over if they are set globally.

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

BANNER

Simply displays a randomly selected banner

```
msf > banner

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

Frustrated with proxy pivoting? Upgrade to layer-2 VP
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
```

CHECK

There aren't many exploits that support it, but there is also a **check** option that will check to see if a target is vulnerable to a particular exploit instead of actually exploiting it.

PSNUFFLE (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/EXTENDING-PSNUFFLE/)

SNMP SWEEPING (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/SNMP-SCAN/)

WRITING YOUR OWN SCANNER (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/WRITING-SCANNER/)

WINDOWS PATCH ENUMERATION (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/PATCH-ENUMERATION/)

VULNERABILITY SCANNING (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/VULNERABILITY-SCANNING/)

SMB LOGIN CHECK (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/SMB-LOGIN-CHECK/)

VNC AUTHENTICATION (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/VNC-AUTHENTICATION/)

WMAP WEB SCANNER (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/WMAP-WEB-SCANNER/)

WORKING WITH NEXPOSE (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/WORKING-WITH-NEXPOSE/)

```
msf exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST    172.16.194.134   yes       The target add
   RPORT    445              yes       Set the SMB se
   SMBPIPE  BROWSER          yes       The pipe name

Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting

msf exploit(ms08_067_netapi) > check

[*] Verifying vulnerable status... (path: 0x0000005a)
[*] System is not vulnerable (status: 0x00000000)
[*] The target is not exploitable.
msf  exploit(ms08_067_netapi) >
```

COLOR

You can enable or disable if the output you get through the msfconsole will contain colors.

```
msf > color
Usage: color >'true'|'false'|'auto'>

Enable or disable color output.
```

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:
```

You can see all the additional options by issuing the **-h** parameter.

```
msf > connect -h
Usage: connect [options]

Communicate with a host, similar to interacting via r
any configured session pivoting.

OPTIONS:

    -C        Try to use CRLF for EOL sequence.
    -P <opt>  Specify source port.
    -S <opt>  Specify source address.
    -c <opt>  Specify which Comm to use.
    -h        Help banner.
    -i <opt>  Send the contents of a file.
    -p <opt>  List of proxies to use.
    -s        Connect with SSL.
    -u        Switch to a UDP socket.
    -w <opt>  Specify connect timeout.
    -z        Just try to connect, then return.

msf >
```

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-fram

##
# This module requires Metasploit: http//metasploit.c
# Current source: https://github.com/rapid7/metasploi
##

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 = {})
```

EXIT

The **exit** command will simply exit msfconsole.

```
msf exploit(ms10_061_spoolss) > exit
root@kali:~#
```

GREP

The **grep** command is similar to Linux grep. It matches a given pattern from the output of another msfconsole command. The following is an example of using **grep** to match output containing the string “http” from a **search** for modules containing the string “oracle”.

```
msf > grep
Usage: grep [options] pattern cmd

Grep the results of a console command (similar to Lir

OPTIONS:

    -A <opt&>  Show arg lines of output After a match
    -B   Show arg lines of output Before a match.
    -c        Only print a count of matching lines.
    -h        Help banner.
    -i        Ignore case.
    -k   Keep (include) arg lines at start of output.
    -m   Stop after arg matches.
    -s   Skip arg lines of output before attempting m
    -v        Invert match.
msf >
msf > grep http search oracle
   auxiliary/scanner/http/oracle_demantra_database_cr
   auxiliary/scanner/http/oracle_demantra_file_retrie
   auxiliary/scanner/http/oracle_ilom_login
   exploit/multi/http/glassfish_deployer
   exploit/multi/http/oracle_ats_file_upload
   exploit/multi/http/oracle_reports_rce
   exploit/windows/http/apache_chunked
   exploit/windows/http/bea_weblogic_post_bof
   exploit/windows/http/oracle9i_xdb_pass
   exploit/windows/http/oracle_beehive_evaluation
   exploit/windows/http/oracle_beehive_prepareaudiotc
   exploit/windows/http/oracle_btm_writetofile
   exploit/windows/http/oracle_endeca_exec
   exploit/windows/http/oracle_event_processing_uploa
   exploit/windows/http/osb_uname_jlist
```

HELP

The **help** command will give you a list and small description of all available commands.

SECURITY.COM/METASPLOIT-UNLEASHED/HOSTING-JAVASCRIPT/)

FINAL EXPLOIT (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/FINAL-EXPLOIT/)

CLIENT SIDE ATTACKS (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/CLIENT-SIDE-ATTACKS/)

BINARY PAYLOADS (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/BINARY-PAYLOADS/)

BINARY LINUX TROJAN (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/BINARY-LINUX-TROJAN/)

CLIENT SIDE EXPLOITS (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/CLIENT-SIDE-EXPLOITS/)

VBSCRIPT INFECTION METHODS (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/VBSCRIPT-INFECTION-METHODS/)

MSF POST EXPLOITATION (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/MSF-POST-EXPLOITATION/)

PRIVILEGE ESCALATION (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/PRIVILEGE-ESCALATION/)

```
msf > help

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

    Command       Description
    -------       -----------
    ?             Help menu
    banner        Display an awesome metasploit banne
    cd            Change the current working director
    color         Toggle color
    connect       Communicate with a host
...snip...

Database Backend Commands
=========================

    Command           Description
    -------           -----------
    db_connect        Connect to an existing database
    db_disconnect     Disconnect from the current dat
    db_export         Export a file containing the co
    db_import         Import a scan result file (file
...snip...
```

INFO

The **info** command will provide detailed information about a particular module including all options, targets, and other information. Be sure to always read the module description prior to using it as some may have un-desired effects.

The info command also provides the following information:

- The author and licensing information
- Vulnerability references (ie: CVE, BID, etc)
- Any payload restrictions the module may have

```
msf  exploit(ms09_050_smb2_negotiate_func_index) > ir

       Name: Microsoft SRV2.SYS SMB Negotiate Process
     Module: exploit/windows/smb/ms09_050_smb2_negoti
    Version: 14774
   Platform: Windows
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Good

Provided by:
  Laurent Gaffie <laurent.gaffie@gmail.com>
  hdm <hdm@metasploit.com>
  sf <stephen_fewer@harmonysecurity.com>

Available targets:
  Id  Name
  --  ----
  0   Windows Vista SP1/SP2 and Server 2008 (x86)

Basic options:
  Name   Current Setting  Required  Description
  ----   ---------------  --------  -----------
  RHOST                   yes       The target addres
  RPORT  445              yes       The target port
  WAIT   180              yes       The number of sec

Payload information:
  Space: 1024

Description:
  This module exploits an out of bounds function tabl
  the SMB request validation code of the SRV2.SYS dri
  Windows Vista, Windows 7 release candidates (not RT
  2008 Server prior to R2. Windows Vista without SP1
  affected by this flaw.

References:
  http://www.microsoft.com/technet/security/bulletin/
  http://cve.mitre.org/cgi-bin/cvename.cgi?name=2009-
  http://www.securityfocus.com/bid/36299
  http://www.osvdb.org/57799
```

```
  http://seclists.org/fulldisclosure/2009/Sep/0039.ht
  http://www.microsoft.com/technet/security/Bulletin/

msf  exploit(ms09_050_smb2_negotiate_func_index) >
```

IRB

Running the **irb** command will drop you into a live Ruby interpreter shell where you can issue commands and create Metasploit scripts on the fly. This feature is also very useful for understanding the internals of the Framework.

```
msf > irb
[*] Starting IRB shell...

>> puts "Hello, metasploit!"
Hello, metasploit!
=> nil
>> Framework::Version
=> "4.8.2-2014022601"
```

JOBS

Jobs are modules that are running in the background. The **jobs** command provides the ability to list and terminate these jobs.

```
msf > jobs -h
Usage: jobs [options]

Active job manipulation and interaction.

OPTIONS:

    -K        Terminate all running jobs.
    -h        Help banner.
    -i   Lists detailed information about a running j
    -k   Terminate the specified job name.
    -l        List all running jobs.
    -v        Print more detailed info.  Use with -i

msf >
```

KILL

The **kill** command will kill any running jobs when supplied with the job id.

```
msf exploit(ms10_002_aurora) > kill 0
Stopping job: 0...

[*] Server stopped.
```

LOAD

The **load** command loads a plugin from Metasploit's **plugin** directory. Arguments are passed as **key=val** on the shell.

```
msf > load
Usage: load  [var=val var=val ...]

Loads a plugin from the supplied path.  If path is nc
in the user's plugin directory (/root/.msf4/plugins)
in the framework root plugin directory (/usr/share/me
The optional var=val options are custom parameters th

msf > load pcap_log
[*] PcapLog plugin loaded.
[*] Successfully loaded plugin: pcap_log
```

LOADPATH

The **loadpath** command will load a third-part module tree for the path so you can point Metasploit at your 0-day exploits, encoders, payloads, etc.

```
msf > loadpath /home/secret/modules

Loaded 0 modules.
```

UNLOAD

Conversely, the **unload** command unloads a previously loaded plugin and removes any extended commands.

```
msf > unload pcap_log
Unloading plugin pcap_log...unloaded.
```

RESOURCE

The **resource** command runs resource (batch) files that can be loaded through msfconsole.

```
msf > resource
Usage: resource path1 [path2 ...]

Run the commands stored in the supplied files.  Resou
ruby code between  tags.

See also: makerc
```

Some attacks, such as Karmetasploit, use resource files to run a set of commands in a **karma.rc** file to create an attack. Later, we will discuss how, outside of Karmetasploit, that can be very useful.

```
msf > resource karma.rc
[*] Processing karma.rc for ERB directives.
resource (karma.rc_.txt)> db_connect postgres:toor@12
resource (karma.rc_.txt)> use auxiliary/server/browse
...snip...
```

Batch files can greatly speed up testing and development times as well as allow the user to automate many tasks. Besides loading a batch file from within msfconsole, they can also be passed at startup using the **-r** flag. The simple example below creates a batch file to display the Metasploit version number at startup.

```
root@kali:~# echo version > version.rc
root@kali:~# msfconsole -r version.rc


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

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


       =[ metasploit v4.8.2-2014021901 [core:4.8 api:
+ -- --=[ 1265 exploits - 695 auxiliary - 202 post ]
+ -- --=[ 330 payloads - 32 encoders - 8 nops      ]

[*] Processing version.rc for ERB directives.
resource (version.rc)> version
Framework: 4.8.2-2014022601
Console  : 4.8.2-2014022601.15168
msf >
```

ROUTE

The **route** command in Metasploit allows you to route sockets through a session or ‘comm’, providing basic pivoting capabilities. To add a route, you pass the target subnet and network mask followed by the session (comm) number.

MODULE-REFERENCE/)

ADMIN HTTP AUXILIARY MODULES (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/ADMIN-HTTP-AUXILIARY-MODULES/)

ADMIN MYSQL AUXILIARY MODULES (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/ADMIN-MYSQL-AUXILIARY-MODULES/)

ADMIN MSSQL AUXILIARY MODULES (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/ADMIN-MSSQL-AUXILIARY-MODULES/)

ADMIN POSTGRES AUXILIARY MODULES (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/ADMIN-POSTGRES-AUXILIARY-MODULES/)

ADMIN VMWARE AUXILIARY MODULES (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/ADMIN-VMWARE-AUXILIARY-MODULES/)

SCANNER DCERPC AUXILIARY MODULES (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/SCANNER-DCERPC-AUXILIARY-MODULES/)

SCANNER DISCOVERY AUXILIARY MODULES (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/SCANNER-DISCOVERY-AUXILIARY-MODULES/)

```
meterpreter > route -h
Route traffic destined to a given subnet through a su

Usage:
  route [add/remove] subnet netmask [comm/sid]
  route [add/remove] cidr [comm/sid]
  route [get]
  route [flush]
  route [print]

Subcommands:
  add - make a new route
  remove - delete a route; 'del' is an alias
  flush - remove all routes
  get - display the route for a given target
  print - show all active routes

Examples:
  Add a route for all hosts from 192.168.0.0 to 192.1
    route add 192.168.0.0 255.255.255.0 1
    route add 192.168.0.0/24 1

  Delete the above route
    route remove 192.168.0.0/24 1
    route del 192.168.0.0 255.255.255.0 1

  Display the route that would be used for the given
    route get 192.168.0.11

meterpreter >
```

SCANNER FTP AUXILIARY MODULES (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/SCANNER-FTP-AUXILIARY-MODULES/)

SCANNER HTTP AUXILIARY MODULES (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/SCANNER-HTTP-AUXILIARY-MODULES/)

SCANNER MYSQL AUXILIARY MODULES (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/SCANNER-MYSQL-AUXILIARY-MODULES/)

SCANNER MSSQL AUXILIARY MODULES (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/SCANNER-MSSQL-AUXILIARY-MODULES/)

SCANNER IMAP AUXILIARY MODULES (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/SCANNER-IMAP-AUXILIARY-MODULES/)

SCANNER NETBIOS AUXILIARY MODULES (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/SCANNER-NETBIOS-AUXILIARY-MODULES/)

SCANNER POP3 AUXILIARY MODULES (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/SCANNER-POP3-AUXILIARY-MODULES/)

SCANNER SMB AUXILIARY MODULES (HTTPS://WWW.OFFENSIVE-

```
meterpreter > route

Network routes
==============

    Subnet           Netmask          Gateway
    ------           -------          -------
    0.0.0.0          0.0.0.0          172.16.1.254
    127.0.0.0        255.0.0.0        127.0.0.1
    172.16.1.0       255.255.255.0    172.16.1.100
    172.16.1.100     255.255.255.255  127.0.0.1
    172.16.255.255   255.255.255.255  172.16.1.100
    224.0.0.0        240.0.0.0        172.16.1.100
    255.255.255.255  255.255.255.255  172.16.1.100
```

SEARCH

The msfconsole includes an extensive regular-expression based search functionality. If you have a general idea of what you are looking for, you can search for it via **search**. In the output below, a search is being made for MS Bulletin MS09-011. The search function will locate this string within the module names, descriptions, references, etc.

Note the naming convention for Metasploit modules uses underscores versus hyphens.

```
msf > search usermap_script

Matching Modules
================

   Name                                Disclosure Dat
   ----                                --------------
   exploit/multi/samba/usermap_script  2007-05-14

msf >
```

SECURITY.COM/METASPLOIT-UNLEASHED/SCANNER-SMB-AUXILIARY-MODULES/)

SCANNER SMTP AUXILIARY MODULES (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/SCANNER-SMTP-AUXILIARY-MODULES/)

SCANNER SNMP AUXILIARY MODULES (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/SCANNER-SNMP-AUXILIARY-MODULES/)

SCANNER SSH AUXILIARY MODULES (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/SCANNER-SSH-AUXILIARY-MODULES/)

SCANNER TELNET AUXILIARY MODULES (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/SCANNER-TELNET-AUXILIARY-MODULES/)

SCANNER TFTP AUXILIARY MODULES (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/SCANNER-TFTP-AUXILIARY-MODULES/)

SCANNER VMWARE AUXILIARY MODULES (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/SCANNER-VMWARE-AUXILIARY-MODULES/)

SCANNER VNC AUXILIARY MODULES (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/SCANNER-VNC-AUXILIARY-MODULES/)

help

You can further refine your searches by using the built-in keyword system.

```
msf > help search
Usage: search [keywords]

Keywords:
  app       :  Modules that are client or server atta
  author    :  Modules written by this author
  bid       :  Modules with a matching Bugtraq ID
  cve       :  Modules with a matching CVE ID
  edb       :  Modules with a matching Exploit-DB ID
  name      :  Modules with a matching descriptive na
  platform  :  Modules affecting this platform
  ref       :  Modules with a matching ref
  type      :  Modules of a specific type (exploit, a

Examples:
  search cve:2009 type:exploit app:client

msf >
```

name

To search using a descriptive name, use the **name** keyword.

SERVER CAPTURE AUXILIARY MODULES (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/SERVER-CAPTURE-AUXILIARY-MODULES/)

RECENT CHANGES TO METASPLOIT UNLEASHED (HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/RECENT-CHANGES/)

```
msf > search name:mysql

Matching Modules
================

   Name
   ----
   auxiliary/admin/mysql/mysql_enum
   auxiliary/admin/mysql/mysql_sql
   auxiliary/analyze/jtr_mysql_fast
   auxiliary/scanner/mysql/mysql_authbypass_hashdump
   auxiliary/scanner/mysql/mysql_hashdump
   auxiliary/scanner/mysql/mysql_login
   auxiliary/scanner/mysql/mysql_schemadump
   auxiliary/scanner/mysql/mysql_version
   exploit/linux/mysql/mysql_yassl_getname
   exploit/linux/mysql/mysql_yassl_hello
   exploit/windows/mysql/mysql_payload
   exploit/windows/mysql/mysql_yassl_hello
msf >
```

platform

You can use **platform** to narrow down your search to modules that affect a specific platform.

```
msf > search platform:aix

Matching Modules
================

   Name                                      Disclosure D
   ----                                      ------------
   payload/aix/ppc/shell_bind_tcp
   payload/aix/ppc/shell_find_port
   payload/aix/ppc/shell_interact
...snip...
```

type

Using the **type** lets you filter by module type such as auxiliary, post, exploit, etc.

```
msf > search type:post

Matching Modules
================

   Name
   ----
   post/linux/gather/checkvm
   post/linux/gather/enum_cron
   post/linux/gather/enum_linux
...snip...
```

author

Searching with the **author** keyword lets you search for modules by your favourite author.

```
msf > search author:dookie

Matching Modules
================

   Name
   ----
   exploit/osx/http/evocam_webserver
   exploit/osx/misc/ufo_ai
   exploit/windows/browser/amaya_bdo
...snip...
```

multiple

You can also combine multiple keywords together to further narrow down the returned results.

```
msf > search cve:2011 author:jduck platform:linux

Matching Modules
================

   Name                                              Disc
   ----                                              ----
   exploit/linux/misc/netsupport_manager_agent  2011-
```

SESSIONS

The **sessions** command allows you to list, interact with, and kill spawned sessions. The sessions can be shells, Meterpreter sessions, VNC, etc.

```
msf > sessions -h
Usage: sessions [options] or sessions [id]

Active session manipulation and interaction.

OPTIONS:

    -C   Run a Meterpreter Command on the session giv
    -K        Terminate all sessions
    -c   Run a command on the session given with -i,
    -h        Help banner
    -i   Interact with the supplied session ID
    -k   Terminate sessions by session ID and/or rang
    -l        List all active sessions
    -q        Quiet mode
    -r        Reset the ring buffer for the session g
    -s   Run a script on the session given with -i, c
    -t   Set a response timeout (default: 15)
    -u   Upgrade a shell to a meterpreter session on
    -v        List sessions in verbose mode
    -x        Show extended information in the sessic

Many options allow specifying session ranges using cc
For example:  sessions -s checkvm -i 1,3-5  or  sessi
```

To list any active sessions, pass the **-l** options to **sessions**.

```
msf exploit(3proxy) > sessions -l

Active sessions
===============

  Id  Description    Tunnel
  --  -----------    ------
  1   Command shell  192.168.1.101:33191 -> 192.168.1
```

To interact with a given session, you just need to use the **-i** switch followed by the Id number of the session.

```
msf exploit(3proxy) > sessions -i 1
[*] Starting interaction with 1...

C:WINDOWSsystem32>
```

SET

The **set** command allows you to configure Framework options and parameters for the current module you are working with.

```
msf auxiliary(ms09_050_smb2_negotiate_func_index) > s
RHOST => 172.16.194.134
msf auxiliary(ms09_050_smb2_negotiate_func_index) > s

Module options (exploit/windows/smb/ms09_050_smb2_neg

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST  172.16.194.134   yes       The target addre
   RPORT  445              yes       The target port
   WAIT   180              yes       The number of se

Exploit target:

   Id  Name
   --  ----
   0   Windows Vista SP1/SP2 and Server 2008 (x86)
```

Metasploit also allows you to set an encoder to use at run-time. This is particularly useful in exploit development when you aren't quite certain as to which payload encoding methods will work with a given exploit.

```
msf  exploit(ms09_050_smb2_negotiate_func_index) > sh

Compatible Encoders
===================

   Name                    Disclosure Date  Rank
   ----                    ---------------  ----
   generic/none                             normal
   x86/alpha_mixed                          low
   x86/alpha_upper                          low
   x86/avoid_utf8_tolower                   manual
   x86/call4_dword_xor                      normal
   x86/context_cpuid                        manual
   x86/context_stat                         manual
   x86/context_time                         manual
   x86/countdown                            normal
   x86/fnstenv_mov                          normal
   x86/jmp_call_additive                    normal
   x86/nonalpha                             low
   x86/nonupper                             low
   x86/shikata_ga_nai                       excellent
   x86/single_static_bit                    manual
   x86/unicode_mixed                        manual
   x86/unicode_upper                        manual
```

unset

The opposite of the **set** command, of course, is **unset**. **unset** removes a parameter previously configured with **set**. You can remove all assigned variables with **unset all**.

```
msf > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf > set THREADS 50
THREADS => 50
msf > set

Global
======

  Name     Value
  ----     -----
  RHOSTS   192.168.1.0/24
  THREADS  50

msf > unset THREADS
Unsetting THREADS...
msf > unset all
Flushing datastore...
msf > set

Global
======

No entries in data store.

msf >
```

SETG

In order to save a lot of typing during a pentest, you can set *global variables* within msfconsole. You can do this with the **setg** command. Once these have been set, you can use them in as many exploits and auxiliary modules as you like. You can also save them for use the next time you start msfconsole. However, the pitfall is forgetting you have saved globals, so always check your options before you **run** or **exploit**. Conversely, you can use the **unsetg** command to unset a global variable. In the

examples that follow, variables are entered in all-caps (ie: LHOST), but Metasploit is case-insensitive so it is not necessary to do so.

```
msf > setg LHOST 192.168.1.101
LHOST => 192.168.1.101
msf > setg RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf > setg RHOST 192.168.1.136
RHOST => 192.168.1.136
```

After setting your different variables, you can run the **save** command to save your current environment and settings. With your settings saved, they will be automatically loaded on startup, which saves you from having to set everything again.

```
msf > save
Saved configuration to: /root/.msf4/config
msf >
```

SHOW

Entering **show** at the msfconsole prompt will display every module within Metasploit.

```
msf > show

Encoders
========

   Name                    Disclosure Date  Rank
   ----                    ---------------  ----
   cmd/generic_sh                           good
   cmd/ifs                                  low
   cmd/printf_php_mq                        manual
...snip...
```

There are a number of **show** commands you can use but the ones you will use most frequently are **show auxiliary**, **show exploits**, **show payloads**, **show encoders**, and **show nops**.

auxiliary

Executing **show auxiliary** will display a listing of all of the available auxiliary modules within Metasploit. As mentioned earlier, auxiliary modules include scanners, denial of service modules, fuzzers, and more.

```
msf > show auxiliary
Auxiliary
=========

   Name
   ----
   admin/2wire/xslt_password_reset
   admin/backupexec/dump
   admin/backupexec/registry
...snip...
```

exploits

Naturally, **show exploits** will be the command you are most interested in running since at its core, Metasploit is all about exploitation. Run **show exploits** to get a listing of all exploits contained in the framework.

```
msf > show exploits

Exploits
========

   Name
   ----
   aix/rpc_cmsd_opcode21
   aix/rpc_ttdbserverd_realpath
   bsdi/softcart/mercantec_softcart
...snip...
```

Using MSFconsole Payloads

Running **show payloads** will display all of the different payloads for all platforms available within Metasploit.

```
msf > show payloads

Payloads
========

   Name                                                D
   ----                                                -
   aix/ppc/shell_bind_tcp
   aix/ppc/shell_find_port
   aix/ppc/shell_interact
...snip...
```

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 particular 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                                                D
   ----                                                -
   generic/custom
   generic/debug_trap
   generic/shell_bind_tcp
...snip...
```

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 add
   RPORT    445              yes       Set the SMB se
   SMBPIPE  BROWSER          yes       The pipe name

Exploit target:

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

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...
```

ADVANCED

If you wish the further fine-tune an exploit, you can see more advanced options by running **show advanced**.

```
msf exploit(ms08_067_netapi) > show advanced

Module advanced options:

   Name           : CHOST
   Current Setting:
   Description    : The local client address

   Name           : CPORT
   Current Setting:
   Description    : The local client port

...snip...
```

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
   ----                     ---------------  ----
   cmd/generic_sh                            good
   cmd/ifs                                   low
   cmd/printf_php_mq                         manual
   generic/none                              normal
   mipsbe/longxor                            normal
   mipsle/longxor                            normal
   php/base64                                great
   ppc/longxor                               normal
   ppc/longxor_tag                           normal
   sparc/longxor_tag                         normal
   x64/xor                                   normal
   x86/alpha_mixed                           low
   x86/alpha_upper                           low
   x86/avoid_utf8_tolower                    manual
   x86/call4_dword_xor                       normal
   x86/context_cpuid                         manual
   x86/context_stat                          manual
   x86/context_time                          manual
   x86/countdown                             normal
   x86/fnstenv_mov                           normal
   x86/jmp_call_additive                     normal
   x86/nonalpha                              low
   x86/nonupper                              low
   x86/shikata_ga_nai                        excellent
   x86/single_static_bit                     manual
   x86/unicode_mixed                         manual
   x86/unicode_upper                         manual
```

NOPS

Lastly, issuing the **show nops** command will display the NOP Generators that Metasploit has to offer.

```
msf > show nops
NOP Generators
==============

   Name             Disclosure Date  Rank    Descript
   ----             ---------------  ----    --------
   armle/simple                      normal  Simple
   mipsbe/better                     normal  Better
   php/generic                       normal  PHP Nop
   ppc/simple                        normal  Simple
   sparc/random                      normal  SPARC NO
   tty/generic                       normal  TTY Nop
   x64/simple                        normal  Simple
   x86/opty2                         normal  Opty2
   x86/single_byte                   normal  Single B
```

USE

When you have decided on a particular module to make use of, issue the **use** command to select it. The **use** command changes your context to a specific module, exposing type-specific commands. Notice in the output below that any global variables that were previously set are already configured.

```
msf > use dos/windows/smb/ms09_001_write
msf auxiliary(ms09_001_write) > show options

Module options:

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST                   yes       The target addre
   RPORT  445              yes       Set the SMB serv

msf auxiliary(ms09_001_write) >
```

At any time you need assistance you can use the msfconsole **help** command to display available options.

‹ PREVIOUS PAGE
Using the MSFconsole Interface (https://www.offensive-security.com/metasploit-unleashed/msfconsole/)

NEXT PAGE ›
Working with Active and Passive Exploits in Metasploit (https://www.offensive-security.com/metasploit-unleashed/exploits/)

READY TO ENROLL?

Register for a course (https://www.offensive-security.com/pre-reg/)

COURSES

Penetration Testing with Kali Linux (PEN-200) (https://www.offensive-security.com/pwk-oscp/)

Offensive Security Wireless Attacks (PEN-210) (https://www.offensive-security.com/wifu-oswp/)

Evasion Techniques and Breaching Defenses (PEN-300) (https://www.offensive-security.com/pen300-osep/)

Advanced Web Attacks and Exploitation (WEB-300) (https://www.offensive-security.com/awae-oswe/)

Advanced Windows Exploitation (EXP-401) (https://www.offensive-security.com/awe-osee/)

Courses and Certifications Overview (https://www.offensive-security.com/courses-and-certifications/)

CERTIFICATIONS

OSWE Web Expert (https://www.offensive-security.com/awae-oswe/)

OSCP Certified Professional (https://www.offensive-security.com/pwk-oscp/)

OSEP Experienced Pentester (https://www.offensive-security.com/pen300-osep/)

OSWP Wireless Professional (https://www.offensive-security.com/wifu-oswp/)

OSEE Exploitation Expert (https://www.offensive-security.com/awe-osee/)

PROVING GROUNDS (HOSTED LABS) (HTTPS://WWW.OFFENSIVE-SECURITY.COM/LABS/)

Proving Grounds Play and Practice (https://www.offensive-security.com/labs/individual/)

Proving Grounds for Teams and Orgs (https://www.offensive-security.com/labs/enterprise/)

User-Generated Content (https://www.offensive-security.com/labs/submit/)

SECURITY SERVICES

Offsec Academy (https://www.offensive-security.com/academy/)

OffSec for Orgs (https://www.offensive-security.com/offsec-for-orgs/)

Authorized Training Partners (https://www.offensive-security.com/offsec-for-orgs/training-partners/)

Penetration Testing Services (https://www.offensive-security.com/penetration-testing/)

Advanced Attack Simulation (https://www.offensive-security.com/penetration-testing/#other-services)

Application Security Assessment (https://www.offensive-security.com/penetration-testing/#asa)

ABOUT OFFSEC (HTTPS://WWW.OFFENSIVE-SECURITY.COM/WHY-OFFSEC/)

Why OffSec (https://www.offensive-security.com/why-offsec/)

Leadership Team (https://www.offensive-security.com/leadership-team/)

Our Core Values (https://www.offensive-security.com/values/)

Try Harder Ethos (/why-offsec/#try-harder)

Blog (https://www.offensive-security.com/blog/)

Bug Bounty Program (https://www.offensive-security.com/bug-bounty-program/)

Contact Us (https://www.offensive-security.com/contact-us/)

KALI AND COMMUNITY (HTTPS://WWW.OFFENSIVE-SECURITY.COM/COMMUNITY-PROJECTS/)

OffSec Community (https://portal.offensive-security.com/sign-up/community)

Kali Linux (https://www.kali.org/)

DOWNLOADS

Kali Linux Virtual Machines (https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-image-download/)

Kali Linux ARM Images (https://www.offensive-security.com/kali-linux-arm-images/)

RESOURCES

Pricing (https://www.offensive-security.com/courses-and-certifications/)

FAQ (https://help.offensive-security.com/hc/en-us)

Kali NetHunter (https://www.kali.org/kali-linux-nethunter/)

Exploit Database (https://www.exploit-db.com/)

VulnHub (https://www.vulnhub.com/)

Google Hacking Database (https://www.exploit-db.com/google-hacking-database)

Metasploit Unleashed (/metasploit-unleashed/)

Kali Linux NetHunter Images (https://www.offensive-security.com/kali-linux-nethunter-download/)

Careers (https://www.offensive-security.com/careers/)

Join Our Email List (https://learn.offensive-security.com/subscribe-newsletter)

Official OffSec Swag (https://hackerwarehouse.com/security/)


(https://twitter.com/offsectraining)


(https://www.facebook.com/offsec/)


(https://www.linkedin.com/company/offensive-security/)