Msfconsole a11y.text Msfconsole msfconsole help command output What is the MSFconsole? a11y.text What is the MSFconsole? The msfconsole is probably the most popular interface to the Metasploit Framework (MSF). It provides an â€œall-in-oneâ€• centralized console and allows you efficient access to virtually all of 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 a11y.text 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 readline 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 > Launching MSFconsole a11y.text Launching MSFconsole The MSFconsole is launched by simply running msfconsole from the command line. MSFconsole is located in the /usr/share/metasploit-framework/msfconsole directory. The -q option removes the launch banner by starting msfconsole in quiet mode. root@kali:# msfconsole -q

msf > How to Use the Command Prompt a11y.text How to Use the Command Prompt You can pass -h to msfconsole to see the other usage options available to you. root@kali : ~ # msfconsole -h

Usage: msfconsole [options]

Common options

-E, --environment ENVIRONMENT The Rails environment. Will use RAIL_ENV environment variable if that is set. Defaults to production if neither option not RAILS_ENV environment variable is set.

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, --version Show version

Module options

--defer-module-loads Defer module loading unless explicitly asked.

-m, --module-path DIRECTORY An additional module path

Console options:

-a, --ask Ask before exiting Metasploit or accept 'exit -y'

-d, --defanged Execute the console as defanged

-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 string as console commands (use ; for multiples)

-h, --help Show this message 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

previous Sets the previously loaded module as the current module

pushm Pushes the active or list of modules onto the module stack

quit Exit the console

reload_all Reloads all modules from all defined module paths

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 information about sessions

set Sets a context-specific variable to a value

setg Sets a global variable to a value

show Displays modules of a given type, or all modules

sleep Do nothing for the specified number of seconds

spool Write console output into a file as well the screen

threads View and manipulate background threads

unload Unload a framework plugin

unset Unsets one or more context-specific variables

unsetg Unsets one or more global variables

use Selects a module by name

version Show the framework and console library version numbers

Database Backend Commands

==========

Command Description

------- -----------

creds List all credentials in the database

db_connect Connect to an existing database

db_disconnect Disconnect from the current database instance

db_export Export a file containing the contents of the database

db_import Import a scan result file (filetype will be auto-detected)

db_nmap Executes nmap and records the output automatically

db_rebuild_cache Rebuilds the database-stored module cache

db_status Show the current database status

hosts List all hosts in the database

loot List all loot in the database

notes List all notes in the database

services List all services in the database

vulns List all vulnerabilities in the database

workspace Switch between database workspaces Tab Completion a11y.text Tab Completion

The MSFconsole is designed to be fast to use and one of the features that helps this goal is tab completion. With the wide array of modules available, it can be difficult to remember the exact name

and path of the particular module you wish to make use of. As with most other shells, entering what you know and pressing â€˜Tabâ€™ will present you with a list of options available to you or auto-complete the string if there is only one option. Tab completion depends on the ruby readline extension and nearly every command in the console supports tab completion. use exploit/windows/dce use .*netapi.* set LHOST show set TARGET set PAYLOAD windows/shell/ exp msf > use exploit/windows/smb/ms

use exploit/windows/smb/ms03_049_netapi

use exploit/windows/smb/ms04_007_killbill

use exploit/windows/smb/ms04_011_lsass

use exploit/windows/smb/ms04_031_netdde

use exploit/windows/smb/ms05_039_pnp

use exploit/windows/smb/ms06_025_rasmans_reg

use exploit/windows/smb/ms06_025_rras

use exploit/windows/smb/ms06_040_netapi

use exploit/windows/smb/ms06_066_nwapi

use exploit/windows/smb/ms06_066_nwwks

use exploit/windows/smb/ms06_070_wkssvc

use exploit/windows/smb/ms07_029_msdns_zonename

use exploit/windows/smb/ms08_067_netapi

use exploit/windows/smb/ms09_050_smb2_negotiate_func_index

use exploit/windows/smb/ms10_046_shortcut_icon_dllloader

use exploit/windows/smb/ms10_061_spoolss

use exploit/windows/smb/ms15_020_shortcut_icon_dllloader

msf > use exploit/windows/smb/ms08_067_netapi The MSFconsole is the most commonly used interface for Metasploit. Making yourself familiar with theseÂ msfconsoleÂ commands will help you throughout this course and give you a strong foundation for working with Metasploit in general. Next