



---

# Pro Console User Guide

Release 4.6

Last Updated: April 10, 2013

# TABLE OF CONTENTS

## About This Guide

Target Audience .....	1
Organization .....	1
Document Conventions .....	2
Support .....	2
Support for Metasploit Pro and Metasploit Express .....	2
Support for the Metasploit Framework and Metasploit Community .....	2

## Metasploit Console Overview

About the Metasploit Console .....	6
Console Tasks .....	7
Launching the Metasploit Console .....	7
Launching the Metasploit Console on Windows .....	7
Launching the Metasploit Console on Linux .....	8
Basic Terms .....	9
Vulnerability .....	9
Exploit .....	9
Payload .....	9
Reverse Shell .....	9
Bind Shell .....	9
Modules .....	9
Exploit Modules .....	10
Auxiliary Modules .....	10
Post-Exploitation Modules .....	10
Payloads .....	11
NOP Generators .....	11
Payload Encoders .....	11
General Metasploit Workflow .....	11
Choose a Module .....	11
Configure the Module .....	12
Set the Options .....	13
Select a Target .....	13
Select a Payload .....	13
Run the Module .....	14

## Administration

Metasploit Console Management .....	15
Logging Input and Output from the Console .....	15
Changing the Log Verbosity .....	15
Logging Input and Output for a Session .....	16
User Account Management .....	16
Changing the Current User .....	16
Viewing a List of Users .....	16
Database Management .....	16
Connecting to the Database .....	17
Verifying the Database Connection .....	17
Disconnecting from the Database .....	17
Viewing the Current Database Status .....	18
Task Management .....	18
Viewing All Tasks .....	18
Viewing Running Tasks .....	18
Viewing the Task Log .....	19
Canceling a Running Task .....	19

## Projects

About Projects .....	20
Working with Projects .....	20
Creating a Project .....	21
Viewing the Current Project .....	21
Changing the Project .....	21
Deleting a Project .....	22

## Host Discovery

About Host Discovery .....	23
Managing Hosts .....	23
Scanning for Hosts .....	23
Specifying SMB Credentials .....	24
Viewing a List of Hosts .....	24
Adding a Host .....	24
Deleting a Host .....	25
Connecting to a Host .....	25
Outputting Host Data to CSV File .....	25
Importing Scan Data .....	25
Viewing Hosts that Are Up .....	27

Viewing Specific Columns from the Hosts Table .....	27
Viewing Loot .....	28
Outputting Host Data .....	28
Managing Vulnerabilities .....	28
Listing Vulnerabilities by Port.....	29
Listing Vulnerabilities by Service .....	29
Searching for a Vulnerability.....	29
Managing Notes .....	29
Adding a Note to a Host.....	29
Deleting a Note from a Host .....	30
Deleting All Notes from All Hosts.....	30

## Bruteforce Attacks

Overview of Bruteforce Attacks .....	31
Setting Up a Bruteforce Attack .....	31
Running a Bruteforce Attack.....	31
Targeting Services for a Bruteforce Attack .....	32
Setting the Payload Type for a Bruteforce Attack.....	33
Defining the Host Blacklist for a Bruteforce Attack .....	33
Performing a Dry Run of a Bruteforce Attack .....	33
Excluding Known Credentials from a Bruteforce Attack .....	33
Quitting the Bruteforce Attack after a Successful Login .....	34
Running a Bruteforce Attack without Opening Sessions .....	34

## Exploitation

Exploitation Overview .....	35
Manual Exploits .....	35
Module Search.....	36
Showing All Exploit Modules.....	36
Loading a Module .....	37
Showing Options for a Module.....	37
Showing Advanced Options for a Module.....	37
Showing the Targets for a Module.....	37
Setting the Target for an Exploit .....	38
Running an Exploit Module.....	38
Automated Exploits.....	38
Running an Automated Exploit .....	39
Defining a Host Blacklist for an Automated Exploit.....	39
Defining a Port Blacklist for an Automated Exploit .....	40
Running a Dry Run of an Automated Exploit.....	40
Setting the Application Evasion Level for an Automated Exploit .....	40

Application Evasion Level Options for DCERPC .....	41
Setting the TCP Evasion Level in an Automated Exploit.....	42
Setting the Payload Connection Type for an Automated Exploit.....	43
Setting the Minimum Rank for an Automated Exploit .....	43

## Reports

Reports Overview .....	45
Reports Types .....	45
Working with Reports .....	46
Supported Tasks.....	46
Generating a Report .....	46
Specifying the Report Type .....	47
Specifying the Report Name.....	47
Viewing Reports.....	47

## Metasploit Console Command Reference

Command Overview .....	48
Metasploit Pro Commands .....	48
Core Commands.....	49
Database Backend Commands.....	51
Accessing the Metasploit Pro Console .....	52
Basic Task Commands.....	52
Pro_bruteforce .....	52
Pro_collect.....	53
Pro_discover.....	54
Pro_exploit.....	55
Pro_project .....	56
Pro_report.....	57
Pro_tasks.....	57
Pro_user .....	58
Version.....	59
Database Back End Commands .....	59
Creds .....	60
Db_autopwn.....	60
Db_add_cred .....	61
Db_add_host .....	62
Db_add_note .....	63
Db_add_port.....	63
Db_connect.....	64
Db_disconnect.....	65
Db_driver .....	65
Db_export .....	66

Db_import .....	67
Db_nmap .....	67
Db_status.....	68
Hosts.....	68
Loot.....	69
Notes .....	69
Services .....	70
Vulns.....	70
Workspace.....	71
Core Commands .....	71
Back.....	72
Banner .....	72
Cd .....	72
Color .....	72
Connect .....	72
Exit.....	73
Help .....	73
Info.....	73
Irb.....	73
Jobs .....	73
Kill .....	74
Load.....	74
Loadpath.....	75
Quit .....	75
Reload_all.....	75
Route .....	75
Save.....	76
Search .....	76
Sessions .....	77
Set .....	78
Setg .....	78
Show.....	79
Sleep.....	80
Spool.....	80
Threads.....	81
Unload .....	82
Unset .....	82
Unsetg .....	82
Use .....	82
Version.....	83

# ABOUT THIS GUIDE

This chapter covers the following topics:

- [Target Audience 1](#)
- [Organization 1](#)
- [Document Conventions 2](#)
- [Support 2](#)

## Target Audience

This guide is for IT and security professionals who use Metasploit Pro as a penetration testing solution.

## Organization

This guide includes the following chapters:

- About this Guide
- Metasploit Console Overview
- Administration
- Projects
- Discovery
- Bruteforce
- Exploits
- Reports
- Metasploit Console Command Reference
- Index

# Document Conventions

The following table describes the conventions and formats that this guide uses:

Convention	Description
<b>Command</b>	Indicates buttons and fields in the user interface that you can use to input data. For example, “ <b>Click Projects &gt; New Project.</b> ”
Text	Indicates information that you need to type, code, or file directories. For example, “Enter the following: <code>chmod +x Desktop/metasploit-4.3.0-linux-x64-installer.</code> ”
<i>Title</i>	Indicates the title of a document or chapter name. For example, “For more information, see the <i>Metasploit Pro Installation Guide.</i> ”

## Support

Rapid7 and the community strive to provide you with a variety of support options. For a list of support options that are available, view the support section for the Metasploit product that you own.

### Support for Metasploit Pro and Metasploit Express

You can visit the Customer Center or e-mail the Rapid7 support team to obtain support for Metasploit Pro and Metasploit Express. To log in to the Customer Center, use the e-mail and password provided by Rapid7.

The following table describes the methods you can use to contact the Rapid7 support team.

Support Method	Contact Information
Customer Portal	<a href="https://www.rapid7.com/for-customers/">https://www.rapid7.com/for-customers/</a>

### Support for the Metasploit Framework and Metasploit Community

An official support team is not available for the Metasploit Framework or for Metasploit Community. However, there are multiple support channels available for you to use, such as the IRC channel and mailing list.



You can visit the [Metasploit Community](#) to submit your question to the community or you can visit the [help page](#) to view the support options that are available.

## Joining the IRC Channel

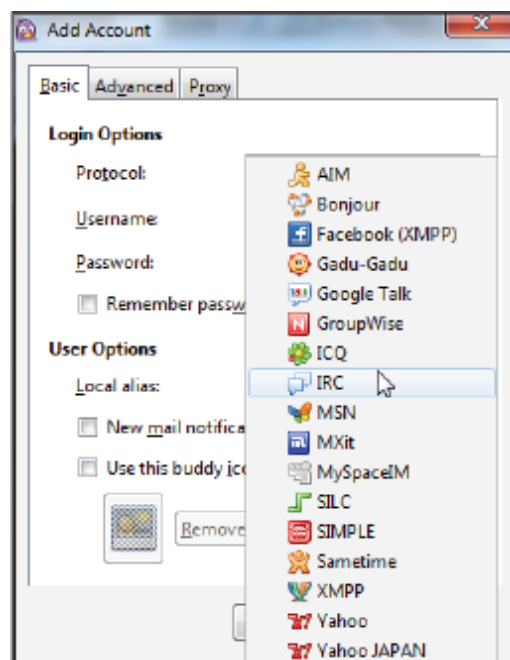
IRC, or Internet Relay Chat, lets you communicate with other members of the Metasploit IRC channel in real time. There are several IRC clients that you can use to connect to the Metasploit IRC channel, such as [Pidgin](#), [Xchat](#), and [Chatzilla](#). Choose the client that works best for you.

After you install an IRC client, use the following channel and server information to connect to the Metasploit channel.

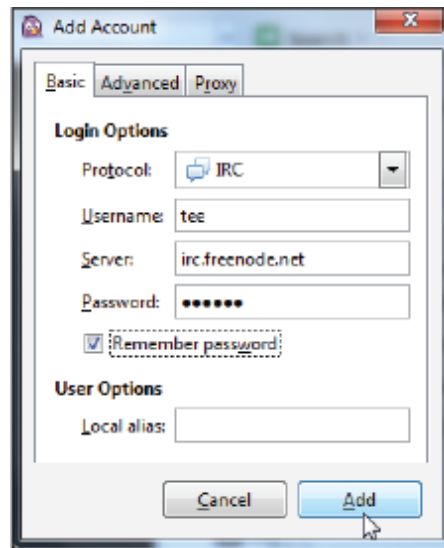
- Server: irc.freenode.net
- Channel: #metasploit

### *Setting Up the Metasploit IRC Channel on Pidgin*

- 1.) Download and install [Pidgin](#).
- 2.) Launch Pidgin.
- 3.) Select **Accounts > Manage Accounts**.
- 4.) Click **Add**.
- 5.) Choose **IRC** from the **Protocol** dropdown menu.



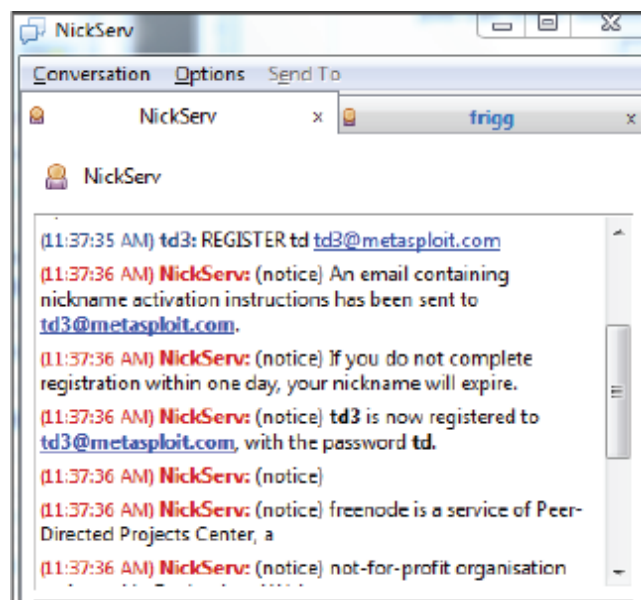
6.) Enter a user name and password for the IRC account.



7.) Verify that the Server field shows `irc.freenode.net`.

8.) Click **Add** to save the changes.

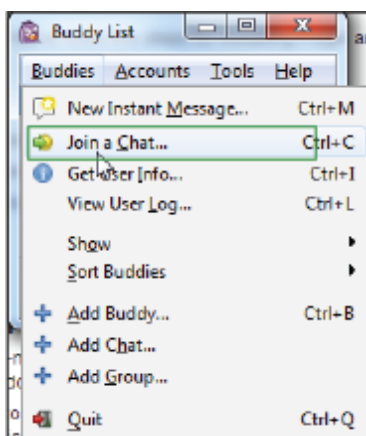
9.) A NickServ window appears and alerts you that your nickname is not , type `REGISTER` `<your IRC account password><your e-mail address>` and press Enter. For example, you can enter something like `REGISTER username username@mail.com`.



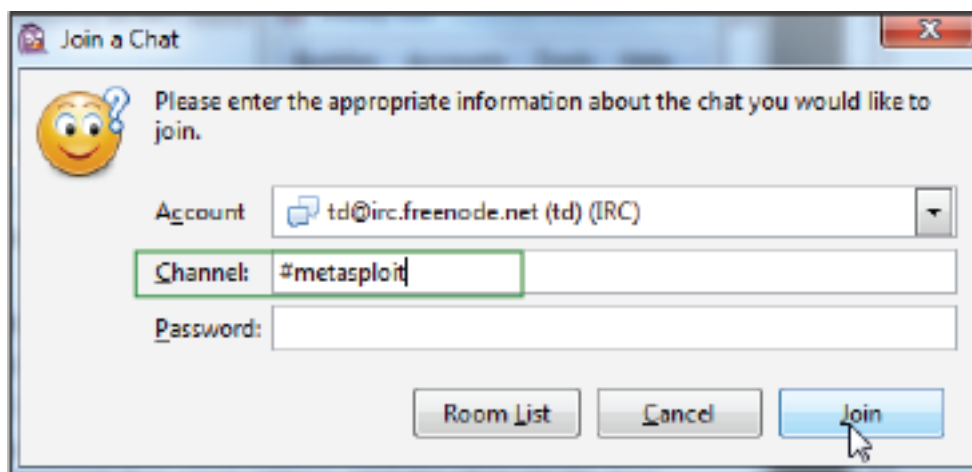
10.) After you press enter, NickServ alerts you that an activation e-mail has been sent to your e-mail address. Check your e-mail and follow the activation instructions.

11.) After you activate your IRC account, go back to the Pidgin Buddy List.

12.) Select **Buddies > Join a Chat**. The Join a Chat window appears.



13.) Enter `#metasploit` in the Channel field. The channel does not require a password.



14.) Join the room.

## Joining the Metasploit Mailing List

The mailing list provides access to active discussions between Metasploit users and developers. Subscribe to the mailing list to view the latest questions and ideas from the Metasploit community.

To join the mailing list, you can send a blank e-mail to [framework-subscribe@mail.metasploit.com](mailto:framework-subscribe@mail.metasploit.com) or you can fill out the [Metasploit mailing list form](#).

# CHAPTER 2

# METASPLOIT CONSOLE

# OVERVIEW

This chapter covers the following topics:

- [About the Metasploit Console 6](#)
- [Launching the Metasploit Console 7](#)
- [Basic Terms 9](#)
- [General Metasploit Workflow 11](#)

## About the Metasploit Console

The Metasploit Console provides the functionality of Metasploit Pro through a command line interface and serves as an alternative to the Metasploit Web UI. If you have traditionally been a Metasploit Framework user, the Metasploit Console provides you with a console that is similar to msfconsole.

Use the Metasploit Console to perform the following tasks:

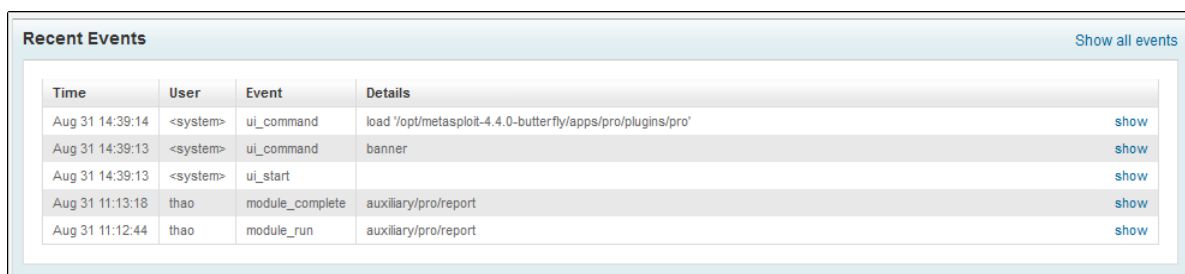
- Create and manage projects.
- Scan and enumerate hosts.
- Import and export data.
- Configure and run modules.
- Run automated exploits.
- View information about hosts.
- Generate an audit report in PDF, Word, RTF, or HTML.
- Collect evidence from exploited systems.

**Note:** You cannot perform all Metasploit Pro tasks through the Metasploit Console. Tasks that are not supported include social engineering and web application scanning.

## Console Tasks

In Metasploit Pro, a task is an action that the system can perform, such as a scan, brute-force attack, exploit, report generation, and data collection. The progress of any task that you perform through the Metasploit Console is viewable from the Recent Events area in the Metasploit Web UI. The system tags console tasks as `ui_command` and the user as `system`.

The following image shows the Recent Events area of the Metasploit Web UI:



The screenshot shows the 'Recent Events' section of the Metasploit Web UI. It features a table with four columns: Time, User, Event, and Details. There are five rows of event data, each with a 'show' link in the Details column. A 'Show all events' link is located in the top right corner of the table area.

Time	User	Event	Details
Aug 31 14:39:14	<system>	ui_command	load '/opt/metasploit-4.4.0-butterfly/apps/pro/plugins/pro'
Aug 31 14:39:13	<system>	ui_command	banner
Aug 31 14:39:13	<system>	ui_start	
Aug 31 11:13:18	thao	module_complete	auxiliary/pro/report
Aug 31 11:12:44	thao	module_run	auxiliary/pro/report

## Supported Tasks

Not all tasks and features that are available through the Metasploit Web UI are available to the Metasploit Console. Some features do not map directly to a command in the Metasploit Console. Social engineering and web application scanning cannot be performed through the Metasploit Console.

## Launching the Metasploit Console

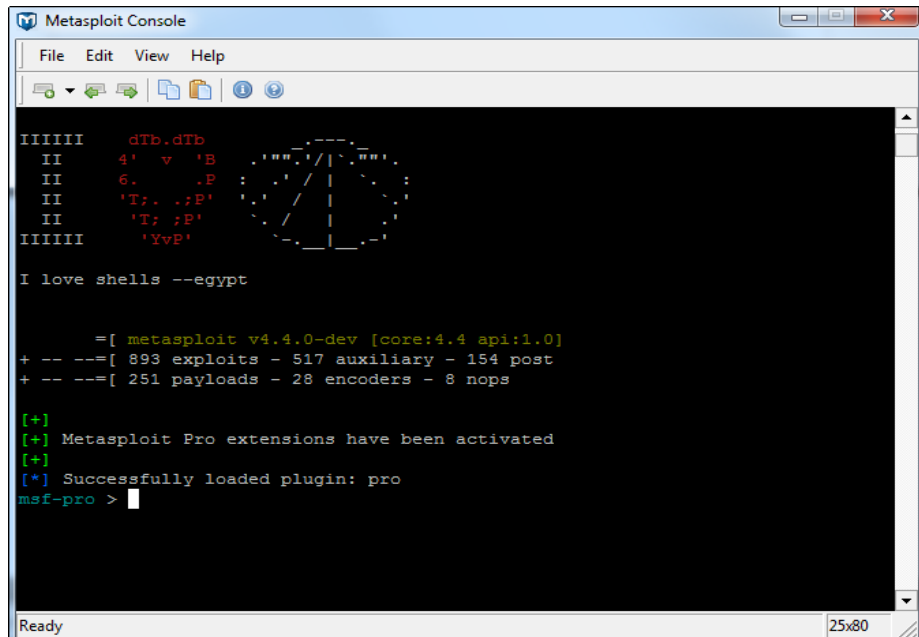
The Metasploit Console is available after you download and install the latest version of [Metasploit](#).

To launch the Metasploit Console, follow the instructions for your operating system.

## Launching the Metasploit Console on Windows

To launch the Metasploit Console on a Windows system, go to **Start > All Programs > Metasploit > Metasploit Console**.

When the Metasploit Console loads, the command line drops to an `msf-pro >` prompt. The following image shows the Metasploit Console after it loads:



```
Metasploit Console
File Edit View Help

IIIIII dTb.dTb
II 4' v 'B
II 6. .P
II 'T; .;P'
II 'T; ;P'
IIIIII 'YvP'

I love shells --egypt

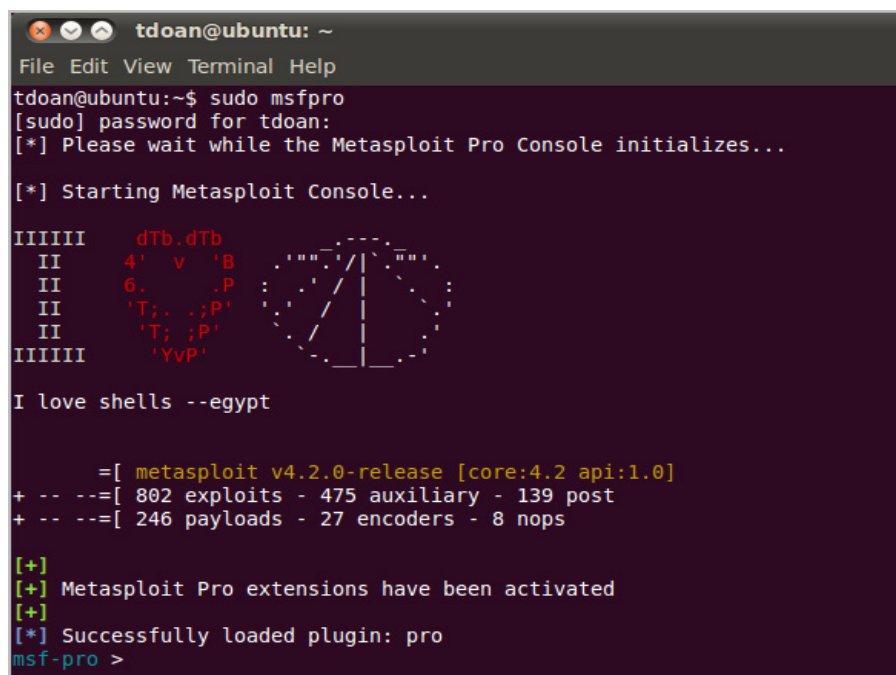
=[ metasploit v4.4.0-dev [core:4.4 api:1.0]
+ -- --[ 893 exploits - 517 auxiliary - 154 post
+ -- --[ 251 payloads - 28 encoders - 8 nops

[+]
[+] Metasploit Pro extensions have been activated
[+]
[*] Successfully loaded plugin: pro
msf-pro >
```

## Launching the Metasploit Console on Linux

To launch the Metasploit Console on a Linux system, open the command line terminal and type `msfpro` when the command prompt appears.

**Note:** Depending on your account privileges, you may need to use `sudo` to launch the Metasploit Console.



```
tdoan@ubuntu: ~
File Edit View Terminal Help
tdoan@ubuntu:~$ sudo msfpro
[sudo] password for tdoan:
[*] Please wait while the Metasploit Pro Console initializes...

[*] Starting Metasploit Console...

IIIIII dTb.dTb
II 4' v 'B
II 6. .P
II 'T; .;P'
II 'T; ;P'
IIIIII 'YvP'

I love shells --egypt

=[ metasploit v4.2.0-release [core:4.2 api:1.0]
+ -- --[ 802 exploits - 475 auxiliary - 139 post
+ -- --[ 246 payloads - 27 encoders - 8 nops

[+]
[+] Metasploit Pro extensions have been activated
[+]
[*] Successfully loaded plugin: pro
msf-pro >
```

# Basic Terms

The following sections describe the basic terms that you need to know to understand penetration testing, security research, and Metasploit.

## Vulnerability

A vulnerability is a security flaw or weakness in an application or system that enables an attacker to compromise the target system. A compromised system can result in privilege escalation, denial-of-service, unauthorized data access, stolen passwords, and buffer overflows.

## Exploit

An exploit is a program that takes advantage of a specific vulnerability and provides an attacker with access to the target system. An exploit typically carries a payload and delivers the payload to the target system.

## Payload

A payload is the actual code that executes on the target system after an exploit successfully executes. There are two types of payloads: reverse shell and bind shell. The major difference between a reverse shell and a bind shell is how the shell enables you to connect to the exploited system.

## Reverse Shell

A reverse shell creates a connection from the target machine back to you as a command prompt.

## Bind Shell

A bind shell attaches a command prompt to a listening port on the exploited system. You can connect to the bind shell to access the exploited system.

## Modules

Modules are the core components of Metasploit. A module is a piece of software that can perform a specific action, such as exploitation, fuzzing, and scanning. Each task that you can perform with the Metasploit Framework is defined within a module.

You can locate modules that are available in the following directory: `<installation directory>/metasploit/msf3/modules`. The modules are categorized by type and then by protocol. For example, you can find FTP fuzzers in the following location: `<installation directory>/metasploit/msf3/modules/auxiliary/fuzzers/ftp`.

The easiest way to find a module is to use the search command to find a module. You can also go to the [Metasploit Exploit web page](#) to search for modules.

There are a few types of modules. The module type depends on the purpose of the module and the type of action that the module performs.

The following modules are available in the Metasploit:

- Exploit
- Auxiliary
- Post-Exploitation
- Payload
- NOP generator
- Payload encoder

## Exploit Modules

An exploit module executes a sequence of commands to target a specific vulnerability found in a system or application. An exploit module takes advantage of a vulnerability to provide the attacker with access to the target system. Exploit modules include buffer overflow, code injection, and web application exploits.

## Auxiliary Modules

An auxiliary module does not execute a payload and perform arbitrary actions that may not be related to exploitation. Examples of auxiliary modules include scanners, fuzzers, and denial of service attacks.

## Post-Exploitation Modules

A post-exploitation module enables you to gather more information or to gain further access to an exploited target system. Examples of post-exploitation modules include hash dumps and application and service enumerators.



## Payloads

A payload is the shell code that runs after an exploit successfully compromises a system. The payload enables you to define how you want to connect to the shell and what you want to do to the target system after you take control of it.

A payload can open a Meterpreter or command shell. Meterpreter is an advanced payload that allows you to write DLL files to dynamically create new features as you need them.

For more information on Meterpreter, see the [Meterpreter User Guide](#).

## NOP Generators

A NOP generator produces a series of random bytes that you can use to bypass standard IDS and IPS NOP sled signatures. Use NOP generators to pad buffers.

## Payload Encoders

A payload encoder enables you to evade IDS and IPS signatures that are looking for specific bytes of a payload.

## General Metasploit Workflow

The following sections provide a brief description of the general work flow for the Metasploit Console after you log in to the Metasploit Console and connect to the database.

### Choose a Module

One of the very first things you must do is identify the purpose of your task. The purpose of your task determines the type of module you need to successfully test or analyze the target system.

For example, if you are still learning about the target systems, you may want to run a scanner to find any open and active ports. If you have already identified the vulnerabilities on the target system, you may want to choose an exploit module. If you want to create a buffer overflow, you might take a look at the NOP generators.

The easiest way to choose a module is to view a list of everything that is available in Metasploit. In the Metasploit Console, type `show all` to view a list of exploits, auxiliary modules, post-exploitation modules, NOP generators, and payloads that are available.

If you know the type of module you want to use, you can narrow down the search by appending any of the following options to the `show` command:

- Exploit
- Auxiliary
- Post
- Nops
- Payloads

For example, if you type `show nops`, the Metasploit Console lists the NOP generators that are available.

After you select the module that you want to run, you can use the `use` command to load the module into the current context. The Metasploit Console shows the current context in the parenthesis next to the `msf >` prompt.

For example, if you want to run a bruteforce attack with the `smb_login` auxiliary module, you enter the following:

```
msf-pro > show auxiliary

===
auxiliary/scanner/smb/smb_login
===

msf-pro > use auxiliary/scanner/smb/smb_login
msf (smb_login) >
```

## Configure the Module

Now that you have chosen a module, you can type `show options` to see the options that you can configure for the module.

```
msf (smb_login) > show options

Module options (auxiliary/scanner/smb/smb_login)

Name Current Setting Required Description
=====
BLANK_PASSWORDS true no Try blank passwords for all users
BRUTEFORCE_SPEED 5 yes How fast to bruteforce
```

If an option is required, you must specify a value for it. Otherwise, you can leave it empty. Generally, the most common options that you may want to set are `RHOSTS`, `RPORT`, `LHOST`, `LPORT`, `THREADS`, `TIMEOUT`, `WORKSPACE`, and `GWHOST`.

Module options vary between modules, so you can use the built-in option descriptions to generally determine what each option does.

## Set the Options

To set the options, you can use the `set` and `unset` commands. The `set` and `unset` commands work specifically within the current module context. After you change out of the current module context, the values stored by `set` and `unset` are lost.

The following example sets the bruteforce speed for the bruteforce attack:

```
msf (smb_login) > show options

Module options (auxiliary/scanner/smb/smb_login)

Name Current Setting Required Description
=====
BLANK_PASSWORDS true no Try blank passwords for all users
BRUTEFORCE_SPEED 5 yes How fast to bruteforce

msf (smb_login) > set bruteforce_speed 3
bruteforce_speed => 3
```

## Select a Target

Before you can run an exploit module, you need to verify the vulnerable targets that are available for that particular exploit. Make sure that the target you are testing is listed as a potentially vulnerable target.

To check the list of potential vulnerable targets, use the `show targets` command.

```
msf > use windows/wins/ms04_045_wins
msf (ms04_045_wins) > show targets

Exploit targets:

Id Targets
== =====
0 Windows 2000 English

msf (ms04_045_wins) > set target 0
TARGET => 0
```

## Select a Payload

If you want to run an exploit module, you need to specify the payload for the attack. The first thing you need to do is display a list of payloads that are available for the exploit. After you

identify the payload that you want to send, you need to use the `set` command to load the payload into the exploit.

The following example shows how to set the payload:

```
msf exploit (ms04_045_wins) > show payloads
msf exploit (ms04_045_wins) > set payload windows/shell_bind_tcp
```

## Run the Module

Finally, now that you have selected and configured a module, you are ready to run the module.

```
msf exploit (ms04_045_wins) > exploit
[*] Started....
```

# CHAPTER 3

## ADMINISTRATION

This chapter covers the following topics:

- [Metasploit Console Management](#) 15
- [User Account Management](#) 16
- [Database Management](#) 16
- [Task Management](#) 18

### Metasploit Console Management

You can use the global settings to manage input and output from the Metasploit Console.

#### Logging Input and Output from the Console

Use the `ConsoleLogging` option to store information that the Metasploit Console inputs and outputs into a log.

```
msf-pro > setg ConsoleLogging y
Console logging is now enabled.
```

#### Changing the Log Verbosity

Use the `LogLevel` option to set the verbosity of the logs. Set the value between 1 and 5.

```
msf-pro > setg LogLevel 3
LogLevel => 3
```

## Logging Input and Output for a Session

Use the `SessionLogging` option to store information that msfconsole inputs and outputs about a session into a log.

```
msf-pro > setg SessionLogging y
Session logging will be enabled for future sessions.
```

## User Account Management

User account management includes the ability to view a list of Metasploit Pro users and switch between user accounts.

### Changing the Current User

Use the `pro_user` command and supply the user name as the argument to change the current user.

```
msf-pro > pro_user joe
{*} Changed pro_user to joe
```

### Viewing a List of Users

Use the `pro_user` command and the `-l` option to view a list of users.

```
msf-pro > pro_user -l

Username Full Name Email Admin?
=====
joe
john
```

## Database Management

Metasploit provides back end database support for PostgreSQL. The database stores information, such as host data, evidence, and exploit results. Any data that you use within Metasploit is stored within the database.

The first time you launch the Metasploit Console, the system automatically sets up the database for you. The console loads the database each subsequent time you launch it.

Commands that manage the database start with a “db\_” prefix.

## Connecting to the Database

To connect to the database instance, you need the user name, password, host name for the database, host port number, and the database name.

**Note:** The Metasploit Console retains the connection with the database unless you disconnect from it.

You can find the database settings that the Metasploit uses in `<msf dir>/config/database.yml`.

```
msf-pro > db_connect username:password@host:port/db
```

If you are a Linux user, you can use the following command:

```
msf-pro > db_connect -y /opt/metasploit/config/database.yml
```

## Verifying the Database Connection

After you run the `db_connect` command, you should verify that you have successfully connected to the database instance. If you have successfully connected to the database, Metasploit Console returns a list of available hosts.

```
msf-pro > hosts

Hosts
=====

addresss mac name os_name os_flavor os_sp purpose info comments
===== == == == == == == == == ==
192.168.0.1
```

## Disconnecting from the Database

Use the `db_disconnect` command to disconnect from the database.

```
msf-pro > db_disconnect
```

## Viewing the Current Database Status

Use the `db_status` command to view the current database status.

```
msf-pro > db_status
```

## Task Management

When you run a task, such as a scan, bruteforce attack, exploit, or report, the Metasploit Console tracks the progress of the task in the task log. To manage the task log, you can view or cancel any task.

### Viewing All Tasks

Use the `pro_tasks` command to view a list of all tasks that you have run since you connected to the database.

```
msf-pro > pro_tasks

Id Project Desc Status Information
-- -
0 default scan running
```

### Viewing Running Tasks

Use the `pro_tasks` command and the `-r` option to view a list of all running tasks.

```
msf-pro > pro_tasks -r

Id Project Desc Status Information
-- -
0 default scan running
```



## Viewing the Task Log

Use the `pro_tasks` command and the `-w` option to view the task log for a specific task. You must define the task ID for the task log that you want to view. To find the task ID, type `pro_tasks` to view the tasks list. The tasks list shows the task ID.

```
msf-pro > pro_tasks

Id Project Desc Status Information
-- -
0 default scan running

msf-pro > pro_tasks -w 0
```

## Canceling a Running Task

Use the `pro_tasks` command and the `-k` option to cancel a task. You must specify the task ID for the task that you want to cancel. To find the task ID, type `pro_tasks` to view the tasks list. The tasks list shows the task IDs.

```
msf-pro > pro_tasks

Id Project Desc Status Information
-- -
0 default scan running

msf-pro > pro_tasks -k 0
```

# CHAPTER 4

# PROJECTS

This chapter covers the following topics:

- [About Projects 20](#)
- [Working with Projects 20](#)

## About Projects

Projects are containers that you can use to segment and organize data that a database stores. Use projects to create a logical separation for each segment that you want to test. For example, you may want to create a project for each subnet, or department, within an organization to limit the hosts to a specific network. Departments like HR, IT, and Accounting may each need a separate project.

Metasploit stores data in the current project.

**Note:** To create or work within projects, you must be connected to a database instance.

## Working with Projects

Projects organize segment a network into manageable spaces. The following sections explain how to create, view, change, and delete a projects.

## Creating a Project

Use the `pro_project` command and the `-a` option to create a project. The project that you create becomes the current project.

```
msf-pro > pro_project -a HR
msf-pro > pro_project -a IT
msf-pro > pro_project -a ACC

default
HR
IT
*ACC
```

## Viewing the Current Project

Use the `pro_project` command to view the current project. An asterisk denotes the current project.

```
msf-pro > pro_project

*default
HR
IT
ACC
```

## Changing the Project

Use the `pro_project` command to change the current project. You can tab complete the project name. Project names are case sensitive.

**Note:** If you need to specify a project that contains spaces, you must enclose the project name in quotes. For example, use `pro_project "IT Dept"`.

```
msf-pro > pro_project

*default
HR
IT
ACC

msf-pro > pro_project HR
<*) Workspace: HR
```

## Deleting a Project

Use the `pro_project` command and the `-d` option to delete a project. This deletes the project, which includes the hosts, credentials, evidence, and any other data related to the project.

```
msf-pro > pro_project

*default
HR
IT
ACC

msf-pro > pro_project -d ACC
<*) Workspace: ACC
```

# CHAPTER 5

# HOST DISCOVERY

This chapter covers the following topics:

- [About Host Discovery](#) 23
- [Managing Hosts](#) 23
- [Managing Vulnerabilities](#) 28
- [Managing Notes](#) 29

## About Host Discovery

Host discovery is the process of that Metasploit performs to identify the ports, services, and operating systems that are in use by hosts on a particular network. You run a scan to find the hosts that are accessible on a network and to help you identify vulnerabilities based on the open ports and services that the scan finds.

## Managing Hosts

You can manage hosts by adding them manually, importing scan data, and running a scan.

## Scanning for Hosts

You can launch a discovery scan to enumerate services and ports on target hosts. A discovery scan performs host discovery, port scanning, and OS fingerprinting.

A discovery scan starts with an Nmap scan to detect available systems and scan ports. Next, the discovery scan sweeps the target network with UDP probes to identify additional systems. After the discovery scan identifies available ports, the discovery scan sweeps the ports with service specific modules to identify active services.

Use the `pro_discover` command to perform a discovery scan.

```
msf-pro > pro_discover 192.168.0.1  
{*] Started task 1
```

## Specifying SMB Credentials

If you have SMB credentials that you want to specify for Windows hosts running Samba or for shared access points, you can use the `pro_discover` command and the `-sd`, `-sp`, and `-su` options.

The `-sd` option defines the SMB domain, the `-su` option specifies the user name, and the `-sp` option specifies the password.

```
msf-pro > pro_discover 192.168.0.1 -sd workgroup -su root -sp root
```

## Viewing a List of Hosts

Use the `hosts` command to view a list of hosts that the database contains. To view a list of hosts, you must have an active connection to the database.

```
msf-pro > hosts  
  
Hosts  
=====  
  
addressss mac name os_name os_flavor os_sp purpose info comments  
===== == == ===== ===== =====  
192.168.0.1  
192.168.0.2
```

## Adding a Host

Use the `hosts` command and the `-a` option to add a host to the current workspace.

```
msf-pro > hosts -a 192.168.0.3  
<[*] Time: 2012-02-01 05:05:05 UTC Host: host=192.168.0.3
```

## Deleting a Host

Use the `hosts` command and the `-d` option to delete a host from the current workspace:

```
msf-pro > hosts -d 192.168.0.3
[*] Deleted 1 hosts
```

## Connecting to a Host

Use the `connect` command to communicate with a host. You must supply the host address and port that you want to connect to.

```
msf-pro > connect 192.168.0.1 22
[*] Connected to 192.168.0.1:22
```

## Outputting Host Data to CSV File

Use the `hosts` command and the `-o` option to output all the information about the hosts in the database to a CSV file. The data includes the IP address, MAC address, host name, operating system, OS flavor, purpose, and comments.

The following example outputs all the hosts in the database to a file called `HRHosts`.

```
msf-pro > hosts -o HRHosts
```

## Importing Scan Data

Use the `db_import` command to import host or scan data into the database. The data must be stored in an XML file. By default, the Metasploit Framework imports files from the `msf3/data` directory.

```
msf-pro > db_import subnetA.xml
[*] Importing 'Metasploit XML' data
[*] Importing host 192.168.0.3
[*] Successfully imported C:/metasploit/msf3/subnetA.xml
```

## Supported Scan Data Formats

You can import report files and scan data from most vulnerability and scanning tools that are available. Metasploit parses the data and imports the hosts and any related host data into the database.

Metasploit supports the following format types:

- Metasploit PWDump Export
- PWDump
- Metasploit XML (all versions)
- Metasploit ZIP (all versions)
- NeXpose Simple XML or XML
- NeXpose Raw XML or XML Export
- Foundstone Network Inventory XML
- Microsoft MBSA SecScan XML
- nCircle IP360 (XMLv3 and ASPL)
- NetSparker XML
- Nessus NBE
- Nessus XML (v1 and v2)
- Qualys Asset XML
- Qualys Scan XML
- Burp Session XML
- Acunetix XML
- AppScan XML
- Nmap XML
- Retina XML
- Amap Log
- IP Address List
- Libcap
- Spiceworks Inventory Summary CSV
- Core Impact XML



## Viewing Hosts that Are Up

Use the `hosts` command and the `-u` option to view a list of hosts that are up.

```
msf-pro > hosts -u

Hosts
=====

addressss mac name os_name os_flavor os_sp purpose info comments
===== ==  ==  ===== =====  =====  =====  =====
192.168.0.1
```

## Viewing Specific Columns from the Hosts Table

Use the `hosts` command and the `-c` option to view specific columns from the database.

```
msf-pro > hosts -c address
msf > hosts -u

Hosts
=====

address
=====
192.168.0.1
192.168.0.2
192.168.0.3
```

## Columns in the Hosts Table

Metasploit stores host data from in the hosts table. You can use the column name to search the database for hosts. For example, if you want to see the names of all the hosts stored in the database, you can type `hosts -c name`, and the console displays a list of all host names in the workspace.

The following table describes the columns that are available for the hosts table:

Column	Description
address	The target host IP address.
comments	Comments about the host.
created_at	The date the host was added to the database.
mac	The host MAC address.

Column	Description
name	The host name.
os_flavor	The host operating system.
os_lang	The language of the operating system.
os_name	The operating system.
os_sp	The operating system version.
purpose	The purpose of the host. The purpose can be client, server, or device.
state	The state of the host. The state can be looted, scanned, cracked, or shelled.
updated_at	The number of days or hours since the host information was updated.

## Viewing Loot

Loot is the collected data that Metasploit stores in the database. You can use the `loot` command to store and retrieve the data that you have collected from target hosts.

```
msf-pro > loot
```

## Outputting Host Data

Use the `hosts` command and the `-o` option to generate a CSV text file that contains the data from the host table.

```
msf-pro > hosts -o subnet1data
[*] Wrote hosts to subnet1data
```

Use the `services` command and the `-s` option to search for hosts running a specific service.

```
msf-pro > hosts -o subnet1data
[*] Wrote hosts to subnet1data
```

## Managing Vulnerabilities

The following sections describe how to view and search for vulnerabilities.

## Listing Vulnerabilities by Port

Use the `vulns` command and the `-p` option to search for vulnerabilities that match a particular port or port range. You can specify a single port, series of ports, or a range of ports.

```
msf-pro > vulns -p 692
[*] Time: 2012-09-25 13:57:10 UTC Vuln: host=10.6.200.108 name=cve-1999-0003 refs=CVE-1999-003
```

## Listing Vulnerabilities by Service

Use the `vulns` command and the `-s` option to search for vulnerabilities that match a particular port or port range. You can specify a single port, series of ports, or a range of ports.

```
msf-pro > vulns -s tooltalk
[*] Time: 2012-09-25 13:57:10 UTC Vuln: host=10.6.200.108 name=cve-1999-0003 refs=CVE-1999-003
```

## Searching for a Vulnerability

Use the `vulns` command and the `-S` option to search for vulnerabilities that match a keyword. The keyword can be a reference ID, vulnerability name, operating system, service, or host name.

```
msf-pro > vulns -S tooltalk
[*] Time: 2012-09-25 13:57:10 UTC Vuln: host=10.6.200.108 name=cve-1999-0003 refs=CVE-1999-003
```

## Managing Notes

The following sections describe how to add and delete notes from a project.

### Adding a Note to a Host

Use the `notes` command and the `-a`, `-t`, and `-n` options to add a note to a host. The `-a` option indicates that you want to add a note. The `-t` option denotes the note type, such as

host.os.<service>\_fingerprint, host.<service>.traceroute, or app. The `-n` option specifies the note that you want to add to the host.

```
msf-pro > notes -a -t apps -n zip file 192.168.184.155
[*] Time: 2012-09-25 13:57:10 UTC Note: host=192.168.184.155 type= apps
data=zip file
```

## Deleting a Note from a Host

Use the `notes` command and the `-d` option to delete a note from a host.

```
msf-pro > notes -d apps 192.168.184.155
[*] Time: 2012-06-27 22:08:22 UTC Note: host=192.168.184.155
type=host.last_boot data={:time=>"Fri Jun 01 23:21:31 2012"}
[*] Deleted 1 note
```

## Deleting All Notes from All Hosts

Use the `notes` command and the `-d` option to delete all notes from the project. Do not perform this step unless you want to clear every note in the project.

```
msf-pro > notes -d
[*] Time: 2012-06-27 22:08:22 UTC Note: host=74.125.227.67
type=host.nmap.traceroute data={"port"=>6050, "proto"=>"tcp",
" hops"=>[{"ttl"=>"1", "ipaddr"=>"74.125.227.67", "rtt"=>"7.00",
"name"=>"dfw06s07-in-f3.1e100.net"}]}
[*] Time: 2012-06-27 22:08:22 UTC Note: host=74.125.227.64
type=host.os.nmap_fingerprint data={:os_vendor=>"FreeBSD",
:os_family=>"FreeBSD", :os_version=>"6.X", :os_accuracy=>86}
[*] Time: 2012-06-27 22:08:22 UTC Note: host=74.125.227.64
type=host.last_boot data={:time=>"Fri Jun 01 23:21:31 2012"}
[*] Deleted 3 notes
```

# CHAPTER 6

# BRUTEFORCE ATTACKS

This chapter covers the following topics:

- [Overview of Bruteforce Attacks](#) 31
- [Setting Up a Bruteforce Attack](#) 31

## Overview of Bruteforce Attacks

A bruteforce attack tries a large number of common user name and password combinations in order to open a session on the target machine. After the bruteforce attack successfully guesses a credential, the system stores the user name and password in the workspace.

In Metasploit Pro, a bruteforce attack launches service specific modules to attempt to crack the credentials for the service. You can choose the services and ports that you want to target, and the bruteforce attack chooses modules that target those services. If the bruteforce attack successfully cracks a credential and opens a session, you can use the session to gain further access and information for the system.

## Setting Up a Bruteforce Attack

To run a bruteforce attack, you must define the services that you want to target on a particular host or network range. In addition to the services, you can configure the bruteforce attack to exclude specific hosts and credentials, perform a dry run, and use a particular payload type.

## Running a Bruteforce Attack

Use the `pro_bruteforce` command to run a bruteforce attack against a single host or a range of hosts. If you do not specify a host, Metasploit uses the workspace's default network range.

Additionally, you can specify additional options to do things like specify services, exclude hosts, and set payloads.

```
msf-pro > pro_bruteforce 192.168.184.137
<*> Started task 1
```

## Targeting Services for a Bruteforce Attack

Use the `pro_bruteforce` command and the `-s` option to run a bruteforce attack against specific services. You must define a comma separated list of services.

```
msf-pro > pro_bruteforce 192.168.184.137 -s smb, ssh
<*> Started task 1
```

## Targeted Services

The following list describes the services that you can target:

- SMB
- Postgres
- DB2
- MySQL
- MSSQL
- Oracle
- HTTP
- HTTPS
- SSH
- SSH\_PUBKEY
- Telnet
- FTP
- POP3
- EXEC
- LOGIN
- SHELL
- VMAuthd
- VNC
- SNMP

## Setting the Payload Type for a Bruteforce Attack

Use the `pro_bruteforce` command to run a bruteforce attack and the `-m` option to specify the payload type. The payload types are `auto`, `bind`, and `reverse`. By default, Metasploit uses `auto`.

```
msf-pro > pro_bruteforce 192.168.184.137 -m bind
[*] Started task 1
```

## Defining the Host Blacklist for a Bruteforce Attack

Use the `pro_bruteforce` command to run a bruteforce attack and the `-b` option to specify a host blacklist. The host blacklist specifies the hosts that you want to exclude from the attack.

```
msf-pro > pro_bruteforce 192.168.184.0/24 -b 192.168.184.139
[*] Started task 1
```

## Performing a Dry Run of a Bruteforce Attack

Use the `pro_bruteforce` command and the `-d` option to perform a dry run of a bruteforce attack. You can define additional options that you want the bruteforce attack to use, such as the payload type, SMB domains, and the hosts and credentials you want to exclude.

```
msf-pro > pro_bruteforce 192.168.184.0/24 -m bind -d
[*] Started task 1
```

## Excluding Known Credentials from a Bruteforce Attack

Use the `pro_bruteforce` command to run a bruteforce attack and the `-I` and `-K` options to specify the credentials that you want to skip. Use `-I` to exclude imported credentials and use `-K` to exclude known credentials. Known credentials are credentials that the system cracked in a previous bruteforce attack or ones that you manually added for a host.

```
msf-pro > pro_bruteforce 192.168.184.0/24 -m bind -I -K
[*] Started task 1
```

## Quitting the Bruteforce Attack after a Successful Login

Use the `pro_bruteforce` command to run a bruteforce attack and the `-q` option to quit the bruteforce attack after a successful login to a host.

```
msf-pro > pro_bruteforce 192.168.184.0/24 -m bind -q  
<*) Started task 1
```

## Running a Bruteforce Attack without Opening Sessions

Use the `pro_bruteforce` command to run a bruteforce attack and the `-G` option to not open any sessions on the target systems.

```
msf-pro > pro_bruteforce 192.168.184.0/24 -m bind -G  
<*) Started task 1
```



# CHAPTER 7

# EXPLOITATION

This chapter covers the following topics:

- [Exploitation Overview 35](#)
- [Manual Exploits 35](#)
- [Automated Exploits 38](#)

## Exploitation Overview

An exploit is a program that takes advantage of a specific vulnerability and provides an attacker with access to the target system. An exploit typically carries a payload and delivers the payload to the target system.

Metasploit uses modules to run exploits. These modules are known as exploit modules. An exploit module executes a specific set of instructions to take advantage of a weakness in a system. Use exploit modules to gain access and send a payload to a vulnerable system. The most common types of exploit modules are buffer overflow and SQL injection exploits.

In Metasploit Pro, you can either use manual exploits or automated exploits to compromise a system.

## Manual Exploits

To run a manual exploit, you must choose and configure an exploit module to run against a target host. You choose the exploit module based on the information you have about the host. For example, if you know that the host runs Windows Service Pack 1, you can run an exploit that targets Windows Service Pack 1 vulnerabilities. Or if you know that the target system has a specific vulnerability that you want to test, you can run the exploit that targets that particular weakness.

## Module Search

You can create keyword expressions to search for a specific module name, path, platform, author, CVE ID, BID, OSDVB ID, module type, or application. The search returns a list of results that match the query.

You use keyword tags to create keyword expressions. Keyword tags use the following format:  
`tag:keyword`.

## Module Search Keywords

The following are keyword tags that you can use:

- name
- path
- platform
- type
- app
- author
- cve
- bid
- osdvp

## Searching for a Module

Use the `search` command to search for a module.

```
msf-pro > search platform:Windows
msf-pro > search type:exploit
msf-pro > search author:hd
msf-pro > search app:client
```

## Showing All Exploit Modules

Use the `show` command to view a list of the exploits that are available.

```
msf-pro > show exploits
```

## Loading a Module

Use the `use` command to load an exploit module. After you issue the `use` command, the Metasploit Console changes the command prompt to show the loaded module.

```
msf-pro > use exploit/windows/wins/ms04_045_wins
msf-pro exploit (ms04_045_wins) >
```

## Showing Options for a Module

Use the `show` command to view a list of options that are available for a particular module

```
msf-pro > use exploit/windows/wins/ms04_045_wins
msf-pro exploit (ms04_045_wins) > show options
```

## Showing Advanced Options for a Module

Use the `show` command to view a list of advanced options that are available for an exploit module.

```
msf-pro > use exploit/windows/wins/ms04_045_wins
msf-pro exploit (ms04_045_wins) > show advanced
```

## Showing the Targets for a Module

Use the `show targets` command to view a list of potentially vulnerable targets. Most modules display a list of targets that may be vulnerable to the exploit. Each target has an ID and operating system.

If you know the victim machine's operating system and version, you can specify the target for the exploit. Some modules provide an automatic targeting option. If you are unsure of the

operating system or version, you can use the automatic targeting option, if it is available, to automatically detect the victim machine's OS and version. t

```
msf-pro > use exploit/windows/wins/ms04_045_wins
msf-pro exploit (ms04_045_wins) > show targets

Exploit targets:

Id Name
-- ----
0 Windows 2000 English
```

## Setting the Target for an Exploit

Use the `set target` command to specify a target for the exploit.

```
msf-pro > use exploit/windows/wins/ms04_045_wins
msf-pro exploit (ms04_045_wins) > show targets

Exploit targets:

Id Name
-- ----
0 Windows 2000 English

msf-pro exploit (ms04_045_wins) > set target 0
```

## Running an Exploit Module

Use the `exploit` or `run` command to run an exploit module.

```
msf-pro > use exploit/windows/wins/ms04_045_wins
msf-pro exploit (ms04_045_wins) > run
```

## Automated Exploits

If you need Metasploit Pro to choose the exploits that it runs against a target system, you should use automated exploits. For example, if you do not know the vulnerabilities that exist on a target system, you can run an automated exploit, which uses the host information to systematically choose exploits to run against the target host.

When you run an automated exploit, Metasploit Pro builds an attack plan based on the service, operating system, and vulnerability information that it has for the target system. Metasploit Pro obtains this information from the discovery scan or from the information that you provide for the target host. The attack plan defines the exploit modules that Metasploit Pro will use to attack the target systems.

To run an automated exploit, you must specify the hosts that you want to exploit and the minimum reliability setting that Metasploit Pro should use. The minimum reliability setting indicates the potential impact that the exploits have on the target system. If you use a high ranking, such as excellent or great, Metasploit Pro uses exploits that will be unlikely to crash the service or system. Exploits that typically have a high reliability ranking include SQL injection exploits, web application exploits, and command execution exploits. Exploits that corrupt memory will most likely not have a high reliability ranking.

You can also specify the payload type that you want the exploit to use. By default, automated exploits use Meterpreter, but you can choose to use a command shell instead.

## Running an Automated Exploit

Use the `pro_exploit` command to run an automated exploit. You can define the evasion level, minimum reliability rank, payload, and ports that the exploits use.

If you do not define any options for the automated exploit, Metasploit Pro uses the default settings.

```
msf-pro > pro_exploit 192.168.184.139

Id Project Desc Status Information
== =====
12 default exploiting

<*) Started task 1
```

## Defining a Host Blacklist for an Automated Exploit

Use the `pro_exploit` command to run an automated exploit and the `-b` option to specify a list of hosts that you want to exclude from the exploit.

```
msf-pro > pro_exploit 192.168.184.0/24 -b 192.168.184.138

Id Project Desc Status Information
== =====
12 default exploiting

<*) Started task 1
```

## Defining a Port Blacklist for an Automated Exploit

Use the `pro_exploit` command to run an automated exploit and the `-pb` option to specify a list of ports that you want to exclude from the exploit.

```
msf-pro > pro_exploit 192.168.184.0/24 -pb 22-23

Id Project Desc Status Information
== =====
12 default exploiting

<*) Started task 1
```

## Running a Dry Run of an Automated Exploit

Use the `pro_exploit` command to run an automated exploit and the `-d` option to perform a dry run of the automated exploit.

```
msf-pro > pro_exploit 192.168.184.0/24 -d

Id Project Desc Status Information
== =====
12 default exploiting

<*) Started task 1
```

## Setting the Application Evasion Level for an Automated Exploit

Use the `pro_exploit` command to run an automated exploit and the `-ea` option to set the evasion level for an automated exploit. The application evasion level affects SMB, DCERPC, and HTTP based exploits. You can assign an evasion level of `none`, `low`, `medium`, and `high`. Higher evasion levels use more aggressive evasion techniques.

```
msf-pro > pro_exploit 192.168.184.0/24 -ea low

Id Project Desc Status Information
== =====
12 default exploiting

<*) Started task 1
```

## Application Evasion Level Options for SMB

The following table describes the application evasion levels for SMB:

Evasion Level	Description
None	Does not apply any evasion techniques.
Low	Obscures the PIPE string, places extra padding between the SMB headers and data, and obscures the path names.
Medium	Segments SMB read/write operations.
High	Sets the maximum size for SMB reads and writes to a value between 4 and 64.

## Application Evasion Level Options for DCERPC

The following table describes the application evasion levels for DCERPC:

Evasion Level	Description
None	Does not apply any evasion techniques.
Low	Adds fake UUIDs before and after the actual UUID targeted by the exploit.
Medium	
High	Sets the maximum fragmentation size of DCERPC calls to a value between 4-64.

## Application Evasion Level Options for HTTP

The following table describes the application evasion levels for HTTP:

Evasion Level	Description
None	Does not apply any evasion techniques.
Low	Adds “header folding”, which splits HTTP headers in separate lines, joined by whitespace by the server. Adds random cases to HTTP methods. Adds between 1-64 fake HTTP headers.
Medium	Adds fake query strings to GET requests (1-64 of them). Adds 1-64 whitespace characters between tokens.

Evasion Level	Description
High	Encodes some characters as percent-u unicoded characters (half, randomly), adds a fake “end” to HTTP requests before the attack, and uses back slashes instead of forward slashes.

## Setting the TCP Evasion Level in an Automated Exploit

Use the `pro_exploit` command to run an automated exploit and the `-et` option to set the TCP evasion level. You can assign an evasion level of `none`, `low`, `medium`, and `high`.

```
msf-pro > pro_exploit 192.168.184.0/24 -ea low

Id Project Desc Status Information
== =====
12 default exploiting

<*) Started task 1
```

## TCP Evasion Level Options

The following table describes the TCP evasion levels:

Evasion Level	Description
None	Does not apply any evasion techniques.
Low	Inserts delays between TCP packets.
Medium	Sends small TCP packets.
High	Inserts delays between TCP packets and sends small TCP packets.



## Setting the Payload Connection Type for an Automated Exploit

Use the `pro_exploit` command to run an automated exploit and the `-m` option to set the payload type for an automated exploit. The payload types are auto, bind, and reverse

```
msf-pro > pro_exploit 192.168.184.0/24 -m bind

Id Project Desc Status Information
== =====
12 default exploiting

<*) Started task 1
```

## Payload Connection Types

The following table describes the payload types:

Payload Type	Description
Auto	Automatically selects the payload connection type for the exploit. Auto chooses bind when the system detects NAT, otherwise, the system uses reverse for most exploits.
Bind	Attaches a command prompt to a listening port on the exploited system. You can connect to the bind shell to access the exploited system.
Reverse	Creates a connection from the target machine back to you as a command prompt.

## Setting the Minimum Rank for an Automated Exploit

Use the `pro_exploit` command to run an automated exploit and the `-r` option to set the payload type for an automated exploit. The minimum rank settings are low, average, normal, good, great, and excellent.

```
msf-pro > pro_exploit 192.168.184.0/24 -r good

Id Project Desc Status Information
== =====
12 default exploiting

<*) Started task 1
```

## Minimum Reliability Rank

The minimum reliability rank indicates the potential impact that the exploits have on the target system. If you use a high ranking, such as excellent or great, Metasploit Pro uses exploits that are unlikely to crash the service or system.

The following table describes the minimum ranks that are available:

Minimum Reliability Rank	Description
Low	Exploits are unlikely to compromise common platforms.
Average	Exploits are unreliable and unlikely to exploit the target system.
Normal	Exploits are reliable, but depend on a specific version. These exploits do not auto-detect the appropriate targets.
Good	Exploits have a default target and are the common case for a particular type of software. For example, English, Windows XP for a desktop application and 2003 for a server.
Great	Exploits have a default target and auto-detects the appropriate target. These exploits use an application-specific return address after a version check.
Excellent	Exploits do not crash the service. Exploits that typically have this ranking are SQL injection, CMD execution, and web application exploits.

# CHAPTER 8

## REPORTS

This chapter covers the following topics:

- [Reports Overview 45](#)
- [Working with Reports 46](#)

### Reports Overview

A report provides valuable information and results that you can use to solve and mitigate security vulnerabilities. A report helps you to identify the vulnerabilities in a target network and pinpoint how an organization can strengthen their security infrastructure.

Metasploit Pro offers several report types that you can use to categorize your findings and test results. The report type that you select depends on the information that you want to present. The Metasploit Console only supports the generation of Audit reports. If you want a more specialized report type, you must generate reports from the Web UI.

Ultimately, reports help you to clearly assess and identify the vulnerabilities and risks that exist on the target. Use this information to provide support and to outline the tactics that an organization can implement to improve its security posture.

### Reports Types

The following table describes the report formats that are available:

Report Type	Description
Audit	Combines the high-level results from the other reports and presents them in a single comprehensive report.
Compromised and Vulnerable Hosts	Lists all hosts on which Metasploit Pro was able to open a session, successfully run a module, or record a vulnerability.
Authenticated Tokens	Lists all cracked hosts and includes all cracked passwords, SMB hashes, and SSH keys discovered.

Report Type	Description
Services	Lists all network services discovered by Metasploit Pro.
Collected Evidence	Lists all looted hosts and includes the files and screen shots collected from the compromised hosts.
Campaigns	Lists all Web Campaigns run as part of the project.
Webapp	Lists all websites and the vulnerabilities, forms, and pages associated with the websites.
PCI Compliance	Uses PCI compliance criteria to analyze the hosts.
FISMA Compliance	Uses FISMA compliance criteria to analyze the hosts.

## Working with Reports

A report provides detailed information and results for the penetration test. Use reports to perform an analysis of the target network and to provide valuable information to help solve and mitigate security vulnerabilities.

A report contains the information that you obtain during a penetration test. Reports help you identify vulnerabilities in a target network and help you to pinpoint how an organization can strengthen their security infrastructure.

## Supported Tasks

You can perform the following reporting tasks through the Metasploit Console:

- Create an Audit report.
- Specify a display name for the report.
- Specify the output type for the report.

## Generating a Report

Use the `pro_report` command to generate an audit report for the current project. If you want to use a different report type, you must generate the report through the Web UI.

If you do not specify options for `pro_report`, Metasploit Pro generates a PDF of the report and automatically generates a report name for the report.

```
msf-pro > pro_oroject
default
* dailyscan
msf-pro > pro_report
```

## Specifying the Report Type

You can generate reports in Word, PDF, HTML, and XML.

Use the `pro_report` command to generate a report and the `-t` option to specify the format for the report.

```
msf-pro > pro_oroject
default
* dailyscan
msf-pro > pro_report -t word
```

## Specifying the Report Name

Use the `pro_report` command to generate a report and the `-n` option to specify the report name. The `-n` option sets the name that displays for the report in the Web UI and in the Metasploit Console.

```
msf-pro > pro_oroject
default
* dailyscan
msf-pro > pro_report -n scanReport
```

## Viewing Reports

You can access the reports from the Web UI or from the following directory: `<metasploit directory>/apps/pro/reports`.

# CHAPTER 9

# METASPLOIT CONSOLE COMMAND REFERENCE

This chapter covers the following topics:

- [Command Overview](#) 48
- [Accessing the Metasploit Pro Console](#) 52
- [Basic Task Commands](#) 52
- [Database Back End Commands](#) 59
- [Core Commands](#) 71

## Command Overview

Commands enable you to interact with Metasploit Pro through the command line interface.

There are three types of commands available in the Metasploit Console:

- Metasploit Pro commands
- Core commands
- Database backend commands

## Metasploit Pro Commands

In Metasploit Pro, a task represents an action that the system can perform, such as a scan, bruteforce attack, exploit, report generation, and data collection. The Metasploit Console provides Metasploit Pro commands that you can use to perform these tasks.

The following table provides general descriptions for Metasploit Pro commands:

Command	Description
pro_bruteforce	Use this command to configure and run a bruteforce attack against a target host or range of hosts.
pro_collect	Use this command to gather evidence, such as passwords, screenshots, SSH keys, and system information, from an active session.
pro_discover	Use this command to scan the target network for available hosts and active ports.
pro_exploit	Use this command to run an exploit against a target host.
pro_project	Use this command to create and manage a project.
pro_report	Use this command to generate a report for the project.
pro_tasks	Use this command to view a list of active tasks.
pro_user	Use this command to view and change the current user.
version	Use this command to view the current Metasploit Framework, Metasploit Console, and Metasploit Pro versions.

## Core Commands

Use the core commands to manage the datastore, active sessions, modules, and variables.

The following table provides general descriptions for core commands:

Command	Description
?	Use this command to display the help menu.
back	Use this command to go back to the original context or the basic <code>msf</code> prompt.
banner	Use this command to display the Metasploit banner.
cd	Use this command to change the current working directory.
color	Use this command to toggle the color of the command prompt.
connect	Use this command to connect to a host during a pivoted session.
exit	Use this command to exit and close the Metasploit Console.
help	Use this command to display the help menu.

Command	Description
info	Use this command to view information about a module or list of modules.
irb	Use this command to drop into irb scripting mode, which enables you to run Ruby scripts from the Metasploit Console.
jobs	Use this command to display and manage the active jobs.
kill	Use this command to kill an active job.
load	Use this command to load a Metasploit Framework plugin, such as the Nessus or Nexpose plugin.
loadpath	Use this command to load the modules from a given directory.
makerc	Use this command to generate a resource file for the commands that were used in the session.
popm	Use this command to control the module stack. Use the <code>pushm</code> command to load the module into the current context and use the <code>popm</code> command to unload the module from the current context. After you pop the module from the stack, the context reverts back to the previous module context.
previous	Use this command to set the previously loaded module as the current module.
pushm	Use this command to control the module stack. This command loads a module into the current context and enables you to configure a module. Use the <code>popm</code> command to unload the module and go back to the previous module context.
quit	Use this command to exit the console.
reload_all	Use this command to unload and reload the all modules from all module paths.
resource	Use this command to run commands from a file.
route	Use this command to utilize an active session to route traffic to a specific subnet.
save	Use this command to save the current datastore.
search	Use this command to search for modules. Use keywords with this command to quickly find modules.
sessions	Use this command to manage and interact with active sessions.
set	Use this command to define a variable for a module.



Command	Description
setg	Use this command to define a global variable.
show	Use this command to display modules and module options.
sleep	Use this command to set the amount of time that you want the system to be idle.
spool	Use this command to create a file that contains the console output.
threads	Use this command to manage background threads.
unload	Use this command to unload a Metasploit Framework plugin.
unset	Use this command to unset a variable.
unsetg	Use this command to unset a global variable.
use	Use this command to select a module and to set it as the current context.
version	Use this command to show the current Metasploit Framework, Metasploit Console, and Metasploit Pro versions.

## Database Backend Commands

Use the database backend commands to manage the connection to the database and the information that the database contains. The database stores data that the system was able to collect from the target hosts, such as service information, looted data, credentials, and notes.

The following table provides general descriptions for database backend commands:

Command	Description
creds	Use this command to list and manage the credentials that are stored in the database.
db_connect	Use this command to connect to the PostgreSQL database.
db_disconnect	Use this command to disconnect from the PostgreSQL database.
db_export	Use this command to export the content, such as collected evidence, from the database to a file.
db_import	Use this command to import scan data to the database.
db_nmap	Use this command to run and configure an Nmap scan.

Command	Description
db_status	Use this command to return the status of the connection between Metasploit and the PostgreSQL database.
hosts	Use this command to list and configure hosts.
loot	Use this command to view a list of looted data for a host.
notes	Use this command to view and define notes for a host.
services	Use this command to view and configure the active services for a host.
vulns	Use this command to view and configure the vulnerabilities for a host.
workspace	Use this command to view and manage workspaces within Metasploit.

## Accessing the Metasploit Pro Console

You can access the Metasploit Pro console from the Start menu. Select **Start > Metasploit > Metasploit Console**. You can also launch `console.bat` from the Metasploit directory to open the Metasploit Pro console

## Basic Task Commands

The following sections provide descriptions and syntaxes for the basic console commands.

The basic commands include the following:

- Bruteforce
- Discovery
- Exploitation
- Evidence collection
- Report generation
- Task log generation
- User information retrieval

### Pro\_bruteforce

This command performs a bruteforce attack on the addresses or address range that you specify. If you do not specify any addresses, the bruteforce attack uses the network range that

you specify for the project. The default scope setting is normal, but you can change the scope to quick, normal, deep, known, and default.

## Pro\_bruteforce Options

The following table describes the options that are available for a bruteforce attack:

Option	Description
-G	Do not get sessions from successful logins.
-I	Do not include imported credentials.
-K	Do not include known credentials.
-b <opt>	Defines the host blacklist.
-d	Performs a dry run of the bruteforce attack.
-h	Displays the help for the specified command.
-l <opt>	Sets the LHOST for all payloads.
-m <opt>	Sets the payload method. Choose from auto, bind, or reverse.
-q	Quits the bruteforce attack after a successful login.
-s <opt>	Defines the service that you want to attempt to bruteforce. Separate each service with a comma.
-sd <opt>	Deletes the SMB domains. Separate each domain with a comma.

## Pro\_bruteforce Syntax

```
pro_bruteforce <address range> <scope> -K -I -b <address> -s <service>
```

## Pro\_bruteforce Example

```
msf> pro_bruteforce <192.168.1.0/24> defaults -K -I -b 192.168.1.1 -s smb
```

## Pro\_collect

This command gathers evidence such as the host name, OS name and version, passwords and hashes, and SSH keys from the session that you specify or for all open sessions.

## Pro\_collect Options

The following table describes the options that are available for evidence collection:

Option	Description
-c	Sets the maximum number of files that you want to download based on the pattern that you define for -f.
-f	Defines the pattern that is used to gather files.
-h	Displays the help for the command that you specify.
-k	Defines the maximum size for individual files (in kilobytes)

## Pro\_collect Syntax

```
pro_collect -f *<file pattern> -c <max files> -k <max file size>
```

## Pro\_collect Example

```
msf > pro_collect -f *.xml -c 15 -k 250
```

## Pro\_discover

This command scans for all hosts in the target range. If you do not define a range, the system uses the network range that you define for the project.

## Pro\_discover Options

The following table describes the options that are available for a discovery scan:

Option	Description
-F	Sets the scan to not enumerate users via Finger.
-I	Sets the scan to perform a port scan. The scan does not identify any services.
-S	Sets the scan to not use SNMP to discover devices.
-U	Sets the scan to not perform UDP discovery.
-b <opt>	Defines the host blacklist. The host blacklist specifies hosts that you do not want to include in the scan.

Option	Description
-d	Performs a dry run of the scan.
-h	Displays the help for the command that you specify.
-p <opt>	Defines custom ports using an Nmap format.
-sd <opt>	Defines the domain for SMB discovery.
-sp <opt>	Defines the password for SMB discovery.
-su <opt>	Defines the user name for SMB discovery.

## Pro\_discover Syntax

```
pro_discover <address>
```

## Pro\_discover Example

```
msf> pro_discover 192.168.1.0/24
```

## Pro\_exploit

This command exploits target hosts. If you do not specify hosts for the exploit, the system uses the network range that you define in the project.

## Pro\_exploit Options

The following table describes the options that are available for and exploit:

Option	Description
-b <opt>	Defines the host blacklist, which does not include the hosts that you specify.
-d	Performs a dry run of the exploit attack.
-ea <opt>	Sets the evasion level for target applications. You can set the evasion level to 1, 2, or 3.
-et <opt>	Sets the evasion level for TCP. You can set the evasion level to 1, 2, or 3.
-h	Displays the help for the option that you specify.
-l <opt>	Sets LHOST for all payloads.

Option	Description
-m <opt>	Sets the payload method to auto, bind, or reverse.
-p <opt>	Defines the custom ports in Nmap format.
-pb <opt>	Quits the attempt to exploit the system after a successful login.
-r <opt>	Sets the minimum rank of exploits that the attack uses.

## Pro\_exploit Syntax

```
pro_exploit <options> <address>
```

## Pro\_exploit Example

```
msf> pro_exploit 192.168.1.1
```

## Pro\_project

This command creates, lists, and deletes projects.

## Pro\_project Options

The following table describes the options that are available for a project:

Option	Description
-a	Creates a project.
-d	Deletes a project.
-h	Displays the help for the command that you specify.

## Pro\_project Syntax

```
pro_project
```

## Pro\_project Example

```
msf> pro_project #lists current project  
msf> pro_project-a pentest1
```

## Pro\_report

This command generates a report for the current test. The report includes the hosts for the active penetration test. You can access the generated report in the following directory:

<Metasploit install directory>/apps/pro/reports.

## Pro\_report Options

The following table describes the options that are available for a report:

Option	Description
-h	Displays the help for the command that you specify.
-t	Defines the report type that you want to generate. You can specify the report type as PDF, Word, or RTF.

## Pro\_report Syntax

```
pro_report -t <report_format>
```

## Pro\_report Example

```
msf> pro_report -t pdf
```

## Pro\_tasks

This command displays the tasks that are currently in progress. You can use the `pro_tasks` command to display a log for the task and to kill the task.

## Pro\_tasks Options

The following table describes the options that are available for Metasploit Pro tasks:

Option	Description
-h	Displays the help for the command that you specify.
-k	Kills the task that you specify.
-r	Displays that tasks that are in progress.
-w <opt>	Displays the task log for the task that you specify.

## Pro\_tasks Syntax

```
pro_tasks
```

## Pro\_tasks Example

```
msf> pro_tasks -r  
msf> pro_tasks -k 1 -w 3
```

## Pro\_user

This command returns the current user for the project.

## Pro\_user Options

The following table describes the options that are available for the user command:

Option	Description
-h	Displays the help for the command that you specify.
-l	Lists the users for the system.



## Pro\_user Syntax

```
pro_user
```

## Pro\_user Example

```
msf> pro_user -l
```

## Version

This command returns the version for the system.

## Version Options

The following table describes the options that are available for the version command:

Option	Description
-h	Displays the help for the command that you specify.

## Version Syntax

```
version
```

## Version Example

```
msf> version
```

## Database Back End Commands

The following sections describe the database back end commands.

For commands that search the database, you can use the Nmap host specification format instead of the full IP address.

## Creds

This command enables you to manage credentials. By default, the user name and password is “blank.”

### Creds options

The following table describes the options that are available for credential management:

Option	Description
-a	Adds a credential for the target hosts that you specify.
-d	Deletes a credential.
-h	Displays the help for the command that you specify.
-p <opt>	Lists the credentials for the port that you specify.
-s <opt>	Lists the credentials for the service that you specify.
-t <opt>	Adds a credential of the type that you define.
-u	Adds a credential for the user that you specify.
-P	Adds a password for the user that you specify.

### Creds Syntax

```
creds <address range>  
creds -a <address range> -p <port> -t <type> -u <user> -p <password>
```

### Creds Example

```
msf> creds a 192.168.1.0/24 -p 445 -u joe -p smith2!  
msf> creds 192.168.1.0/24 #shows credentials for the specified host
```

## Db\_autopwn

This command automatically runs Nmap to scan for hosts, stores hosts in the database, and runs hosts against exploits in the Metasploit Framework.

## Db\_autopwn Options

The following table describes the options that are available for automated exploits:

Option	Description
-h	Displays the help for the command that you specify.
-t	Lists the exploits that match the criteria that you specify.
-x	Selects modules based on vulnerability references.
-p	Selects modules based on open ports.
-e	Launches exploits against all matched targets.
-r	Uses a reverse bind connect shell.
-q	Disables output for exploit modules.
-R <rank>	Runs modules with the minimal rank.
-I <range>	Exploits hosts within the range that you specify.
-X <range>	Excludes hosts from the range that you specify.
-PI <range>	Exploits hosts with the open ports that you specify.
-PX <range>	Excludes the hosts with the open ports that you specify from the exploit.
-m <regex>	Runs the modules with names that match the regular expression that you define.
-T <secs>	Sets the maximum run time for an exploit in seconds.

## Db\_autopwn Syntax

```
db_autopwn <options>
```

## Db\_autopwn Example

```
msf> db_autopwn -I 192.168.1.0/24
```

## Db\_add\_cred

This command adds a credential for a service.

## Db\_add\_cred Options

The following table describes the options that are available for credential management for a specific host and port combination.

Option	Description
-h	Displays the help for the command that you specify.

## Db\_add\_cred Syntax

```
db_add_cred <host> <port> <user> <password> <type> <active>
```

## Db\_add\_Cred Example

```
msf> db_add_cred 192.168.1.1 445 joe ps123
```

## Db\_add\_host

This command adds a host to the database.

## Db\_add\_host Options

The following table describes the options that are available for you to add a host to a project:

Option	Description
-h	Displays the help for the command that you specify.

## Db\_add\_host Syntax

```
db_add_cred <host> <port> <user> <password> <type> <active>
```

## Db\_add\_host Example

```
msf> db_add_cred 192.168.1.1 445 joe ps123
```

## Db\_add\_note

This command adds a note to a host.

The type column uses a hierarchical format similar to OIDs, with the top level of the tree listed first and each successive element connected with a period. The last item in the type name is the actual value. For example, the type "host.os.updates.last\_updated\_time" indicates a value called "last\_updated\_time" within the "updates" branch of the "os" child of the "host" tree. A new sub-category is created when more than two types can be grouped within it.

## Db\_add\_note Options

The following table describes the options that are available for you to add a note to a host:

Option	Description
-h	Displays the help for the command that you specify.
<type>	Defines a free form option. This is typically set to <code>host.os.fingerprint</code> or <code>smb.users</code> .

## Db\_add\_note Syntax

```
db_add_note <host address> <type> <note>
```

## Db\_add\_note Example

```
msf> db_add_note 192.168.1.1 type windows only host
```

## Db\_add\_port

This command defines a port for a host.

## Db\_add\_port Options

The following table describes the options that are available for you to define a port for a host:

Option	Description
-h	Displays the help for the command that you specify.

## Db\_add\_port Syntax

```
db_add_port <host> <port> <protocol> <name>
```

## Db\_add\_port Example

```
msf> db_add_port 192.168.1.1 445
```

## Db\_connect

This command enables you to connect to a database.

## Db\_connect Options

The following table describes the options that are available for you to connect to a database:

Option	Description
-h	Displays the help for the command that you specify.

## Db\_connect Syntax 1

```
db_connect <username:password>@<host:port>/<database>
```

## Db\_connect Syntax 2

```
db_connect -y <path/to/database.yml>
```

## Db\_connect Example

```
msf> db_connect user:pass123@192.168.1.1/metasploit
```

## Db\_disconnect

This command disconnects you from the current database.

## Db\_disconnect Options

The following table describes the options that are available for you to disconnect from a database:

Option	Description
-h	Displays the help for the command that you specify.

## Db\_disconnect Syntax

```
db_disconnect
```

## Db\_driver

This command defines a database driver.

## Db\_driver Options

The following table describes the options that are available for you to define a database driver:

Option	Description
-h	Displays the help for the command that you specify.

## Db\_driver Syntax

```
db_driver <driver name>
```

## Db\_export

This command exports a file that contains the contents of a database.

## Db\_export Options

The following table describes the options that are available for you to export data from a database:

Option	Description
-h	Displays the help for the command that you specify.
-f	Specifies the file format that the system uses to export data from the database.
-a <file name>	Specifies the name for the file that the system exports.



## Db\_export Syntax

```
db_export -f <format> -a <filename>
```

## Db\_export Example

```
msf> db_export -f xml -a dbexport
```

## Db\_import

This command imports a scan result file. Use this command in place of deprecated commands, such as `db_import_amap_log`, `db_import_amap_mlog`, `db_import_ip360_xml`, `db_import_ip_list`, `db_import_msfe_xml`, `db_import_nessus_nbe`, `db_import_nessus_xml`, `db_import_nmap_xml`, and `db_import_qualys_xml`, to import files.

## Db\_import Options

The following table describes the options that are available for you to import data to a database:

Option	Description
-h	Displays the help for the command that you specify.

## Db\_import Syntax

```
db_import <filename>
```

## Db\_nmap

This command executes Nmap and automatically records the output.

## Db\_nmap Options

The following table describes the options that are available for you to use Nmap:

Option	Description
-h	Displays the help for the command that you specify.

## db\_nmap Syntax

```
db_nmap
```

## Db\_status

This command displays the current database status.

### Db\_status Options

The following table describes the options that are available for you to display the database status:

Option	Description
-h	Displays the help for the command that you specify.

## Db\_status Syntax

```
db_status
```

## Hosts

This command lists all hosts that the database contains.

### Hosts Options

The following table describes the options that are available for you to display the hosts in a database:

Option	Description
-h	Displays the help for the command that you specify.

## hosts Syntax

```
hosts
```

## Loot

This command lists all the loot that the database contains.

## Loot Options

The following table describes the options that are available for you to display the evidence that the database contains:

Option	Description
-h	Displays the help for the command that you specify.

## loot Syntax

```
loot <address1 address 2> <-t <type1, type2>>
```

## Notes

This command lists all notes that the database contains.

## Notes Options

The following table describes the options that are available for you to display the notes that the database contains:

Option	Description
-h	Displays the help for the command that you specify.

## Notes Syntax

```
notes
```

## Services

This command lists all the services that the database contains.

### Services Options

The following table describes the options that are available for you to display the services that the database contains:

Option	Description
-h	Displays the help for the command that you specify.

## services Syntax

```
services
```

## Vulns

This command lists all the vulnerabilities that the database contains.

### Vulns Options

The following table describes the options that are available for you to display the vulnerabilities that the database contains:

Option	Description
-h	Displays the help for the command that you specify.

## Vulns Syntax

```
vulns
```

## Workspace

This command enables you to switch between database workspaces. Workspaces can be used to manage and maintain related information. When viewing all workspaces, the current workspace is denoted with an asterisk (\*).

## Workspace Options

The following table describes the options that are available for you to add, delete, and view database workspaces:

Option	Description
-h	Displays the help for the command that you specify.
-a <name>	Adds the workspace that you specify.
-d <name>	Deletes the workspace that you specify.

## workspace Syntax

```
workspace /#lists all workspaces  
workspace -a <name> /#adds a workspace  
workspace -d <name> /#deletes a workspace
```

## Workspace Example

```
msf> workspace  
msf> workspace -a w2 -d w3
```

## Core Commands

The following sections describe the core commands that are available with the Metasploit Console.

## Back

Use this command to switch between contexts.

## Banner

Use this command to display the Metasploit banner.

## Cd

Use this command to change the current working directory.

## Color

Use this command to toggle the color.

## Connect

Use this command to communicate with a host.

## Connect Options

The following table describes the options that are available for you to connect with a host:

Option	Description
-C	Attempts to use CRLF for EOL sequence.
-P <opt>	Defines the source port.
-S <opt>	Defines the source address.
-h	Displays the help for the command that you specify.
-i <opt>	Sends the contents of the file that you specify.
-p <opt>	Displays a list of proxies that you can use.
-s	Uses SSL to connect.
-u	Switches to a UDP socket.
-w <opt>	Defines the connect timeout in seconds.
-z	Attempts to only make a connection.

## Connect Syntax

```
connect <options> <host> <port>
```

## Connect Example

```
msf> connect 192.168.1.1 445
```

## Exit

Use this command to exit the console.

## Help

Use this command to display the help menu.

## Info

Use this command to display the information for a module, such as the version, rank, options, description, and reference information.

## Info Syntax

```
info <module name>
```

## Irb

Use this command to start a live Ruby interpreter shell. Use the IRB shell to run commands and create scripts.

## Jobs

Use this command to display and manage jobs.

## Jobs Options

The following table describes that options that you can use to manage tasks:

Option	Description
-K	Ends all jobs that are running.
-h	Displays the help for the command that you specify.
-i <opt>	Displays information about the job that you specify.
-k <opt>	Ends the job that you specify.
-l	Displays a list of running jobs.
-v	Prints information about a job.

## Jobs Syntax

```
jobs <options>
```

## Jobs Example

```
msf> jobs -K
msf> jobs -l
msf> jobs -k job1
```

## Kill

This command ends the job that you specify. You can use the jobs command to view a list of jobs that are running.

## Kill Syntax

```
kill <job name>
```

## Load

This command loads a Metasploit Framework plugin. You can find Metasploit plugins in the following directory: `Metasploit/apps/pro/msf3/plugins`.



## Load Syntax

```
load </path to MSF plugins>
```

## Load Example

```
msf> Load /Metasploit/apps/pro/msf3/plugins/lab
```

## Loadpath

This command loads modules from the directory that you specify. The directory that you specify must contain the subdirectories for module types.

## Loadpath Syntax

```
loadpath /path/to/modules/
```

## Quit

This command exits the console.

## Reload\_all

This command reloads the modules from all module paths.

## Route

This command uses a supplied session to route traffic to a specific subnet.

## Route Syntax

```
route <add/remove/get/flush/print> subnet netmask <comm/sid>
```

## Route Example

```
msf> route add 192.168.1.0 255.255.255.0 2 #default session number is local
msf> route print #shows active routing table
```

## Save

This command saves the active data stores and the current settings, such as the global variables. You can access the settings every time that you log in to the Metasploit Pro console.

## Save Syntax

```
save
```

## Search

This command searches for specific modules. You can use regular expression of the built-in keyword search to perform a search.

## Search Keywords

The following table describes the keywords that you can use to perform a search:

Keyword	Description
name	Returns modules that match the name that you specify.
path	Returns modules that match the path or reference name that you specify.
platform	Returns modules that affect the platform that you specify.
type	Returns the modules that match the type that you specify. The type can be exploit, auxiliary, or post.
app	Returns the modules that match the application type that you specify. The application can be client or server attacks.
author	Returns the modules that match the author that you specify.

Keyword	Description
cve	Returns the modules that match the CVE ID that you specify.
bid	Returns the modules that match the Bugtraq ID that you specify.
osvdb	Returns the modules that match the OSVDB ID that you specify.

## Search Syntax

```
search <keywords>
```

## Search Example

```
msf> search cve:2008 type:exploit
```

## Sessions

This command enables you to list, configure, and close a session.

## Sessions Options

The following table describes the options that are available for you to interact with sessions:

Option	Description
-K	Ends all active sessions.
-c <opt>	Runs a command on all sessions or on a session given with -i.
-d <opt>	Detaches an interactive session.
-h	Displays the help for the command that you specify.
-i <opt>	Interacts with the session ID that you specify. For example, <code>sessions -i 2</code> .
-k <opt>	Ends the session that you specify.
-l	Displays a list of active sessions.
-q	Runs quiet mode.

Option	Description
-r	Resets the ring buffer for the session that you specify or for all sessions.
-s <opt>	Runs a script on the session that you specify or for all sessions.
-u <opt>	Upgrades a WIN32 shell to a Meterpreter session.
-v	Lists verbose fields.

## Sessions Syntax

```
sessions <options>
```

## Sessions Example

```
msf> sessions -l
```

## Set

This command sets a variable to a value that you define for a module.

## Setg Syntax

```
set <variable> <value>
```

## Setg Example

```
msf> set LHOST 192.168.1.1
```

## Setg

This command sets a global variable to a value that you define. After you create a global variable, you can reuse them in different projects.

## Setg Syntax

```
setg <variable> <value>
```

## Setg Example

```
msf> setg LHOST 192.168.1.1
```

## Show

This command displays a list of the modules that are available. Additionally, you can use this command to view the payloads and plugins that are available in the Metasploit Framework.

If you want more granular control over a module, use the `use` command to set the context to that module. You can use the `advanced`, `evasion`, `targets`, and `actions` commands to view more information about a specific module.

## Show Options

The following table describes the commands that you can use to list modules:

Option	Description
all	Lists all encoders, nops, exploits, payloads, plugins, and auxiliary modules.
encoders	Lists all encoders that are available.
nops	Lists all NOP generators that are available.
exploits	Lists all exploits that are available.
payloads	Lists the payloads that are available across all platforms.
auxiliary	Lists all auxiliary modules that are available.
plugins	Lists all plugins that are available with the Metasploit Framework.
options	Lists the global options that are available. If you use the <code>options</code> command outside the context of a module, you can view all global options.

## Show Syntax

```
show <option>
```

## Show Example

```
msf> show exploits #returns a list of all exploit modules  
use admin/db2/db2rcmd #sets the module context  
msf> show advanced #returns the advanced settings for the module
```

## Sleep

This command defines the amount of time, in seconds, that the system can sleep or perform no tasks.

## Sleep Syntax

```
sleep <time>
```

## Sleep Example

```
msf> sleep 10
```

## Spool

This command writes console output in a file and displays the output onto the screen.

## Spool Syntax

```
spool <file name>
```

## Spool Example

```
msf> spool /tmp/console.log
```

## Threads

This command enables you to view and modify background threads,

## Threads Options

The following table describes the options that are available for you to modify background threads:

Option	Description
-K	Terminates all non-critical threads.
-h	Displays the help for the command that you specify.
-i <opt>	Displays information for the thread that you specify.
-k <opt>	Terminates the thread ID that you specify.
-l	Displays a list of all background threads.
-v	Prints information about the thread that you specify.

## Threads Syntax

```
threads <options>
```

## Threads Example

```
msf> threads -l #lists all background threads  
msf> threads -k 1 #kills the thread
```

## Unload

This command unloads a Metasploit Framework plugin.

## Unload Syntax

```
unload <plugin name>
```

## Unset

This command unsets a variable. For global variables, use the `unsetg` command.

## Unset Syntax

```
unset var1
```

## Unsetg

This command unsets a global variable.

## Unsetg Syntax

```
unsetg gvar1
```

## Use

This command selects a module by the name that you specify.



## Use Syntax

```
use <module name>
```

## Use Example

```
msf> use admin/db2/db2rcmd
```

## Version

This command displays the Metasploit Framework and console library version numbers.

## Version Syntax

```
version
```

# INDEX

## A

auxiliary module 10

## B

banner 72

## C

ConsoleLogging 15  
creds 60  
CSV file 25

## D

database 16, 17  
db\_add\_cred 61  
db\_add\_host 62  
db\_add\_note 63  
db\_add\_port 63  
db\_autopwn 60  
db\_connect 17, 64  
db\_disconnect 65  
db\_driver 65  
db\_export 66  
db\_import 25, 67  
db\_nmap 67  
db\_status 68

## E

exploit 9, 10, 35  
exploit module 10, 35

## H

hosts 23

## I

import  
    data 25  
irb 73

## L

loadpath 75

LogLevel 15  
loot 28

## M

module 9

## N

NOP generator 11

## P

payload 9, 11  
payload encoder 11  
post-exploitation 10  
post-exploitation module 10  
PostgreSQL 16  
pro\_bruteforce 52  
pro\_collect 53  
pro\_discover 54  
pro\_exploit 55  
pro\_project 56  
pro\_report 57  
pro\_tasks 57  
pro\_user 58

## R

reload\_all 75  
report 45, 46

## S

scan data 25  
SessionLogging 16  
setg 78  
spool 80

## T

threads 81

## U

unload 82  
unset 82  
unsetg 82

## V

verbosity 15  
Version 59  
vulnerability 9  
vulns 70



workspace 71