Chapter 4 – Metasploit Framework
Part 1 – Introduction & Metasploit Fundamentals

What is Metasploit?

Metasploit Framework (MSF) is a penetration testing framework used to discover, validate, and assess vulnerabilities in authorized environments. It provides exploits, payloads, auxiliary modules, post-exploitation modules, encoders, listeners, and reporting capabilities.

Main Components

• Exploits: Target known vulnerabilities.
• Payloads: Code executed after successful exploitation.
• Auxiliary: Scanning and enumeration modules.
• Post: Actions after a session exists.
• Encoders: Modify payload representation (not a reliable AV bypass).
• NOP Generators: Internal support for some exploits/payloads.

Workflow

Reconnaissance → Scanning → Enumeration → Identify Vulnerability → Search Module → Configure → Execute (authorized) → Session → Post-Exploitation → Cleanup → Report.

Directory Structure

/usr/share/metasploit-framework/
modules/, exploits/, payloads/, auxiliary/, post/, encoders/, data/

Module Naming

Example: exploit/windows/smb/ms17_010_eternalblue
Type → Platform → Service → Vulnerability.

Database

Metasploit can use PostgreSQL to store hosts, services, credentials, vulnerabilities, notes, loot, and workspaces.

Normal Use

Use reconnaissance tools first, identify a vulnerability, configure the appropriate Metasploit module, verify exploitability in an authorized lab, perform post-exploitation if appropriate, and document findings.

Special Cases

If no module exists, a custom exploit may be required. Zero-day vulnerabilities generally require independent research. Manual exploitation may be preferable in some situations.

Common Beginner Mistakes

Skipping reconnaissance, choosing incompatible payloads, failing to configure required options, and assuming every target can be exploited.

Quick Revision

Metasploit is a framework—not just an exploit tool. Understand the vulnerability before using modules. Database features improve organization and reporting.

Chapter 4 – Metasploit Framework

Part 2 – msfconsole Basics

What is msfconsole?

The primary interface for Metasploit. It is used to search modules, configure options, manage sessions, and perform authorized penetration testing tasks.

Core Commands

msfconsole – Start Metasploit.
version – Show framework version.
help or ? – Display help.
clear – Clear the screen.
exit or quit – Exit the console.

Module Commands

search – Search for modules.
info <module> – View module details.
use <module> – Load a module.
back – Return to the main prompt.

Viewing Information

show options – Display configurable parameters.
show payloads – List compatible payloads.
show targets – Display supported targets.

Useful Features

Tab completion speeds up typing. Arrow keys recall previous commands. Help is available for individual commands using help command_name.

Workflow

Start msfconsole → Search module → Read info → Use module → Show options → Choose payload (if applicable) → Configure → Run only in authorized environments.

Beginner Mistakes

Skipping the info command, forgetting required options, loading the wrong module, or assuming every search result is appropriate.

Quick Revision

Always review module information before use. Search first, load second, configure third, execute only with proper authorization.

Chapter 4 – Metasploit Framework

Part 3 – Module Management & Configuration

Overview

This section explains how to configure Metasploit modules correctly before running them. Proper configuration helps avoid common mistakes and ensures required options are set.

Essential Commands

show options – Display configurable module options.
set OPTION VALUE – Set a module-specific option.
setg OPTION VALUE – Set a global option.
getg OPTION – View a global option.
unset OPTION – Remove a module option.
unsetg OPTION – Remove a global option.

Advanced Commands

show advanced – View advanced settings.
show missing – Display required options that are not configured.

Execution

check – Test whether the module reports the target as vulnerable (if supported).
run / exploit – Execute the loaded module in an authorized environment.
run -j – Run as a background job.
jobs – List background jobs.
jobs -k ID – Stop a background job.
rerun – Execute the current module again.
reload – Reload the current module.

Target & Payload Selection

show targets – List supported targets.
set TARGET N – Select a target profile.
show payloads – Display compatible payloads.

Best Practices

Review module information first, configure all required options, verify settings with show options or show missing, use check when supported, and execute only in authorized environments.

Quick Revision

Search → Info → Use → Show Options → Set Values → Show Missing → Check (if available) → Run.

Chapter 4 – Metasploit Framework

Part 4 – Payloads, Encoders & Listeners

What is a Payload?

A payload is the code that runs after an exploit successfully achieves code execution. Payloads determine what happens next during an authorized assessment.

Payload Types

Single: Self-contained payload.
Staged: Small initial stage followed by a second stage.
Stageless: Entire payload delivered as one unit.
Meterpreter: Advanced interactive payload.
Command Shell: Basic shell access.

Essential Commands

show payloads – List compatible payloads.
set payload PAYLOAD_NAME – Select a payload.
show options – View payload-specific options.

Reverse vs Bind

Reverse payloads initiate a connection back to the tester's listener. Bind payloads wait for the tester to connect. Choose the type appropriate for the authorized lab environment.

Encoders & NOPs

show encoders – List available encoders.
set ENCODER encoder_name – Select an encoder (if supported).
show nops – Display NOP generators.

Encoders modify payload representation but are not a reliable antivirus bypass.

Listeners

use exploit/multi/handler – Load the handler.
set payload ... – Choose the matching payload.
show options – Configure required settings.
run or exploit – Start the handler.

Workflow

Load exploit → Show payloads → Select payload → Configure options → Start handler (if needed) → Execute only in an authorized environment.

Quick Revision

Always select a compatible payload, review payload options, understand reverse vs bind, and ensure the handler matches the payload.

Chapter 4 – Metasploit Framework

Part 5 – Auxiliary Modules (Complete Deep Dive)

What are Auxiliary Modules?

Auxiliary modules perform reconnaissance, enumeration, scanning, service verification, and other non-exploitation tasks during authorized security assessments.

Common Categories

Scanner – Discover hosts and services.
Enumeration – Gather detailed information.
Login Testing – Verify credentials where authorized.
Fuzzer – Test software robustness in controlled environments.
Server – Provide supporting services for specialized lab workflows.

Essential Commands

search type:auxiliary – Find auxiliary modules.
search http – Search by service.
info MODULE – View module details.
use MODULE – Load a module.
show options – View configuration.
set OPTION VALUE – Configure options.
run – Execute the module.

Threading & Jobs

set THREADS N – Configure concurrent workers.
run -j – Run in the background.
jobs – List background jobs.
jobs -k ID – Stop a background job.

Workflow

Search → Info → Use → Show Options → Configure → Run → Review Results → Back.

Best Practices

Read module information first, configure required options, use conservative thread counts where appropriate, document findings, and perform only authorized testing.

Quick Revision

Auxiliary modules are designed for reconnaissance and enumeration rather than exploitation. They help you gather information before selecting the next assessment step.

Chapter 4 – Metasploit Framework

Part 6 – Meterpreter Fundamentals (Conceptual Guide)

What is Meterpreter?

Meterpreter is an advanced interactive environment provided by certain Metasploit payloads after successful exploitation during authorized security assessments. It supports information gathering, file interaction, process inspection, network review, and post-exploitation workflows.

Meterpreter vs Command Shell

Meterpreter offers a richer, extensible interface, while a traditional command shell provides basic operating system command execution.

Sessions

A session represents an active connection to a target during an authorized assessment. Multiple sessions may be managed simultaneously.

Typical Activities

Identify the operating system, determine the current user context, inspect running processes, review network configuration, gather evidence, and document findings.

Evidence & Best Practices

Collect only information relevant to the assessment, minimize changes to the target, document observations, close sessions when testing is complete, and include evidence in the final report.

Common Mistakes

Making unnecessary changes, failing to document findings, collecting irrelevant data, and leaving sessions open after testing.

Quick Revision

Meterpreter is designed for post-exploitation in authorized environments. The goal is validation, evidence collection, and reporting—not simply obtaining access.

Chapter 4 – Metasploit Framework

Part 7 – Post-Exploitation Concepts (Professional Methodology)

Overview

Post-exploitation is the phase after successful vulnerability verification during an authorized assessment. The focus is on understanding impact, collecting evidence, assessing risk, documenting findings, and cleaning up.

Primary Objectives

Understand the affected system, validate business impact, identify additional security weaknesses, collect evidence, and recommend remediation.

Information Gathering

Document the operating system, installed software, patch level, user context, hostname, network configuration, running services, and security controls.

Risk & Evidence

Assess business impact, determine risk severity, and collect reproducible evidence such as system information, configuration details, screenshots (where permitted), and timestamps.

Key Concepts

Persistence and pivoting should only be evaluated when explicitly authorized and within the agreed scope. The emphasis is on defensive assessment rather than maintaining access.

Cleanup & Reporting

Remove temporary assessment artifacts, close sessions, restore the environment if needed, and prepare a report containing an executive summary, technical findings, evidence, risk ratings, and remediation recommendations.

Quick Revision

Professional post-exploitation is about validation, evidence, documentation, cleanup, and reporting—not simply obtaining access.

Chapter 4 – Metasploit Framework

Part 8 – Workspaces, Loot & Reporting

Overview

Workspaces help organize penetration testing projects by separating hosts, services, vulnerabilities, notes, credentials, and collected artifacts into independent assessment databases.

Workspace Management

workspace – List workspaces.
workspace -a NAME – Create a workspace.
workspace NAME – Switch workspace.
workspace -d NAME – Delete a workspace.

Database Objects

hosts – View discovered hosts.
services – View discovered services.
vulns – Review recorded vulnerabilities.
notes – Display notes.
notes -a – Add notes.
creds – View stored credentials.
loot – View collected assessment artifacts.

Database Commands

db_status – Check database connectivity.
db_import FILE – Import supported scan results into the active workspace.

Reporting Workflow

Create workspace → Verify database → Import scan results → Review hosts and services → Record notes → Validate findings → Collect evidence → Prepare the final report.

Best Practices

Use a separate workspace for every engagement, keep notes throughout testing, verify database status before starting, organize evidence carefully, and write findings as they are validated.

Quick Revision

Workspaces separate projects, the database stores assessment information, and good documentation is just as important as technical testing.

Chapter 4 – Metasploit Framework

Part 9 – Automation & Resource Scripts

Overview

Automation reduces repetitive work by using resource scripts, global settings, logging, and repeatable workflows during authorized security assessments.

Resource Scripts

resource file.rc – Execute a resource script.
A resource script is a plain text file containing Metasploit console commands that run sequentially.

Creating Scripts

makerc file.rc – Record commands entered in msfconsole into a reusable resource script.
Review generated scripts before reuse.

Logging

spool assessment.log – Start logging console output.
spool off – Stop logging.
Logs help with troubleshooting and report writing.

Automation Workflow

Create workspace → Verify database → Set global options → Import scan results → Enable logging → Begin assessment.

Best Practices

Keep scripts small and modular, use descriptive names, test scripts in a lab, review changes regularly, and store reusable scripts in a dedicated folder.

Quick Revision

Use resource scripts for repeatable setup, makerc to capture workflows, spool for logging, and automation to improve consistency rather than replace understanding.

Chapter 4 – Metasploit Framework

Part 10 – Common Mistakes, Troubleshooting & Best Practices

Overview

This chapter summarizes common Metasploit issues, troubleshooting methodology, and professional best practices for authorized security assessments.

Troubleshooting Workflow

Read the error message → Verify configuration → Check connectivity → Review module information → Test again → Document results.

Common Issues

show missing – Review required options.
show options – Verify configured values.
info MODULE – Read module documentation.
db_status – Verify database connectivity.
jobs / jobs -k ID – Review or stop background jobs.
workspace – Confirm the active project.

Professional Best Practices

Always work with authorization, create a new workspace for each engagement, keep logs and notes, collect only necessary evidence, minimize impact, and produce clear documentation.

Interview Highlights

Know the difference between exploits and payloads, understand Meterpreter conceptually, explain workspaces, auxiliary modules, and resource scripts.

Master Cheat Sheet

Key commands: msfconsole, help, search, info, use, show options, show payloads, set/setg, run, exploit, workspace, hosts, services, vulns, notes, creds, loot, db_status, db_import, resource, makerc, spool, jobs, exit.

Final Revision

Metasploit is a framework supporting the full lifecycle of an authorized assessment. Strong organization, documentation, and reporting are as important as technical skills.