

Defense Evasion with Meterpreter

Meterpreter Deep Dive



William Hardy
Security Engineer



Why Learn Defense Evasion ?



Required skill for penetration testers and red-teamers

Better detection and protection technologies & increased adoption rates

Know your enemies to improve your defenses

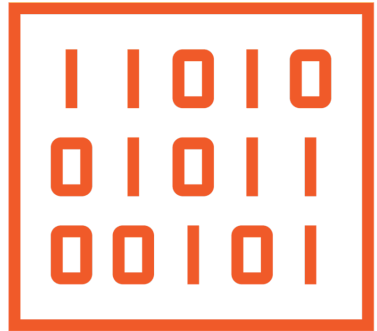
Developing better security technologies



Meterpreter Deep Dive



Meterpreter Components



Stager



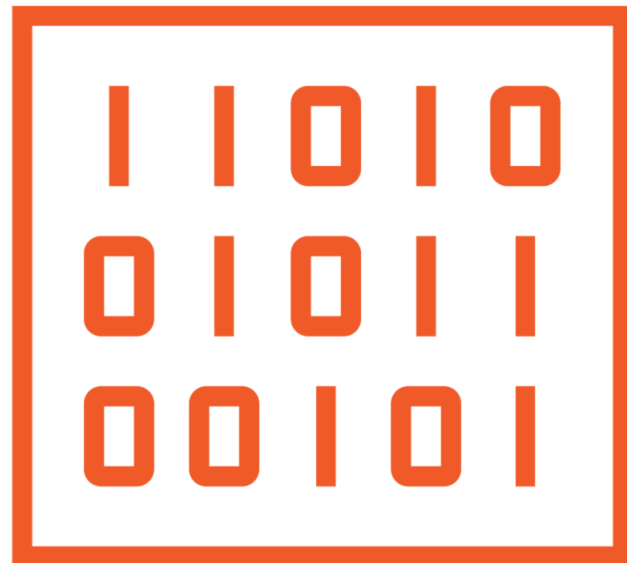
Handler



Payload



Stager



Shellcode

Tied to a specific CPU architecture

Position-independent

Written in assembly

Can be used to

- Exploit memory corruption bugs in software
- Backdoor executable files



Handler



Runs on the attacker's machine

Stager connects to the handler

Sends the meterpreter payload to the victim's machine

Enables interaction with the victim's machine

Payload



A reflective DLL file sent back to the stager by the handler

Bootstrapped by the stager

Contains the actual meterpreter code



Reflective DLL Injection



Loads the DLL from memory without writing it to disk

Essentially performs the job of the operating system loader

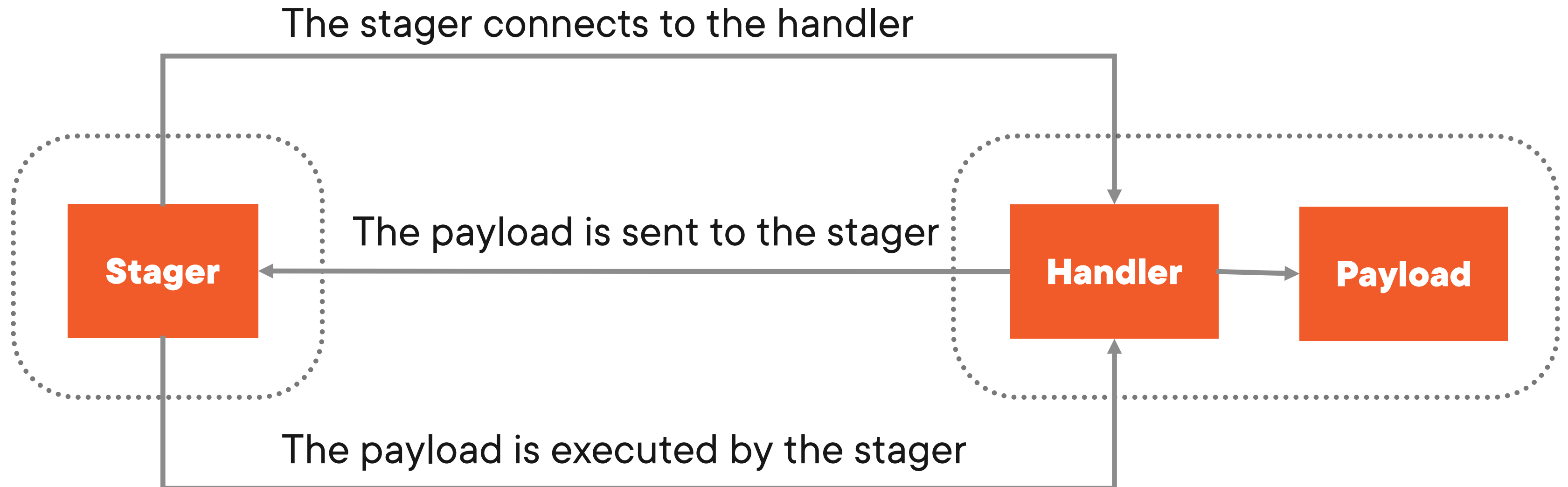
Very evasive, requires memory forensics to be detected

Used by meterpreter payloads and other extensions such as «stdapi», «incognito» or «kiwi»



Victim process

Attacker machine



Demo



Meterpreter Initialization

- Generate the HTTP stager
- Write a shellcode runner for our stager
- Configure the handler
- Execute the shellcode runner
- Step through the code as it executes

