SECURITY 101 | 13 MIN READ

A step-by-step guide to the Metasploit Framework

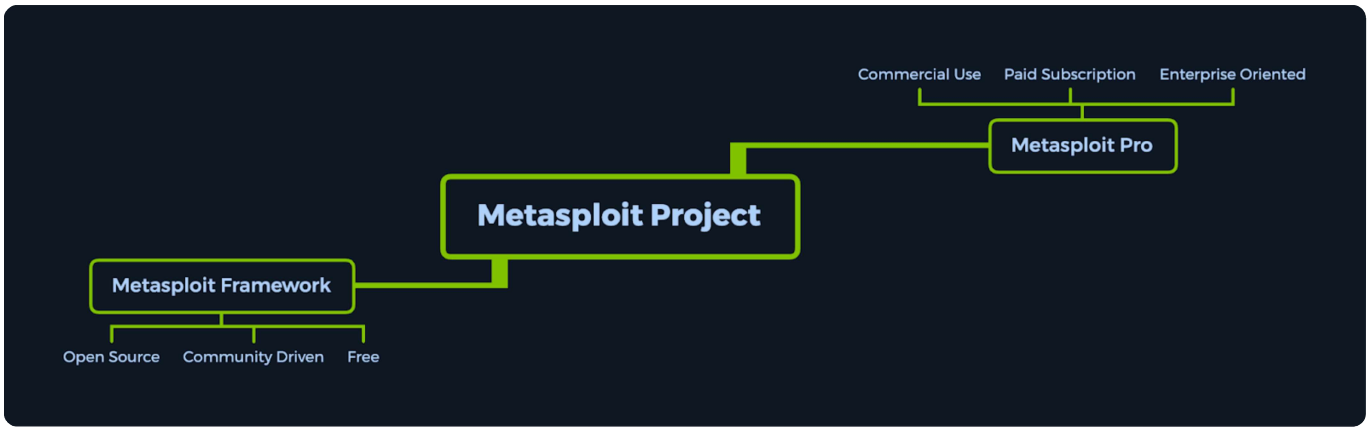
Follow this Metasploit Framework tutorial for a comprehensive overview of module types, targets, payloads, and much more!

Dimitris, Oct 16 2024

This post is based on the Hack The Box (HTB) Academy module on Using the Metasploit Framework. This module equips learners with the skills to use Metasploit for network enumeration, attacks, testing security vulnerabilities, evading detection, performing privilege escalation attacks, and performing post-exploitation.

You can learn more by browsing the catalog of free or advanced cybersecurity courses on the HTB Academy!

What is Metasploit?



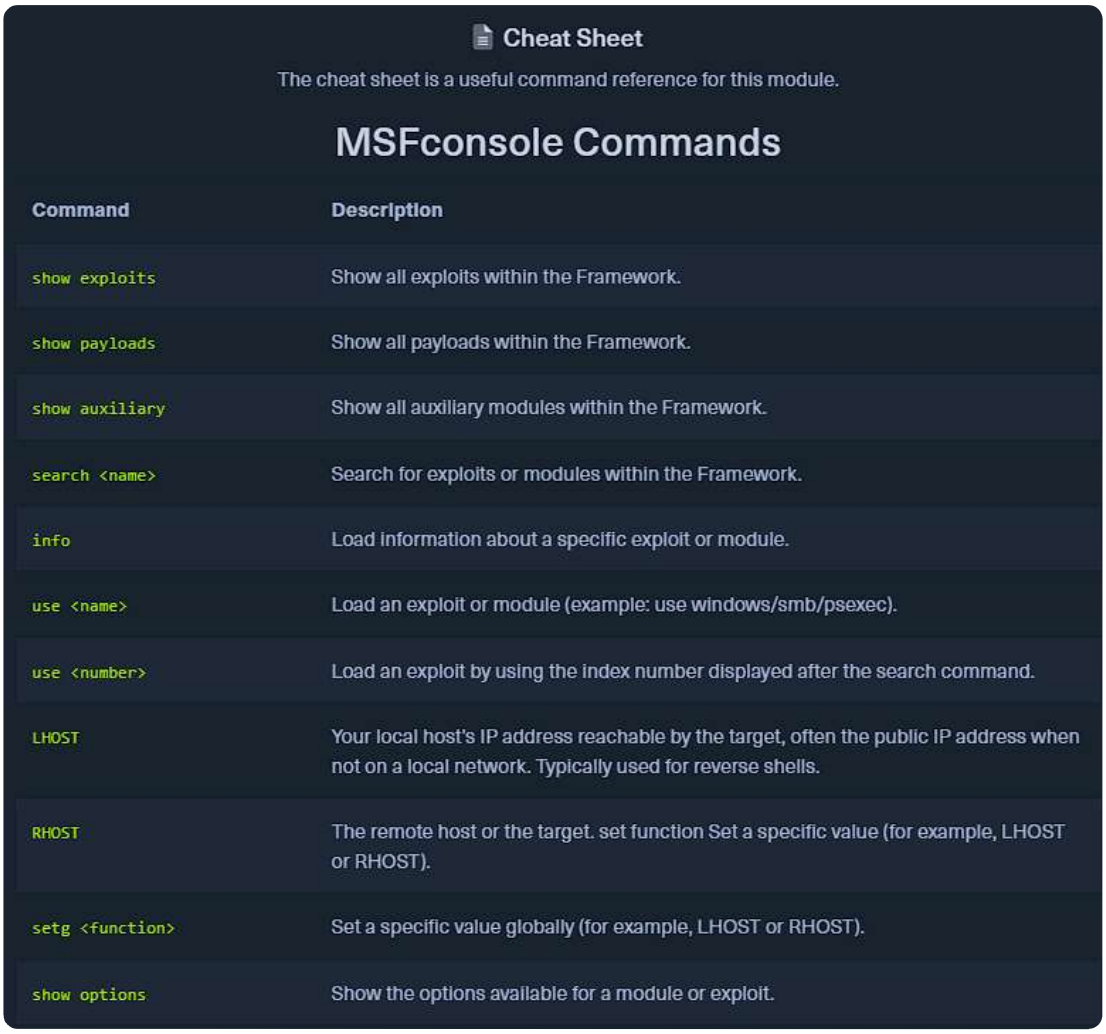
The Metasploit Project is a Ruby-based, modular penetration testing platform that allows you to write, test, and execute exploit code. This exploit code can be custom-made by you, or taken from a database containing the latest discovered and modularized exploits.

At its core, the Metasploit Project is a collection of commonly used tools that provide a complete environment for penetration testing and exploit development. The Metasploit Framework includes a suite of tools that you can use to test security vulnerabilities, enumerate networks, execute attacks, and evade detection.

Understanding the Metasploit Framework

The Metasploit Framework Console (msfconsole) is the most popular interface to the Metasploit Framework (MSF). It provides an "all-in-one" centralized console and allows you to access virtually all options available in the MSF.

Msfconsole may seem intimidating at first, but once you learn the command syntax, you can start to appreciate the power of this interface.



Cheat Sheet

The cheat sheet is a useful command reference for this module.

MSFconsole Commands

Command	Description
`show exploits`	Show all exploits within the Framework.
`show payloads`	Show all payloads within the Framework.
`show auxiliary`	Show all auxiliary modules within the Framework.
`search <name>`	Search for exploits or modules within the Framework.
`info`	Load information about a specific exploit or module.
`use <name>`	Load an exploit or module (example: use windows/smb/psexec).
`use <number>`	Load an exploit by using the index number displayed after the search command.
`LHOST`	Your local host's IP address reachable by the target, often the public IP address when not on a local network. Typically used for reverse shells.
`RHOST`	The remote host or the target. set function Set a specific value (for example, LHOST or RHOST).
`setg <function>`	Set a specific value globally (for example, LHOST or RHOST).
`show options`	Show the options available for a module or exploit.

For a full cheat sheet, see the HTB Academy module on Using the Metasploit Framework

Broadly, MSF offers the following features:

- The only supported way to access most of the features within Metasploit.
- Console-based interface to MSF.
- Has the most features and is the most stable MSF interface.
- Full readline support, tabbing, and command completion.
- Allows execution of external commands in msfconsole.

Should we rely on tools when penetration testing?

There is some debate in the security community about whether to use automated tools during a security assessment. Not using automated tools gives the security analyst or penetration tester no chance to “prove” themselves when interacting with a vulnerable environment.

However, some security experts disagree, arguing that tools help us learn better by offering a more user-friendly approach to the vast variety of vulnerabilities that exist in the wild, while saving us time for the more intricate parts of an assessment.

Some testers use automated tools to locate the "low-hanging fruit", so they can spend more time manually testing for more complicated or rarer vulnerabilities.

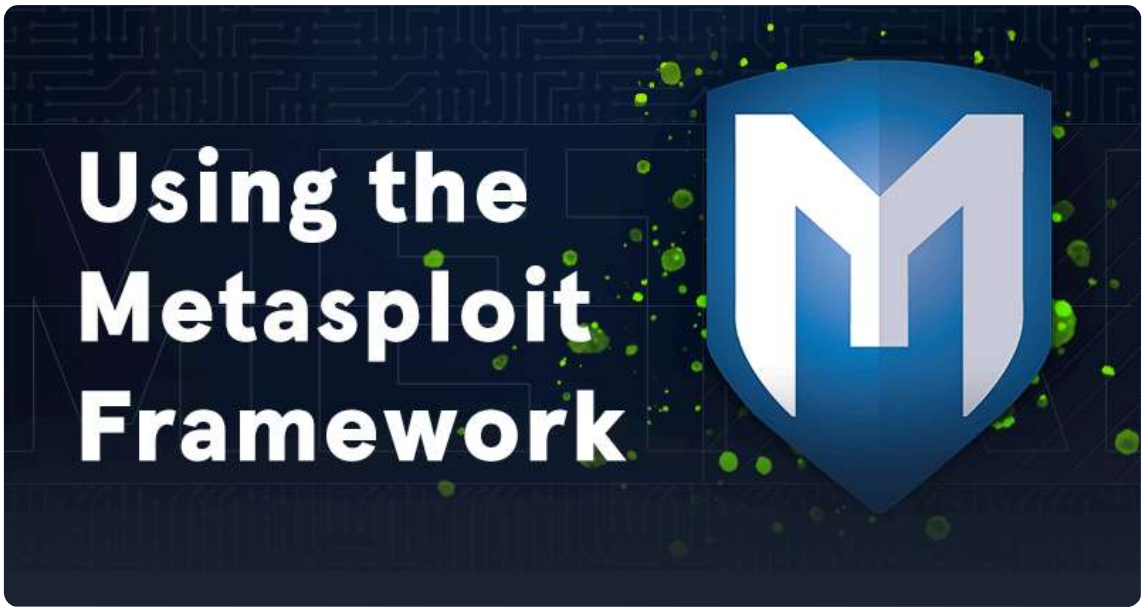
Either way, relying on tools can lead to some downsides, such as:

- Putting the tester into a comfort zone that will be hard to break out of when learning new skills.
- Creating a security risk, as the tools are published online for everyone (including attackers) to see and use.
- Causing a "tunnel vision" effect, where it feels as though "If the tool can't do it, neither can I."

We can combat these downsides by analyzing and learning our tools inside and out, so we can keep our tracks covered and avoid a cataclysmic event during our assessment.

As long as we follow the rules here (see Metasploit best practices for penetration testers), tools like Metasploit can be a valuable educational platform for beginners and a needed time-saver mechanism for professionals.

Do not get tunnel vision. Use the tool as a tool, not as a backbone or life support for our complete assessment.



In this module, you'll learn:

- An overview of the Metasploit Framework.
- Metasploit module types.
- Setting targets and payloads.
- Pivoting between sessions and jobs.
- Importing new modules.
- Creating a custom payload.
- Creating a custom module.
- Using Meterpreter and Kiwi.

PRACTICE USING THE METASPLOIT FRAMEWORK

Metasploit uses and benefits

So, who actually uses Metasploit? The easy-to-use framework has been adopted by security professionals and cybercriminals alike and is especially popular in the offensive security community.

Because the software is popular with cybercriminals and widely available, this reinforces the need for security professionals to become familiar with the framework even if they don't use it.

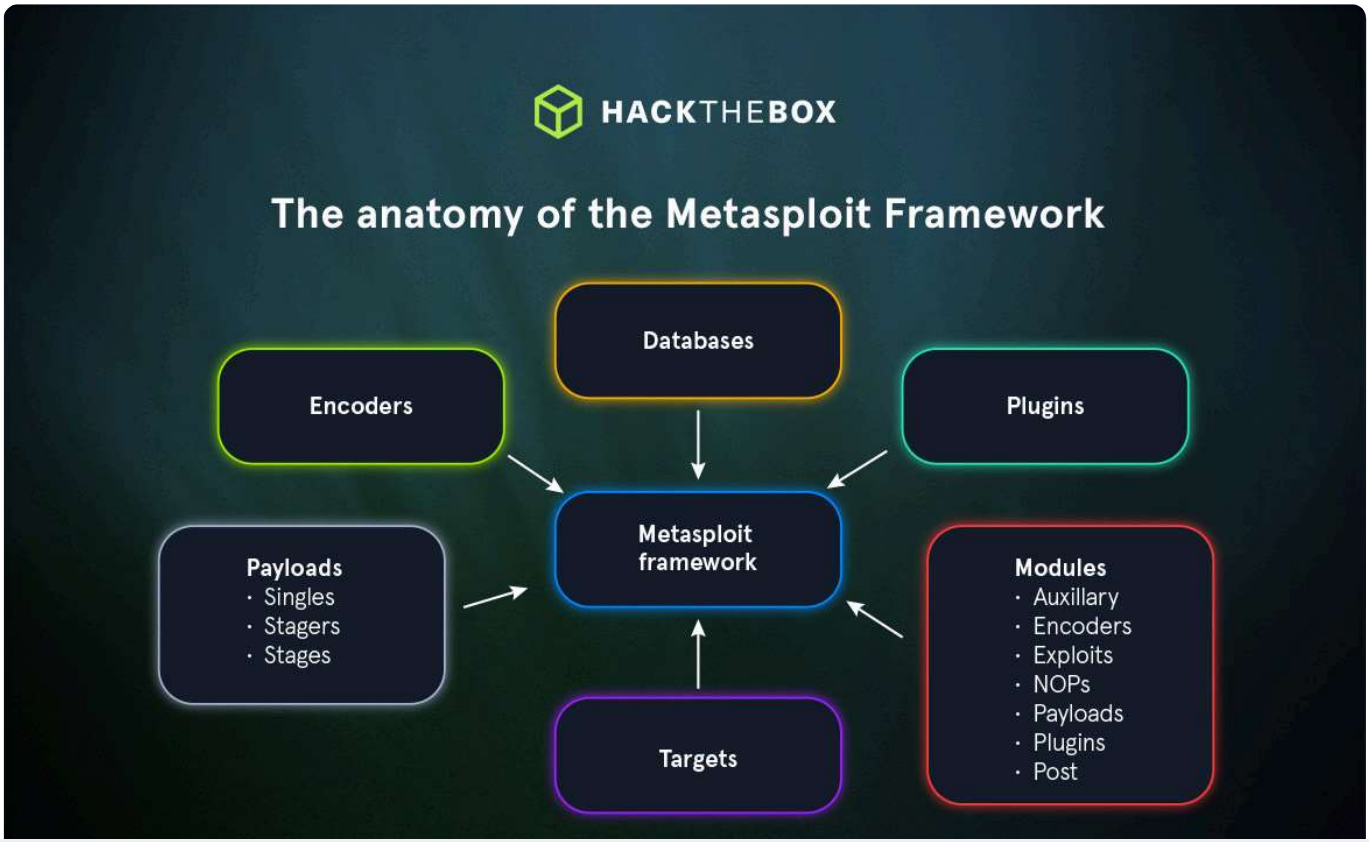
Using MSF has the following benefits:

- **Open source:** many adopt the MSF because it's open source and actively developed. This deep customization gives pentesters access to the source code and the ability to add their own modules.
- **Ease of use:** easily switch between payloads, this provides great flexibility when attempting to penetrate systems.
- **Clean exits:** MSF is able to exit cleanly without being detected, even if the target system is not expected to restart after the pentest.
- **Visual GUI:** manage vulnerabilities and create workspaces at a button's click.

Despite the industry debates revolving around the level of security knowledge needed to operate a “swiss army knife” type tool, such as Metasploit, these kinds of frameworks allow for in-depth exploration and auditing that one might not have the necessary time given different circumstances.

MSF Components

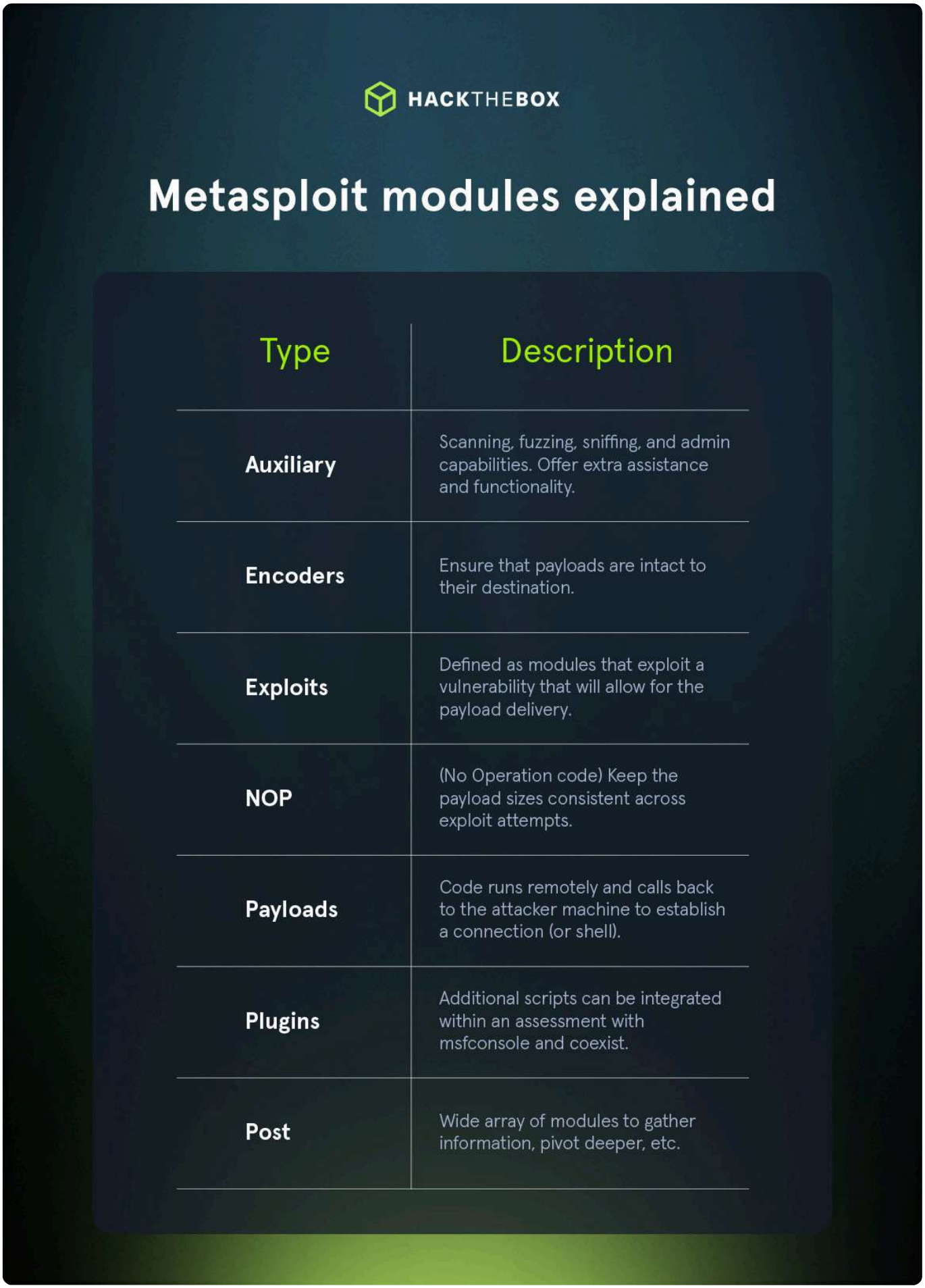
The Metasploit framework has several components that work together to offer a comprehensive penetration testing platform. We will dive into these components in detail to help you understand how the tool works.



Modules

Metasploit modules are prepared scripts with a specific purpose and functions that have already been developed and tested in the wild. Modules are organized into a variety of types, including Auxiliary, Exploits, and Payloads.

As an example, the exploit category consists of proofs of concept (POCs) that can be used to exploit existing vulnerabilities in a largely automated manner.



Type	Description

Auxiliary	Scanning, fuzzing, sniffing, and admin capabilities. Offer extra assistance and functionality.
Encoders	Ensure that payloads are intact to their destination.
Exploits	Defined as modules that exploit a vulnerability that will allow for the payload delivery.
NOP	(No Operation code) Keep the payload sizes consistent across exploit attempts.
Payloads	Code runs remotely and calls back to the attacker machine to establish a connection (or shell).
Plugins	Additional scripts can be integrated within an assessment with msfconsole and coexist.
Post	Wide array of modules to gather information, pivot deeper, etc.

Metasploit also offers a well-developed search function for the existing modules. We can use this function to quickly search through all the modules using specific tags to find a suitable one for our target.

```
msf6 > help search

Usage: search [<options>] [<keywords>:<value>]

Prepending a value with '-' will exclude any matching results.

If no options or keywords are provided, cached results are displayed.

OPTIONS:

  -h                    Show this help information

  -o <file>             Send output to a file in csv format

  -S <string>           Regex pattern used to filter search results

  -u                    Use module if there is one result

  -s <search_column>    Sort the research results based on <search_column> in ascending order

  -r                    Reverse the search results order to descending order



Keywords:

  aka              :  Modules with a matching AKA (also-known-as) name

  author           :  Modules written by this author

  arch             :  Modules affecting this architecture

  bid              :  Modules with a matching Bugtraq ID

  cve              :  Modules with a matching CVE ID

  edb              :  Modules with a matching Exploit-DB ID

  check            :  Modules that support the 'check' method

  date             :  Modules with a matching disclosure date

  description      :  Modules with a matching description

  fullname         :  Modules with a matching full name

  mod_time         :  Modules with a matching modification date

  name             :  Modules with a matching descriptive name

  path             :  Modules with a matching path

  platform         :  Modules affecting this platform

  port             :  Modules with a matching port

  rank             :  Modules with a matching rank (Can be descriptive (ex: 'good') or numeric with comparison operators (e

  ref              :  Modules with a matching ref

  reference        :  Modules with a matching reference

  target           :  Modules affecting this target

  type             :  Modules of a specific type (exploit, payload, auxiliary, encoder, evasion, post, or nop)

Supported search columns:
```

```
  date             :  Sort modules by their disclosure date. Alias for disclosure_date
  disclosure_date  :  Sort modules by their disclosure date
  name             :  Sort modules by their name
  type             :  Sort modules by their type
  check            :  Sort modules by whether or not they have a check method
Examples:
  search cve:2009 type:exploit
  search cve:2009 type:exploit platform:-linux
  search cve:2009 -s name
  search type:exploit -s type -r
```

Targets

Each module has a list of Targets, which are the unique Operating Systems (OS) that the module has been tested to run on. Use the “show targets” command within an exploit module view to display all available vulnerable targets for that specific exploit.

Check whether the module supports automatic targeting, because it will default to the target listed in the 0 position unless instructed otherwise.

```
msf6 > show targets
[-] No exploit module selected.
```

Payloads

A Metasploit Payload is a type of module that typically aids in returning a shell to the attacker.

These payloads are sent with the exploit, and designed to bypass standard functioning procedures of the vulnerable service and then run on the target OS to attempt to return a reverse connection to the attacker and establish a foothold.

There are three types of payload modules in the Metasploit Framework:

1. **Singles:** contain the exploit and the entire shellcode for the selected task. These payloads are by design more stable than their counterparts because they contain everything all-in-one.

2. **Stagers:** works with Stage payloads to perform a specific task. After deployment, a Stager establishes a connection between the attacker's machine and the victim host, so that it can run stages on the remote host.

3. **Stages:** components that are downloaded by Stager payload modules. The various payload Stages provide advanced features with no size limits, such as Meterpreter, VNC Injection, and others.

Encoders

Encoders assist with making payloads compatible with different processor architectures while at the same time helping with antivirus evasion. Encoders come into play when changing the payload to run on different OS and architectures.

Databases

Databases in msfconsole keep track of your results. During complex machine assessments, things can get complicated due to the sheer amount of search results, entry points, detected issues, and discovered credentials, among other returned data. This is where Databases come into play.

Msfconsole has built-in support for the PostgreSQL database system. With it, we have direct, quick, and easy access to scan results with the added ability to import and export results in conjunction with third-party tools. Database entries can also be used to configure Exploit module parameters directly with the already existing findings.

Plugins

Plugins are readily available software that has already been released by third parties and have approval from the creators of Metasploit to integrate their software inside the framework.

These can represent commercial products that have a Community Edition for free use but with limited functionality, or they can be individual projects developed by individuals.

The use of plugins makes a pentester's life even easier, bringing the functionality of other well-known software into the msfconsole or Metasploit Pro environments.

How to use Metasploit

Ready to start utilizing the MSF during your penetration tests? Here's a step-by-step guide on how to start using this popular tool:

Step 1: Download and install Metasploit

The Metasploit Framework works on several different platforms including Windows, Kali Linux, and macOS. Many security-oriented Linux distributions such as Parrot Security and Kali Linux come with msfconsole preinstalled.

Once installed, type msfconsole in your terminal of choice. After launching the msfconsole, the console presents the MSF splash art and the command line prompt, waiting for our first command.

Step 3: Update the database

Metasploit benefits from a huge database of exploits and vulnerabilities. To benefit from this database, keeping it up-to-date must be a priority. Type the "db update" command to ensure you have the latest database version.

Step 4: Select your module

As discussed earlier, modules make up the core components of the Metasploit Framework. To browse modules for scanning or exploiting networks, locate modules in the following directory:

```
/path/to/metasploit/apps/pro/msf3/modules
```

To find a specific exploit, use the "search" command. For instance, to find an exploit that targets the "SMB" protocol, type "search smb".

Once you have the exploit you're looking for, use the "use" command to select it. For instance, to run the "eternalromance" exploit, type `use exploit/windows/smb/MS17-010_eternalromance`.

Step 5: Set the target

Enter the IP address of the target system you want to test. Set the target with the "set" command. For instance, if the host's IP address is 192.168.1.100, type "set RHOST 192.168.1.100."

Don't forget to check whether your module is compatible with your target system using the "show targets" command.

Important: You should never run pentesting tools on a system you do not have permission to test or exploit. Depending on the laws in your area, doing so could be illegal. If you don't have a target to practice on, you can set up the Metasploitable project from GitHub: https://github.com/rapid7/metasploitable3.

Now that you have msfconsole set up, you can begin using it. To get more practice, check out HTB's modules on Getting Started (with pentesting), Using the Metasploit Framework, and Shells & Payloads.

Metasploit best practices for penetration testers

As penetration testers, we should never rely solely on a tool to do our jobs for us, which is why these best practices should always be top of mind:

1. Don't neglect your practical skills

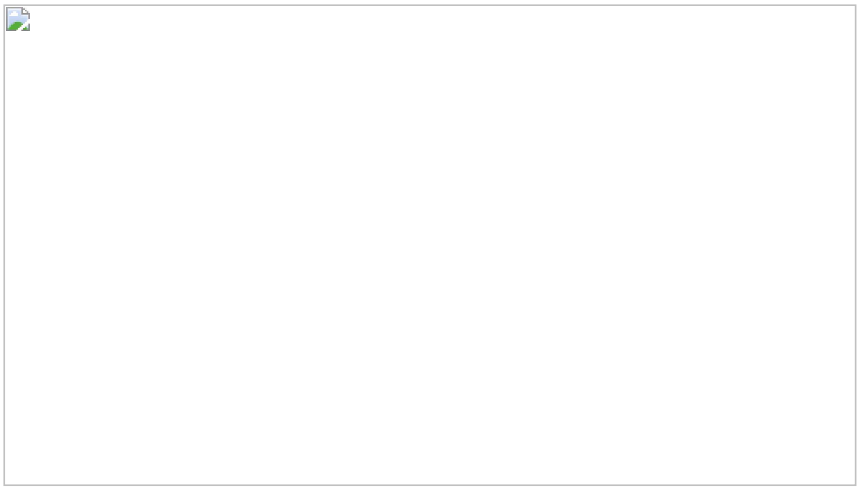
Many people often think that the failure of an exploit disproves the existence of the suspected vulnerability.

However, this is only proof that the Metasploit exploit you used did not work, not that the vulnerability does not exist. This is because many exploits may require customization for the target hosts to make the exploit work.

Therefore, consider automated tools such as the Metasploit framework as support tools, rather than a substitute for our manual skills.

2. Understand the fundamentals

Before executing any form of exploit, we must understand the fundamentals of penetration testing. This is not only key to excelling in your work, but also to spotting things that tools can overlook.



You should learn how to manually perform exploits before using tools. When beginners get into the testing side of security, they usually default to a linear, tool-based problem-solving workflow.

However, real-world penetration testing requires an element of raw human intuition. If human intuition wasn't needed, then paid software would have solved security by now. You can't rely exclusively on programmatic or tool-oriented thinking because creativity, adaptability, and out-of-the-box thinking are critical.

MSF is constantly being patched and updated, including updates for the latest CVEs. This means that you must keep it updated to make the most out of the tool, and to avoid overlooking recent vulnerabilities.

We recommend checking for updates with the “db update” command every time you use it.

Remember to check the Metasploit documentation if you have issues.

4. Keep track of your progress

You can become a better penetration tester by making notes on your findings throughout the process. This will help you spot areas for improvement, and prove your value to key stakeholders.

Don’t forget to take screenshots whenever you run an exploit or achieve access to something. They say “pictures are worth a thousand words”, and these screenshots can help you write any reports you have to produce.

👉 Free penetration testing report template.

Learn Metasploit with Hack The Box

The HTB Academy module introduces the fundamentals of the Metasploit Framework with a retrospective analysis of the usage of automated tools in today's penetration testing environments.

Our Metasploit module includes more information about using the MSF, as well as details about setting targets and payloads, importing new modules, and how to create custom modules and payloads. You will also learn about Kiwi and Meterpreter, two msfconsole extensions for more advanced testing.

LEARN MORE WITH HTB

Latest News

THREAT INTELLIGENCE | 5 MIN READ

War Room: CVE-2025-14847—Mongobleed explained

diskordia, Jan 07, 2026

ARTIFICIAL INTELLIGENCE | 5 MIN READ

Adversarial AI meets its match: The complete AI Red Teamer Path from HTB and Google

diskordia, Jan 07, 2026

NEWS | 3 MIN READ

Inclusivity, opportunity, impact: CodeFest 2025 powered by NAB + HTB

HTB-Bot, Jan 06, 2026

The latest news and updates, direct from Hack The Box

Read More

The #1 platform to build attack-ready teams and organizations.

Get a demo

FORRESTER® WAVE LEADER

CYBER ESSENTIALS

ISO 27001 ISO 27701 ISO 9001

4.8 Capterra 4.8

Products

Solutions

Resources

Company