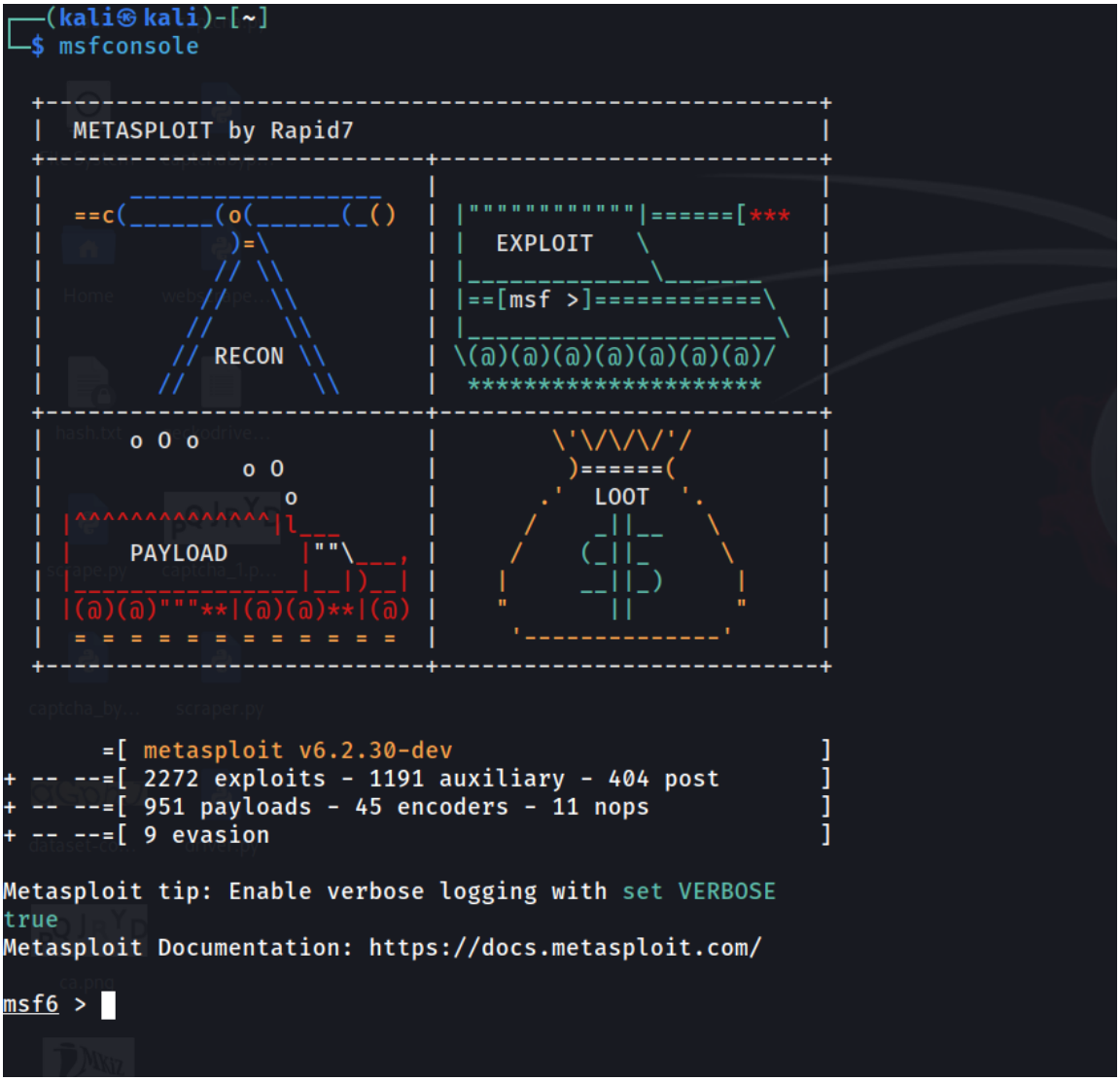
Metasploit - Introduction - Tryhackme

Introduction

- Metasploit is a widely used exploitation framework.
- It is a powerful tool that can support all phases of a penetration testing engagement, from information gathering to post-exploitation.
- The main components of metasploit-framework are:
  - **msfconsole:** The main command-line interface
  - **Modules:** Supporting modules such as exploits, scanners, payloads etc.
  - **Tools:** Tools that help with vulnerability research, vulnerability assessment, or penetration testing, for example - msfvenom, pattern_create etc.

Main Components of Metasploit

- Launch Metasploit using `msfconsole`.

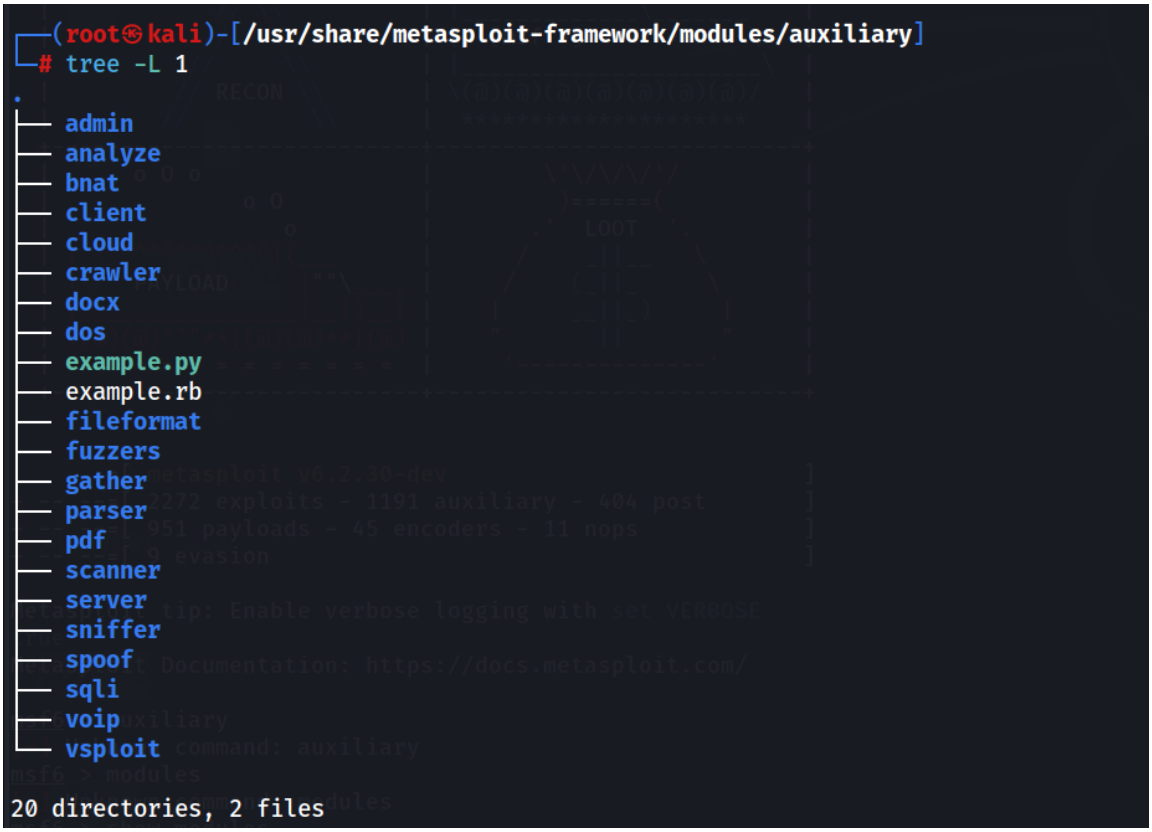


- It is our main interface to interact with the different modules of the Metasploit Framework.
- Modules are small components within the Metasploit framework that are built to perform a specific task, such as:
  - Exploiting a vulnerability
  - Scanning a target
  - Performing a brute-force attack
- Some important keyterms:
  - **Exploit:** It is a piece of code that uses a vulnerability present on the target system.
  - **Vulnerability:** It is a design, coding, or logic flaw affecting the target system.

- **Payload:** Exploits take the advantage of a vulnerability, but if we want the exploit to have the result we want like - gaining access to the target system, reading confidential information, loading a malware or backdoor on the target system etc. we need to use a payload. They are the code that runs on the target system.

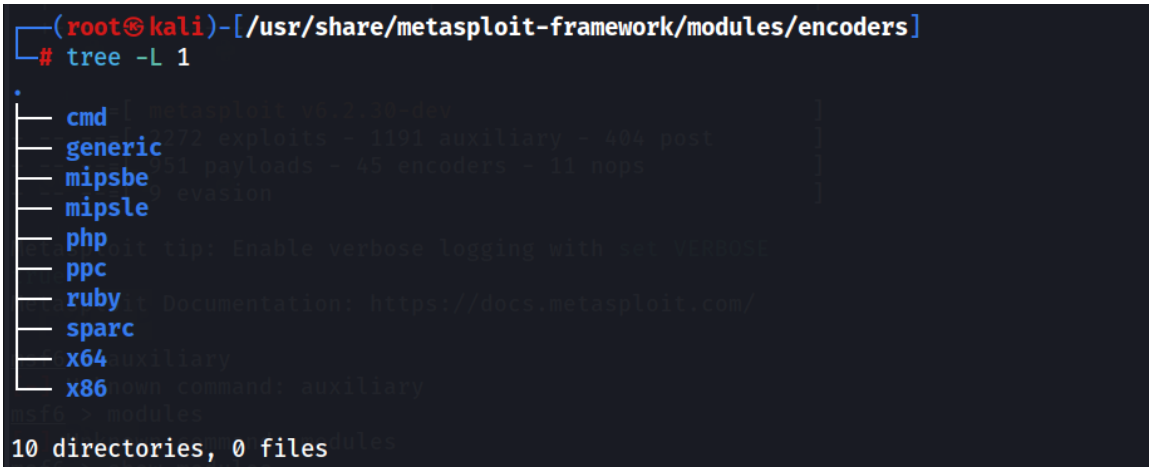
Auxiliary

- Any supporting module, such as scanners, crawlers and fuzzers, can be found here.



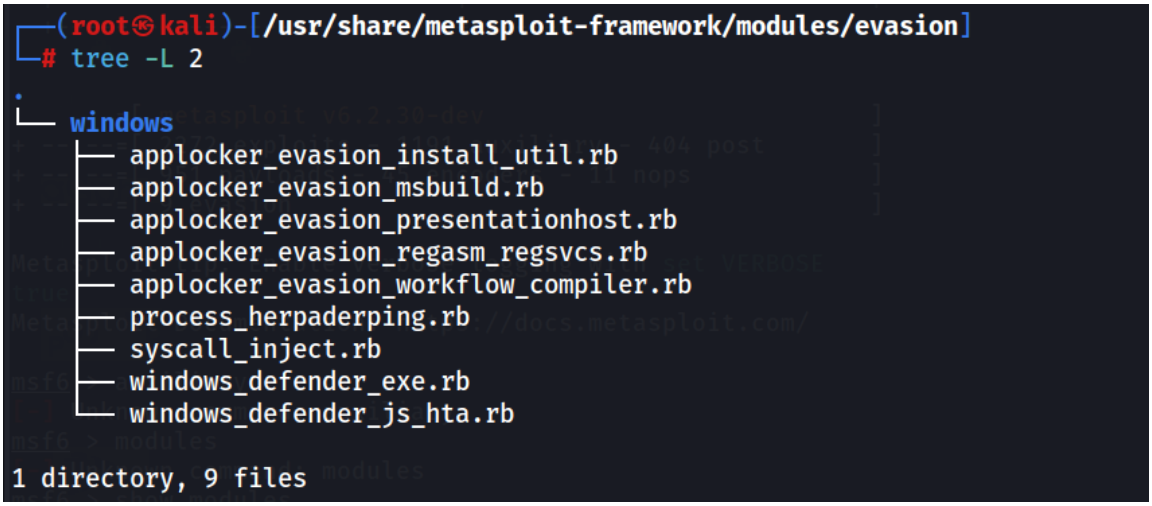
Encoders

- They allow us to encode the exploit and payload in the hope that a signature-based antivirus solution may miss them.
- Encoders can have a limited success rate as antivirus solutions can perform additional checks.



Evasion

- Encoders encode the payload, but they should not be considered a direct attempt to evade antivirus software. Whereas **evasion** modules will try that, with more or less success.



Exploits

```
┌──(root㉿kali)-[/usr/share/metasploit-framework/modules/exploits]
└─# tree -L 1
.
├── aix
├── android
├── apple_ios
├── bsd
├── bsdi
├── dialup
├── example_linux_priv_esc.rb
├── example.py
├── example.rb
├── example_webapp.rb
├── firefox
├── freebsd
├── hpux
├── irix
├── linux
├── mainframe
├── multi
├── netware
├── openbsd
├── osx
├── qnx
├── solaris
├── unix
└── windows

20 directories, 4 files
```

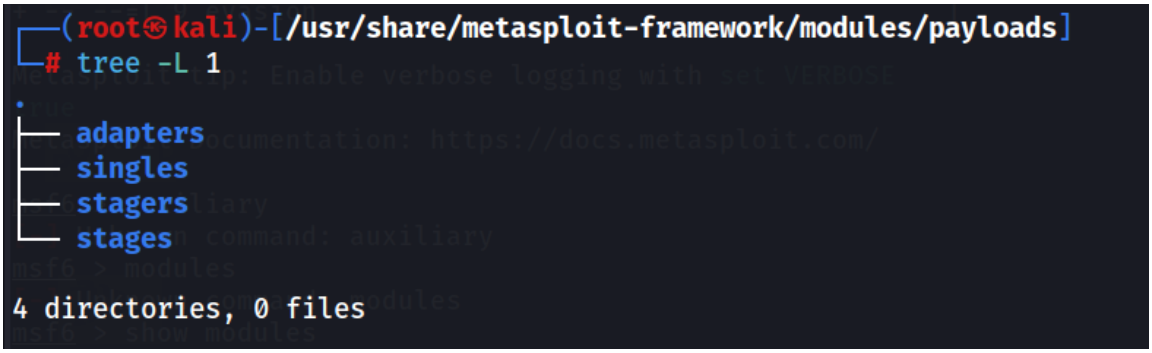
NOPs

- NOPs (No Operation) do nothing.
- They are often used as a buffer to achieve consistent payload sizes.

```
┌──(root㉿kali)-[/usr/share/metasploit-framework/modules]
└─# cd nops && tree -L 1
.
├── aarch64
├── armle
├── cmd
├── mipsbe
├── php
├── ppc
├── sparc
├── tty
├── x64
└── x86

10 directories, 0 files
```

Payloads

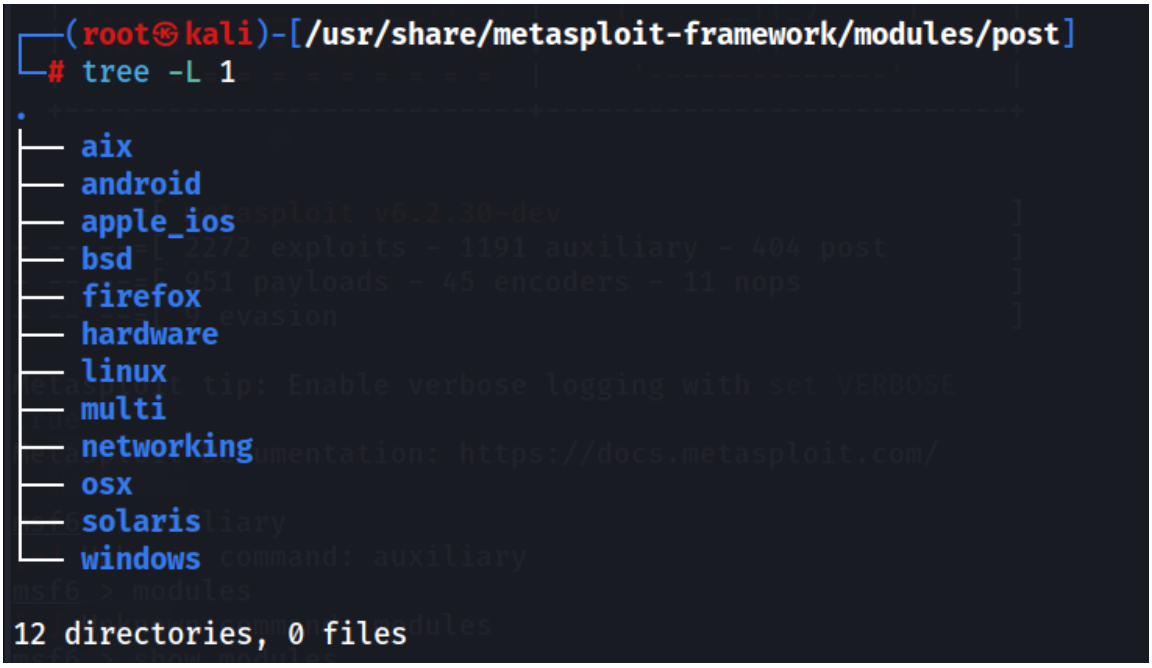
- They are codes that will run on the target system.
- Metasploit offers the ability to send different payloads that can open shells on the target system.



- **Adapters:** An adapter wraps single payloads to convert them into different formats. Example- A normal single payload can be wrapped inside a Powershell adapter, which will make a single powershell command that will execute the payload.
- **Singles:** Self-contained payloads (add user, launch notepad.exe, etc.) that do not need to download an additional component to run.
- **Stagers:** Responsible for setting up a connection channel between Metasploit and the target system. Useful when working with *staged payloads.* Staged Payloads first upload a stager on the target system then download the rest of the payload (stage).
- **Stages:** Downloaded by the stager.
- Way to identify single and staged payloads-
    1. generic/shell_reverse_tcp : It is a single payload as indicated by “_” b/w “shell” and “reverse” .
    2. windows/x64/shell/reverse_tcp: It is a staged payload as indicated by “/ “ b/w “shell” and “reverse” .

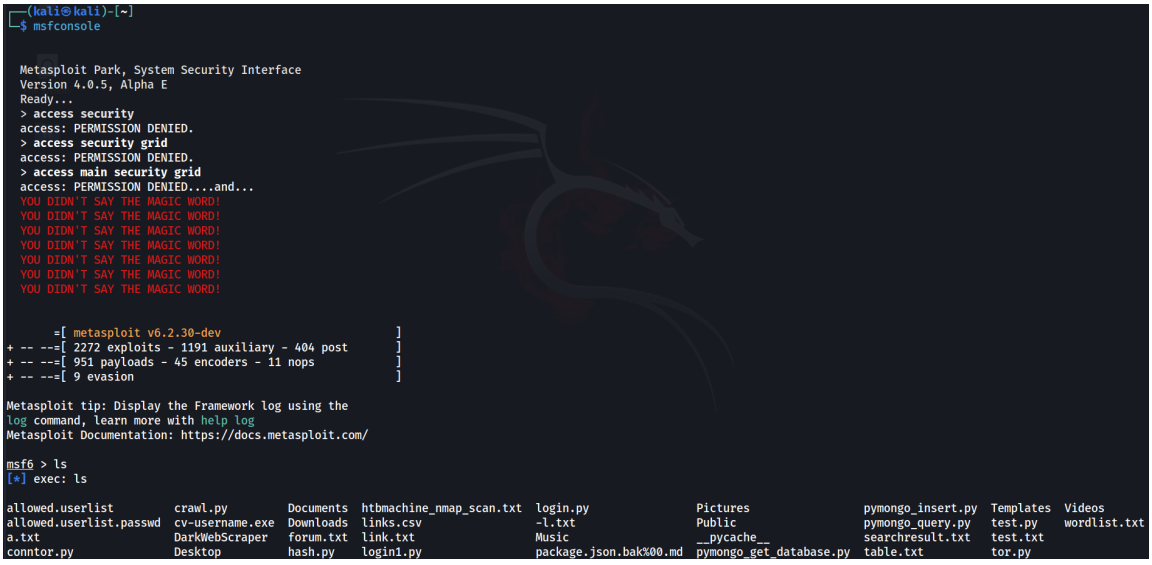
Post

- They are useful on the final stage of the penetration testing process, post-exploitation.



Msfconsole

- The Metasploit console (msfconsole) can be used just like a regular command-line shell.
- For example, on running the command `ls` , it lists the contents of the folder from which Metasploit was launched.



- It supports most linux commands, but does not supports redirection.

```
msf6 > ping -c 2 www.google.com
[*] exec: ping -c 2 www.google.com

PING www.google.com (142.250.206.132) 56(84) bytes of data.
64 bytes from del11s21-in-f4.1e100.net (142.250.206.132): icmp_seq=1 ttl=115 time=30.9 ms
64 bytes from del11s21-in-f4.1e100.net (142.250.206.132): icmp_seq=2 ttl=115 time=30.3 ms

--- www.google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1006ms
rtt min/avg/max/mdev = 30.292/30.586/30.880/0.294 ms
msf6 >
```

- The **help** command can be used on its own or for a specific command.

```
msf6 > help

Core Commands
=============

    Command       Description
    -------       -----------
    ?             Help menu
    banner        Display an awesome metasploit banner
    cd            Change the current working directory
    color         Toggle color
    connect       Communicate with a host
    debug         Display information useful for debugging
    exit          Exit the console
    features      Display the list of not yet released features that can be opted in to
    get           Gets the value of a context-specific variable
    getg          Gets the value of a global variable
    grep          Grep the output of another command
    help          Help menu
    history       Show command history
    load          Load a framework plugin
    quit          Exit the console
    repeat        Repeat a list of commands
    route         Route traffic through a session
    save          Saves the active datastores
    sessions      Dump session listings and display information about sessions
    set           Sets a context-specific variable to a value
    setg          Sets a global variable to a value
    sleep         Do nothing for the specified number of seconds
    spool         Write console output into a file as well the screen
    threads       View and manipulate background threads
    tips          Show a list of useful productivity tips
    unload        Unload a framework plugin
    unset         Unsets one or more context-specific variables
    unsetg        Unsets one or more global variables
    version       Show the framework and console library version numbers
```

```
msf6 > help set
Usage: set [options] [name] [value]

Set the given option to value.  If value is omitted, print the current value.
If both are omitted, print options that are currently set.

If run from a module context, this will set the value in the module's
datastore.  Use -g to operate on the global datastore.

If setting a PAYLOAD, this command can take an index from `show payloads'.
```

- We can also use the `history` command to see commands that we typed earlier.
- In Msfconsole all parameter settings are lost if we change the module we have decided to use.
- If we want to use a module we can use the **use** command with the module name.
- After that we can use the `show options` command to see the options we can set for that particular module.

```
msf6 > use exploit/windows/smb/ms17_010_eternalblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > show options

Module options (exploit/windows/smb/ms17_010_eternalblue):

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   RHOSTS                          yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT          445              yes       The target port (TCP)
   SMBDomain                       no        (Optional) The Windows domain to use for authentication. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Sta
                                             ndard 7 target machines.
   SMBPass                         no        (Optional) The password for the specified username
   SMBUser                         no        (Optional) The username to authenticate as
   VERIFY_ARCH    true             yes       Check if remote architecture matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standar
                                             d 7 target machines.
   VERIFY_TARGET  true             yes       Check if remote OS matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target
                                              machines.


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.0.2.6         yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target
```

- The `show` command can be used in any context followed by a module type (auxiliary, payload, exploit etc.) to list available modules.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > show payloads

Compatible Payloads
===================

   #   Name                                           Disclosure Date  Rank    Check  Description
   -   ----                                           ---------------  ----    -----  -----------
   0   payload/generic/custom                                          normal  No     Custom Payload
   1   payload/generic/shell_bind_tcp                                  normal  No     Generic Command Shell, Bind TCP Inline
   2   payload/generic/shell_reverse_tcp                               normal  No     Generic Command Shell, Reverse TCP Inline
   3   payload/generic/ssh/interact                                    normal  No     Interact with Established SSH Connection
   4   payload/windows/x64/custom/bind_ipv6_tcp                        normal  No     Windows shellcode stage, Windows x64 IPv6 Bind TCP Stager
   5   payload/windows/x64/custom/bind_ipv6_tcp_uuid                   normal  No     Windows shellcode stage, Windows x64 IPv6 Bind TCP Stager with UUID Support
   6   payload/windows/x64/custom/bind_named_pipe                      normal  No     Windows shellcode stage, Windows x64 Bind Named Pipe Stager
   7   payload/windows/x64/custom/bind_tcp                             normal  No     Windows shellcode stage, Windows x64 Bind TCP Stager
   8   payload/windows/x64/custom/bind_tcp_rc4                         normal  No     Windows shellcode stage, Bind TCP Stager (RC4 Stage Encryption, Metasm)
   9   payload/windows/x64/custom/bind_tcp_uuid                        normal  No     Windows shellcode stage, Bind TCP Stager with UUID Support (Windows x64)
   10  payload/windows/x64/custom/reverse_http                         normal  No     Windows shellcode stage, Windows x64 Reverse HTTP Stager (wininet)
   11  payload/windows/x64/custom/reverse_https                        normal  No     Windows shellcode stage, Windows x64 Reverse HTTP Stager (wininet)
   12  payload/windows/x64/custom/reverse_named_pipe                   normal  No     Windows shellcode stage, Windows x64 Reverse Named Pipe (SMB) Stager
   13  payload/windows/x64/custom/reverse_tcp                          normal  No     Windows shellcode stage, Windows x64 Reverse TCP Stager
   14  payload/windows/x64/custom/reverse_tcp_rc4                      normal  No     Windows shellcode stage, Reverse TCP Stager (RC4 Stage Encryption, Metasm)
   15  payload/windows/x64/custom/reverse_tcp_uuid                     normal  No     Windows shellcode stage, Reverse TCP Stager with UUID Support (Windows x64)
   16  payload/windows/x64/custom/reverse_winhttp                      normal  No     Windows shellcode stage, Windows x64 Reverse HTTP Stager (winhttp)
   17  payload/windows/x64/custom/reverse_winhttps                     normal  No     Windows shellcode stage, Windows x64 Reverse HTTPS Stager (winhttp)
   18  payload/windows/x64/exec                                        normal  No     Windows x64 Execute Command
   19  payload/windows/x64/loadlibrary                                 normal  No     Windows x64 LoadLibrary Path
   20  payload/windows/x64/messagebox                                  normal  No     Windows MessageBox x64
   21  payload/windows/x64/meterpreter/bind_ipv6_tcp                   normal  No     Windows Meterpreter (Reflective Injection x64), Windows x64 IPv6 Bind TCP Stager
   22  payload/windows/x64/meterpreter/bind_ipv6_tcp_uuid              normal  No     Windows Meterpreter (Reflective Injection x64), Windows x64 IPv6 Bind TCP Stager
with UUID Support
   23  payload/windows/x64/meterpreter/bind_named_pipe                 normal  No     Windows Meterpreter (Reflective Injection x64), Windows x64 Bind Named Pipe Stage
r
   24  payload/windows/x64/meterpreter/bind_tcp                        normal  No     Windows Meterpreter (Reflective Injection x64), Windows x64 Bind TCP Stager
```

- In the above example, it only lists the payloads that can be used with ms17-010 Eternalblue exploit.
- We can leave the context using the `back` command.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > back
msf6 >
```

- Information on any module can be obtained using the `info` command.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > info

       Name: MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
     Module: exploit/windows/smb/ms17_010_eternalblue
   Platform: Windows
       Arch: x64
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Average
  Disclosed: 2017-03-14

Provided by:
  Equation Group
  Shadow Brokers
  sleepya
  Sean Dillon <sean.dillon@risksense.com>
  Dylan Davis <dylan.davis@risksense.com>
  thelightcosine
  wvu <wvu@metasploit.com>
  agalway-r7
  cdelafuente-r7
  cdelafuente-r7
  agalway-r7

Available targets:
  Id  Name
  --  ----
  0   Automatic Target
  1   Windows 7
  2   Windows Embedded Standard 7
  3   Windows Server 2008 R2
  4   Windows 8
  5   Windows 8.1
  6   Windows Server 2012
  7   Windows 10 Pro
```

- We can also use the **info** command followed by a module's path from the msfconsole prompt.
- Example -

```
msf6 > info exploit/windows/smb/ms17_010_eternalblue

       Name: MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
     Module: exploit/windows/smb/ms17_010_eternalblue
   Platform: Windows
       Arch: x64
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Average
  Disclosed: 2017-03-14

Provided by:
  Equation Group
  Shadow Brokers
  sleepya
  Sean Dillon <sean.dillon@risksense.com>
  Dylan Davis <dylan.davis@risksense.com>
  thelightcosine
  wvu <wvu@metasploit.com>
  agalway-r7
  cdelafuente-r7
  cdelafuente-r7
  agalway-r7

Available targets:
  Id  Name
  --  ----
  0   Automatic Target
  1   Windows 7
```

Search

- We can use the `search` command to search for modules by using CVE numbers, exploit names (eternalblue, heartbleed, etc.)

```
msf6 > search eternalblue

Matching Modules
================

   #  Name                                      Disclosure Date  Rank     Check  Description
   -  ----                                      ---------------  ----     -----  -----------
   0  exploit/windows/smb/ms17_010_eternalblue  2017-03-14       average  Yes    MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
   1  exploit/windows/smb/ms17_010_psexec       2017-03-14       normal   Yes    MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
   2  auxiliary/admin/smb/ms17_010_command      2017-03-14       normal   No     MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
   3  auxiliary/scanner/smb/smb_ms17_010                         normal   No     MS17-010 SMB RCE Detection
   4  exploit/windows/smb/smb_doublepulsar_rce  2017-04-14       great    Yes    SMB DOUBLEPULSAR Remote Code Execution


Interact with a module by name or index. For example info 4, use 4 or use exploit/windows/smb/smb_doublepulsar_rce
```

- Then we can use the module with **use** command followed by the number preceding the module name.

```
msf6 > use 0
[*] Using configured payload windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) >
```

- We can also narrow down our search using keywords such as **type** and **platform.**
- For example -

```
msf6 > search type:exploit eternalblue

Matching Modules
================

   #  Name                                      Disclosure Date  Rank     Check  Description
   -  ----                                      ---------------  ----     -----  -----------
   0  exploit/windows/smb/ms17_010_eternalblue  2017-03-14       average  Yes    MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
   1  exploit/windows/smb/ms17_010_psexec       2017-03-14       normal   Yes    MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
   2  exploit/windows/smb/smb_doublepulsar_rce  2017-04-14       great    Yes    SMB DOUBLEPULSAR Remote Code Execution


Interact with a module by name or index. For example info 2, use 2 or use exploit/windows/smb/smb_doublepulsar_rce
```

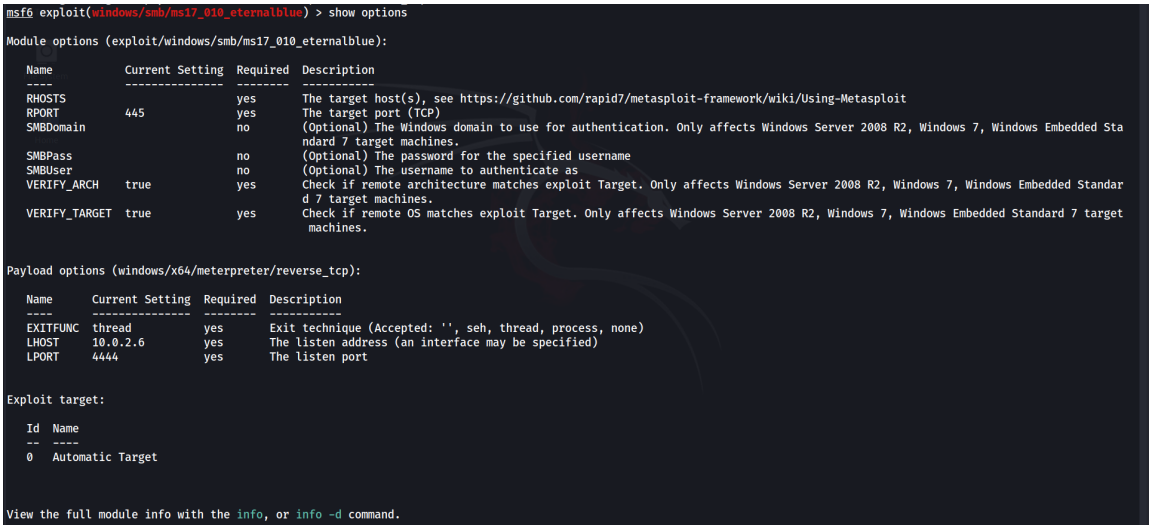
```
msf6 > search platform:windows reverse_tcp

Matching Modules
================

   #    Name                                                                Disclosure Date  Rank       Check  Description
   -    ----                                                                ---------------  ----       -----  -----------
   0    exploit/windows/http/cayin_xpost_sql_rce                            2020-06-04       excellent  Yes    Cayin xPost wayfinder_seqid SQLi to RCE
   1    payload/java/jsp_shell_reverse_tcp                                                   normal     No     Java JSP Command Shell, Reverse TCP Inline
   2    exploit/windows/smb/ms04_007_killbill                               2004-02-10       low        No     MS04-007 Microsoft ASN.1 Library Bitstring Heap Overflow
   3    exploit/windows/imap/novell_netmail_auth                            2007-01-07       average    No     Novell NetMail IMAP AUTHENTICATE Buffer Overflow
   4    exploit/multi/postgres/postgres_copy_from_program_cmd_exec          2019-03-20       excellent  Yes    PostgreSQL COPY FROM PROGRAM Command Execution
   5    payload/cmd/windows/powershell/powershell_reverse_tcp                                normal     No     Powershell Exec
   6    payload/cmd/windows/powershell/powershell_reverse_tcp_ssl                            normal     No     Powershell Exec
   7    payload/cmd/windows/powershell/x64/powershell_reverse_tcp                            normal     No     Powershell Exec
   8    payload/cmd/windows/powershell/x64/powershell_reverse_tcp_ssl                        normal     No     Powershell Exec
   9    payload/cmd/windows/powershell/dllinject/reverse_tcp_allports                        normal     No     Powershell Exec, Reverse All-Port TCP Stager
   10   payload/cmd/windows/powershell/meterpreter/reverse_tcp_allports                      normal     No     Powershell Exec, Reverse All-Port TCP Stager
   11   payload/cmd/windows/powershell/patchupdllinject/reverse_tcp_allports                 normal     No     Powershell Exec, Reverse All-Port TCP Stager
   12   payload/cmd/windows/powershell/patchupmeterpreter/reverse_tcp_allports               normal     No     Powershell Exec, Reverse All-Port TCP Stager
   13   payload/cmd/windows/powershell/peinject/reverse_tcp_allports                         normal     No     Powershell Exec, Reverse All-Port TCP Stager
   14   payload/cmd/windows/powershell/vncinject/reverse_tcp_allports                        normal     No     Powershell Exec, Reverse All-Port TCP Stager
   15   payload/cmd/windows/powershell/dllinject/reverse_tcp                                 normal     No     Powershell Exec, Reverse TCP Stager
   16   payload/cmd/windows/powershell/meterpreter/reverse_tcp                               normal     No     Powershell Exec, Reverse TCP Stager
   17   payload/cmd/windows/powershell/patchupdllinject/reverse_tcp                          normal     No     Powershell Exec, Reverse TCP Stager
   18   payload/cmd/windows/powershell/patchupmeterpreter/reverse_tcp                        normal     No     Powershell Exec, Reverse TCP Stager
   19   payload/cmd/windows/powershell/peinject/reverse_tcp                                  normal     No     Powershell Exec, Reverse TCP Stager
   20   payload/cmd/windows/powershell/vncinject/reverse_tcp                                 normal     No     Powershell Exec, Reverse TCP Stager
   21   payload/cmd/windows/powershell/dllinject/reverse_tcp_dns                             normal     No     Powershell Exec, Reverse TCP Stager (DNS)
   22   payload/cmd/windows/powershell/meterpreter/reverse_tcp_dns                           normal     No     Powershell Exec, Reverse TCP Stager (DNS)
   23   payload/cmd/windows/powershell/patchupdllinject/reverse_tcp_dns                      normal     No     Powershell Exec, Reverse TCP Stager (DNS)
   24   payload/cmd/windows/powershell/patchupmeterpreter/reverse_tcp_dns                    normal     No     Powershell Exec, Reverse TCP Stager (DNS)
   25   payload/cmd/windows/powershell/peinject/reverse_tcp_dns                              normal     No     Powershell Exec, Reverse TCP Stager (DNS)
   26   payload/cmd/windows/powershell/vncinject/reverse_tcp_dns                             normal     No     Powershell Exec, Reverse TCP Stager (DNS)
   27   payload/cmd/windows/powershell/dllinject/reverse_tcp_rc4_dns                         normal     No     Powershell Exec, Reverse TCP Stager (RC4 Stage Encryption
 DNS, Metsrv)
```

- Another useful information returned is the **Rank** column.
- Exploits are based on their reliability.

Ranking	Description
ExcellentRanking	The exploit will never crash the service. This is the case for SQL Injection, CMD execution, RFI, LFI, etc. No typical memory corruption exploits should be given this ranking unless there are extraordinary circumstances (WMF Escape()).
GreatRanking	The exploit has a default target AND either auto-detects the appropriate target or uses an application-specific return address AFTER a version check.
GoodRanking	The exploit has a default target and it is the “common case” for this type of software (English, Windows 7 for a desktop app, 2012 for server, etc). Exploit does not auto-detect the target.
NormalRanking	The exploit is otherwise reliable, but depends on a specific version that is not the “common case” for this type of software and can’t (or doesn’t) reliably autodetect.
AverageRanking	The exploit is generally unreliable or difficult to exploit, but has a success rate of 50% or more for common platforms.
LowRanking	The exploit is nearly impossible to exploit (under 50% success rate) for common platforms.
ManualRanking	The exploit is unstable or difficult to exploit and is basically a DoS (15% success rate or lower). This ranking is also used when the module has no use unless specifically configured by the user (e.g.: exploit/unix/webapp/php_eval).

Working With Modules

- After we enter the context of a module, we will need to set parameters.
- Its a good practice to use the `show options` command to list the required parameters.
- All the parameters are set using the syntax: `set PARAMETER_NAME VALUE`
- The `show options` command will list all available parameters.



- As we can see in the above screenshot, some of these parameters require a value for the exploit to work.
- Parameters that we will often use:
  - **RHOSTS: “**Remote hosts”, the IP address of a target system. They support the CIDR notation, or a network range. We can also use a file, where targets are listed (one target per line).
  - **RPORT: “**Remote port”, the port on the target system the vulnerable application is running on.
  - **PAYLOAD:** The payload we will use with the exploit.
  - **LHOST: “**Localhost” , the attacking machine.
  - **LPORT: “**Local port” , the port that we will use for the reverse shell to connect back to.
  - **SESSION:** Each connection established to the target system using Metasploit will have a session ID.

- We can clear parameters using the `unset` command or clear all set parameters at once using the `unset all` command.
- We can use the `setg` command to set values that will be used for all modules.
- We can clear any value set by `setg` with `unsetg`.
- Once all module parameters are set, we can launch the module using the `exploit` command.
- The `exploit -z` command will run the exploit and background the session as soon as it opens and returns us the context prompt from which we have run the exploit.
- Once a vulnerability has been successfully exploited, a session is created.
- The `sessions` command can be used from the msfconsole prompt or any context to see the existing sessions.
- To interact with any session, we can use the `sessions -i` command followed by the desired session number.