Metasploit - Exploitation - Tryhackme

Introduction

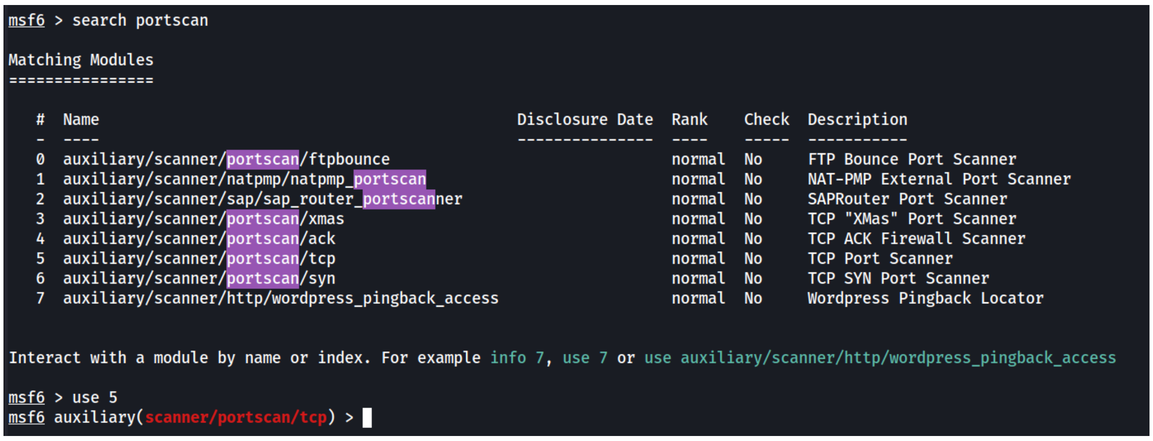
The topics to be covered:

- How to scan target systems using Metasploit.
- How to use Metasploit database feature.
- How to use Metasploit to conduct a vulnerability scan.
- How to use Metasploit to exploit vulnerable services on target systems.
- How `msfvenom` can be used to create payloads and obtain a Meterpreter session on the target system.

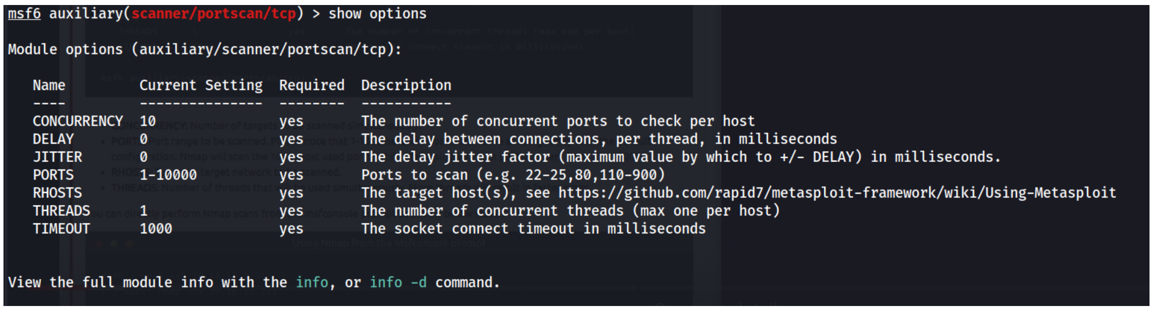
Scanning

Port Scanning

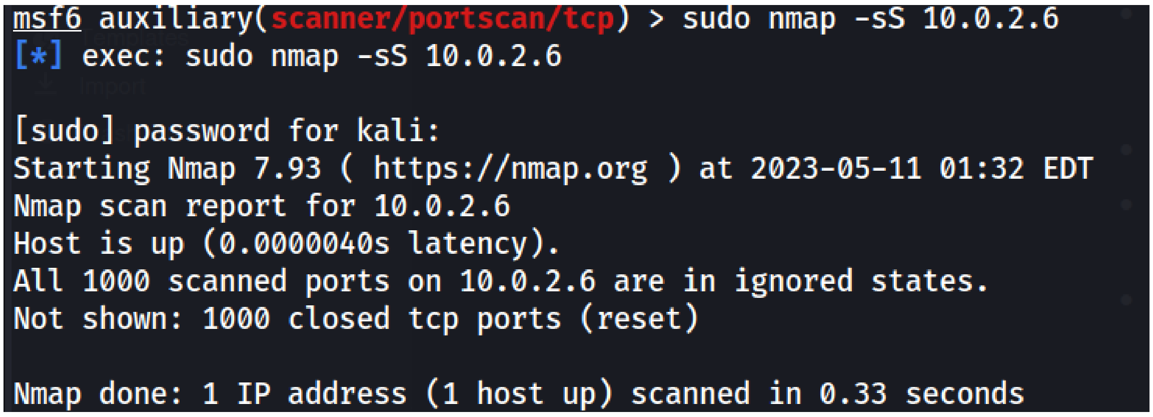
- Metasploit has a number of modules to scan open ports on the target system and network.
- We can list portscanning modules using - `search portscan` command in msfconsole.


```
msf6 > search portscan

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

   #  Name                                              Disclosure Date  Rank    Check  Description
   -  ----                                              ---------------  ----    -----  -----------
   0  auxiliary/scanner/portscan/ftpbounce                                normal  No     FTP Bounce Port Scanner
   1  auxiliary/scanner/natpmp/natpmp_portscan                            normal  No     NAT-PMP External Port Scanner
   2  auxiliary/scanner/sap/sap_router_portscanner                        normal  No     SAPRouter Port Scanner
   3  auxiliary/scanner/portscan/xmas                                     normal  No     TCP "XMas" Port Scanner
   4  auxiliary/scanner/portscan/ack                                      normal  No     TCP ACK Firewall Scanner
   5  auxiliary/scanner/portscan/tcp                                      normal  No     TCP Port Scanner
   6  auxiliary/scanner/portscan/syn                                      normal  No     TCP SYN Port Scanner
   7  auxiliary/scanner/http/wordpress_pingback_access                    normal  No     Wordpress Pingback Locator


Interact with a module by name or index. For example info 7, use 7 or use auxiliary/scanner/http/wordpress_pingback_access

msf6 > use 5
msf6 auxiliary(scanner/portscan/tcp) > 
```

- The module requires setting up a few options.


```
msf6 auxiliary(scanner/portscan/tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   CONCURRENCY  10               yes       The number of concurrent ports to check per host
   DELAY        0                yes       The delay between connections, per thread, in milliseconds
   JITTER       0                yes       The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
   PORTS        1-10000          yes       Ports to scan (e.g. 22-25,80,110-900)
   RHOSTS                        yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   THREADS      1                yes       The number of concurrent threads (max one per host)
   TIMEOUT      1000             yes       The socket connect timeout in milliseconds


View the full module info with the info, or info -d command.
```

- **CONCURRENCY-** Number of targets to be scanned simultaneously.
- **PORTS-** Port range to be scanned. 1-1000 here will not be the same as using Nmap. Nmap scans the top 1000 most used ports whereas Metasploit will scan the port numbers from 1 to 1000.
- **RHOSTS-** Target or target network to be scanned.
- **THREADS-** Number of threads that will be used simultaneously. More threads will result in faster scans.
- We can also perform Nmap scans directly from the msfconsole prompt.

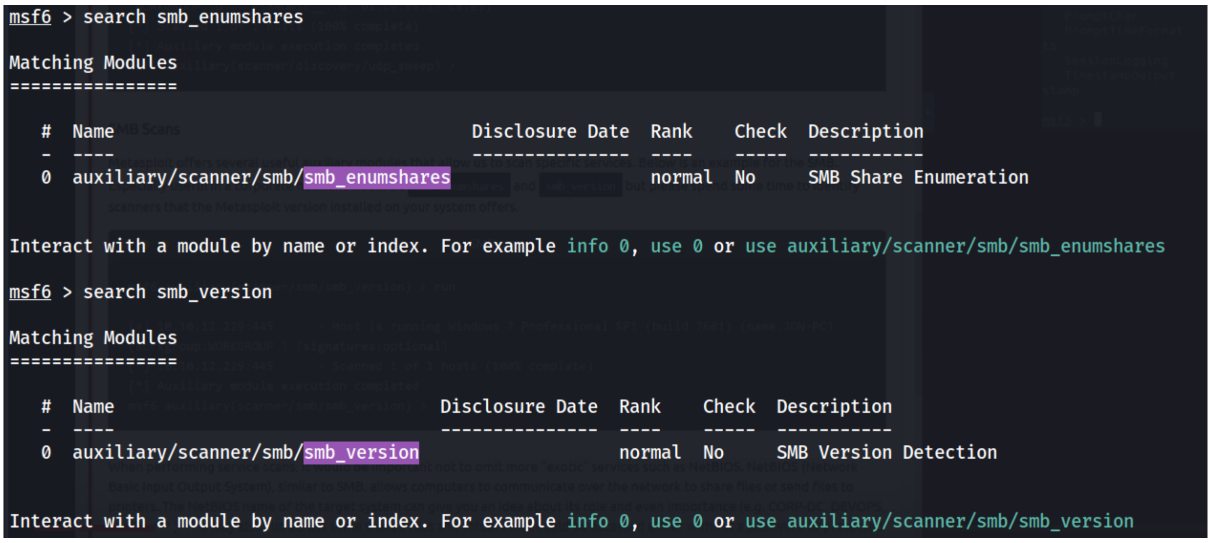

```
msf6 auxiliary(scanner/portscan/tcp) > sudo nmap -sS 10.0.2.6
[*] exec: sudo nmap -sS 10.0.2.6

[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-11 01:32 EDT
Nmap scan report for 10.0.2.6
Host is up (0.0000040s latency).
All 1000 scanned ports on 10.0.2.6 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 1 IP address (1 host up) scanned in 0.33 seconds
```

UDP Service Identification

- The `scanner/discovery/udp_sweep` module allows us to quickly identify services running over UDP.
- This module doesn't conducts an extensive scan of all possible UDP services but does provides a quick way to identify services such as DNS or NetBIOS .

SMB Scans

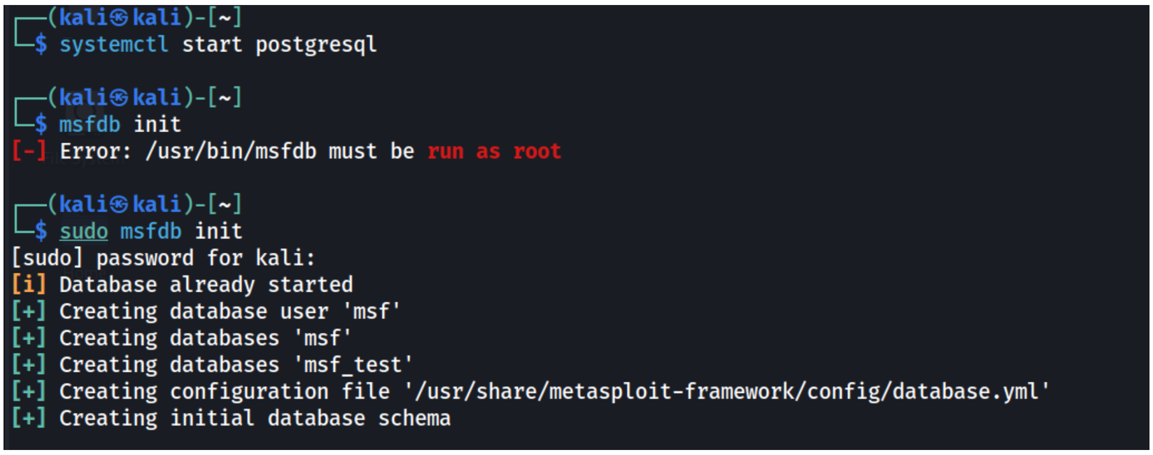
- Especially useful in corporate networks would be `smb_enumshares` and `smb_version`



```
msf6 > search smb_enumshares

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

   #  Name                                   Disclosure Date  Rank    Check  Description
   -  ----                                   ---------------  ----    -----  -----------
   0  auxiliary/scanner/smb/smb_enumshares                    normal  No     SMB Share Enumeration


Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/smb/smb_enumshares

msf6 > search smb_version

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

   #  Name                                Disclosure Date  Rank    Check  Description
   -  ----                                ---------------  ----    -----  -----------
   0  auxiliary/scanner/smb/smb_version                    normal  No     SMB Version Detection


Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/smb/smb_version
```

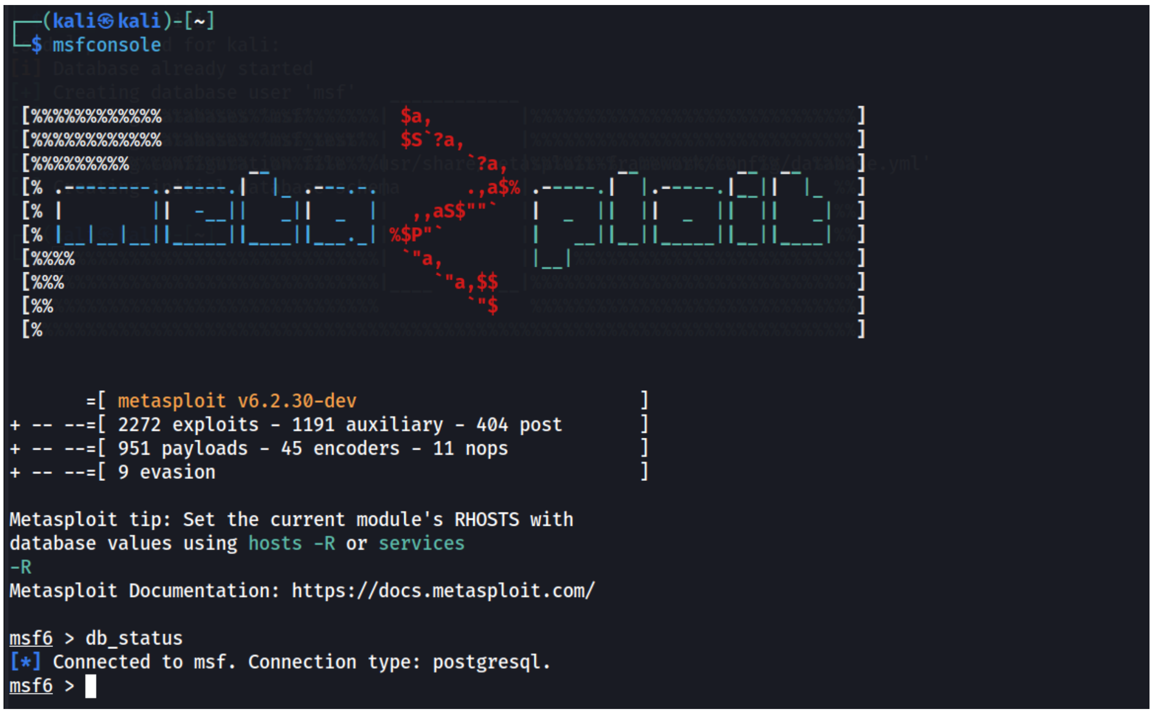
The Metasploit Database

- Metasploit has a database function to simplify project management and avoid possible confusion when setting up parameter values.
- We first need to start the PostgreSQL database, which Metasploit will use with the command - `systemctl start postgresql`
- Then we will need to initialize the Metasploit Database using the command - `msfdb init`



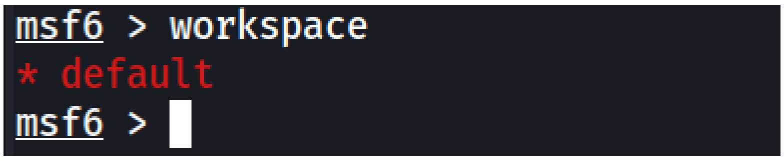
```
┌──(kali㉿kali)-[~]
└─$ systemctl start postgresql

┌──(kali㉿kali)-[~]
└─$ msfdb init
[-] Error: /usr/bin/msfdb must be run as root

┌──(kali㉿kali)-[~]
└─$ sudo msfdb init
[sudo] password for kali:
[i] Database already started
[+] Creating database user 'msf'
[+] Creating databases 'msf'
[+] Creating databases 'msf_test'
[+] Creating configuration file '/usr/share/metasploit-framework/config/database.yml'
[+] Creating initial database schema
```

- We can now launch `msfconsole` and check the database status using the command - `db_status`



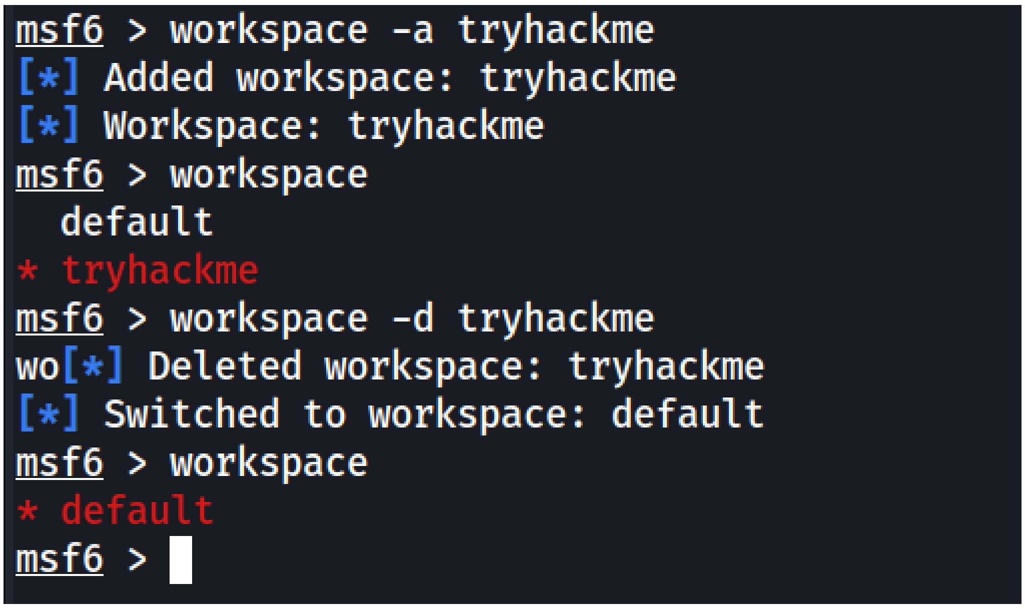
```
┌──(kali㉿kali)-[~]
└─$ msfconsole


       =[ metasploit v6.2.30-dev                          ]
+ -- --=[ 2272 exploits - 1191 auxiliary - 404 post       ]
+ -- --=[ 951 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Set the current module's RHOSTS with
database values using hosts -R or services
-R
Metasploit Documentation: https://docs.metasploit.com/

msf6 > db_status
[*] Connected to msf. Connection type: postgresql.
msf6 >
```

- The database feature allows us to create workspaces to isolate different projects.
- When first launched we will be in the default workspace.
- We can list the workspace using the command - `workspace`



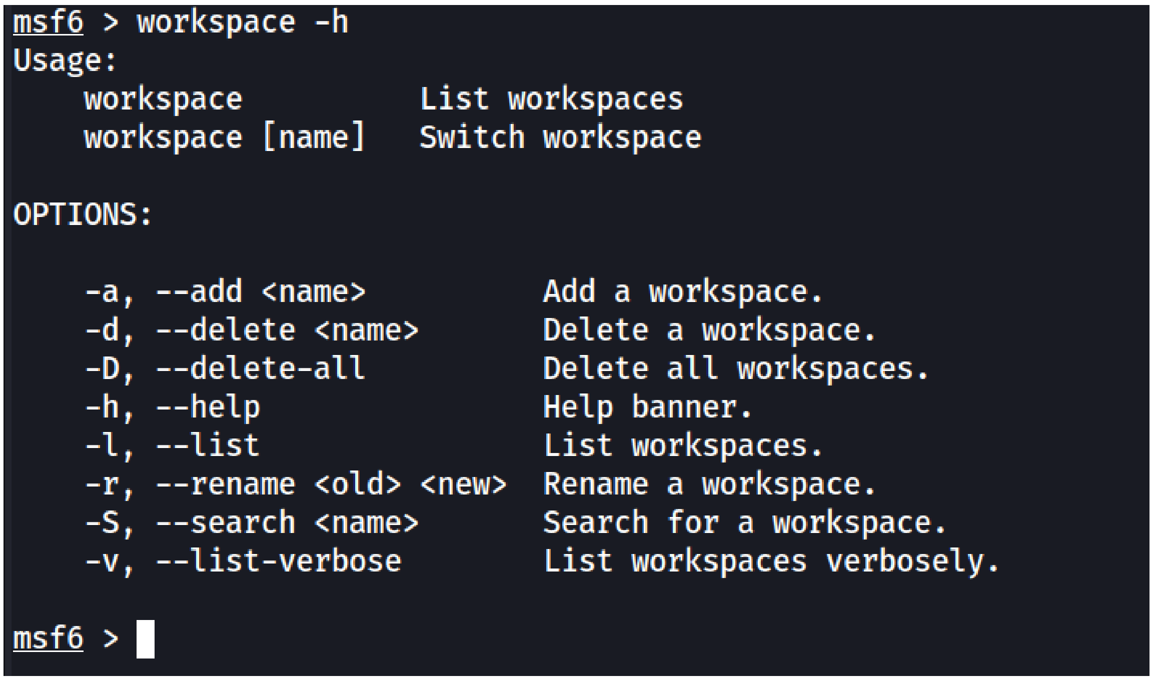
```
msf6 > workspace
* default
msf6 >
```

- We can add a workspace using the `-a` parameter.
- We can delete a workspace using the `-d` parameter.



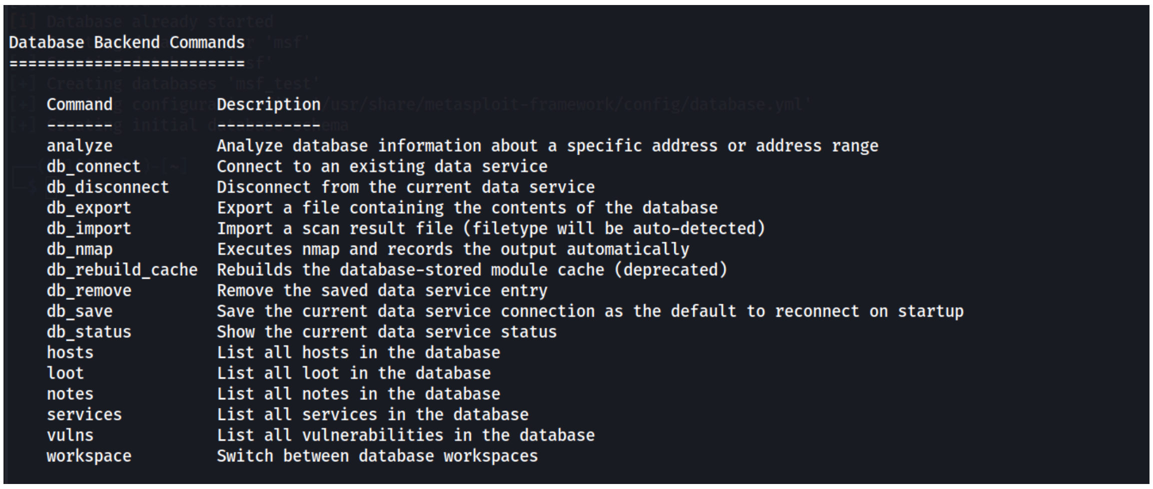
```
msf6 > workspace -a tryhackme
[*] Added workspace: tryhackme
[*] Workspace: tryhackme
msf6 > workspace
  default
* tryhackme
msf6 > workspace -d tryhackme
wo[*] Deleted workspace: tryhackme
[*] Switched to workspace: default
msf6 > workspace
* default
msf6 > 
```

- A new database name is printed in red starting with the `*` symbol.
- We can navigate b/w workspaces using the `workspace` command followed by the name of the desired workspace.
- We can use the command - `workspace -h` to list the available options for the `workspace` command.



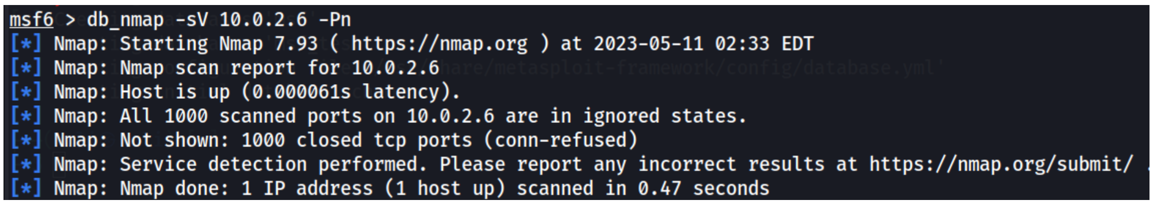
```
msf6 > workspace -h
Usage:
    workspace          List workspaces
    workspace [name]   Switch workspace

OPTIONS:

    -a, --add <name>          Add a workspace.
    -d, --delete <name>       Delete a workspace.
    -D, --delete-all          Delete all workspaces.
    -h, --help                Help banner.
    -l, --list                List workspaces.
    -r, --rename <old> <new>  Rename a workspace.
    -S, --search <name>       Search for a workspace.
    -v, --list-verbose        List workspaces verbosely.

msf6 > 
```

- Once Metasploit is launched with a database, the `help` command, will show the Database Backend Commands Menu.



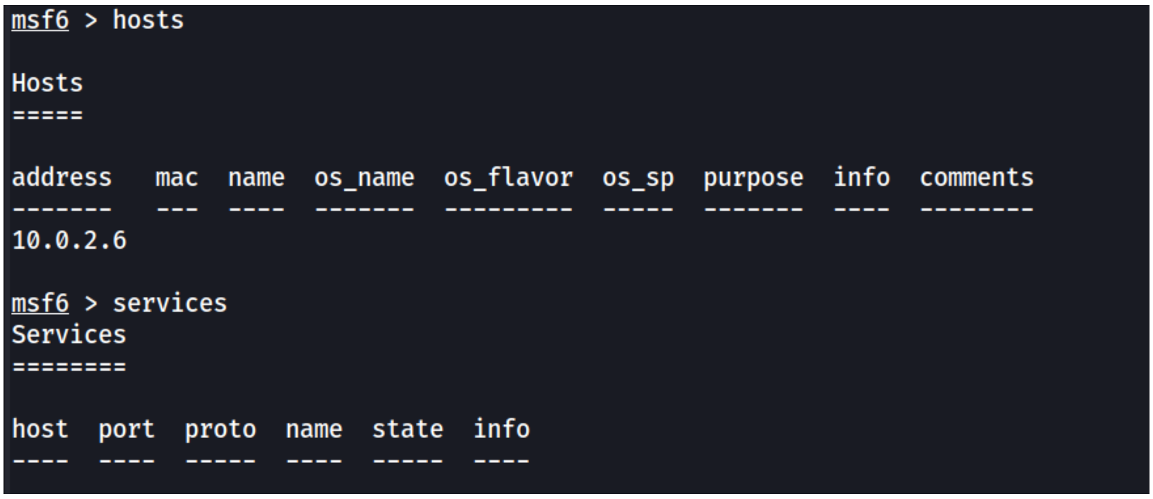
```
Database Backend Commands
=========================

    Command           Description
    -------           -----------
    analyze           Analyze database information about a specific address or address range
    db_connect        Connect to an existing data service
    db_disconnect     Disconnect from the current data service
    db_export         Export a file containing the contents of the database
    db_import         Import a scan result file (filetype will be auto-detected)
    db_nmap           Executes nmap and records the output automatically
    db_rebuild_cache  Rebuilds the database-stored module cache (deprecated)
    db_remove         Remove the saved data service entry
    db_save           Save the current data service connection as the default to reconnect on startup
    db_status         Show the current data service status
    hosts             List all hosts in the database
    loot              List all loot in the database
    notes             List all notes in the database
    services          List all services in the database
    vulns             List all vulnerabilities in the database
    workspace         Switch between database workspaces
```

- If we run a nmap scan using the `db_nmap` command, all the results will be stored in the database.



```
msf6 > db_nmap -sV 10.0.2.6 -Pn
[*] Nmap: Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-11 02:33 EDT
[*] Nmap: Nmap scan report for 10.0.2.6
[*] Nmap: Host is up (0.000061s latency).
[*] Nmap: All 1000 scanned ports on 10.0.2.6 are in ignored states.
[*] Nmap: Not shown: 1000 closed tcp ports (conn-refused)
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds
```

- We can reach information relevant to hosts and services running on the target systems with the `hosts` and `services` commands.



```
msf6 > hosts

Hosts
=====

address   mac  name  os_name  os_flavor  os_sp  purpose  info  comments
-------   ---  ----  -------  ---------  -----  -------  ----  --------
10.0.2.6

msf6 > services
Services
========

host  port  proto  name  state  info
----  ----  -----  ----  -----  ----
```

- We can use `hosts -h` and `services -h` command to get to the help menu of these commands.

- Once the host information is stored in the database, we can use `hosts -R` command to add this value to the RHOSTS parameter.
- If there is more than one host saved to the database, all IP addresses will be used when the `hosts -R` command is executed.
- The service command used with the `-S` parameter allows us to search for specific services in the environment. For example - `services -S netbios`
- We may want to look for:
  - **HTTP -** Could potentially host a web application where we can find vulnerabilities like SQL injection or Remote Code Execution (RCE).
  - **FTP -** Could allow anonymous login and provide access to interesting files.
  - **SMB-** Could be vulnerable to SMB exploits like MS17-010.
  - **SSH -** Could have default or easy to guess credentials.
  - **RDP -** Could be vulnerable to Bluekeep or allow desktop access if weak credentials were used.

Vulnerability Scanning

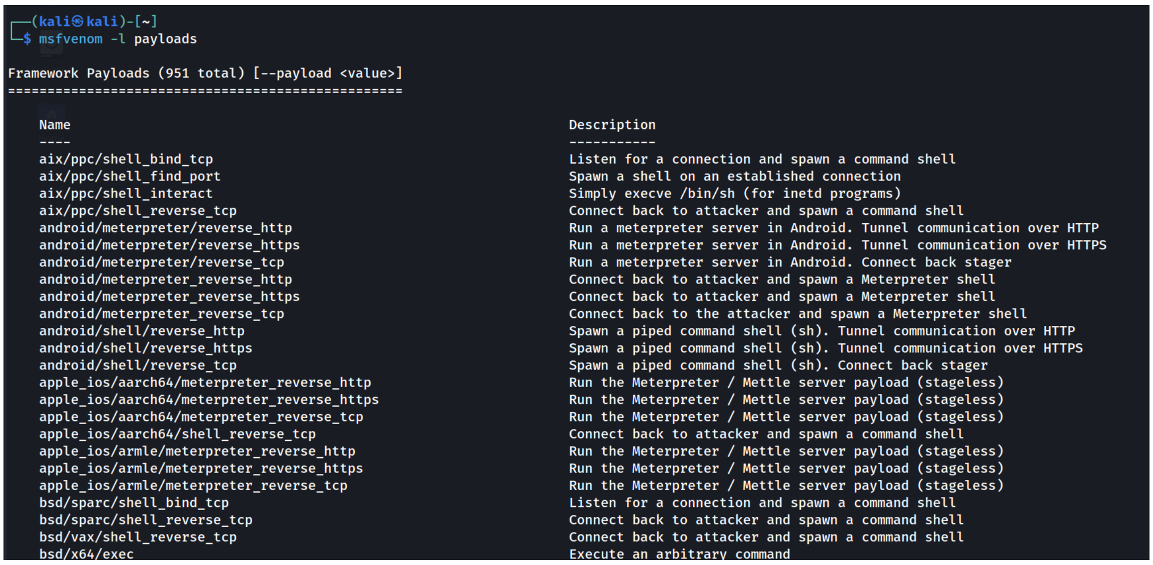
- Metasploit allows us to quickly identify some critical vulnerabilities that could be considered as “low hanging fruit”.
- The term “low hanging fruit” refers to the easily identifiable and exploitable vulnerabilities that could potentially allow us to gain a foothold on the system.
- We can use the `info` command for any module to have a better understanding of its use and purpose.

Exploitation

- Exploits are the most populated module category in Metasploit.
- We can search for an exploit, use that exploit and set payloads to be used with that particular exploit.
- Some payloads open new parameters that we may need to set, running the `show options` command once more can show these. For example, A reverse payload will at lease require us the set the LHOST option.

MsfVenom

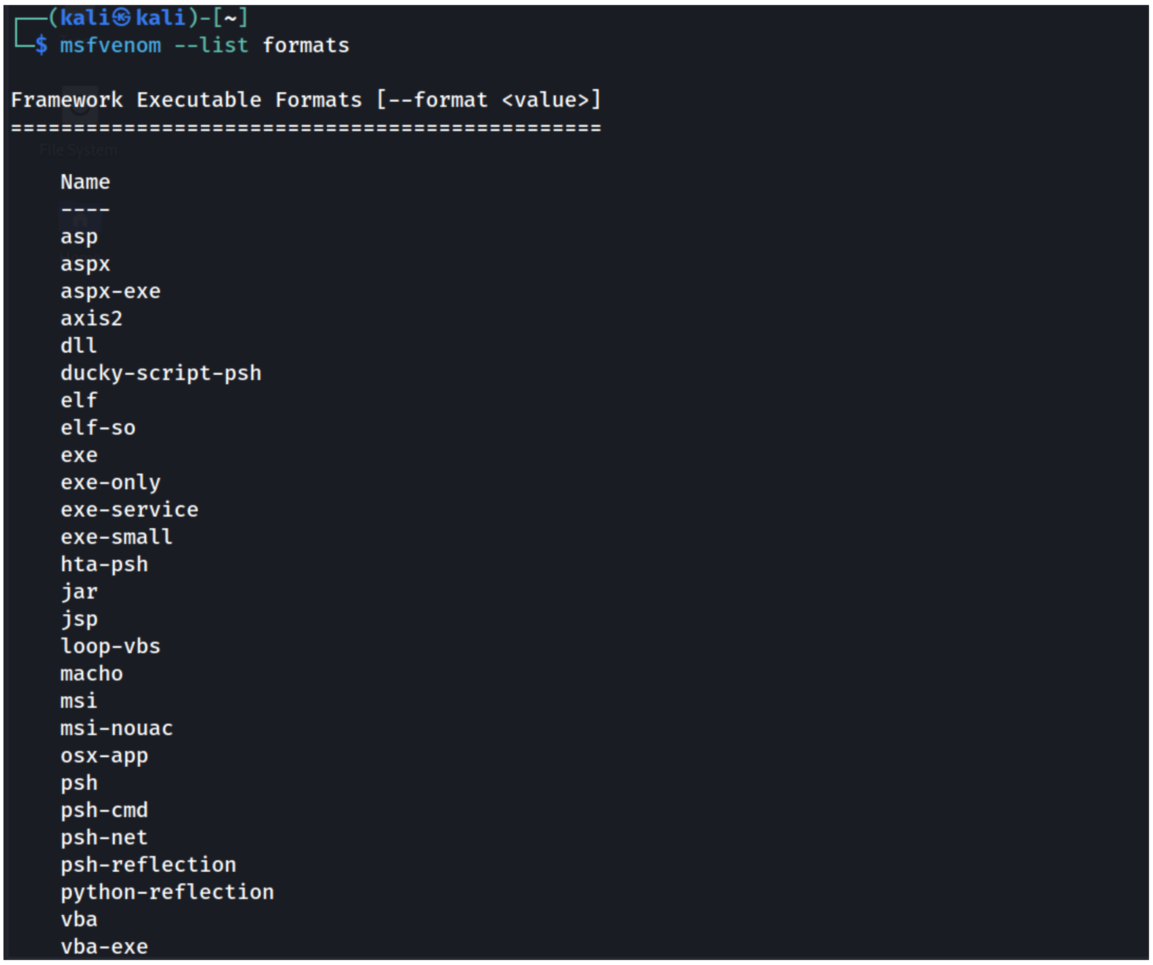
- Msfvenom allows us to access all payloads available in the Metasploit Framework.
- It allows us to create payloads in many different formats (PHP, exe, dll, elf, etc.) and for many different target systems (Apple, Windows, Android, Linux etc.)



```
┌──(kali㉿kali)-[~]
└─$ msfvenom -l payloads

Framework Payloads (951 total) [--payload <value>]
==================================================

    Name                                                Description
    ----                                                -----------
    aix/ppc/shell_bind_tcp                              Listen for a connection and spawn a command shell
    aix/ppc/shell_find_port                             Spawn a shell on an established connection
    aix/ppc/shell_interact                              Simply execve /bin/sh (for inetd programs)
    aix/ppc/shell_reverse_tcp                           Connect back to attacker and spawn a command shell
    android/meterpreter/reverse_http                    Run a meterpreter server in Android. Tunnel communication over HTTP
    android/meterpreter/reverse_https                   Run a meterpreter server in Android. Tunnel communication over HTTPS
    android/meterpreter/reverse_tcp                     Run a meterpreter server in Android. Connect back stager
    android/meterpreter_reverse_http                    Connect back to attacker and spawn a Meterpreter shell
    android/meterpreter_reverse_https                   Connect back to attacker and spawn a Meterpreter shell
    android/meterpreter_reverse_tcp                     Connect back to the attacker and spawn a Meterpreter shell
    android/shell/reverse_http                          Spawn a piped command shell (sh). Tunnel communication over HTTP
    android/shell/reverse_https                         Spawn a piped command shell (sh). Tunnel communication over HTTPS
    android/shell/reverse_tcp                           Spawn a piped command shell (sh). Connect back stager
    apple_ios/aarch64/meterpreter_reverse_http          Run the Meterpreter / Mettle server payload (stageless)
    apple_ios/aarch64/meterpreter_reverse_https         Run the Meterpreter / Mettle server payload (stageless)
    apple_ios/aarch64/meterpreter_reverse_tcp           Run the Meterpreter / Mettle server payload (stageless)
    apple_ios/aarch64/shell_reverse_tcp                 Connect back to attacker and spawn a command shell
    apple_ios/armle/meterpreter_reverse_http            Run the Meterpreter / Mettle server payload (stageless)
    apple_ios/armle/meterpreter_reverse_https           Run the Meterpreter / Mettle server payload (stageless)
    apple_ios/armle/meterpreter_reverse_tcp             Run the Meterpreter / Mettle server payload (stageless)
    bsd/sparc/shell_bind_tcp                            Listen for a connection and spawn a command shell
    bsd/sparc/shell_reverse_tcp                         Connect back to attacker and spawn a command shell
    bsd/vax/shell_reverse_tcp                           Connect back to attacker and spawn a command shell
    bsd/x64/exec                                        Execute an arbitrary command
```

Output formats

- We can either generate stand-alone payloads (e.g. a Windows executable for Meterpreter) or get a usable raw format (e.g. Python).
- The `msfvenom --list formats` command can be used to list supported output formats.



```
┌──(kali㉿kali)-[~]
└─$ msfvenom --list formats

Framework Executable Formats [--format <value>]
===============================================

    Name
    ----
    asp
    aspx
    aspx-exe
    axis2
    dll
    ducky-script-psh
    elf
    elf-so
    exe
    exe-only
    exe-service
    exe-small
    hta-psh
    jar
    jsp
    loop-vbs
    macho
    msi
    msi-nouac
    osx-app
    psh
    psh-cmd
    psh-net
    psh-reflection
    python-reflection
    vba
    vba-exe
```

Encoders

- Encoders do not aim to bypass antivirus installed on the target system.
- They encode the payload, while it can be effective against some antivirus software.