Tools

MANY IT SECURITY TOOLS
DURATION : 0’30

WEBSITE

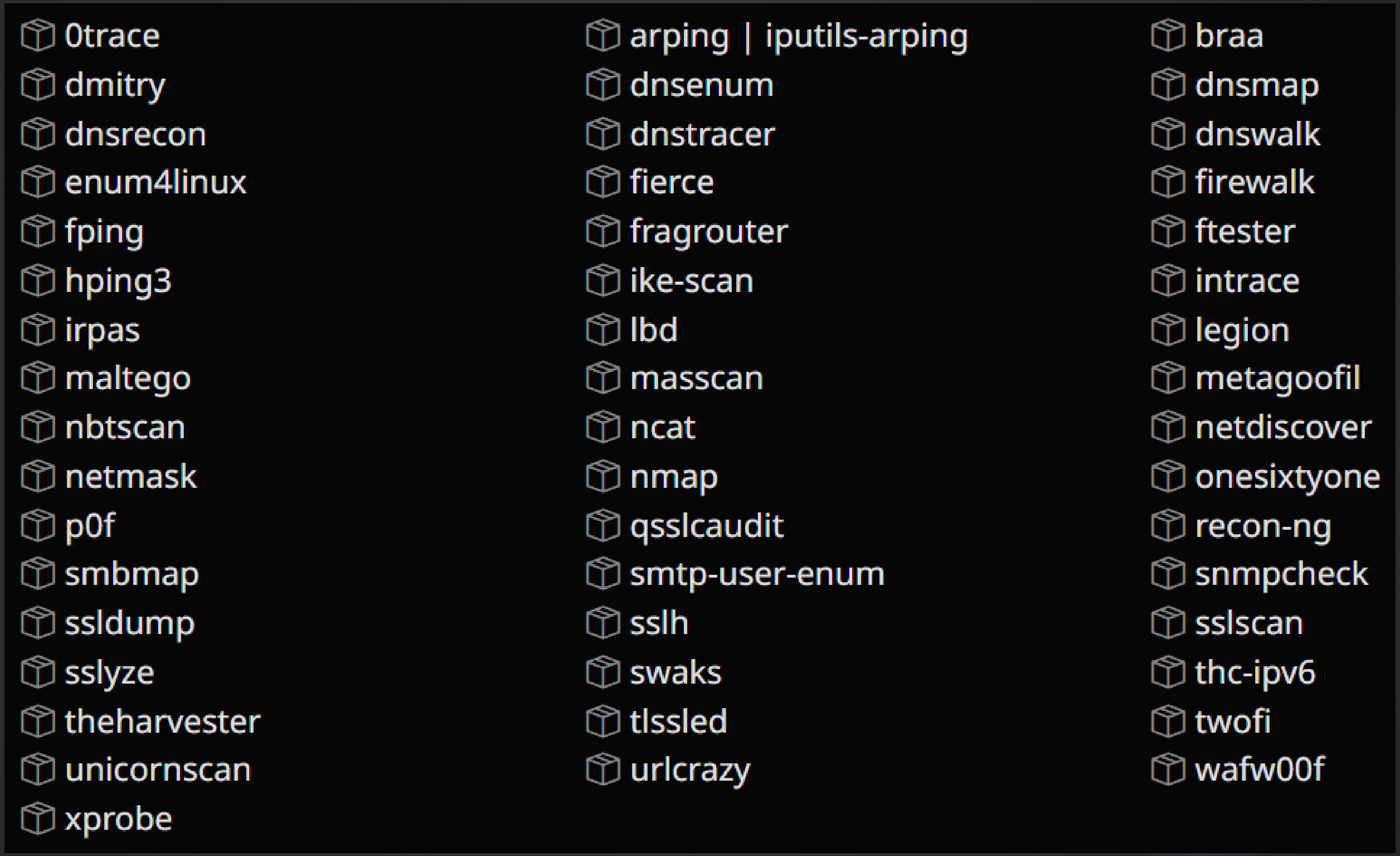
HELP TO FIND VULNERABILITIES

Vulmon

- Vulmon is a vulnerability search engine with vulnerability intelligence features.
- **Search Examples:**
  - Query for latest Apache Tomcat vulnerabilities: https://vulmon.com/searchpage?q=apache+tomcat&sortby=bydate
  - Query for latest Apache Tomcat 8.5.10 vulnerabilities: https://vulmon.com/searchpage?q=Apache+Tomcat+8.5.10&sortby=bydate
  - Query for latest Apache Tomcat XSS vulnerabilities: https://vulmon.com/searchpage?q=apache+tomcat+xss&sortby=bydate
  - Query for Apache Tomcat vulnerabilities sorted by risk score (high to low): https://vulmon.com/searchpage?q=apache+tomcat&sortby=byriskscore
  - Query for latest sql injection vulnerabilities: https://vulmon.com/searchpage?q=sql+injection&sortby=bydate

TOOLS : Information Gathering

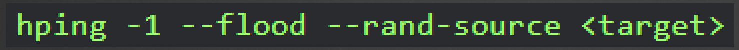
TOOLS : Info Gathering LIST



- 0trace
- arping | iputils-arping
- braa
- dmitry
- dnsenum
- dnsmap
- dnsrecon
- dnstracer
- dnswalk
- enum4linux
- fierce
- firewalk
- fping
- fragrouter
- ftester
- hping3
- ike-scan
- intrace
- irpas
- lbd
- legion
- maltego
- masscan
- metagoofil
- nbtscan
- ncat
- netdiscover
- netmask
- nmap
- onesixtyone
- p0f
- qsslcaudit
- recon-ng
- smbmap
- smtp-user-enum
- snmpcheck
- ssldump
- sslh
- sslscan
- sslyze
- swaks
- thc-ipv6
- theharvester
- tlssled
- twofi
- unicornscan
- urlcrazy
- wafw00f
- xprobe

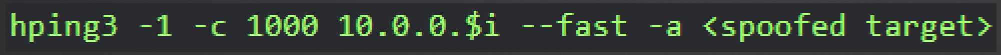
hping3

- hping is a command-line oriented TCP/IP packet assembler/analyzer.
- The interface is inspired to the ping, but hping isn't only able to send ICMP echo requests.
- It supports TCP, UDP, ICMP and RAW-IP protocols, has a traceroute mode, the ability to send files between a covered channel, and many other features.
- Example:
  - ICMP Flood :



```
hping -1 --flood --rand-source <target>
```

  - Smurf attack (DDOS by spoofing IP address):

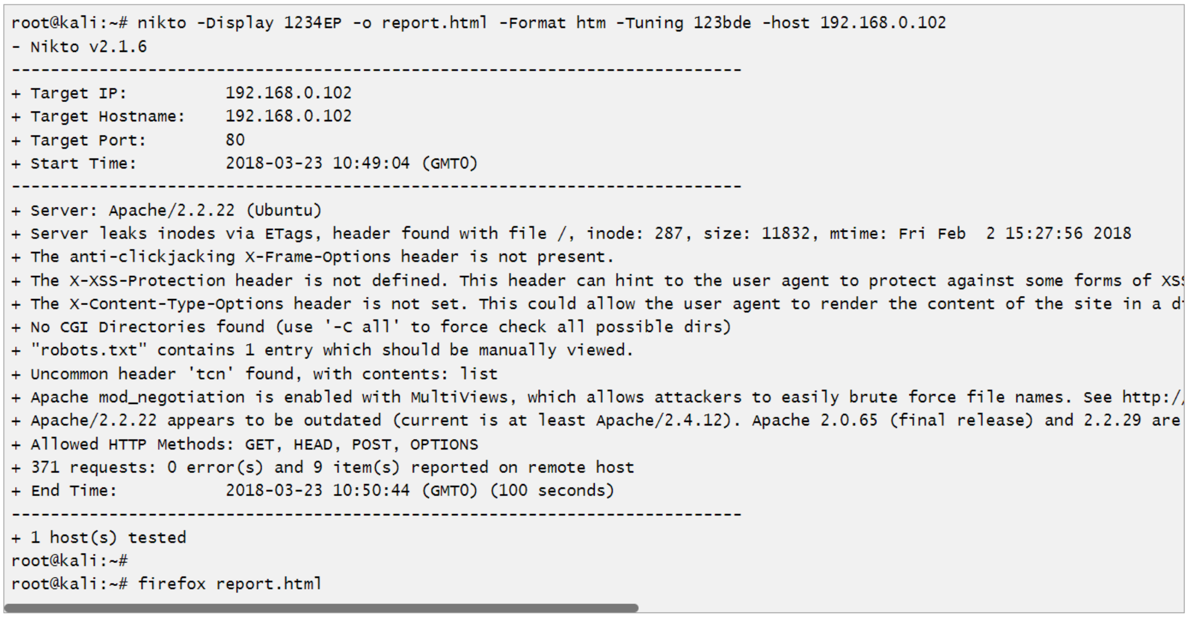


```
hping3 -1 -c 1000 10.0.0.$i --fast -a <spoofed target>
```

- **More information:** https://tools.kali.org/information-gathering/hping3

Nikto

- Nikto is a web server scanner which performs tests against web servers for multiple items:
  - including potentially dangerous files / programs;
  - checks for outdated versions of over many servers and version specific problems on over many servers.
- It also checks for server configuration items such as the presence of multiple index files, HTTP server options, and will attempt to identify installed web servers and software.
- Scan items and plugins are frequently updated and can be automatically updated.

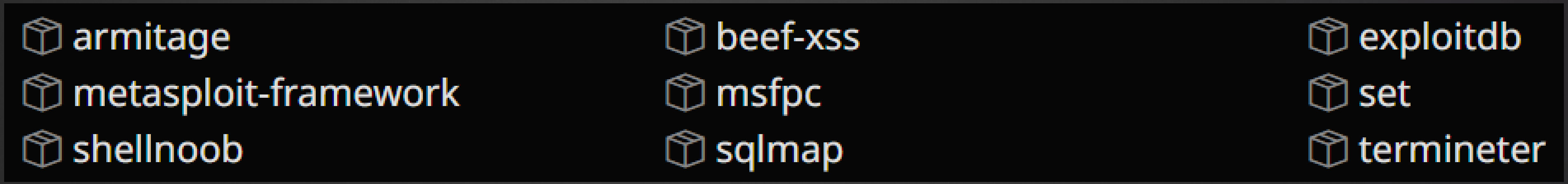


```
root@kali:~# nikto -Display 1234EP -o report.html -Format htm -Tuning 123bde -host 192.168.0.102
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.0.102
+ Target Hostname:    192.168.0.102
+ Target Port:        80
+ Start Time:         2018-03-23 10:49:04 (GMT0)
---------------------------------------------------------------------------
+ Server: Apache/2.2.22 (Ubuntu)
+ Server leaks inodes via ETags, header found with file /, inode: 287, size: 11832, mtime: Fri Feb  2 15:27:56 2018
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a d
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ "robots.txt" contains 1 entry which should be manually viewed.
+ Uncommon header 'tcn' found, with contents: list
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. See http:/
+ Apache/2.2.22 appears to be outdated (current is at least Apache/2.4.12). Apache 2.0.65 (final release) and 2.2.29 are
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ 371 requests: 0 error(s) and 9 item(s) reported on remote host
+ End Time:           2018-03-23 10:50:44 (GMT0) (100 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
root@kali:~#
root@kali:~# firefox report.html
```

- **More information: https://tools.kali.org/information-gathering/nikto**

TOOLS : Exploitation

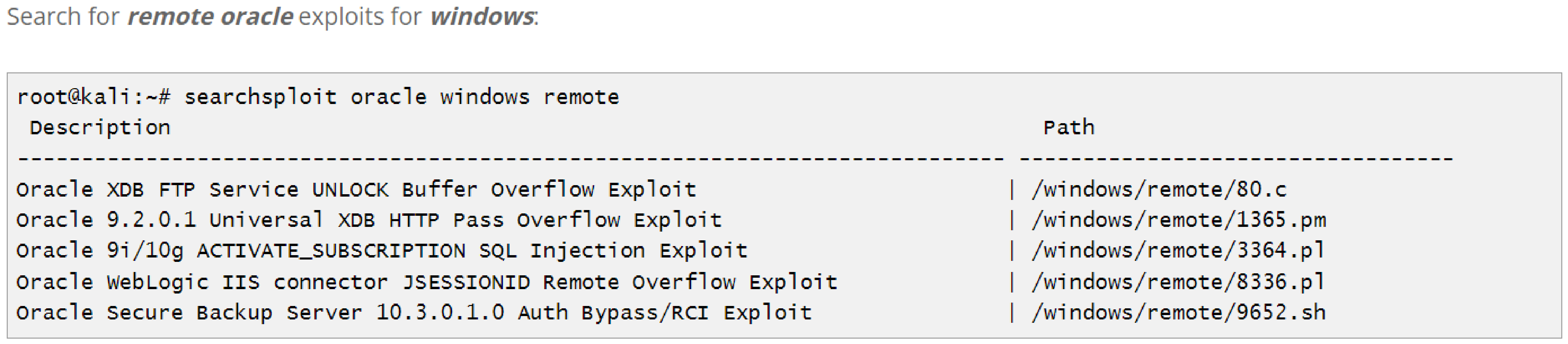
TOOLS : Exploitation LIST



armitage
beef-xss
exploitdb
metasploit-framework
msfpc
set
shellnoob
sqlmap
termineter

ExploitDB : Searchsploit

- Searchsploit is an utility to search an exploit in a database.



Search for ***remote oracle*** exploits for ***windows***:

```
root@kali:~# searchsploit oracle windows remote
 Description                                                                    Path
----------------------------------------------------------------------------- ----------------------------------
Oracle XDB FTP Service UNLOCK Buffer Overflow Exploit                        | /windows/remote/80.c
Oracle 9.2.0.1 Universal XDB HTTP Pass Overflow Exploit                      | /windows/remote/1365.pm
Oracle 9i/10g ACTIVATE_SUBSCRIPTION SQL Injection Exploit                    | /windows/remote/3364.pl
Oracle WebLogic IIS connector JSESSIONID Remote Overflow Exploit             | /windows/remote/8336.pl
Oracle Secure Backup Server 10.3.0.1.0 Auth Bypass/RCI Exploit               | /windows/remote/9652.sh
```

- **More information:**
  - https://tools.kali.org/exploitation-tools/exploitdb
  - https://www.exploit-db.com/searchsploit

Metasploit

- Metasploit is a complete penetration testing framework to perform various simple and complex tasks.
- Unlike many tools Metasploit can perform multiple tasks throughout the penetration testing like cycle:
  - *Information gathering*
  - *Enumeration*
  - *Gaining access*
  - *Privilege escalation*
  - *Maintening access*
  - *Covering tracks*
- **More information:**
  - https://www.metasploit.com

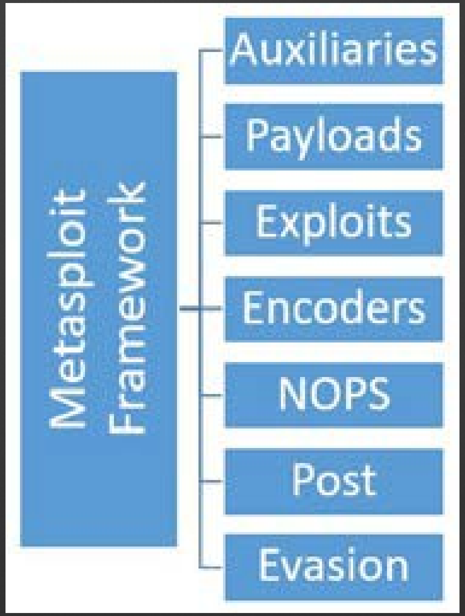
Metasploit

- Metasploit composents and modules.

Penetration testing phase	Use of Metasploit
Information gathering	**Auxiliary/*/ :** portscan/syn, portscan/tcp, smb_version, db_nmap, scanner/ftp/ftp_version, gather/shodan_search
Enumeration	**Auxiliary/scanner/ :** smb/smb_enumshares, smb/smb_enumusers, smb/smb_login
Gaining access	All Metasploit exploits and payloads
Privilege escalation	**meterpreter >** use priv **meterpreter >** getsystem
Maintening access	**meterpreter >** run persistence
Covering tracks	Metasploit Anti-Forensics Project

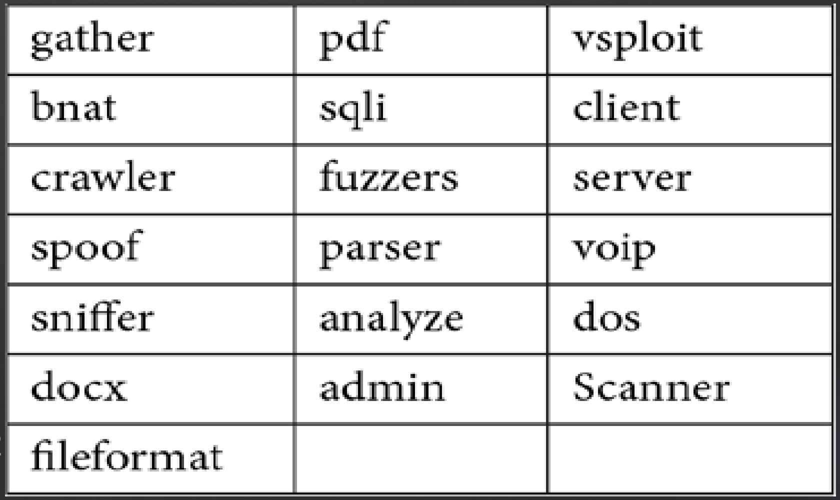
Metasploit : Anatomy and structure

- The simplest method to learn the structure of Metasploit Framework is to browse and explore directory. In Kali Metasploit can be located at /usr/share/metasploit-framework
- The Metasploit Framework structure is as shown in the following screenshot:



Metasploit : Auxiliaries

- Auxiliary modules in the Metasploit Framework are nothing but small pieces of code that are meant to perform a specific task.
- For example, you might need to perform a simple task of verifying whether the FTP servers allow anonymous access.
- Such tasks can be very easily accomplished using the auxiliary modules present in the Metasploit Framework. There are more than 1,000 auxiliary modules spread across 19 categories in the Metasploit Framework.
- The following table shows various categories of auxiliary modules present in the Metasploit Framework:

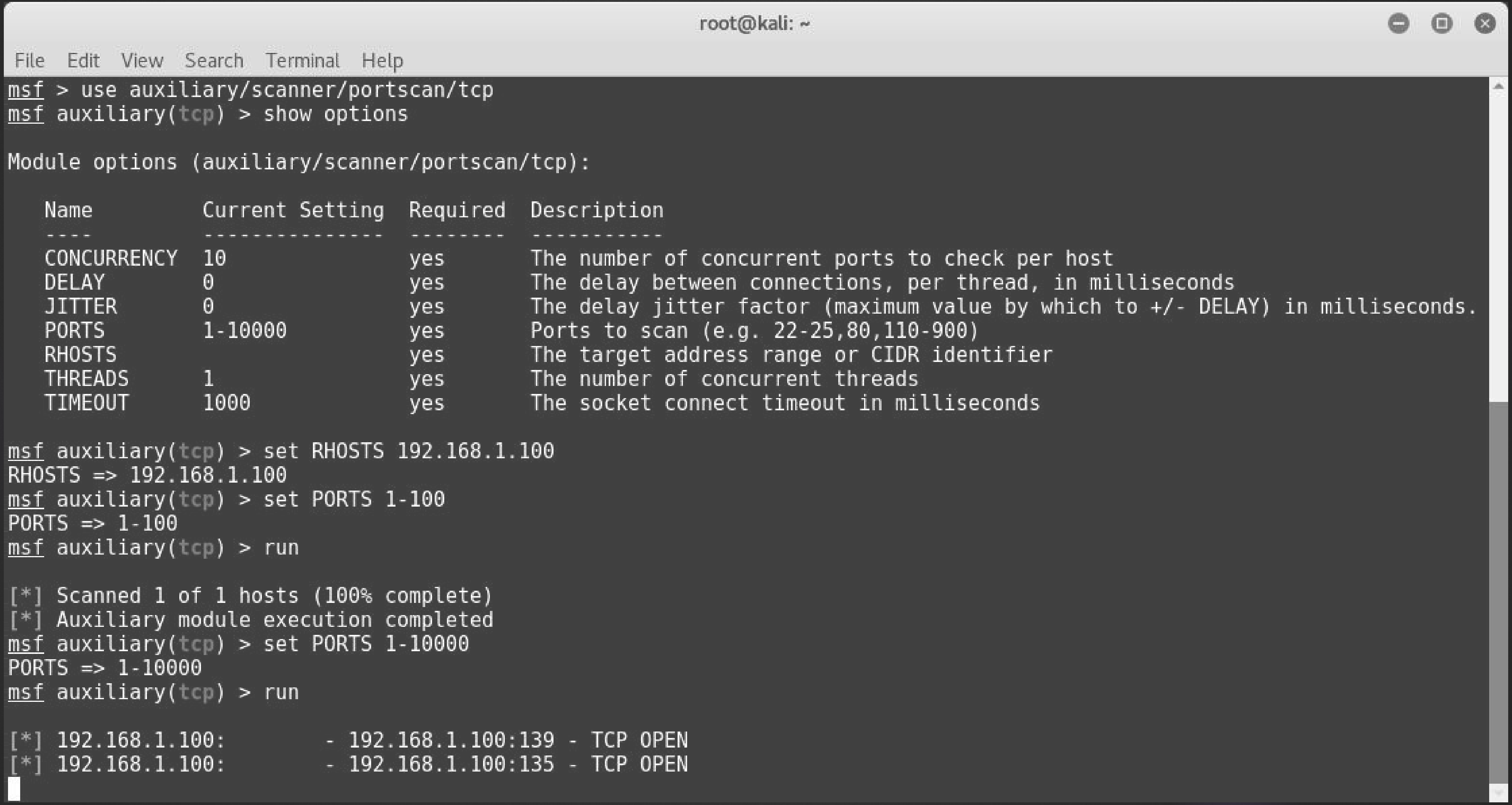


gather	pdf	vsploit
bnat	sqli	client
crawler	fuzzers	server
spoof	parser	voip
sniffer	analyze	dos
docx	admin	Scanner
fileformat		

Metasploit : Auxiliaries

- You may not need to know each and every module individually.
- You just need to search for the right module in the required context and use it accordingly.
- Example:
  1. Open up a terminal window and start Metasploit using the msfconsole command.
  2. Select the portscan/tcp auxiliary module to perform a port scan against a target system.
  3. Using the show command, list all the parameters that need to be configured in order to run this auxiliary module.
  4. Using the set RHOSTS command, set the IP address of our target system.
  5. Using the set PORTS command, select the port range you want to scan on your target system.
  6. Using the run command, execute the auxiliary module with the parameters configured earlier.
  7. You can see the use of all the previously mentioned commands in the following screenshot:

Metasploit : Auxiliaries



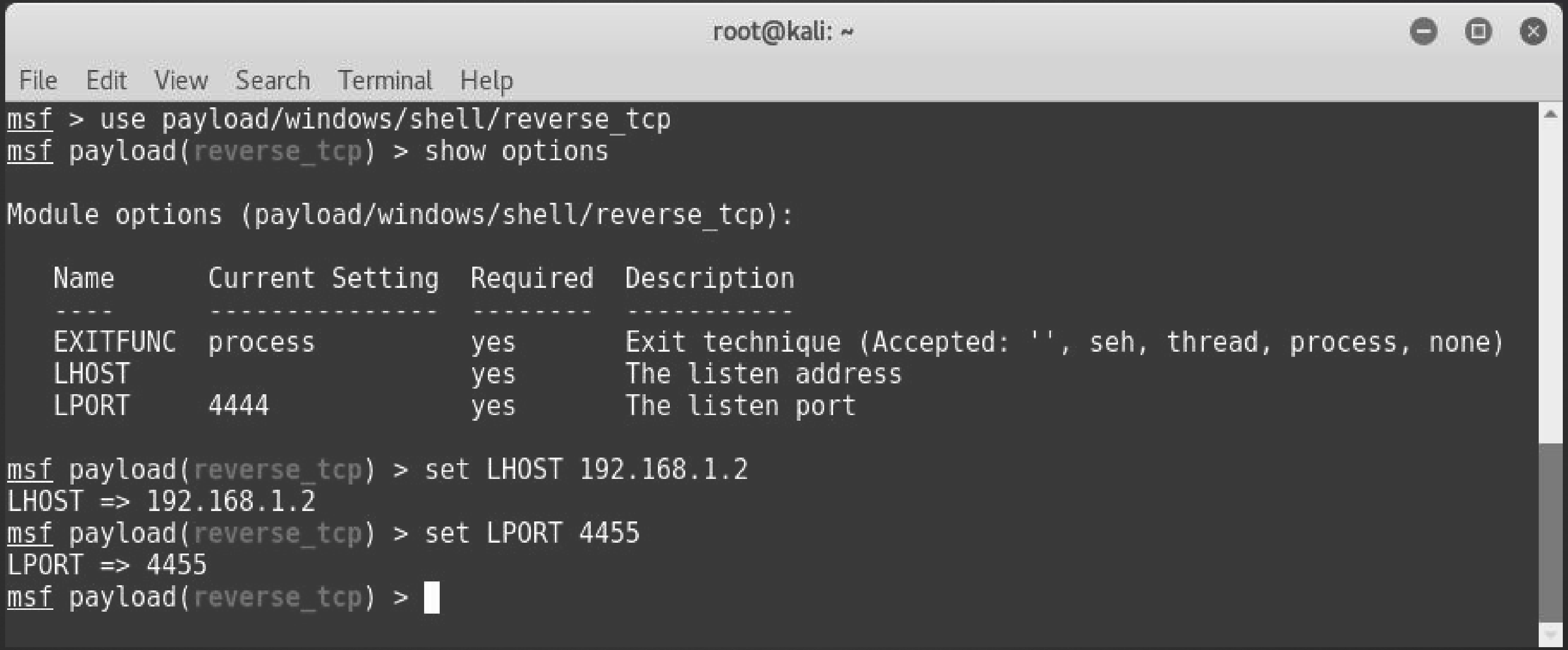
```
msf > use auxiliary/scanner/portscan/tcp
msf auxiliary(tcp) > show options

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 address range or CIDR identifier
   THREADS      1                yes       The number of concurrent threads
   TIMEOUT      1000             yes       The socket connect timeout in milliseconds

msf auxiliary(tcp) > set RHOSTS 192.168.1.100
RHOSTS => 192.168.1.100
msf auxiliary(tcp) > set PORTS 1-100
PORTS => 1-100
msf auxiliary(tcp) > run

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(tcp) > set PORTS 1-10000
PORTS => 1-10000
msf auxiliary(tcp) > run

[*] 192.168.1.100:        - 192.168.1.100:139 - TCP OPEN
[*] 192.168.1.100:        - 192.168.1.100:135 - TCP OPEN
```

Metasploit : Payloads

- To understand what a payload does, let's consider a real-world example.
  - A military unit of a country develops a new missile that can travel a range of 500 km at very high speed.
  - The missile is of no use unless it's armed with the right kind of ammunition.
  - The military unit decided to load high explosive material (the payload) within the missile to cause the required damage to the enemy.
  - The payload can be changed based on the severity of damage that is to be caused by the missile.
- Payloads in the Metasploit let us decide what action is to be performed on the target system once the exploit is successful.
  - Singles: These are sometimes also referred to as inline or non-staged payloads.
    Payloads in this category are a completely self-contained unit of the exploit and require shellcode, which means they have everything that is required to exploit the vulnerability on the target.
    The disadvantage of such payloads is their size.
  - Stagers: There are certain scenarios where the size of the payload matters a lot.
    A payload with even a single byte extra may not function well on the target system.
    The stager's payload comes in handy in such a situation because it makes it possible to establish simply sets up a connection between the attacking system and the target system.
  - Stages: Once the stager payload has set up a connection between the attacking system and the target system, the stages payloads are then downloaded on the target system. They contain the required shellcode to exploit the vulnerability on the target system.

Metasploit : Payloads

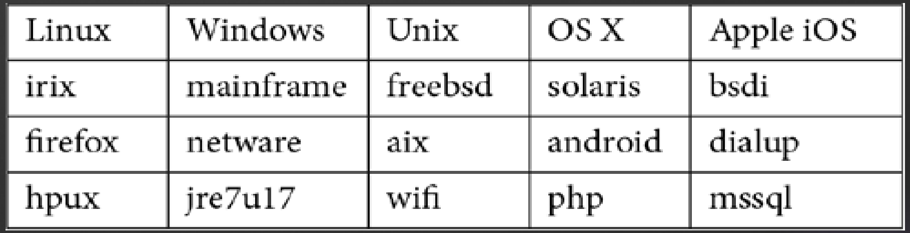
- The following screenshot shows a sample payload that can be used to obtain a reverse TCP shell from a compromised Windows system:



```
msf > use payload/windows/shell/reverse_tcp
msf payload(reverse_tcp) > show options

Module options (payload/windows/shell/reverse_tcp):

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

msf payload(reverse_tcp) > set LHOST 192.168.1.2
LHOST => 192.168.1.2
msf payload(reverse_tcp) > set LPORT 4455
LPORT => 4455
msf payload(reverse_tcp) >
```

Metasploit : Exploits

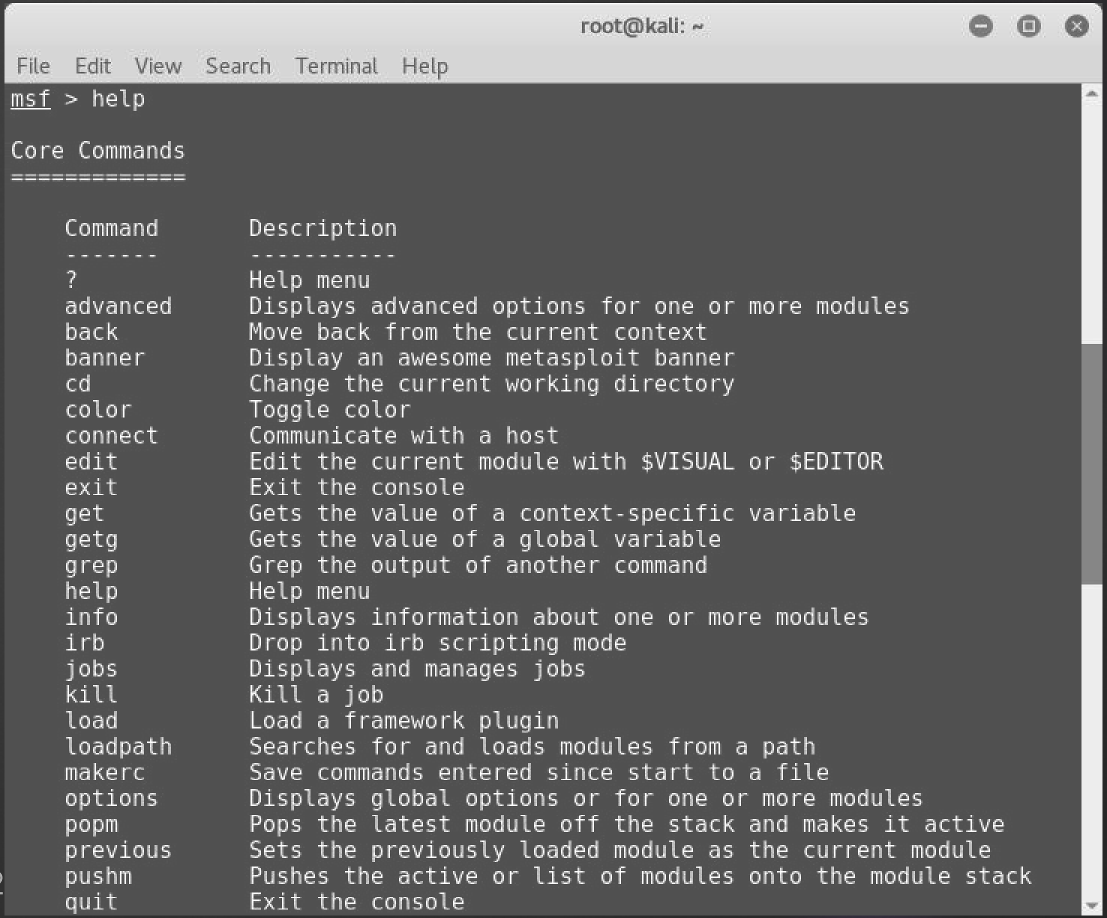
- An exploit is nothing but the actual piece of code that gives the required access to the target system.
- There are more than 2,500 exploits spread across more than 20 categories based on platform supported by exploit but which is the one that needs to be used?
- The decision to use a particular exploit against a target can be made only after extensive enumeration and vulnerability assessment of our target.
- Proper enumeration and a vulnerability assessment of the target will give us the following information based on which we can choose the correct exploit:
  - Operating system of the target system (including exact version and architecture)
  - Open ports on the target system TCP and UDP
  - Services along with versions running on the target system
  - Probability of a particular service being vulnerable
- The following table shows the various categories of exploits available in the Metasploit Framework:



Linux	Windows	Unix	OS X	Apple iOS
irix	mainframe	freebsd	solaris	bsdi
firefox	netware	aix	android	dialup
hpux	jre7u17	wifi	php	mssql

Metasploit : Useful commands

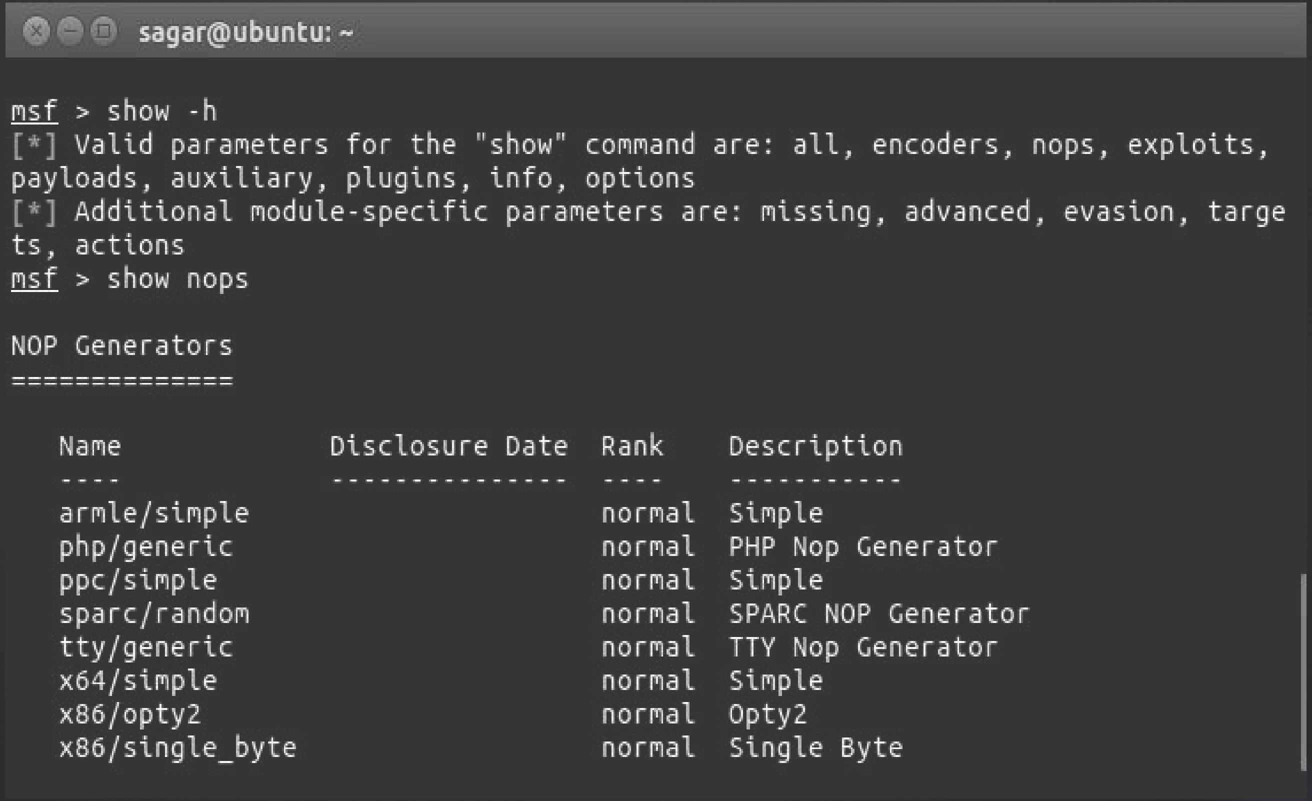
- The **help** command: As the name suggests, the help command offers additional information.



```
msf > help

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

    Command       Description
    -------       -----------
    ?             Help menu
    advanced      Displays advanced options for one or more modules
    back          Move back from the current context
    banner        Display an awesome metasploit banner
    cd            Change the current working directory
    color         Toggle color
    connect       Communicate with a host
    edit          Edit the current module with $VISUAL or $EDITOR
    exit          Exit the console
    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
    info          Displays information about one or more modules
    irb           Drop into irb scripting mode
    jobs          Displays and manages jobs
    kill          Kill a job
    load          Load a framework plugin
    loadpath      Searches for and loads modules from a path
    makerc        Save commands entered since start to a file
    options       Displays global options or for one or more modules
    popm          Pops the latest module off the stack and makes it active
    previous      Sets the previously loaded module as the current module
    pushm         Pushes the active or list of modules onto the module stack
    quit          Exit the console
```

Metasploit : Useful commands

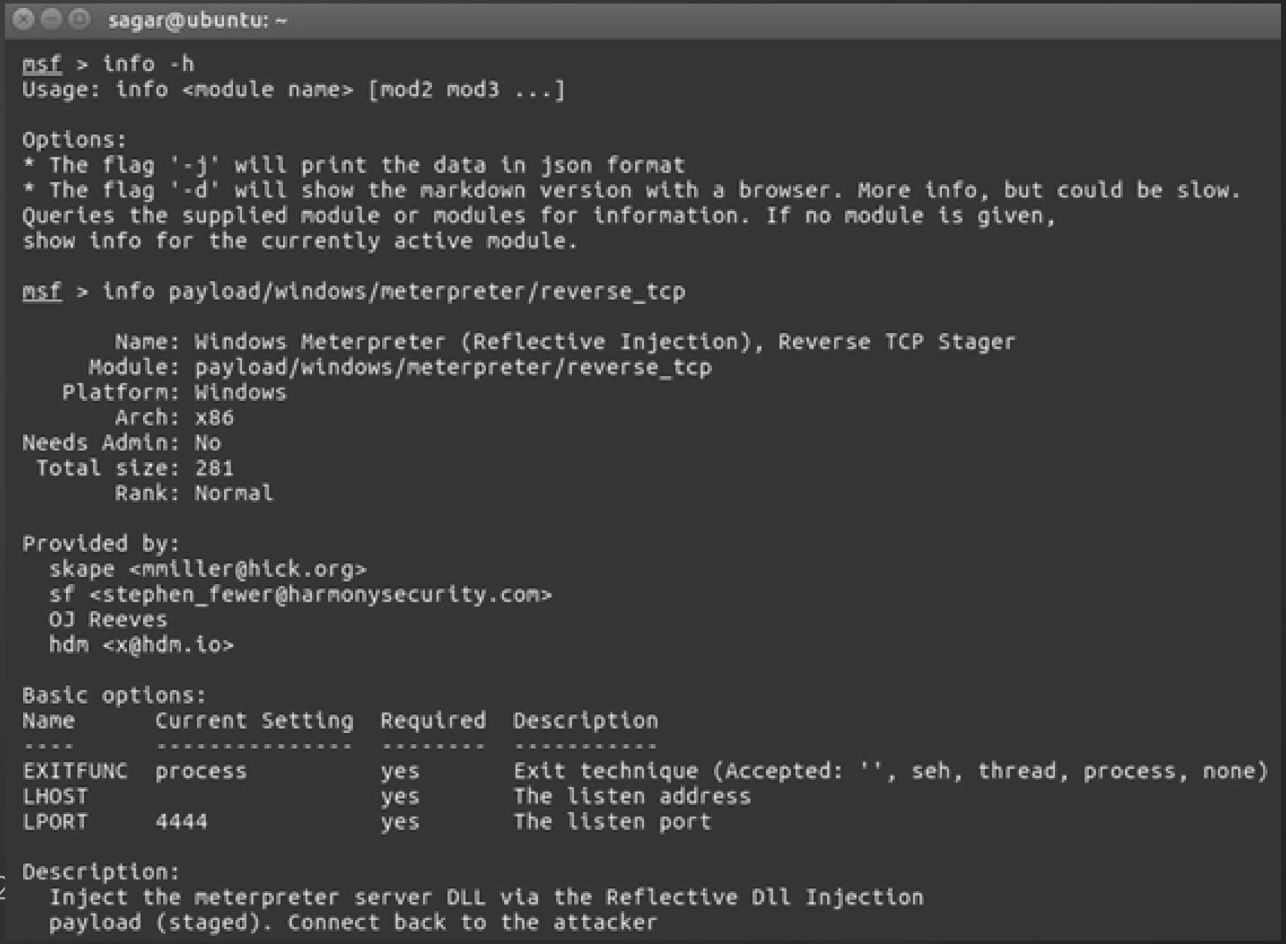
- The **show** command is used to display the available modules within the Metasploit Framework or to display additional information while using a particular module.



```
msf > show -h
[*] Valid parameters for the "show" command are: all, encoders, nops, exploits,
payloads, auxiliary, plugins, info, options
[*] Additional module-specific parameters are: missing, advanced, evasion, targe
ts, actions
msf > show nops

NOP Generators
==============

   Name              Disclosure Date  Rank    Description
   ----              ---------------  ----    -----------
   armle/simple                       normal  Simple
   php/generic                        normal  PHP Nop Generator
   ppc/simple                         normal  Simple
   sparc/random                       normal  SPARC NOP Generator
   tty/generic                        normal  TTY Nop Generator
   x64/simple                         normal  Simple
   x86/opty2                          normal  Opty2
   x86/single_byte                    normal  Single Byte
```

Metasploit : Useful commands

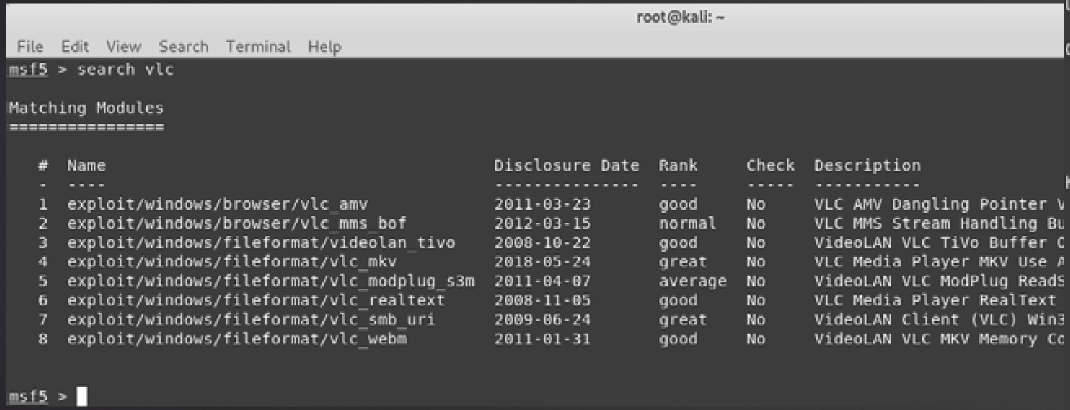
- The **info** command is used to display details about a particular module within the Metasploit Framework.



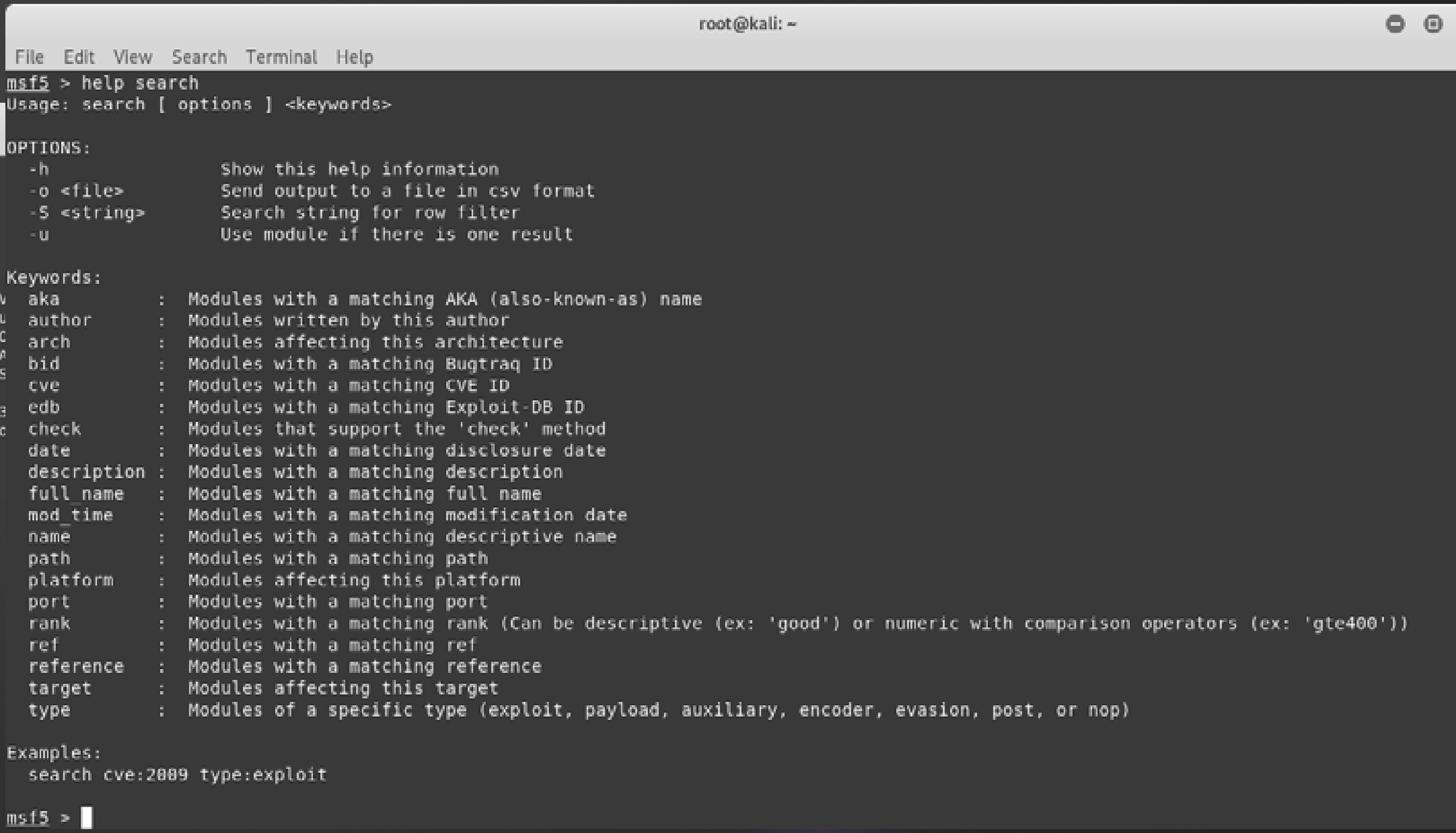
```
msf > info -h
Usage: info <module name> [mod2 mod3 ...]

Options:
* The flag '-j' will print the data in json format
* The flag '-d' will show the markdown version with a browser. More info, but could be slow.
Queries the supplied module or modules for information. If no module is given,
show info for the currently active module.

msf > info payload/windows/meterpreter/reverse_tcp

       Name: Windows Meterpreter (Reflective Injection), Reverse TCP Stager
     Module: payload/windows/meterpreter/reverse_tcp
   Platform: Windows
       Arch: x86
Needs Admin: No
 Total size: 281
       Rank: Normal

Provided by:
  skape <mmiller@hick.org>
  sf <stephen_fewer@harmonysecurity.com>
  OJ Reeves
  hdm <x@hdm.io>

Basic options:
Name      Current Setting  Required  Description
----      ---------------  --------  -----------
EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST                      yes       The listen address
LPORT     4444             yes       The listen port

Description:
  Inject the meterpreter server DLL via the Reflective Dll Injection
  payload (staged). Connect back to the attacker
```

Metasploit : Useful commands

- The Metasploit Framework is a package of many exploits and payloads. At times, it can be quite overwhelming to find the exact exploit or module. This is when the **search** command comes in handy.



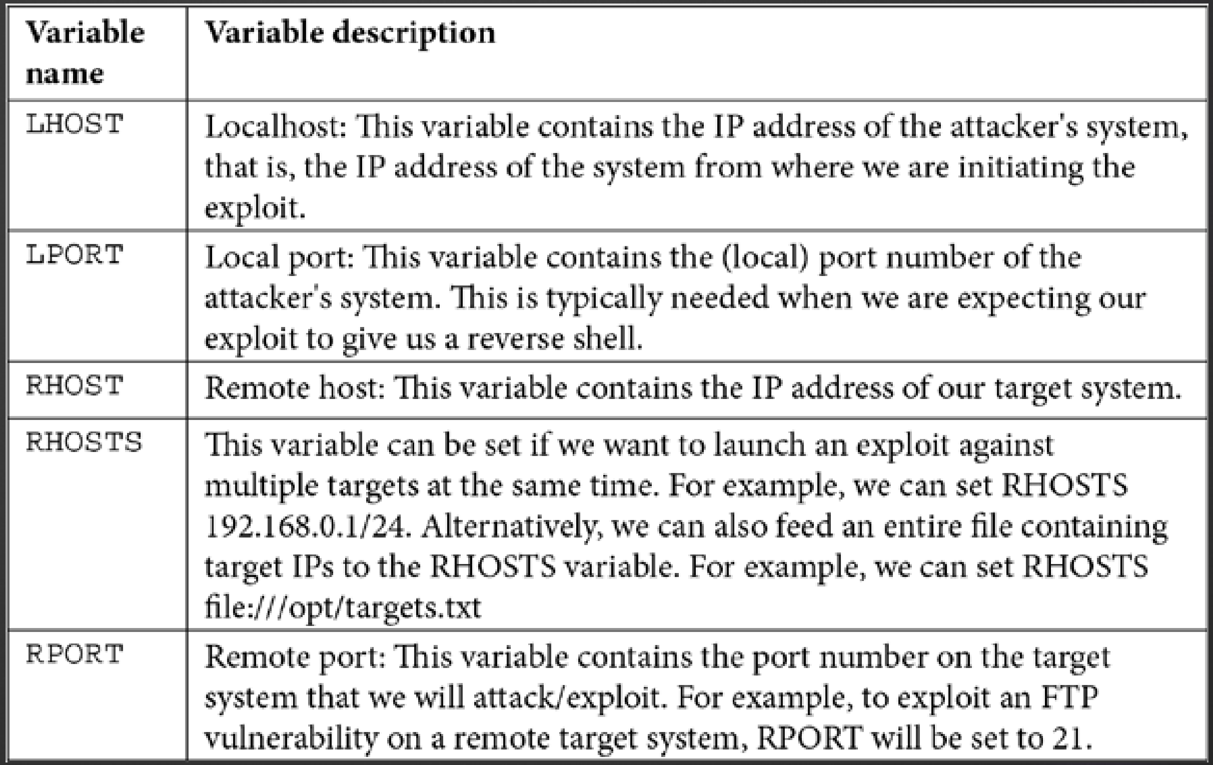
```
msf5 > search vlc

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

   #  Name                                          Disclosure Date  Rank     Check  Description
   -  ----                                          ---------------  ----     -----  -----------
   1  exploit/windows/browser/vlc_amv               2011-03-23       good     No     VLC AMV Dangling Pointer V
   2  exploit/windows/browser/vlc_mms_bof           2012-03-15       normal   No     VLC MMS Stream Handling Bu
   3  exploit/windows/fileformat/videolan_tivo      2008-10-22       good     No     VideoLAN VLC TiVo Buffer C
   4  exploit/windows/fileformat/vlc_mkv            2018-05-24       great    No     VLC Media Player MKV Use A
   5  exploit/windows/fileformat/vlc_modplug_s3m    2011-04-07       average  No     VideoLAN VLC ModPlug ReadS
   6  exploit/windows/fileformat/vlc_realtext       2008-11-05       good     No     VLC Media Player RealText
   7  exploit/windows/fileformat/vlc_smb_uri        2009-06-24       great    No     VideoLAN Client (VLC) Win3
   8  exploit/windows/fileformat/vlc_webm           2011-01-31       good     No     VideoLAN VLC MKV Memory Co


msf5 >
```



```
msf5 > help search
Usage: search [ options ] <keywords>

OPTIONS:
  -h                Show this help information
  -o <file>         Send output to a file in csv format
  -S <string>       Search string for row filter
  -u                Use module if there is one result

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
  full_name   :  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 (ex: 'gte400'))
  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)

Examples:
  search cve:2009 type:exploit

msf5 >
```

Metasploit : Variables

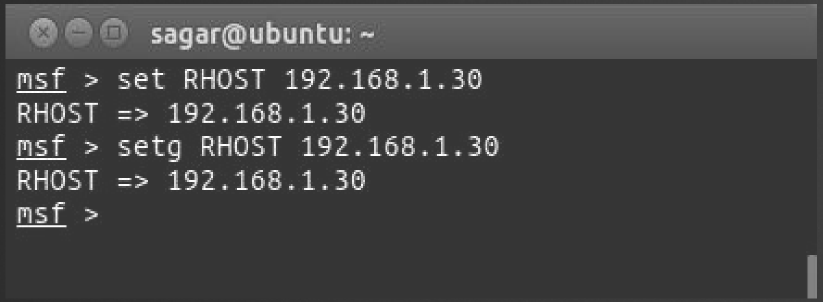
- For most exploits that we use within the Metasploit Framework, we need to set values to some of the variables. The following are some of the common and most important variables in the Metasploit Framework:



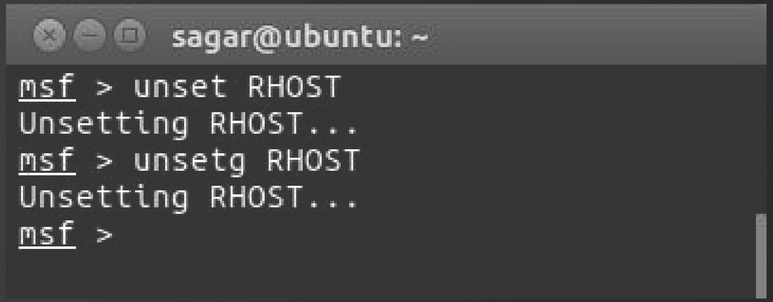
Variable name	**Variable description**
`LHOST`	Localhost: This variable contains the IP address of the attacker's system, that is, the IP address of the system from where we are initiating the exploit.
`LPORT`	Local port: This variable contains the (local) port number of the attacker's system. This is typically needed when we are expecting our exploit to give us a reverse shell.
`RHOST`	Remote host: This variable contains the IP address of our target system.
`RHOSTS`	This variable can be set if we want to launch an exploit against multiple targets at the same time. For example, we can set RHOSTS 192.168.0.1/24. Alternatively, we can also feed an entire file containing target IPs to the RHOSTS variable. For example, we can set RHOSTS file:///opt/targets.txt
`RPORT`	Remote port: This variable contains the port number on the target system that we will attack/exploit. For example, to exploit an FTP vulnerability on a remote target system, RPORT will be set to 21.

Metasploit : Variables

- The **set** command assigns a new value to one of the (local) variables (such as RHOST, RPORT, LHOST, and LPPORT) and **unset** unassigns the value.
- The **setg** command assigns a new value to the (global) variable on a permanent basis, so that it can be used repeatedly whenever required and **unsetg** unassigns the value.



```
msf > set RHOST 192.168.1.30
RHOST => 192.168.1.30
msf > setg RHOST 192.168.1.30
RHOST => 192.168.1.30
msf >
```



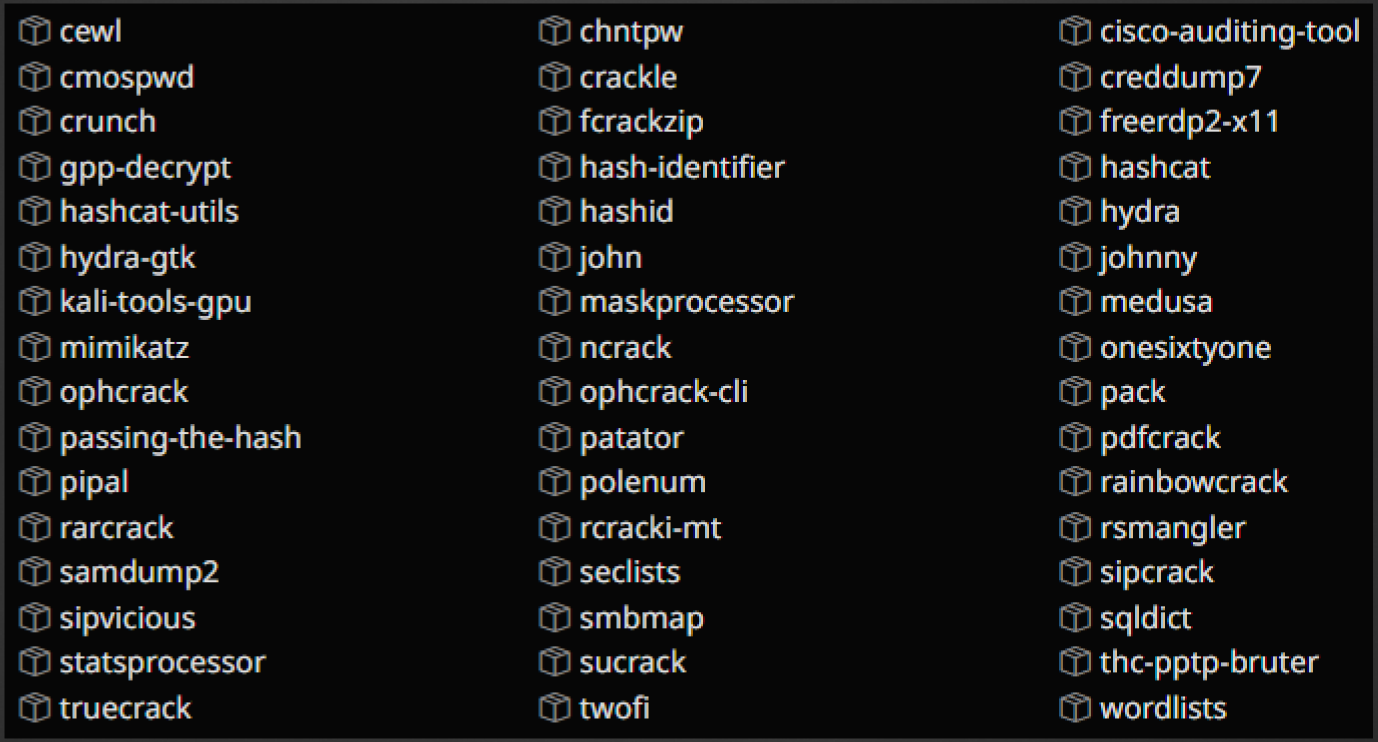
```
msf > unset RHOST
Unsetting RHOST...
msf > unsetg RHOST
Unsetting RHOST...
msf >
```

Metasploit

- **Test it : https://tryhackme.com/room/metasploitintro**

TOOLS : PASSWORDS

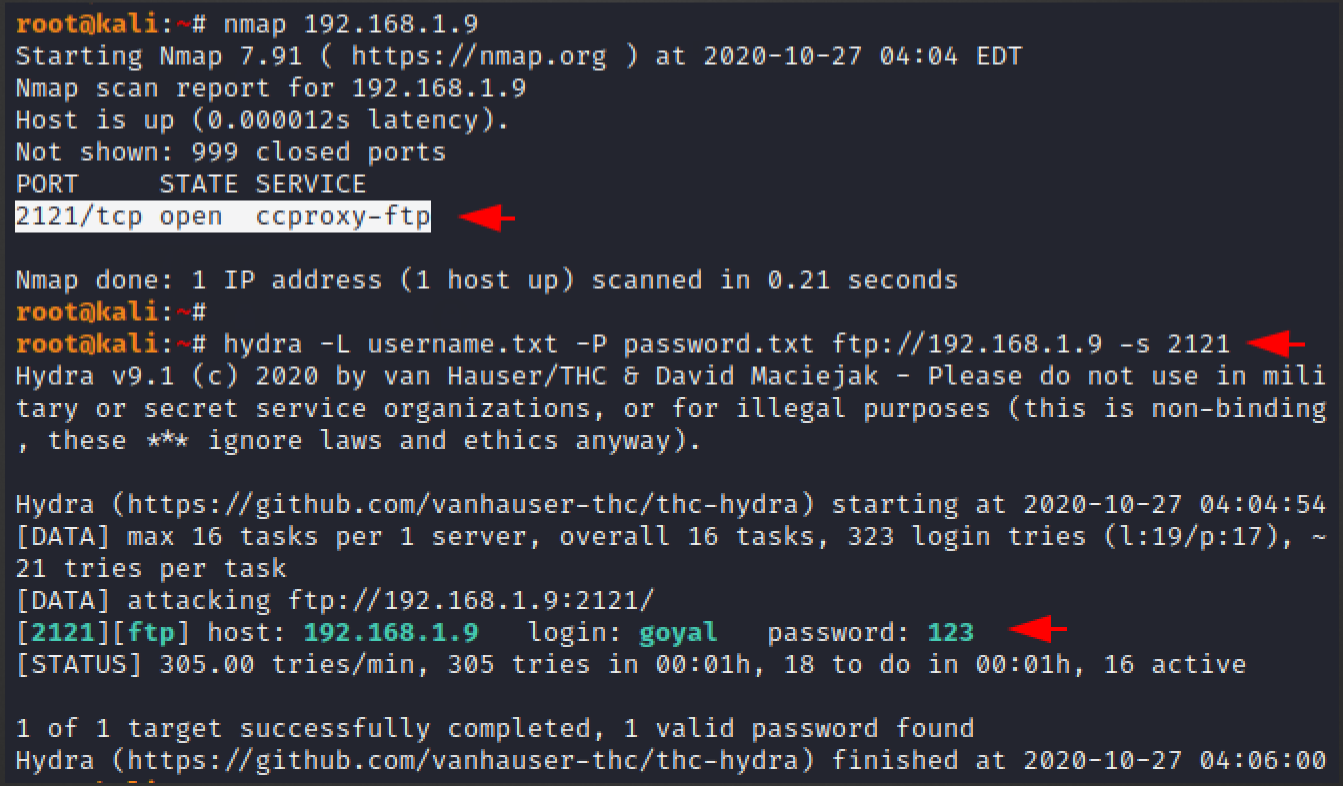
TOOLS : PASSWORD LIST



cewl
cmospwd
crunch
gpp-decrypt
hashcat-utils
hydra-gtk
kali-tools-gpu
mimikatz
ophcrack
passing-the-hash
pipal
rarcrack
samdump2
sipvicious
statsprocessor
truecrack
chntpw
crackle
fcrackzip
hash-identifier
hashid
john
maskprocessor
ncrack
ophcrack-cli
patator
polenum
rcracki-mt
seclists
smbmap
sucrack
twofi
cisco-auditing-tool
creddump7
freerdp2-x11
hashcat
hydra
johnny
medusa
onesixtyone
pack
pdfcrack
rainbowcrack
rsmangler
sipcrack
sqldict
thc-pptp-bruter
wordlists

HYDRA

- Hydra is a brute force online password cracking program.

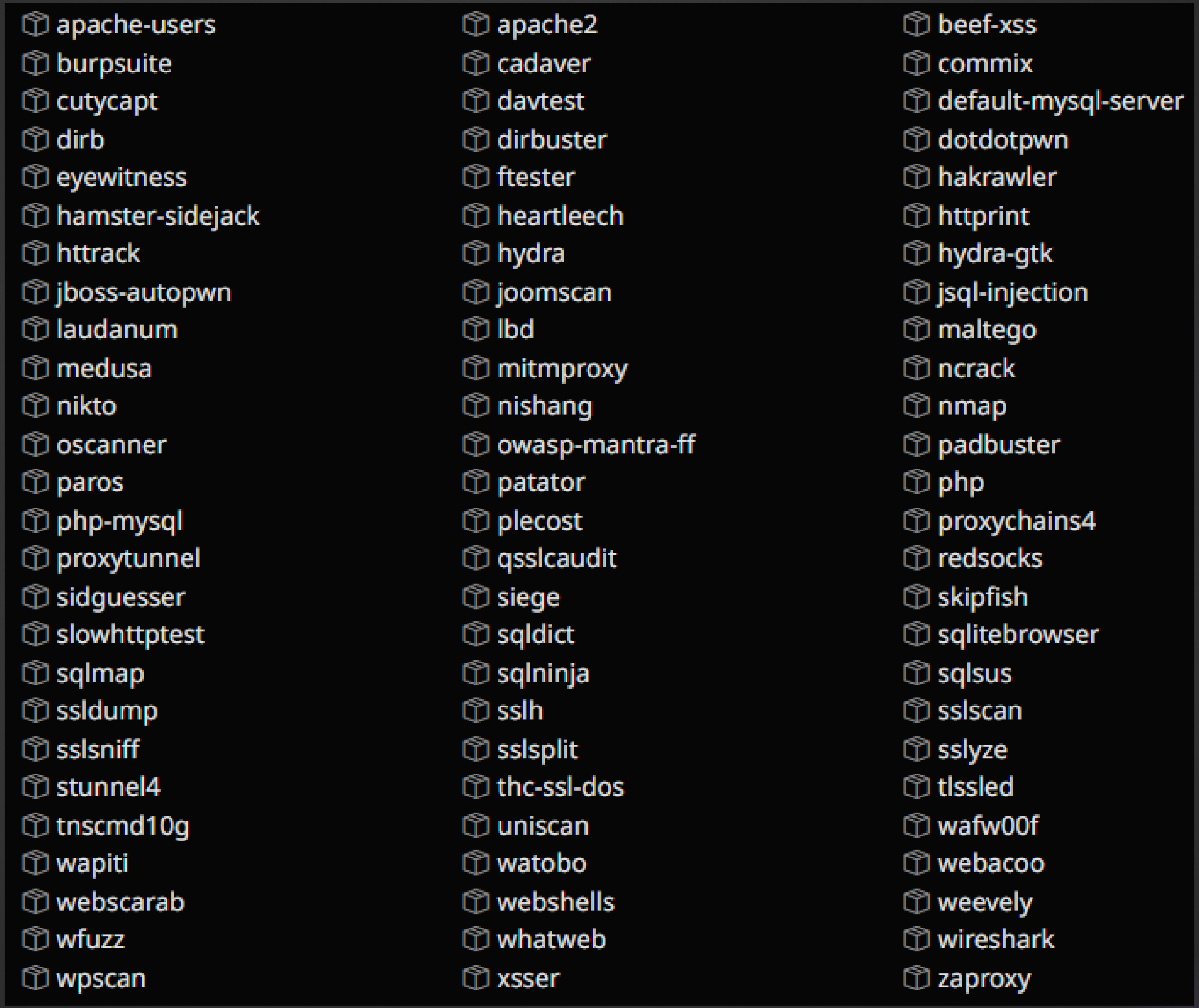


```
root@kali:~# nmap 192.168.1.9
Starting Nmap 7.91 ( https://nmap.org ) at 2020-10-27 04:04 EDT
Nmap scan report for 192.168.1.9
Host is up (0.000012s latency).
Not shown: 999 closed ports
PORT     STATE SERVICE
2121/tcp open  ccproxy-ftp

Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds
root@kali:~#
root@kali:~# hydra -L username.txt -P password.txt ftp://192.168.1.9 -s 2121
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in mili
tary or secret service organizations, or for illegal purposes (this is non-binding
, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-10-27 04:04:54
[DATA] max 16 tasks per 1 server, overall 16 tasks, 323 login tries (l:19/p:17), ~
21 tries per task
[DATA] attacking ftp://192.168.1.9:2121/
[2121][ftp] host: 192.168.1.9   login: goyal   password: 123
[STATUS] 305.00 tries/min, 305 tries in 00:01h, 18 to do in 00:01h, 16 active

1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-10-27 04:06:00
```

- **Test it : https://www.tryhackme.com/room/hydra**

TOOLS : WEB

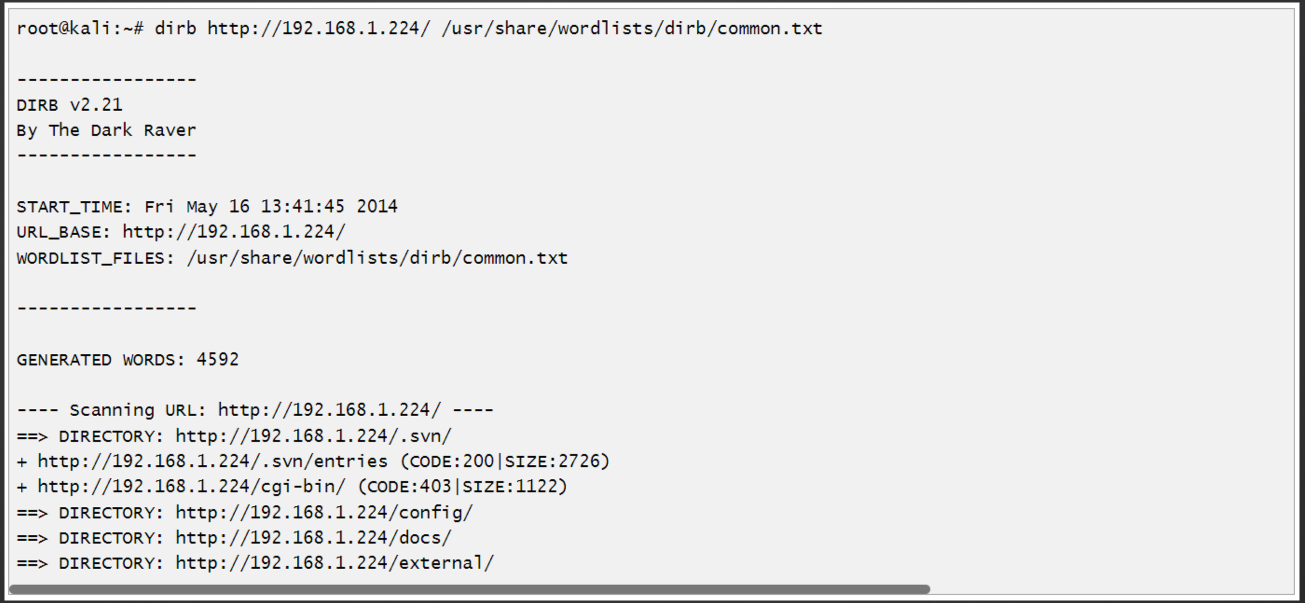
TOOLS : WEB LIST



- apache-users
- apache2
- beef-xss
- burpsuite
- cadaver
- commix
- cutycapt
- davtest
- default-mysql-server
- dirb
- dirbuster
- dotdotpwn
- eyewitness
- ftester
- hakrawler
- hamster-sidejack
- heartleech
- httprint
- httrack
- hydra
- hydra-gtk
- jboss-autopwn
- joomscan
- jsql-injection
- laudanum
- lbd
- maltego
- medusa
- mitmproxy
- ncrack
- nikto
- nishang
- nmap
- oscanner
- owasp-mantra-ff
- padbuster
- paros
- patator
- php
- php-mysql
- plecost
- proxychains4
- proxytunnel
- qsslcaudit
- redsocks
- sidguesser
- siege
- skipfish
- slowhttptest
- sqldict
- sqlitebrowser
- sqlmap
- sqlninja
- sqlsus
- ssldump
- sslh
- sslscan
- sslsniff
- sslsplit
- sslyze
- stunnel4
- thc-ssl-dos
- tlssled
- tnscmd10g
- uniscan
- wafw00f
- wapiti
- watobo
- webacoo
- webscarab
- webshells
- weevely
- wfuzz
- whatweb
- wireshark
- wpscan
- xsser
- zaproxy

DIRB

- DIRB is a web content scanner.
- It looks for existing (and/or hidden) Web Objects.
- It works by launching a dictionary based attack against a web server and analyzing the response.

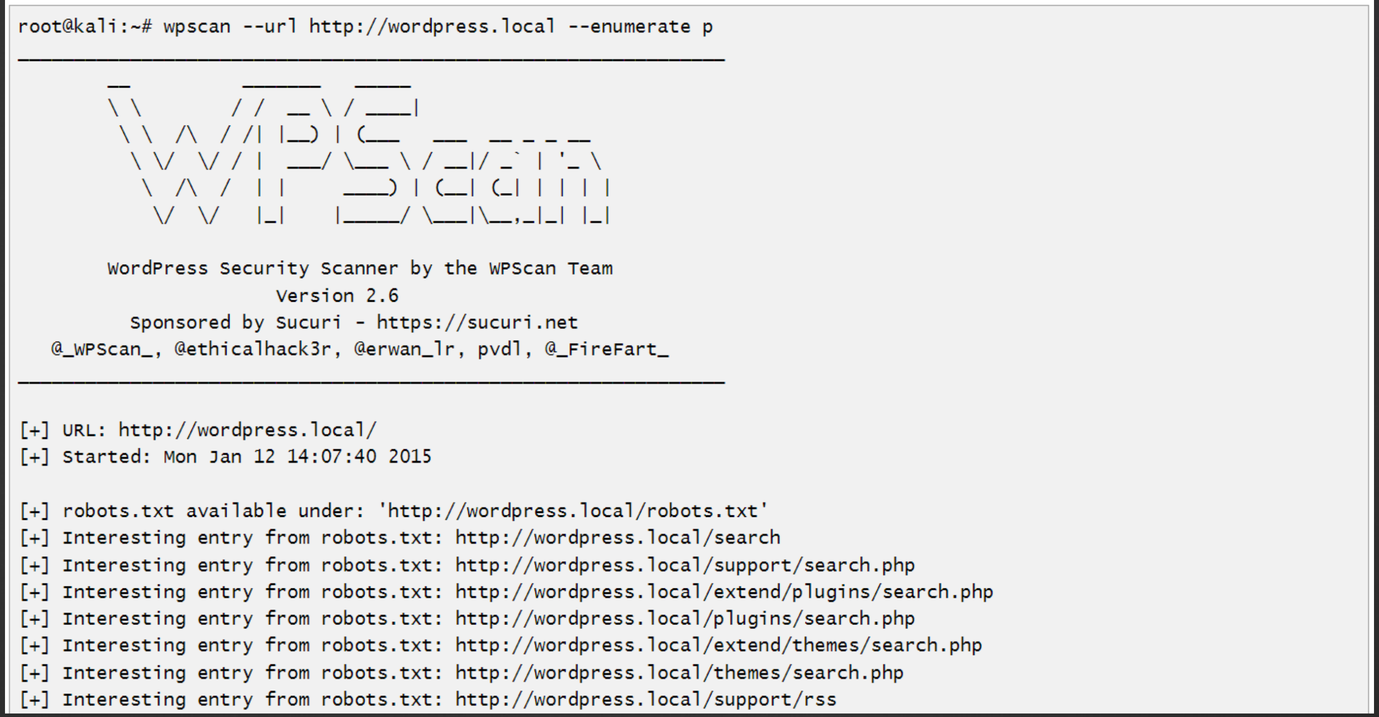


```
root@kali:~# dirb http://192.168.1.224/ /usr/share/wordlists/dirb/common.txt

-----------------
DIRB v2.21
By The Dark Raver
-----------------

START_TIME: Fri May 16 13:41:45 2014
URL_BASE: http://192.168.1.224/
WORDLIST_FILES: /usr/share/wordlists/dirb/common.txt

-----------------

GENERATED WORDS: 4592

---- Scanning URL: http://192.168.1.224/ ----
==> DIRECTORY: http://192.168.1.224/.svn/
+ http://192.168.1.224/.svn/entries (CODE:200|SIZE:2726)
+ http://192.168.1.224/cgi-bin/ (CODE:403|SIZE:1122)
==> DIRECTORY: http://192.168.1.224/config/
==> DIRECTORY: http://192.168.1.224/docs/
==> DIRECTORY: http://192.168.1.224/external/
```

- **More information: https://tools.kali.org/web-applications/dirb**

WPScan

- WPScan is a black box WordPress vulnerability scanner that can be used to scan remote WordPress installations to find security issues.



```
root@kali:~# wpscan --url http://wordpress.local --enumerate p
_______________________________________________________________
        __          _______   _____
        \ \        / /  __ \ / ____|
         \ \  /\  / /| |__) | (___   ___  __ _ _ __
          \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
           \  /\  /  | |     ____) | (__| (_| | | | |
            \/  \/   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                       Version 2.6
          Sponsored by Sucuri - https://sucuri.net
   @_WPScan_, @ethicalhack3r, @erwan_lr, pvdl, @_FireFart_
_______________________________________________________________

[+] URL: http://wordpress.local/
[+] Started: Mon Jan 12 14:07:40 2015

[+] robots.txt available under: 'http://wordpress.local/robots.txt'
[+] Interesting entry from robots.txt: http://wordpress.local/search
[+] Interesting entry from robots.txt: http://wordpress.local/support/search.php
[+] Interesting entry from robots.txt: http://wordpress.local/extend/plugins/search.php
[+] Interesting entry from robots.txt: http://wordpress.local/plugins/search.php
[+] Interesting entry from robots.txt: http://wordpress.local/extend/themes/search.php
[+] Interesting entry from robots.txt: http://wordpress.local/themes/search.php
[+] Interesting entry from robots.txt: http://wordpress.local/support/rss
```

- **More information: https://tools.kali.org/web-applications/wpscan**