

Metasploit para novatos

Estos ejemplos son para una versión de Metasploit preinstalada en sistemas kali linux.

1. Inicialización de la base de datos y primer arranque de metasploit

Para el uso de metasploit se recomienda inicializar la base de datos la primera vez que se arranque con el comando:

msfdb init

Una vez que se inicialice la base de datos, cada vez que se quiera usar metasploit se puede arrancar con el siguiente comando:

msfdb run

2. Escaneo de máquina objetivo

Lo primero que debemos hacer es escanear los puertos de la máquina objetivo para identificar qué servicios tiene arrancados y si alguno de estos es vulnerable.

2.1 NMAP

Primero se puede comenzar realizando un escaneo de namp desde metasploit con el siguiente comando:

```
db_nmap <ip_maquina_objetivo>
```

Como resultado deberíamos recibir un listado de puertos abiertos indicando que servicio provee cada uno:

```
msf6 > db_nmap 192.168.56.9
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-02 11:03 EDT
[*] Nmap: Nmap scan report for 192.168.56.9
[*] Nmap: Host is up (0.0011s latency).
[*] Nmap: Not shown: 977 closed tcp ports (reset)
[*] Nmap: PORT      STATE SERVICE
[*] Nmap: 21/tcp    open  ftp
[*] Nmap: 22/tcp    open  ssh
[*] Nmap: 23/tcp    open  telnet
[*] Nmap: 25/tcp    open  smtp
[*] Nmap: 53/tcp    open  domain
[*] Nmap: 80/tcp    open  http
[*] Nmap: 111/tcp   open  rpcbind
[*] Nmap: 139/tcp   open  netbios-ssn
[*] Nmap: 445/tcp   open  microsoft-ds
[*] Nmap: 512/tcp   open  exec
[*] Nmap: 513/tcp   open  login
[*] Nmap: 514/tcp   open  shell
[*] Nmap: 1099/tcp  open  rmiregistry
[*] Nmap: 1524/tcp  open  ingreslock
[*] Nmap: 2049/tcp  open  nfs
[*] Nmap: 2121/tcp  open  ccproxy-ftp
[*] Nmap: 3306/tcp  open  mysql
[*] Nmap: 5432/tcp  open  postgresql
[*] Nmap: 5900/tcp  open  vnc
[*] Nmap: 6000/tcp  open  X11
[*] Nmap: 6667/tcp  open  irc
[*] Nmap: 8009/tcp  open  ajp13
[*] Nmap: 8180/tcp  open  unknown
[*] Nmap: MAC Address: 08:00:27:22:00:BD (Oracle VirtualBox virtual NIC)
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 13.45 seconds
msf6 > █
```

2.2 Metasploit port scanner

Alternativamente también se puede usar el módulo de escaneo de metasploit, para ello podemos seleccionarlo con el siguiente comando:

```
use auxiliary/scanner/portscan/tcp
```

Una vez seleccionado el módulo, hay que configurar sus parámetros, podemos ver los parámetros disponibles con el siguiente comando.

```
show options
```

```
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        0          yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
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.

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

Para configurar los parámetros se debe usar el comando set:

```
set <parámetro> <valor>
```

Por ejemplo, en este caso se debe establecer un valor para RHOST para indicarle a metasploit que máquina de la red debe escanear:

```
msf6 auxiliary(scanner/portscan/tcp) > set RHOST 192.168.56.9
RHOST => 192.168.56.9
```

Una vez configurados los parámetros se puede ejecutar el módulo con el comando run:

```
msf6 auxiliary(scanner/portscan/tcp) > run
[+] 192.168.56.9: - 192.168.56.9:22 - TCP OPEN
[+] 192.168.56.9: - 192.168.56.9:21 - TCP OPEN
[+] 192.168.56.9: - 192.168.56.9:25 - TCP OPEN
[+] 192.168.56.9: - 192.168.56.9:23 - TCP OPEN
[+] 192.168.56.9: - 192.168.56.9:53 - TCP OPEN
[+] 192.168.56.9: - 192.168.56.9:80 - TCP OPEN
[+] 192.168.56.9: - 192.168.56.9:111 - TCP OPEN
[+] 192.168.56.9: - 192.168.56.9:139 - TCP OPEN
[+] 192.168.56.9: - 192.168.56.9:445 - TCP OPEN
[+] 192.168.56.9: - 192.168.56.9:513 - TCP OPEN
[+] 192.168.56.9: - 192.168.56.9:514 - TCP OPEN
[+] 192.168.56.9: - 192.168.56.9:512 - TCP OPEN
[+] 192.168.56.9: - 192.168.56.9:1099 - TCP OPEN
[+] 192.168.56.9: - 192.168.56.9:1524 - TCP OPEN
[+] 192.168.56.9: - 192.168.56.9:2049 - TCP OPEN
[+] 192.168.56.9: - 192.168.56.9:2121 - TCP OPEN
[+] 192.168.56.9: - 192.168.56.9:3306 - TCP OPEN
[+] 192.168.56.9: - 192.168.56.9:3632 - TCP OPEN
[+] 192.168.56.9: - 192.168.56.9:5432 - TCP OPEN
[+] 192.168.56.9: - 192.168.56.9:5900 - TCP OPEN
[+] 192.168.56.9: - 192.168.56.9:6000 - TCP OPEN
[+] 192.168.56.9: - 192.168.56.9:6667 - TCP OPEN
[+] 192.168.56.9: - 192.168.56.9:6697 - TCP OPEN
[+] 192.168.56.9: - 192.168.56.9:8009 - TCP OPEN
[+] 192.168.56.9: - 192.168.56.9:8180 - TCP OPEN
[+] 192.168.56.9: - 192.168.56.9:8787 - TCP OPEN
[*] 192.168.56.9: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/portscan/tcp) > 
```

2.3 Escaneo Profundo

Podemos realizar un escaneo profundo de la máquina en cuestión con el siguiente comando:

```
db_nmap -sV <ip_máquina_objetivo>
```

```
msf6 auxiliary(scanner/portscan/tcp) > db_nmap -sV 192.168.56.9
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-02 11:17 EDT
[*] Nmap: Nmap scan report for 192.168.56.9
[*] Nmap: Host is up (0.00050s latency).
[*] Nmap: Not shown: 977 closed tcp ports (reset)
[*] Nmap: PORT      STATE SERVICE      VERSION
[*] Nmap: 21/tcp    open  ftp          vsftpd 2.3.4
[*] Nmap: 22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
[*] Nmap: 23/tcp    open  telnet       Linux telnetd
[*] Nmap: 25/tcp    open  smtp         Postfix smtpd
[*] Nmap: 53/tcp    open  domain      ISC BIND 9.4.2
[*] Nmap: 80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
[*] Nmap: 111/tcp   open  rpcbind     2 (RPC #100000)
[*] Nmap: 139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
[*] Nmap: 445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
[*] Nmap: 512/tcp   open  exec        netkit-rsh rexecd
[*] Nmap: 513/tcp   open  login       OpenBSD or Solaris rlogind
[*] Nmap: 514/tcp   open  shell       Netkit rshd
[*] Nmap: 1099/tcp  open  java-rmi   GNU Classpath grmiregistry
[*] Nmap: 1524/tcp  open  bindshell   Metasploitable root shell
[*] Nmap: 2049/tcp  open  nfs         2-4 (RPC #100003)
[*] Nmap: 2121/tcp  open  ftp         ProFTPD 1.3.1
[*] Nmap: 3306/tcp  open  mysql       MySQL 5.0.51a-3ubuntu5
[*] Nmap: 5432/tcp  open  postgresql PostgreSQL DB 8.3.0 - 8.3.7
[*] Nmap: 5900/tcp  open  vnc         VNC (protocol 3.3)
[*] Nmap: 6000/tcp  open  X11         (access denied)
[*] Nmap: 6667/tcp  open  irc         UnrealIRCd
[*] Nmap: 8009/tcp  open  ajp13      Apache Jserv (Protocol v1.3)
[*] Nmap: 8180/tcp  open  http        Apache Tomcat/Coyote JSP engine 1.1
[*] Nmap: MAC Address: 08:00:27:22:00:BD (Oracle VirtualBox virtual NIC)
[*] Nmap: Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
[*] 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 25.27 seconds
msf6 auxiliary(scanner/portscan/tcp) > 
```

Si además queremos más información podemos añadir el flag -A:

```
db_nmap -sV -A <ip_maquina_objetivo>
```

```
msf6 auxiliary(scanner/portscan/tcp) > db_nmap -sV -A 192.168.56.9
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-02 11:23 EDT
[*] Nmap: Nmap scan report for 192.168.56.9
[*] Nmap: Host is up (0.00072s latency).
[*] Nmap: Not shown: 977 closed tcp ports (reset)
[*] Nmap: PORT      STATE SERVICE      VERSION
[*] Nmap: 21/tcp    open  ftp          vsftpd 2.3.4
[*] Nmap: |_ftp-anon: Anonymous FTP login allowed (FTP code 230)
[*] Nmap: |_ftp-syst:
[*] Nmap: |  STAT:
[*] Nmap: |  FTP server status:
[*] Nmap: |    Connected to 192.168.56.102
[*] Nmap: |    Logged in as ftp
[*] Nmap: |    TYPE: ASCII
[*] Nmap: |    No session bandwidth limit
[*] Nmap: |    Session timeout in seconds is 300
[*] Nmap: |    Control connection is plain text
[*] Nmap: |    Data connections will be plain text
[*] Nmap: |    vsFTPD 2.3.4 - secure, fast, stable
[*] Nmap: |_End of status
[*] Nmap: 22/tcp   open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
[*] Nmap: |_ssh-hostkey:
[*] Nmap: | 1024 60:0f:cf:e1:05:f6:a7:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
[*] Nmap: | 2048 56:56:24:0f:21:id:de:a7:b2:ae:61:b1:24:3d:e8:f3 (RSA)
[*] Nmap: 23/tcp   open  telnet       Linux telnetd
[*] Nmap: 25/tcp   open  smtp         Postfix smtpd
[*] Nmap: |_ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no such thing outside US/countryName=XX
[*] Nmap: |_Not valid before: 2010-03-17T14:07:45
[*] Nmap: |_Not valid after: 2010-04-16T14:07:45
[*] Nmap: |_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN
[*] Nmap: |_ssl-date: 2025-05-02T15:23:44+00:00; 0s from scanner time.
[*] Nmap: |_sslv2:
[*] Nmap: |  SSLv2 supported
[*] Nmap: |  ciphers:
[*] Nmap: |    SSL2_RC4_128_WITH_MD5
[*] Nmap: |    SSL2_RC2_128_CBC_WITH_MD5
[*] Nmap: |    SSL2_DES_192_EDE3_CBC_WITH_MD5
```

3. Explotando vulnerabilidades

Ahora que sabemos los puertos abiertos y servicios de la máquina, podemos proceder a explotar las vulnerabilidades de esta.

3.1 Localización de exploits con searchsploit

Para localizar si alguno de los servicios de la máquina tiene exploits disponibles podemos usar searchsploit. Si queremos buscar por título del exploit podemos usar el comando searchsploit con el flag -t:

```
searchsploit -t <nombre_del_servicio>
```

Por ejemplo, si queremos buscar una vulnerabilidad para el servicio UnrealIRCd:

```
(thejuanvisu㉿kali)-[~]
$ searchsploit -t unrealircd

Exploit Title | Path
-----|-----
UnrealIRCd 3.2.8.1 - Backdoor Command Execution (Metasploit) | linux/remote/16922.rb
UnrealIRCd 3.2.8.1 - Local Configuration Stack Overflow | windows/dos/18011.txt
UnrealIRCd 3.2.8.1 - Remote Downloader/Execute | linux/remote/13853.pl
UnrealIRCd 3.x - Remote Denial of Service | windows/dos/27407.pl

Shellcodes: No Results
```

Si quisieramos buscar un exploit en función a un CVE podemos usar el comando con el flag -cve seguido de la fecha seguida de un guion y el identificador:

```
searchsploit --cve <fecha>-<identificador>
```

Por ejemplo, para buscar los exploits asociados al CVE-2010-2075:

```
(thejuanvisu㉿kali)-[~]
$ searchsploit --cve 2010-2075

Exploit Title | Path
-----|-----
UnrealIRCd 3.2.8.1 - Backdoor Command Execution (Metasploit) | linux/remote/16922.rb
UnrealIRCd 3.2.8.1 - Remote Downloader/Execute | linux/remote/13853.pl

Shellcodes: No Results

(thejuanvisu㉿kali)-[~]
$
```

Con los resultados obtenidos podemos ver que hay al menos 2 exploits que se podrían aplicar a esta máquina aprovechando las vulnerabilidades del servicio UnrealIRCd.

3.2 Exploit Automático con metasploit

Para ejecutar el exploit de forma automática contra la máquina objetivo debemos primero saber donde está ubicado el módulo automático para cargarlo en metasploit. Para ello podemos buscar el CVE en google seguido de metasploit, donde nos aparecerán páginas, como la de rapid7 con la ubicación de este:

Architectures
cmd

References

- [Source Code](#)
- [History](#)

Module Options

To display the available options, load the module within the Metasploit console and run the commands 'show options' or 'show advanced':

```
msf > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf //() > show actions
...actions...
msf //() > set ACTION < action-name >
msf //() > show options
...show and set options...
msf //() > run
```

Ahora que sabemos donde está el módulo podemos cargarlo usando el comando use:

```
use <ubicación_del_modulo>
```

```
File Actions Edit View Help
msf6 auxiliary(scanner/portscan/tcp) > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) >
```

Una vez cargado usamos el siguiente comando para ver los parámetros del módulo

```
show options
```

```
File Actions Edit View Help
msf6 auxiliary(scanner/portscan/tcp) > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show options
Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):
Name  Current Setting  Required  Description
      _____
      CHOST          no        The local client address
      CPORT          no        The local client port
      Proxies        no        A proxy chain of format type:host:port[,type:host:port][...]
      RHOSTS         yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
      RPORT          6667      yes       The target port (TCP)

Exploit target:
Id  Name
--  --
0  Automatic Target

View the full module info with the info, or info -d command.
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) >
```

From:

<https://knoppia.net/> - **Knoppia**



Permanent link:

https://knoppia.net/doku.php?id=metasploit:ms_dummies&rev=1746203124

Last update: **2025/05/02 16:25**