

## **Práctica 38. Paso del hash.... (Por HxC Mods-Adm)**

En esta ocasión vamos a "probar" un ataque devastador que sufrieron los Windows NT.... y todavía operativo con W2000 y 2003.

Se trata de montar un proxy "especial" que nos permitirá conectarnos a un equipo remoto como administradores (o como cualquier otro usuario válido) sin necesidad de conocer la contraseña del admin., bueno, bueno, no es así de mágico... tiene trampa, pero en su momento fue "matador" porque los hashes de las SAM de NT no estaban cifradas con syskey y bastaba con colocar nuestro esnifer SMB/NetBios preferido a escuchar las autenticaciones para ser capaces de tomar el control de cualquier máquina NT.

Bien, por partes... necesitaremos

- Un atacante, en mi caso será un Windows 2000 Advanced Server,
- Una víctima, que ha de ser un windows NT y/o Windows 2000, XP
- Pwdump2, 3 ó 4 al gusto, yo usaré pwdump4 pero es lo mismo para éste caso.
- Un exploit RPCDCOM, que está de moda
- Y... un proxy transparente, pero con mucho salero.... smbproxy

Claro, también necesitaremos algo de suerte y tiempo para practicar...

A por ellos que son pocos y .... no voy a explicar de nuevo el proceso de autenticación de los sistemas Windows, ya hemos descrito con anterioridad todo ello, pero si comentaré por qué era un ataque devastador (y todavía lo es, pero hay que tener "algo de suerte")

Ya conocemos que los hashes y usuarios de una máquina Windows se guardan en un archivo llamado SAM (olvidemos de momento los servidores de dominio), dicho archivo es intocable y "esos" datos están cifrados en MD4, con la aparición de syskey se le dobla la seguridad aunque desgraciadamente para windows, al depender todavía de LANManager, sigue siendo muy vulnerable.

Antes de que apareciese syskey, la información que guarda la SAM se encontraba cifrada sólo mediante el algoritmo MD4, por lo que si se obtenía la SAM por cualquier método, nos quedaría la "pesada" tarea de pasar un craqueador por diccionario o fuerza bruta para obtener el pass en texto plano, eso sigue siendo igual ahora, pero veamos un "nuevo concepto"

Si ya tengo los hashes *¿Por qué no pasar directamente el hash al Sistema Operativo "víctima" en lugar de perder el tiempo en craquearlo?*

Con esto nos evitamos la labor de reventarlo, que puede ser cuestión de minutos o de semanas, dependiendo de lo fuerte que sea la contraseña y lo rápido que sea el equipo que craquea.

Es decir, el truco consiste en iniciar la sesión como administrador en la víctima SIN CONOCER EL PASSWORD, sólo con saber el nombre de usuario y el hash cifrado en MD4 y ahorrarnos la molestia de pasarlo por "el molino" (es decir, por el craqueador de contraseñas, tipo john, LC4 o cain)

Hoy en día la técnica sigue siendo válida, lo comprbaréis inmediatamente, pero el segundo cifrado que syskey le pasa a los hashes imposibilita que se haga "directamente" y tendremos que ayudarnos de pwdump para obtener los hashes en Md4.

### **Paradoja:**

Pwdump necesita que seamos administradores, o más, para que funcione, por eso nos apoyaremos en nuestro querido exploit RPC, gracias a él lograremos una shell de system y por tanto podremos ejecutar pwdump, pero no es el único modo:

¿que nos impide crearnos un script que ejecute pwdump en la máquina remota y luego que nos envíe el resultado a un ftp?, deseando tener suerte y que sea el admin. el que lo ejecute sin darse cuenta, vamos que seguro que se os ocurren más métodos... o.....

¿Por qué no un correo malicioso? O ¿Un bug diferente al RPC? O por las bravas.... lo empaquetamos y se lo enviamos disfrazado de MP3 o de cualquier cosa que se nos ocurra.... hasta si me apuras y tenemos acceso físico a la máquina, nos lo llevamos en un disquete y lo "pasamos" en un descuido o mediante ingeniería social.

Todo esto, no era preciso antes de syskey, bastaba con esnifar las conversaciones SMB para obtener el hash deseado e inmediatamente ponerlo en el proxy, segundos se tardaba compañeros....

### Como usar Pwdump

Pwdump es fácil de usar, hay varias, existen otros similares como LSAdump, al final se consigue el mismo objetivo.... volcar los hashes de las cuentas de usuario y contraseñas.

Supongamos que usamos pwdump3, la sintaxis es:

**C:\pwdump3 ip.del.objetivo**

O sea, algo así: pwdump3 172.28.0.25 (siendo la ip 172.28.0.25 la del equipo en el que estamos loguados)

Si queremos que el volcado se guarde en un archivo, escribiremos:

**C:\pwdump3 172.28.0.25 z.txt**

Así que supongamos que los hashes obtenidos mediante pwdump:

```
Administrador:500:AAAB261B2008C113AAD3B435B51404EE:5E8C03CCBC34F2E2E6CFC57102C91C09:::  
Invitado:501:AAD3B435B51404EEAAD3B435B51404EE:31D6CFE0D16AE931B73C59D7E0C089C0:::  
IUSR_PC-CASA2:1001:55C5F8F18F646475EB753E0C14807592:2FB3F8B4B3885FB81BAE27667EA0FFC7:::  
IWAM_PC-CASA2:1002:A5ACD144C59B5ED6EB5FA977F603B77C:BE8D3DE476F41CC6CE716145DCA85458:::  
TsInternetUser:1000:3252D2D1C68F7174EA9206043F8DDC01:5293589F5ADF8EF4F7D8E8EFD89BF54C:::  
v:1003:FDCFC2AFB2D1BE34AAD3B435B51404EE:9AF153E5C3A2B66026B89C11898FEAF7:::
```

Pongamos que esos datos los tenemos guardados en un archivo llamado z.txt

Estos hashes corresponden a un equipo Windows 2000 server con dirección ip 172.28.0.25, la ip no es importante, perfectamente podría haber sido una de Internet, la pongo para que se entienda que no es el mismo equipo en la que se ejecuta el proxy... veamos como sería mi configuración:

Equipo Atacante: 172.28.0.9

Equipo Víctima: 172.28.0.25, que es el que hemos obtenido sus hashes

Vale, ahora a por el proxy....

Se trata de una utilidad muy antigua llamada smbproxy, esta pequeña maravilla "bindeará" un puerto 139 o 445 y establecerá una conexión con la víctima e inyectará sus propios hashes en una autenticación SMB (que el solito crea) y una vez conseguido la inyección en la SAM bastará con hacer un net use para tener el disco duro de la víctima en nuestro equipo.

### **Bien, pero antes hay que preparar el terreno....**

1º) Vamos a rebajar al máximo los niveles de autenticación LM/NTLM y SMB en el equipo atacante  
2º) Hay que parar "algunos" servicios del equipo atacante, al igual que ocurre con smbrelay, smbproxy no puede detectar un servidor "legítimo".

El primer punto no será preciso, pero por si acaso... por si nos da por esnifar o por si el "destino" admite cifrado SMB, pues nosotros no!! Que somos más chulos que un ocho.

Para ello vamos a Inicio-Configuración-Panel de Control-Herramientas administrativas-Directivas de seguridad local

En la consola de Seguridad local pinchamos en Directivas Locales-Opciones de seguridad y buscamos todas las directivas que tengan algo que ver con el Cifrado SMB, Nivel de autenticación, etc. Deshabilitamos los cifrados, Las firmas SMB, el envío de contraseñas cifradas, el nivel de autenticación que sea el más bajo que os permita (LM/NTLM)

Leed bien las directivas, a veces son engañosas, p.e. hay una que pone:

Enviar contraseña no cifrada para conectarse a servidores SMB de otros fabricantes.... bueno pues en este caso debemos poner **Habilitada**, puesto que emplea una lógica típica de M\$, pusieron el "no" delante de "cifrada..." y la cagan, que les costaría ponerlo bien.

El segundo punto es más fácil.... sólo tenemos que deshabilitar los servicios de servidor de nuestro propio equipo, para ello:

Inicio-Ejecutar-cmd.exe (una shell....)

Y ponemos net stop server

Claro tendremos que ser administradores, pero es que es nuestro equipo ..... os preguntará si realmente deseáis parar el servicio de servidor y sus dependencias, que son el DFS y El Examinador de equipos., le decis que sí....

Una vez detenidos los servicios, vuestra máquina está fuera de la red... no tendréis conexión y otros equipos no tendrán conexión SMB-NetBios con vosotros... hasta que montemos el proxy, claro.

Ahora el proxy...

Abrimos 2 ventanas cmd.exe en nuestro equipo... bueno realmente una ya la tenemos de antes... eso que necesitamos 2:

En una escribimos:

***Smbproxy -s 172.28.0.25 -f z.txt -v***

## Explicación:

-s ip.del.equipo.victima  
-f nombre del fichero que tiene los hashes  
-v Salida detallada

Saldrá algo así:

SMBproxy by Patrik Karlson <patrik.karlsson@ixsecurity.com>

-----  
INFO: Proxy started on ip 127.0.0.1 port 139  
INFO: Proxying traffic to 172.28.0.25  
INFO: Loaded 6 Hashes  
INFO: Waiting for connection

Parecerá que está colgado pero como dice en su última línea, está esperando una conexión....

Ahora "pasamos" a la otra shell de nuestro propio equipo, y vamos a conectarnos a un recurso, por ejemplo a C\$

## Escribimos:

*Net use \* ||127.0.0.1|c\$ jaja-jojo /u:administrador*

## Explicando:

Net use permite crear una conexión de red

El asterisco (\*) indica que se utilice la primera letra de unidad disponible, podríamos haber puesto x:, k:, etc. Pero como así no nos preocuparemos, asignará la primera que sea válida (D:,E:,F:,G:.... depende de las que ya existan...)

127.0.0.1 es la dirección del proxy.... claro SOMOS NOSOTROS

c\$ Identifica a la unidad C:, es un recurso administrativo "que suele estar", sino, podríamos probar con admin\$, o descubrir sus recursos compartidos....

Jaja-jojo es un password cualquiera, obviamente no es el que corresponde a esa máquina, pero hay que darle uno, no importa el que sea. El verdadero pass de esa máquina es adm (si lo pasáis por el crack lo veréis rápidamente al tratarse de tan pocas letras.

/u:administrador, está claro, no? El usuario con el que intentamos autenticarnos.

Si todo fue bien... veremos que el comando se ha completado correctamente (joder, que le pusimos un pass que no era...) y en la otra ventana de la shell veremos que se estableció la conexión y se ha empezado a recibir-enviar datos....

Por supuestísimo que tendremos una unidad D-E-F-G-H.... en nuestro PC que corresponderá al disco del equipo víctima....

### Finalizando....

En una cosa os he mentado...., en esto:

*Equipo Proxy: 172.28.0.9 (en este caso es el mismo, pero podría ser otro... ya me entendéis, no?, aunque bien pensado, da igual.... se trata de un proxy transparente por lo que la ip no queda oculta...*

Realmente en windows no es así, el equipo que inicia el proxy de smbproxy será siempre el loopback (127.0.0.1) por lo que no podremos "usar otro" como proxy... no es así en LiNux, aja!!! Esto tambien funciona en LiNux, que sí se puede usar otra máquina "en medio" como proxy, buscad SMBClient y SMBProxy los linuxeros, ya sabéis que soy un "chico windows" y que de "lo otro" ni papa.

No penséis que sólo es posible conectarse a una unidad de red por éste método... también podríamos haber accedido al registro, acceso a otros servers usando pipes, en definitiva cualquier cosa que se comunique mediante smb y netbios.

Bueno, existe "otro" método, usando LC3 (es importante la versión, ha de ser Lc3 y no otra) y un script en perl que convierte los hashes capturados por LC3 al formato pwdump que es lo que necesita smbproxy....

Además hay otro problema con lc3, y es que el sniffing que hace SMBCapture sólo soporta medios conmutados (hub) por lo que adios a internet...yo lo he probado y el script es perl funciona a las mil maravillas, si queréis podéis hacerlo con vuestro propio equipo importando los hashes del registro o de la máquina local, luego le pasáis el script de perl y ya tenéis los hashes en formato pwdump.

Enlaces;

Smbproxy: <http://www.cqure.net/tools/smbproxy-win32bin-1.0.0.zip>

Pwdump y LC3

<http://www.atstake.com/research/lc/download.html>

bueno, ahí está Lc4, el otro os va a costar un poquito más encontrarlo.... pero estar estará, yo lo tengo, si no lo encontráis lo subiré aun ftp, es un trial de 15 días, suficiente para la práctica.... luego "a la tienda" o al ahorra-más.