

En este mini-tutorial, voy a explicar de la forma más sencilla posible el uso de algunas herramientas que nos pueden facilitar la tarea a la hora de realizar un análisis forense en entornos Windows.

Existen muchísimas herramientas destinadas a éste propósito, comerciales y gratuitas. En este tutorial vamos a ver como enfocáramos un análisis partiendo de herramientas gratuitas.

Si Dios quiere se irá revisando en un futuro para ir añadiendo contenido.

Éste tutorial está en su revisión 0.1 así que no le pidáis mucho.. ☺

Cuando un usuario no “autorizado” toma el control de un sistema, éste puede instalar múltiples backdoors (puertas traseras) que le permitan entrar al sistema en un futuro, aunque “parcheemos” la vulnerabilidad original.

Se denomina “**análisis forense**” al proceso de analizar una copia completa de un sistema que ha sufrido una intrusión o ataque.

El “**análisis forense**” permite obtener la mayor cantidad posible de información sobre:

- ✚ El método utilizado por el atacante para introducirse en el sistema
- ✚ Las actividades ilícitas realizadas por el intruso en el sistema
- ✚ El alcance y las implicaciones de dichas actividades
- ✚ Las “puertas traseras” instaladas por el intruso

Realizando un “**análisis forense**” nos permitirá, entre otras cosas, recuperarnos del incidente de una manera más segura y evitaremos en la medida de lo posible que se repita la misma situación en cualquiera de nuestras máquinas.

Un buen análisis forense debe dar respuestas a varias cuestiones, entre las que se encuentran las siguientes:

- ✚ ¿En que fecha exacta se ha realizado la intrusión o cambio?
- ✚ ¿Quién realizó la intrusión?
- ✚ ¿Cómo entró en el sistema?
- ✚ ¿Qué daños ha producido en el sistema?

Si una vez realizado el análisis forense no conocemos con exactitud las respuestas a estas preguntas, no tendremos un análisis al 100 % de veracidad. Esto puede derivar en futuros ataques, bien por la misma persona, o bien por diferentes medios de intrusión que desconozcamos.

### **Estudio preliminar (Primer paso)**

El primer paso de cualquier análisis forense es el análisis preeliminar. Nos deben o debemos explicar con la mayor exactitud posible qué ha ocurrido, qué se llevaron o intentaron llevar y cuándo ocurrió.

También tendremos que recoger información sobre la organización, ya sea organización, casa, etc...

Recogeremos información sobre la tipología de red y de gente directa o indirectamente implicada.

También podríamos recoger información sobre el tipo de escenario y el/los sistema/s afectado/s.

### **¿Apagamos el equipo?**

Podemos presentarnos con dos casos. El primero es el de no apagar el equipo. Si no apagamos el equipo, podremos ver todos los procesos en ejecución, los consumos de memoria, las conexiones de red, los puertos abiertos, los servicios que corren en el sistema, etc.

También se nos presenta el problema de que si apagamos el equipo, se perderá **información volátil** que puede ser esencial para el curso de la investigación.

La parte mala de esta situación es que el sistema, al poder estar **contaminado**, éste puede ocultar la información. También se nos presenta el problema de que si no apagamos el sistema, éste puede comprometer a toda la red.

Si no apagamos el sistema tendremos que controlar este aspecto de la seguridad, y aislarlo completamente de la red, lo cual llega a ser prácticamente imposible en determinados escenarios.

### Captura de la Evidencia (Segundo paso)

El segundo paso consiste en la captura de la/s evidencia/s. Por *evidencia entendemos toda información que podamos procesar en un análisis*. Por supuesto que el único fin del análisis de la/s evidencia/s es saber con la mayor exactitud qué fue lo que ocurrió.

Bueno, y ¿qué entendemos por evidencia? Podemos entender evidencia como:

- ✚ El último acceso a un fichero o aplicación (unidad de tiempo)
- ✚ Un Log en un fichero
- ✚ Una cookie en un disco duro
- ✚ El uptime de un sistema (Time to live o tiempo encendido)
- ✚ Un fichero en disco
- ✚ Un proceso en ejecución
- ✚ Archivos temporales
- ✚ Restos de instalación

En definitiva, cualquier cosa que nos ayude a esclarecer cualquier pregunta de las formuladas anteriormente.

Una consideración a tener en cuenta es que el proceso del análisis debe estar *perfectamente documentado*, ya que si por causas ajenas tenemos que dejar la investigación, otra persona, y partiendo de los datos documentados y las herramientas necesarias, pueda llegar al mismo punto, y proseguir con la investigación.

### ¿Por qué (casi) no se utilizan herramientas gráficas en un análisis forense?

Una de las cosas más importantes a la hora de realizar un *análisis forense* es la de no alterar el escenario a analizar. Esta es una tarea prácticamente imposible, porque como mínimo, alteraremos la memoria del sistema al utilizar cualquier herramienta.

Las herramientas que utilicemos deben de ser lo menos intrusivas en el sistema, de ahí que se “huya” de las herramientas gráficas, las que requieren instalación, las que escriben en el registro, etc..

Lo normal y lógico sería utilizar herramientas ajenas al sistema comprometido, ya sean herramientas guardadas en cualquier soporte (CD-ROM, USB, etc...). Esto lo hacemos para no tener que utilizar las herramientas del sistema, ya que pueden estar manipuladas y arrojar falsos positivos, lecturas erróneas, etc...

### Ya hemos capturado los datos volátiles. ¿Cuál es el siguiente paso?

Una vez que hayamos capturado toda información volátil, es necesario realizar la captura de los datos físicos del sistema, es decir, de los discos físicos. Aunque parezca una *burrada* lo lógico en este caso es pegarle el típico *botonazo* al equipo. Con esto conseguimos garantizar que el sistema no *consigue* borrar posibles evidencias, tales como archivos temporales, copias de troyanos, Logs, etc...

Una de las técnicas más fiables es la duplicación del sistema, bien sea clonándolo o haciendo una imagen exacta.

La herramienta por excelencia es la aplicación *dd*. Ésta herramienta tiene su versión para WIN32, es decir, para sistemas Windows.

Podríamos arrancar el sistema comprometido con un Live CD de Linux, y enviar por red mediante *netcat* a otro equipo una imagen del disco. Podríamos hacerlo de la siguiente manera:

```
dd id=/dev/hda | nc 181.102.99.2 55555
```

Lógicamente en el equipo que recibe tendríamos que tener netcat en funcionamiento tal que así:

```
nc -l -p 55555 > hda.dd
```

Una vez realizada la imagen, lo suyo sería volcarla en una máquina con las mismas características, es decir, con el mismo hardware. Aunque muchas veces esto no sería posible.

**Comencemos con el análisis!**

El primer paso que voy a realizar será una captura de los datos físicos de la máquina, es decir, recoger diversos datos como:

- ✚ Dueño de la máquina (Organización)
- ✚ Tipo de BIOS
- ✚ Uptime del sistema (Time to live)
- ✚ Directorios del sistema
- ✚ Número de tarjetas de red
- ✚ Número de servipacks o updates del sistema
- ✚ Ubicación del archivo de paginación
- ✚ Tipo de procesador
- ✚ Fabricante y modelo del sistema
- ✚ Número de procesadores
- ✚ Tamaño de la memoria RAM
- ✚ Versión del sistema operativo
- ✚ Etc...

Esto lo podemos hacer con varias herramientas, de las cuales yo sólo voy a describir dos.

**SystemInfo (Nativa de Windows)**

Aplicación nativa de Windows que nos muestra información acerca de la configuración del sistema y el tipo y versión del sistema operativo. También nos muestra información relevante a seguridad, propiedades del Hardware, memoria RAM, espacio total del disco e información sobre las tarjetas de red. Su sintaxis es la siguiente:

**Systeminfo [/s equipo [/u dominio\nombreUsuario [/p contraseña]]] [/fo {TABLE | LIST | CSV}]  
[/nh]**

Podríamos redireccionar la salida del comando a un fichero de texto y fechado, para saber con exactitud cuándo se tomó la evidencia. El comando que podríamos poner sería el siguiente:

**Systeminfo /FO list >SystemInfo.txt &date /t >>SystemInfo.txt &time /t >>SystemInfo.txt**

Un ejemplo de lo que nos daría una salida de este comando lo podemos ver en éste fichero [adjunto](#)

**PsInfo (SysInternals)**

Esta herramienta es similar a la nativa de Windows, y podremos conseguir prácticamente los mismos resultados. Ambas herramientas permiten su uso tanto en local como en remoto. Su sintaxis es la siguiente:

**psinfo [[\\computer [, computer [...]] | @file [-u user [-p psswd]]] [-h] [-s] [-d] [-c [-t delimiter]]  
[filter]**

No necesita más explicación.

El **segundo paso** que voy a realizar será recopilar información acerca de los servicios que hay corriendo en la máquina con sus estadísticas. En este punto voy a utilizar el comando nativo de Windows **net** y el comando **SC**.

Con el comando **net statistics** voy a recabar información acerca de los bytes recibidos por el sistema, el número de inicios de sesión fallidos, las cuentas de uso fallidas, etc... Toda esta información la almacenaremos en un archivo de texto con fecha y hora incluida para su posterior análisis. El comando resultante podría ser el siguiente:

**Net statistics Workstation >Estadisticas.txt &date /t >>Estadisticas.txt &time /t >>Estadisticas.txt**

El comando SC me va a permitir conseguir una lista de los servicios que actualmente están corriendo en la máquina. Aunque SC posee muchos comandos para poder regular su salida, un comando resultante válido podría ser el siguiente:

```
SC query >ServiceOpen.txt &date /t >>ServiceOpen.txt &time /t >>ServiceOpen.txt
```

El **tercer paso** que voy a realizar será la recopilación de supuestos procesos maliciosos, puertos de escucha, identificación de aplicaciones no autorizadas y la finalización de procesos legítimos. Para saber cuantas conexiones tengo abiertas puedo utilizar el comando nativo de Windows **netstat**. Utilizaré la opción **-a** para conocer todas las conexiones y puertos de escucha, y la juntaré con la opción **-b** para conocer el ejecutable que crea la conexión necesaria para llegar al TCP/IP. El comando resultante fechado para su posterior análisis quedaría:

```
netstat -ab >Conexiones.txt &date /t >>Conexiones.txt &time /t Conexiones.txt
```

Para saber los procesos que tenemos actualmente corriendo en nuestro sistema utilizaremos la aplicación nativa de Windows tasklist, o en su defecto pslist (sysinternals). También utilizaremos la herramienta Fport. Ambas herramientas (tasklist, pslist) permiten realizar esta operación en local como en remoto. El comando resultante quedaría:

```
Tasklist >Procesos.txt &date /t >>Procesos.txt &time /t >>Procesos.txt
```

También vamos a recopilar información sobre los servicios que dependen de los procesos que están en funcionamiento. También utilizaremos el comando Tasklist, y como resultado el comando sería:

```
Tasklist /SVC >ProcesosYServicios.txt &date /t >>ProcesosYServicios.txt &time /t  
>>ProcesosYServicios.txt
```

Muchas veces cuando en el sistema hay determinados rootkits, virus o troyanos, éste no nos muestra una salida “coherente”, de ahí a que siempre que podamos utilicemos aplicaciones que sean lo menos intrusivas en el sistema. Si somos un poco “paranoicos” en ese tema, para ver los puertos abiertos en un sistema podemos utilizar la herramienta **fports**. Por regla general, no nos debemos fiar de un sistema en el que haya corriendo este tipo de virus.

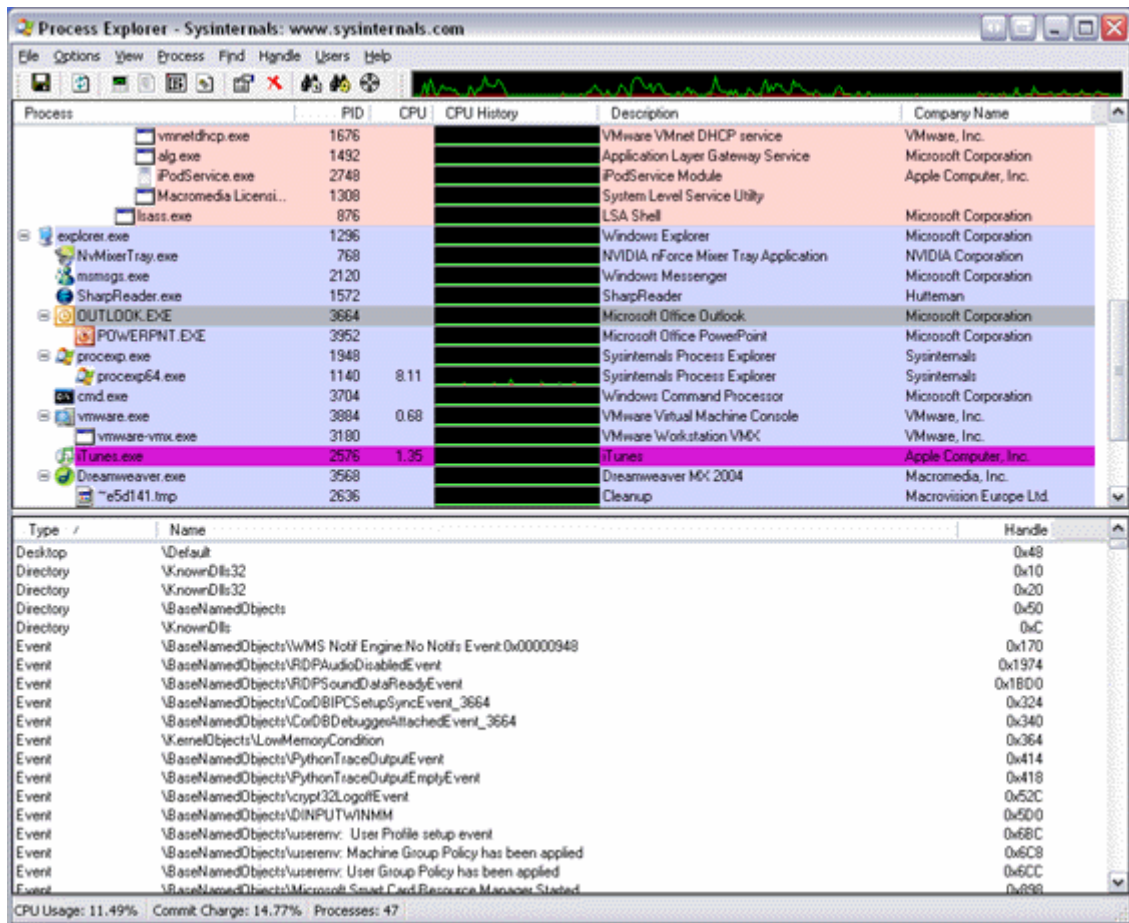
Básicamente **fports** nos muestra la misma salida que si ejecutásemos el comando nativo netstat con el filtro **-a** y **-n**. También puede identificar puertos desconocidos que estén abiertos, con sus correspondientes procesos y PID. La salida a este comando sería la siguiente:

```
C:\>fport  
FPort v2.0 - TCP/IP Process to Port Mapper  
Copyright 2000 by Foundstone, Inc.
```

<http://www.foundstone.com>

```
Pid Process Port Proto Path  
392 svchost -> 135 TCP C:\WINNT\system32\svchost.exe  
8 System -> 139 TCP  
8 System -> 445 TCP  
508 MSTask -> 1025 TCP C:\WINNT\system32\MSTask.exe  
392 svchost -> 135 UDP C:\WINNT\system32\svchost.exe  
8 System -> 137 UDP  
8 System -> 138 UDP  
8 System -> 445 UDP  
224 lsass -> 500 UDP C:\WINNT\system32\lsass.exe  
212 services -> 1026 UDP C:\WINNT\system32\services.exe
```

También podremos ver los procesos que corren en la máquina de forma gráfica con la herramienta de sysinternals ProcessExplorer.exe, la cual nos ofrece prácticamente la misma información pero de forma gráfica. Una captura de pantalla de la aplicación:



Ya sólo nos queda recabar información sobre los módulos que cargan estos procesos. Podemos averiguar por ejemplo qué DLL están asociadas a un determinado proceso. Así tendremos un control más exhaustivo sobre los procesos. Para recabar esta información podemos utilizar la herramienta de sysinternals ListDLLs.exe.

Por ejemplo, si quisiésemos averiguar qué DLL dependen del proceso con PID 1548 utilizaríamos la siguiente sintaxis:

```
ListDLLs.exe 1548 >DLL1548.txt &date /t >>DLL1548.txt &time /t>>DLL1548.txt
```

Puede que necesitemos también extraer el contenido de la memoria de un proceso para su posterior análisis. En tal caso utilizaremos la aplicación **PMDUMP**, la cual se encarga de hacernos el trabajo simplemente con este comando:






```
Pmdump <pid> <fichero>
```

El **cuarto paso** que voy a realizar será una recopilación de los últimos accesos a ficheros, clasificados por fechas. Esta lista me servirá de referencia a la hora de realizar el análisis, y podré comprobar qué ficheros se modificaron en el día o los días en los que el sistema estuvo comprometido.

Podremos utilizar varias herramientas destinadas a tal fin, pero yo voy a utilizar sólo dos.

En una primera instancia podré utilizar el comando nativo de Windows **DIR**, con algunas reglas para que me muestre los ficheros modificados conforme a la fecha. Podría utilizar el siguiente comando:

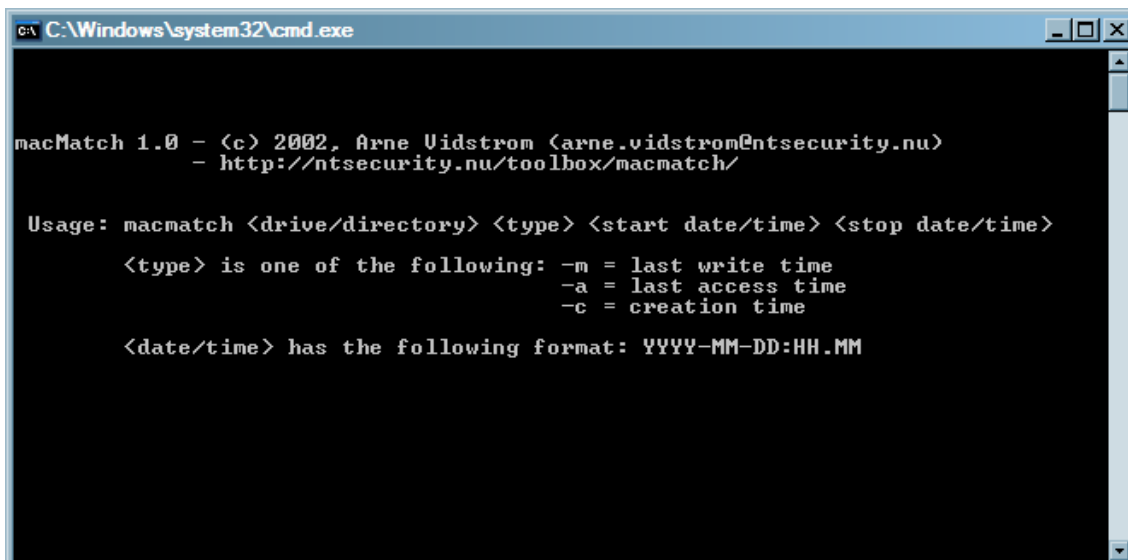
**DIR /t: a /a /s /o: d c:\ >Directory.txt &date /t >>Directory.txt &time /t >>Directory.txt**

-  /t:a Nos muestra el campo del último acceso (Fecha)
-  /a Muestra todos los ficheros
-  /s Muestra todos los archivos del directorio especificado, incluidos los subdirectorios
-  /o Lista los archivos indicados
-  d Muestra los más antiguos primero (Por fecha y hora)

Como siempre poniéndole la fecha al final para saber cuándo tomamos esa prueba.

En varias ocasiones esta lista puede ser larguísima y el fichero puede ocuparnos unos cuantos megas. La herramienta que voy a describir a continuación puede ayudarnos a buscar archivos en fechas concretas. La herramienta en sí se llama MacMatch.exe. Ésta herramienta básicamente buscará ficheros modificados en un intervalo de tiempo, que lógicamente se lo daremos nosotros.

Una captura de imagen con la sintaxis:



```
C:\Windows\system32\cmd.exe

macMatch 1.0 - (c) 2002, Arne Uidstrom (arne.uidstrom@ntsecurity.nu)
- http://ntsecurity.nu/toolbox/macmatch/

Usage: macmatch <drive/directory> <type> <start date/time> <stop date/time>

<type> is one of the following:
-m = last write time
-a = last access time
-c = creation time

<date/time> has the following format: YYYY-MM-DD:HH.MM
```

Por ejemplo, si quisiésemos saber qué ficheros se “tocaron” o modificaron entre el 10 y el 12 de Noviembre, sobre las 15 horas, el comando a poner sería el siguiente:

**Macmatch.exe c:\ -a 2005-11-10:15.00 2005-11-12:15.59**

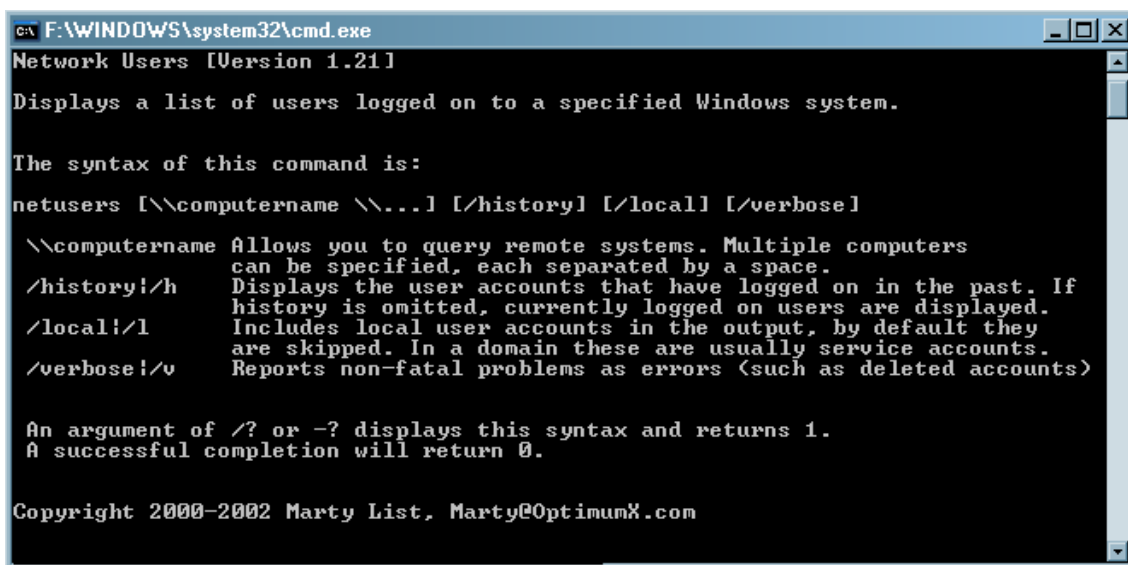
El **quinto paso** que voy a realizar será una recopilación de diversos aspectos de los usuarios. Trataré de recabar la siguiente información:

- ✚ Relación de usuarios creados en el sistema
- ✚ Relación de las últimas sesiones (LogOn) fallidas en el sistema
- ✚ Relación de las últimas sesiones establecidas remotamente en el sistema
- ✚ Relación de las actividades de los usuarios remotos
- ✚ Tiempo de logeo en el sistema
- ✚ Histórico de los usuarios logeados localmente en el sistema

**Para utilizar estas herramientas es necesario verificar que estén activados los controles de auditoría.**

Para realizar estas operaciones voy a utilizar tres aplicaciones. Una aplicación llamada netusers, otra llamada ntlst y una aplicación llamada psloggedon.

Con netusers podremos comprobar los usuarios que están conectados actualmente a una máquina remota o local. Su sintaxis es la siguiente:



```

C:\F:\WINDOWS\system32\cmd.exe
Network Users [Version 1.21]
Displays a list of users logged on to a specified Windows system.

The syntax of this command is:
netusers [\computername \...] [/history] [/local] [/verbose]

\computername Allows you to query remote systems. Multiple computers
                can be specified, each separated by a space.
/history!h      Displays the user accounts that have logged on in the past. If
                history is omitted, currently logged on users are displayed.
/local!l        Includes local user accounts in the output, by default they
                are skipped. In a domain these are usually service accounts.
/verbose!v      Reports non-fatal problems as errors (such as deleted accounts)

An argument of /? or -? displays this syntax and returns 1.
A successful completion will return 0.

Copyright 2000-2002 Marty List, Marty@OptimumX.com
```

Para ver los usuarios que actualmente están conectados a la máquina local tan solo tendríamos que poner el comando **netusers >usuarios.txt &date /t >>usuarios.txt &time /t>>usuarios.txt**.

Si queremos que nos muestre un histórico de usuarios que se han logueado anteriormente, pondríamos **netusers /history >UsersHistory.txt &date /t >>UsersHistory.txt &time /t >>UsersHistory.txt**.

Otra aplicación que nos sirve para el mismo propósito es PsLoggedOn. Esta herramienta la podemos utilizar tanto en local como en remoto, y la salida que nos muestra por defecto es la siguiente:

- ✓ Usuarios locales autenticados en el sistema
- ✓ Usuarios logueados a través de recursos compartidos
- ✓ Hora de inicio de sesión

Bueno, hasta aquí todo de lujo no? Ahora me queda mirar en el archivo de seguridad del visor de sucesos los sucesos auditados, como inicios de sesión fallidos, inicios de sesión correctos, alguna operación con privilegios, etc...

Si tuviésemos que mirar uno a uno todos esos sucesos, prácticamente nos sería casi imposible de terminar, debido a la longitud del fichero. Aquí es donde actúa la herramienta NtLast.

Por medio de esta herramienta podremos averiguar de forma sencilla todos y cada uno de los sucesos del sistema.

En una primera instancia voy a sacar un fichero con los últimos 100 inicios de sesión exitosos, incluidas las sesiones **nulas** en caso de que el sistema no tuviese parcheada esta opción. El comando resultante sería:

```
NtLast -null -v -n 100 >>SuccessfulLogons.txt &date /t >>SuccessfulLogons.txt &time /t >>SuccessfulLogons.txt
```

Ahora voy a sacar un listado con los últimos 100 inicios de sesión fallidos. El comando resultante sería:

```
NtLast -v -n 100 >>LogonsFailed.txt &date /t >>LogonsFailed.txt &time /t >>LogonsFailed.txt
```

También me gustaría sacar un listado con los últimos 100 inicios de sesión remotos. El comando resultante sería:

```
NtLast -v -n 100 -r >>RemoteLogin.txt &date /t >>RemoteLogin.txt &time /t >>RemoteLogin.txt
```

NtLast es una aplicación que puede dar mucho juego a la hora de sacar listados. Además de todo esto, que ya es mucho, podremos sacar todo esto pero referente a un solo usuario, por si tuviésemos alguna pista y tuviésemos que estrechar el cerco.

Por poner algún ejemplo, y si en la empresa X tuviésemos sospechas de que el usuario Juanito está cometiendo actividades impropias o no encomendadas, podríamos hacer lo siguiente:

```
NtLast -v -n 100 -r -u Juanito >>RemoteLoginJuanito.txt &date /t >>RemoteLoginJuanito.txt &time /t >> RemoteLoginJuanito.txt
```

Con este comando lo que obtenemos es una lista con los últimos 100 inicios de sesión remotos del usuario Juanito.

### **Bibliografía**

Windows XP Security Guide (Microsoft)  
The Services and Service Accounts Security Planning Guide (Microsoft)  
The Security Risk Management Guide (Microsoft)  
Guide to Securing Microsoft Windows XP (National Security Agency) (NSA)  
Red-Iris Seguridad  
Google ☺

