

Identificación y eliminación de malware en Windows XP

Ramiro Encinas Alarza – 2008
ramiro.encinas@gmail.com

Índice:

– Definición de malware.....	1
– Vías y formas de infección por malware.....	2
– Procesos fiables comunes.....	3
– Analizando procesos.....	5
– Dentro de un proceso.....	7
– Otras opciones útiles de Process Explorer.....	11
– Análisis longitudinal de procesos.....	11
– Verificación de archivos no firmados digitalmente.....	15
– Inicialización automática de malware.....	17
– Rootkits.....	19
– Eliminación de malware.....	22
– Conclusión.....	25
– Referencias.....	26

Definición de malware

Puede denominarse como malware todo software que, una vez ejecutado en un sistema operativo realiza acciones no solicitadas por el usuario y que, normalmente, tiene fines ilegales y/o negativos hacia éste.

El malware, desde el punto de vista de la atención del usuario, se puede dividir en dos grupos:

- *Los que permanecen ocultos al usuario:* que son virus del tipo puertas traseras (backdoors), gusanos (worms), polimórficos, rootkits y demás fauna oculta.
- *Los que se presentan al usuario:* en forma de ventanas no solicitadas, avisos o formularios que pretenden modificar el comportamiento del usuario. Los fines son publicitarios, engañosos para hacerlo formar parte en procesos ilícitos, publicidad engañosa, SPAM, phishing, ingeniería social, etc.

Vías y formas de infección por malware

Las vías físicas por donde accede el malware son cualquiera de los dispositivos de entrada y salida que tiene un sistema operativo. Actualmente, las vías más comunes son Internet (dispositivos de red) y los dispositivos USB, como los pendrives. Atrás quedaron los tiempos de infección mediante el sector de arranque de los disquetes, aunque todavía se da algún caso.

Podemos clasificar en dos grupos las formas de infectar con malware un sistema operativo:

- *Con intervención del usuario:* el usuario con los privilegios suficientes ejecuta un proceso maligno de forma no intencionada. La forma más común es abrir algún archivo ejecutable infectado descargado de la red o de un dispositivo de almacenamiento (pendrive, disquete, etc).

Otra de las formas más sigilosas y peligrosas que existen es que el usuario abra algún archivo no ejecutable especialmente diseñado con una aplicación vulnerable. En estos casos y dependiendo del tipo de vulnerabilidad, la aplicación afectada fomenta la infección según el contenido “especial” del archivo no ejecutable que fue abierto, todo esto sin que se entere el usuario.

Este último caso es bastante frecuente en muchas aplicaciones y, por ejemplo, desde hace tiempo existía la famosa vulnerabilidad de Adobe Acrobat Reader (versiones inferiores a la 8.1.1) junto con Microsoft Internet Explorer 7. Este tipo de infección se produce al abrir un documento PDF modificado a propósito con una versión de Adobe Acrobat Reader vulnerable. El contenido de este documento PDF puede utilizar la función (mailto:) de Microsoft Internet Explorer 7 parametrizada de una forma especial que permite la ejecución de cualquier archivo (malware incluido), ya sea de forma local o red, y al margen del usuario.

- *Sin intervención del usuario:* algún servicio del sistema operativo recibe un ataque puntual desde otro host a través de un dispositivo de entrada activo. Los objetivos más comunes en este tipo de ataque son servicios de red vulnerables. Mediante el servicio vulnerado se ejecuta el proceso vírico con los mismos privilegios que tiene el servicio (normalmente son muy elevados).

Otra vía de ataque sin intervención del usuario es la autoejecución del contenido del archivo *autorun.ini* ubicado en algunos CD's y pendrives al introducirlos en el sistema. Para evitar este comportamiento hay que desactivar la reproducción automática desde las plantillas administrativas del sistema con la herramienta *gpedit.msc*.

En cualquier caso, es necesario que algún componente del sistema operativo víctima sea vulnerable de alguna manera. El componente más fácil de vulnerar es el usuario.

El malware, en muchas ocasiones, una vez vulnerado algún servicio de entrada, tiende a su persistencia en el sistema (como los backdoors) alojándose en el disco duro en forma de archivo ejecutable (con extensión exe, dll, drv, ocx, sys) o de script (vbs, bat, cmd) ejecutándose en cada inicio mediante la inserción o modificación de alguna entrada en el registro de Windows o mediante el uso de alguno de los procedimientos de inicialización de procesos de Windows. Esto lo veremos más tarde en el apartado **Inicialización automática del malware**.

Procesos fiables comunes

En un sistema operativo como Windows XP, en condiciones normales, existen muchísimos procesos en ejecución. Cada uno de estos procesos realiza una función o funciones distintas según su naturaleza dentro de una jerarquía.

Identificar un proceso de malware puede ser sencillo o muy complicado, depende de las herramientas e intuición según la experiencia. En cualquier caso es conveniente conocer bien los procesos de Windows XP más importantes que se suponen “fiables”. También hay que conocer los procesos que no son de Windows XP y también son fiables; éstos últimos suelen ser de otros fabricantes. El resto de procesos que quedan serán sospechosos.

Los procesos de Windows XP, (o de sistema) son bastantes y están repartidos por todos los niveles del sistema operativo, desde el núcleo hasta las aplicaciones de usuario. La mayoría de los archivos de estos procesos se encuentran en la carpeta *C:\Windows\system32* y los podemos clasificar de la siguiente forma:

- *Componentes centrales:* para empezar, el proceso padre de todos los procesos es el kernel de Windows representado en el archivo *C:\WINDOWS\system32\Ntoskrnl.exe*. Este es el archivo del proceso maestro que se encarga entre otras cosas de la gestión principal de procesos y administración de la memoria. Con él se encuentra *C:\WINDOWS\system32\hal.dll* que se encarga, durante el arranque de la máquina, de recibir de la BIOS los datos de los dispositivos hardware reconocidos y ofrecerlos a Windows de acuerdo a la configuración de hardware del registro de Windows.

En la misma carpeta también tenemos a *Win32k.sys* que se encarga del subsistema gráfico y otras funciones base del subsistema Win32, además de *Ntdll.dll* (muy atacado) que procesa las funciones internas y los servicios del sistema a nivel ejecutivo y por último tenemos a *Kernel32.dll*, *Advapi32.dll*, *User32.dll* y *Gdi32.dll* que terminan de completar el subsistema central e interactúan con *Ntoskrnl.exe* y *Win32k.sys* para ofrecer las API's principales de Windows a las aplicaciones de los usuarios.

- *Drivers o controladores de sistema:* son procesos que controlan los distintos dispositivos hardware instalados en la máquina. Se distinguen dos tipos:
 - Los drivers ya incorporados por Microsoft en la instalación de Windows XP. Los fabricantes de los drivers aportan a Microsoft los archivos de los drivers para incluirlos en el CD de instalación de Windows XP. Los drivers de este tipo suelen localizarse en la carpeta *C:\Windows\system32\drivers*. Ejemplos de estos drivers son los controladores de la mayoría de las tarjetas gráficas, modems, tarjetas de red, etc. Otros drivers más estandarizados (IDE, PCI, USB, etc.) y otros componentes centrales ya los incorpora Microsoft en el CD de Windows XP.
 - Los archivos de los drivers que instala el usuario cuando compra un dispositivo para la máquina (como una tarjeta de video) pueden alojarse también en *C:\Windows\system32*. En la Figura 1, mediante el administrador de dispositivos, podemos ver la ubicación de los drivers de la tarjeta gráfica NVIDIA GeForce FX5200.

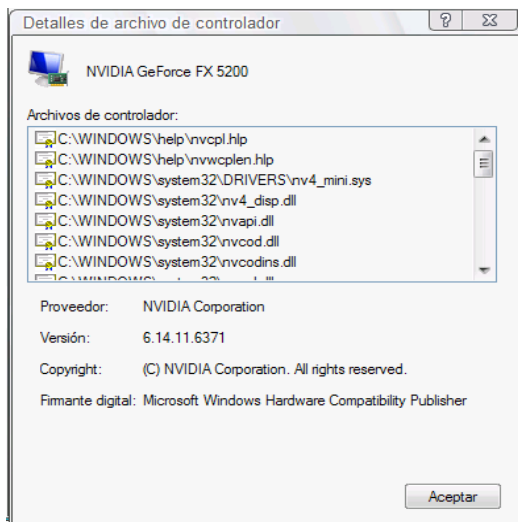


Figura 1. Archivos del driver de NVIDIA GeForce FX5200

Como vemos en la Figura 1, los archivos del driver de NVIDIA están identificados por su Proveedor, Versión, Copyright y están firmados digitalmente en la lista de compatibilidad de hardware de Microsoft. Por tanto estos archivos y sus correspondientes procesos son, en principio, fiables.

- *Procesos de administración de sesión:* aquí nos encontramos con tres procesos muy importantes que son:
 - El subsistema de administración de sesión: *C:\Windows\system32\smss.exe*
 - El subsistema Win32 en modo usuario: *C:\Windows\system32\csrss.exe*
 - El administrador de inicio de sesión: *C:\Windows\system32\winlogon.exe*
- *Servicios de Sistema:* este grupo es el más amplio. Aquí tenemos al servicio LSASS ubicado en *C:\Windows\System32\lsass.exe* que administra la seguridad del sistema, controla a las cuentas de los usuarios, ofrece el almacenamiento protegido y el inicio de sesión en red y a *C:\Windows\System32\services.exe* que es el padre de los servicios comunes de Windows (los famosos *svchost.exe* y alguno que otro más).

Cada *svchost.exe* activo en memoria es un grupo de servicios y los grupos principales son:

- DCOMLAUNCH: Iniciador de procesos DCOM, representado en el archivo *C:\Windows\system32\rpcss.dll*. En este grupo también se inicia el servicio de Terminal Server *C:\Windows\system32\termsrv.dll*.
- RPCSS: llamada a procedimiento remoto *C:\WINDOWS\system32\rpcss.dll*.
- NETSVCS: varios servicios de red como cliente DHCP, BITS, NLA y otros más como Audio de Windows, Administración de discos lógicos, etc.
- NETWORKSERVICE: servicio de caché DNS *C:\Windows\system32\dnsrslvr.dll*.
- LOCALSERVICE: alberga al servicio Cliente web, Registro Remoto, etc.
- IMGsvc: contiene el servicio de Adquisición de Imágenes de Windows (WIA).

Aparte de los *svchost.exe*, *services.exe* tiene más servicios hijos que pueden ser cargados por el usuario, como antivirus o agentes de aplicaciones. También hay que destacar aquí otros servicios como *ALG C:\Windows\system32\alg.exe* que da compatibilidad a otros productos que no son de Microsoft para el firewall de Windows y también ofrece el servicio de conexión compartida de internet, y el servicio de cola de impresión representado en *C:\WINDOWS\system32\spoolsv.exe*.

- *Aplicaciones gráficas de usuario:* en este nivel el usuario interactúa a nivel gráfico con el escritorio de Windows. El proceso principal es *C:\Windows\explorer.exe* y a partir de él se crean los procesos de las aplicaciones en modo gráfico que ejecuta el usuario en su sesión.

Analizando procesos

Aparte de los procesos anteriores, que es lo que normalmente carga Windows XP en condiciones normales, también existen otros muchos procesos que tenemos que identificar. Existen muchas herramientas tanto de Microsoft como de otros fabricantes para identificar procesos.

La mejor herramienta que existe para esta tarea es Process Explorer de Sysinternals. Esta herramienta es gratuita y se puede descargar desde <http://technet.microsoft.com/en-us/sysinternals/default.aspx>

Abrimos Process Explorer y vamos a visualizar los datos relevantes desde View, Select Columns... en la pestaña Process Image marcamos Description, Company Name, Version y Command Line. En la pestaña Process Memory marcamos Working Set Size. En la pestaña DLL marcamos Description, Version, Time Stamp, Name, Company Name, Verified Signer y Mapped Size. También en el menú View, marcaremos la opción Show Lower Pane. Dentro de View, en la opción Lower Pane View, marcaremos DLLs. Con esto tendremos una visión general de los procesos que están corriendo tal y como vemos en la Figura 2:

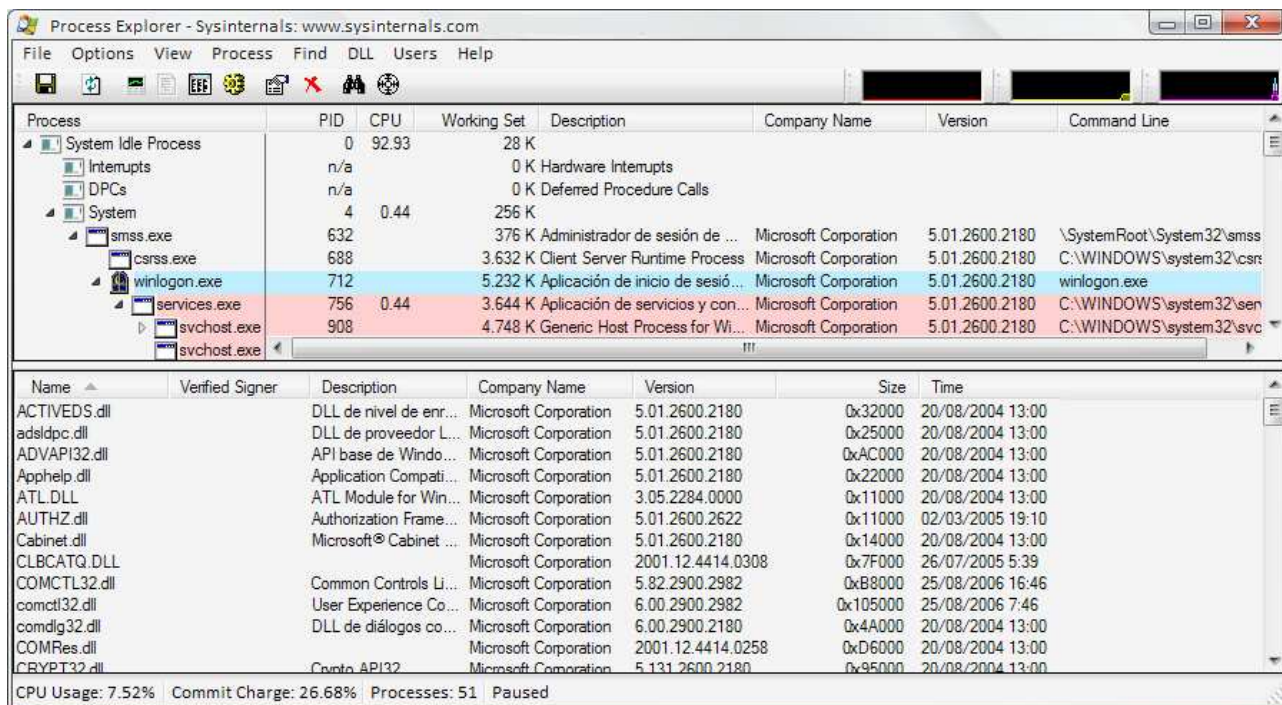


Figura 2. Vista principal de procesos de Process Explorer

Como podemos ver en la Figura 2, tenemos los procesos jerarquizados en la columna izquierda del panel superior, y a la derecha de ellos su número de proceso o PID y más información. En este punto los procesos más fiables son los que hemos descrito antes.

Es importante revisar con mucha atención los nombres de los procesos para comprobar que son correctos y no pretenden engañarnos con otros similares. Existe malware que aprovecha la similitud con los nombres de procesos legítimos para elegir su nombre; por ejemplo: el proceso `csrss.exe` es legítimo de Windows y si encontramos un proceso llamado `scrss.exe` es muy probable que sea malware debido a la similitud con el primero, porque cambia el orden de dos letras para confundirnos. Aparte de la deducción anterior, es conveniente mirar más información del proceso sospechoso para cerciorarse.

También hay que poner especial atención a procesos que no tengan nombre o si alguno de sus campos tiene caracteres extraños.

Un campo que denota confianza es *Company Name* donde viene reflejado el nombre del fabricante del proceso. En caso de que no conozcamos el fabricante podemos buscarlo en Internet. Si *Company Name* está en blanco, podemos sospechar del proceso.

El campo *Description* da información acerca del proceso. Hay muchos procesos que no tienen descripción y no por ello son malware, pero en cualquier caso también lo tendremos en cuenta.

Si nos encontramos con el campo *Version* en blanco hay que sospechar. La mayoría del malware tiene este campo en blanco. Todo proceso fiable debería tener indicada su versión.

El campo *Working Set* da el tamaño en kilobytes del proceso en memoria. Dependiendo del proceso este dato puede variar mucho. Para hacernos una idea, los `svchost.exe` suelen oscilar entre 2 y 7 Mb excepto el `svchost.exe` de NETSVCS (que contiene bastantes servicios) que suele oscilar entre los 20 y los 30Mb.

Cuando el sistema está recién iniciado, en reposo y sin ninguna aplicación abierta, los *Working Set* y el tiempo de CPU no suelen variar. Un comportamiento sospechoso se produce cuando el sistema está en reposo y vemos uno o más procesos que realizan grandes fluctuaciones o grandes consumos de CPU y *Working Set* (descartando que sea Windows Update o algún otro agente que se active sólo). En estos casos también podemos notar ralentización del sistema.

El campo *Command Line* nos muestra la ruta completa y el nombre de archivo ejecutado del proceso. Esto es muy útil para ubicar en el disco duro al archivo del proceso y poder eliminarlo después en caso de ser un proceso malware. Los archivos ejecutables de procesos sospechosos suelen ubicarse en carpetas temporales (TMP, TEMP). También son sospechosos los procesos cuyo archivo ejecutable tiene nombres poco habituales (con símbolos o sintaxis poco comunes).

Otro comportamiento muy sospechoso es la creación y eliminación automática de procesos en poco intervalo de tiempo. En Process Explorer los procesos que se inician toman el color verde y los procesos que se eliminan tienen el color rojo antes de desaparecer.

Los procesos comprimidos o cifrados merecen especial atención (color púrpura). Los procesos malware de este tipo ofuscan su contenido para no revelarlo y no dejar evidencias. Un ejemplo de evidencia es que su contenido contenga alguna cadena de texto con una URL sospechosa donde pueda enviar información sin conocimiento del usuario.

Sospechosos son también los procesos Rundll32.exe, pero no por el proceso en sí (es un proceso fiable de Windows). Rundll32.exe puede ejecutar archivos DLL que pueden ser malware (una DLL es una librería que sólo puede ejecutarse con la llamada de un proceso padre, y en este caso el padre es Rundll32.exe). Es extraño encontrarse a Rundll32.exe durante mucho tiempo en memoria debido a que normalmente es utilizado en los asistentes de instalación de nuevo hardware. Si nos encontramos con este proceso, es conveniente saber cuál es el archivo DLL que está ejecutando e identificarlo.

Una ventaja que tiene el panel principal de procesos de Process Explorer es que podemos ordenarlos por campo pinchando en el título del campo en cuestión con el botón izquierdo del ratón. De esta forma podemos ordenar la vista de procesos, por ejemplo por versión, y entonces quedarán juntos los procesos que no tengan versión. De la misma forma se puede proceder con cualquier otro campo para agilizar la revisión.

Los procesos malware también suelen ocultarse dentro de los procesos svchost.exe. Veremos como localizarlos e identificarlos en el próximo apartado.

Dentro de un proceso

Cuando seleccionamos un proceso con Process Explorer, tendremos en el panel inferior una lista de referencias a otros archivos y DLL's encontradas en la memoria del proceso. Estos archivos se dividen en dos tipos: de imagen (image) y de datos (data). Los archivos de imagen son ejecutables (suelen tener extensión EXE y DLL) y los de datos suelen tener extensión NLS.

En la lista podemos ver más información como la versión, el fabricante, la descripción, etc. Como mencionamos anteriormente, tenemos que revisar estos campos para ver si encontramos algo sospechoso.

Los archivos con extensión NLS no suelen tener más información y parecen sospechosos. Estos archivos son utilizados por los procesos para los mapeos de códigos de lenguaje y suelen ser fiables. Si hacemos doble click izquierdo sobre alguno de los archivos de la lista del panel inferior, aparecerá una ventana nueva donde veremos dos pestañas. La primera pestaña es Image donde tenemos más información que tenemos que revisar, como la ruta del archivo (Path) más otras propiedades, y el botón "Verify" que sirve para comprobar la firma digital del archivo. La pestaña Strings nos muestra las cadenas de texto encontradas tanto en la memoria del proceso con en su archivo, y aquí podemos buscar textos sospechosos. Explicaremos después tanto "Verify" como "Strings".

Otra vista del panel inferior es la de los manipuladores o "Handles" de un proceso. Esta vista podemos activarla con CTRL+H o mediante el menú principal en View, Lower Pane View, Handles. De esta forma podremos ver todos los manipuladores abiertos del proceso. Los manipuladores son rutinas abiertas por el proceso y pueden ser de muchos tipos, por ejemplo: el tipo "File" indica un archivo abierto, el tipo "Key" indica el acceso a una clave de registro, etc. Conviene revisar todos los handles, aunque algunos son bastante difíciles de interpretar.

Dejando el panel inferior y volviendo al panel superior donde figuran los procesos, Process Explorer permite ver el contenido de un proceso, es decir, un montón de propiedades incluyendo a todos los procesos a los que ha llamado (threads o hilos) y que están activos en memoria. Para acceder a esta información realizaremos doble click con el botón izquierdo del ratón sobre el proceso en cuestión y tendremos algo parecido a la Figura 3. En nuestro ejemplo, analizaremos las entrañas del svchost.exe del grupo de servicios NETSVCS.

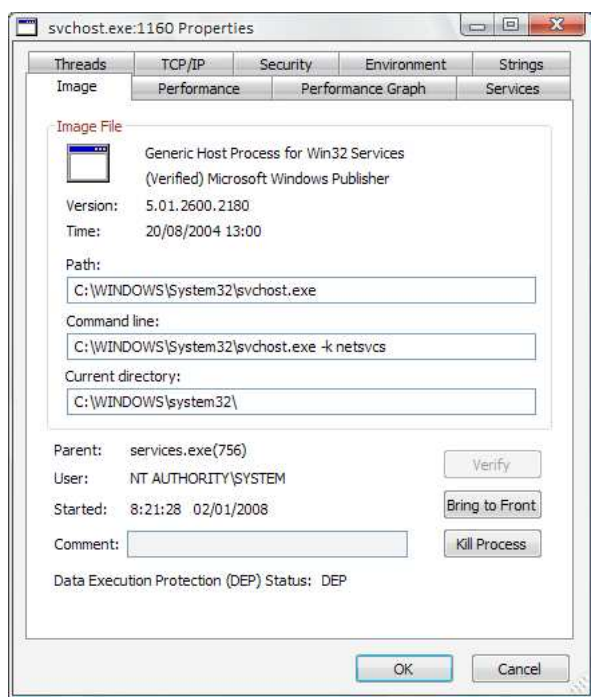


Figura 3. *Propiedades de svchost.exe*

Vemos en la pestaña Image el icono del archivo del proceso (es sospechoso que un proceso no tenga icono, pero en el caso de svchost.exe es normal), descripción del proceso, su ubicación, sus rutas para localizarlo, el proceso padre que lo creó (en este caso es services.exe), el usuario que inició el proceso, y la fecha/hora de creación del proceso. Como dato curioso, el Command Line de nuestro proceso es “svchost.exe -k netsvcs”. Netsvcs, como indicamos anteriormente, es un grupo de servicios iniciados por un svchost.exe. Netsvcs, en este caso es el parámetro junto con “-k” para iniciar este grupo de servicios.

En esta ventana tenemos un botón importante para comprobar la legitimidad de un proceso que es “Verify”. Pulsando este botón, Process Explorer se pondrá en contacto vía Internet con la Web del fabricante del proceso para comprobar si éste está firmado digitalmente (Verified). En este caso sí está firmado. Los procesos malware no están firmados, pero eso no quiere decir necesariamente que sea malware. En el caso de los procesos de Microsoft, la mayoría están firmados.

La pestaña Services muestra los servicios de nuestro svchost.exe y en ella encontraremos sus nombres, la ruta y una descripción de cada uno. Al ser servicios de sistema de Windows, serán sospechosos los que tengan algún campo en blanco o mal escrito. Desde aquí puede seleccionarse cualquier servicio, pausarlo y recuperarlo. Si paramos un servicio, desaparecerá de la lista y no podremos iniciarlo de nuevo desde esta ventana.

La pestaña Strings es muy útil porque muestra las cadenas de texto encontradas en la memoria y en el archivo del proceso. Es sospechoso encontrar alguna URL extraña en esta ventana, así como mensajes que denoten falta de fiabilidad. Esta ventana dispone de un botón de búsqueda “Find” por si queremos encontrar algún texto sospechoso, como direcciones de red (http, www, etc.).

Una de las pestañas más importantes es Threads (Figura 4). Un thread o hilo es una llamada a alguna función de otro proceso compartido que es invocado desde nuestro proceso. Podemos ver la lista de threads abiertos actualmente. Es aquí donde buena parte del malware se aloja.

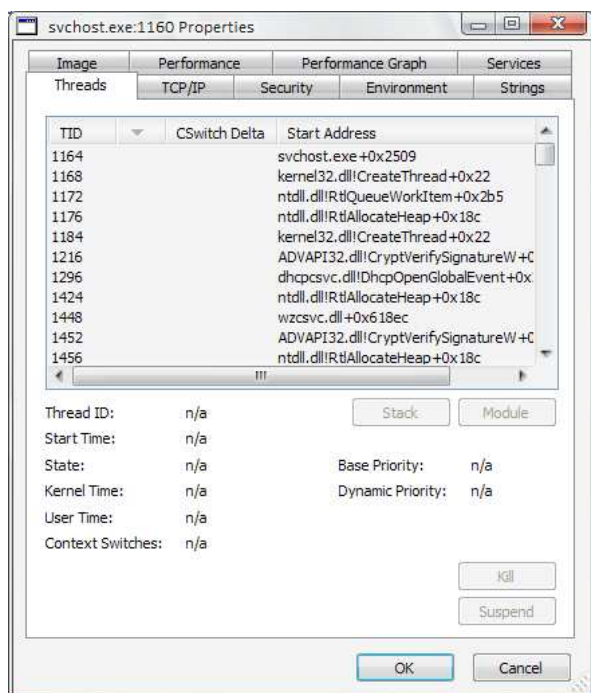


Figura 4. *Threads o hilos de svchost.exe*

Conviene revisar con atención la lista de hilos. Cada hilo tiene un número de identificación TID y el campo Start Address donde viene el nombre del hilo junto con su dirección base y alguna función. Seleccionando un hilo podemos ver la fecha y la hora de su creación, el estado, el tiempo que lleva activo a nivel de kernel y el tiempo activo a nivel de usuario entre otras informaciones.

En el momento de creación de un proceso, también se crean todos sus hilos, por tanto hay que poner especial atención a la fecha y la hora de creación de un hilo respecto a la fecha y la hora de creación del proceso padre (información dada en la pestaña Image del proceso), puesto que el malware puede aparecer “puntualmente” como un hilo creado con posterioridad a la creación del proceso padre. Aquí revisaremos los nombres de los hilos en busca de nombres extraños no documentados.

Si encontramos algún nombre de hilo sospechoso, podemos seleccionar dicho hilo y pulsar “Suspend” para congelarlo y analizarlo con detalle; entonces podemos ver más información de él como la ruta de donde proviene y poder identificarlo en el disco duro para eliminarlo después.

También podemos eliminar un hilo con “Kill” sin afectar al proceso padre, revisar las propiedades del archivo del hilo con “Module”, y con “Stack” veremos una lista de las llamadas a funciones de otros procesos realizadas por el hilo seleccionado. Así llegamos al último paso de la disección de un proceso y sus hilos.

La pestaña TCP/IP nos muestra las conexiones abiertas tanto TCP y UDP que el proceso tiene abiertas en ese momento. La Figura 5 muestra las conexiones abiertas de otro svchost.exe.

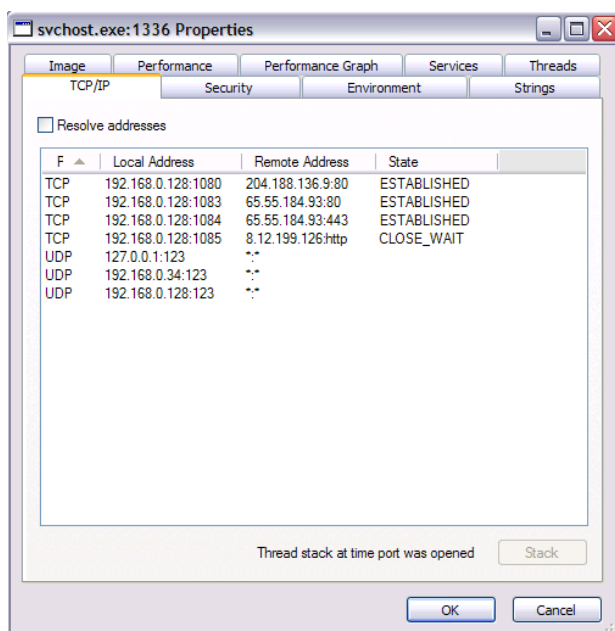


Figura 5. Conexiones abiertas por svchost.exe


Tenemos que identificar las direcciones y los puertos para comprobar que son de confianza. Además, una opción muy interesante cuando seleccionamos una conexión es “Stack”, donde aparecerá otra ventana con una lista de hilos que utilizaron la conexión seleccionada para enviar/recibir datos desde que ésta se inició.

Como podemos ver, el contenido de un proceso es complejo debido a la cantidad de información de todo tipo que contiene. Estratificar y entender toda esta información es la clave para confiar o no en un proceso.

Hay que tener en cuenta que en muchas ocasiones el malware no siempre está activo en memoria y que, dependiendo de factores de tiempo o de eventos del usuario, el malware puede activarse permaneciendo en memoria muy poco tiempo, el justo para realizar su cometido o parte de él. En estos casos, la persistencia en la observación es una tarea necesaria para pillar al malware “in fraganti”. Podemos forzar la actualización de los eventos con F5 o cambiar su frecuencia en el Menú View, Update Speed. Si estamos atentos, nos damos prisa y “pescamos al vuelo” algún proceso nuevo creado que sea sospechoso, como mencionamos anteriormente, podemos pausarlo y analizarlo tranquilamente.

Otras opciones útiles de Process Explorer

Hay que resaltar un par de opciones bastante interesantes a la hora de localizar procesos con Process Explorer.

Una de ellas es la localización del proceso de alguna ventana abierta en el escritorio de Windows. El procedimiento es: pinchamos primero con el botón izquierdo en el botón “diana”  y sin soltarlo lo arrastramos hasta la superficie de la ventana en cuestión. En ese momento Process Explorer seleccionará, en la lista de procesos, el proceso que inició dicha ventana. Esto es especialmente útil cuando aparece alguna ventana no solicitada y queremos averiguar cuál es el proceso responsable y comprobar de dónde proviene.

Otra opción muy interesante es “Find handle or DLL” ubicada en el menú principal “Find” o pinchando en el botón con los prismáticos. Como indica su nombre, busca DLL's o handles (manipuladores) dentro de todos los procesos corriendo en el sistema. Comentaremos esta función más adelante con un caso práctico.

Análisis longitudinal de procesos

Todas las indicaciones anteriores orientadas al Process Explorer son válidas para tareas puntuales que requieren un análisis de procesos a corto plazo, pero si dudamos de algún proceso sospechoso, es conveniente completar su auditoría observándolo durante más tiempo para ver si manifiesta algún comportamiento no deseado durante horas o incluso días.

Para ello, Windows XP tiene la herramienta “Registros de contador” ubicada en el Panel de control, Herramientas administrativas, Rendimiento, Registros y alertas de rendimiento. Procedamos con nuestro análisis pinchando con el botón derecho en Registros de contador y elegimos “Nueva configuración de registro”, indicamos un nombre, por ej. vamos a auditar el proceso de Internet Explorer; introducimos IE como nombre y aparecerá la ventana principal. En la ventana principal pinchamos en “Agregar contadores” y tendremos una ventana como la Figura 6.

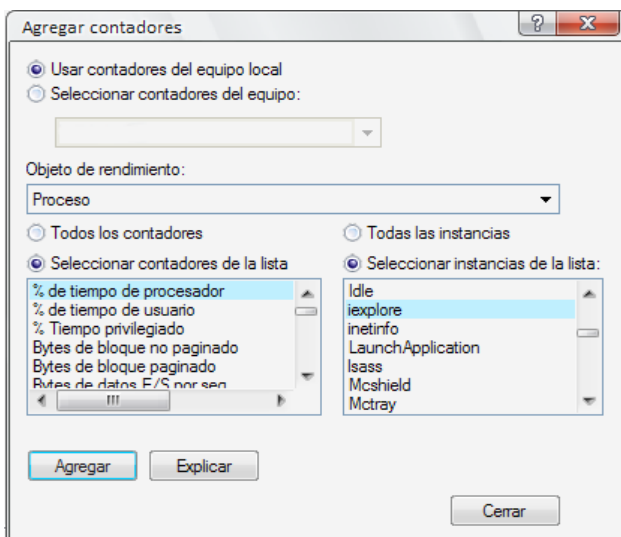


Figura 6. Agregando contadores para analizar

Podemos configurar las opciones como indica la Figura 6. Elegimos el Objeto de rendimiento llamado “Proceso” y así tendremos casi todos los procesos activos de la máquina en la opción/ventana “Seleccionar instancias de la lista:”, donde seleccionaremos “iexplore” que es el nombre del proceso de Internet Explorer.

En la ventana/opción “Seleccionar contadores de la lista” tenemos bastantes contadores. Agregaremos “% de tiempo de procesador”, “Bytes de datos E/S por segundo”, “Cantidad de manipuladores” (handles) y “Número de subprocesos (threads o hilos)”. Agregados todos los contadores indicados (cuidado porque en esta ventana no se ven los contadores agregados) pinchamos en “Cerrar” y nos quedamos en la ventana anterior donde, ahora sí, vemos los contadores agregados e indicaremos un intervalo de toma de muestra de 1 segundo. Después pinchamos en la pestaña “Archivos de registro” y nos aseguramos de que el tipo de archivo de registro es “Archivo binario”. En la pestaña “Programación” establecemos cuándo comenzará la toma de datos y durante cuánto tiempo. Cuanto más tiempo, más datos tendremos, así que un mínimo de una hora no está mal.

Iniciada la toma de datos toca esperar. Cuando termine o paremos el registro tendremos los resultados en el archivo “C:\PerfLogs\IE_000001.blg”. Este archivo no se puede ver directamente. Para ver el resultado gráfico tenemos que ir a “Registros de contador”, pinchar con el botón derecho en “IE” (en el panel derecho) y elegir “Guardar configuración como...”. De esta forma guardaremos un archivo .htm que podemos visualizar después como muestran las Figuras 7, 8 y 9.

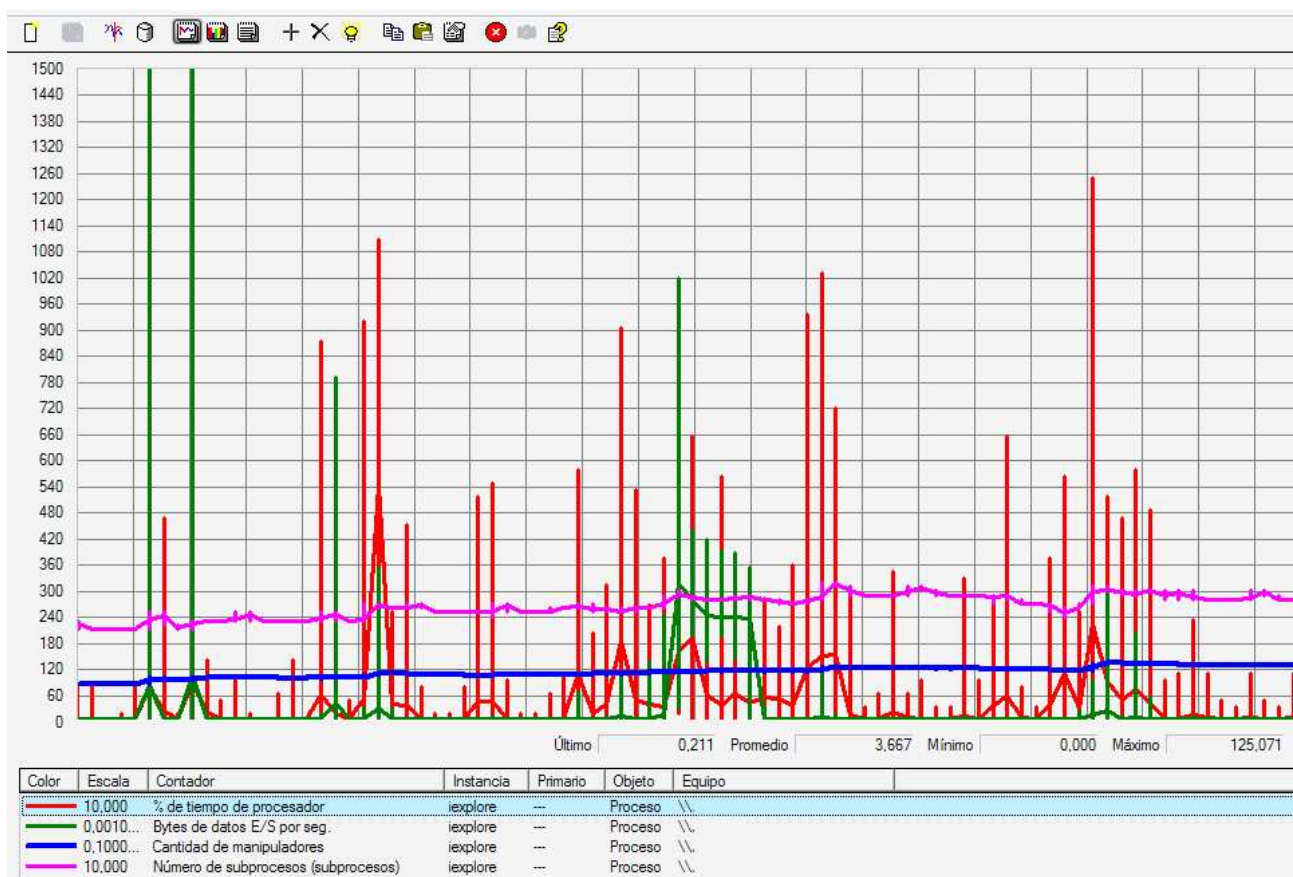


Figura 7. Comportamiento de *iexplore.exe* con usuario navegando

La muestra tomada en la Figura 7 es una sesión activa de Internet Explorer con un usuario navegando de forma normal en dos o tres sitios a intervalos durante una hora. A primera vista tanto la cantidad de manipuladores (azul) como el número de subprocesos (púrpura) no varían mucho, tienen un comportamiento normal y siguen una lenta tendencia a incrementarse lentamente (cosa lógica en Windows).

Los picos verdes (Bytes de datos E/S por segundo) son muy significativos, indican la cantidad de información enviando/recibiendo, y sería muy sospechoso tener estos picos en el caso de un Internet Explorer abierto y ocioso (Figura 8). Los picos rojos tienen un patrón más regular (% de tiempo de procesador) y suelen acompañar a los picos verdes (lógicamente), aunque también hay momentos donde sólo hay picos rojos que denotan procesamiento interno (actividades en segundo plano, cachés, etc.) de Internet Explorer.

La forma ideal para detectar actividad malware en un Internet Explorer es que no exista intervención del usuario en éste durante la captura de datos (Figura 8). En este caso, lo normal es que el movimiento de los contadores sea casi nulo, exceptuando algún % de tiempo de procesador mínimo y una ligera caída de manipuladores y subprocesos con una tendencia continua. No hay E/S ninguna. Cualquier actividad de E/S (verde) sería sospechosa.

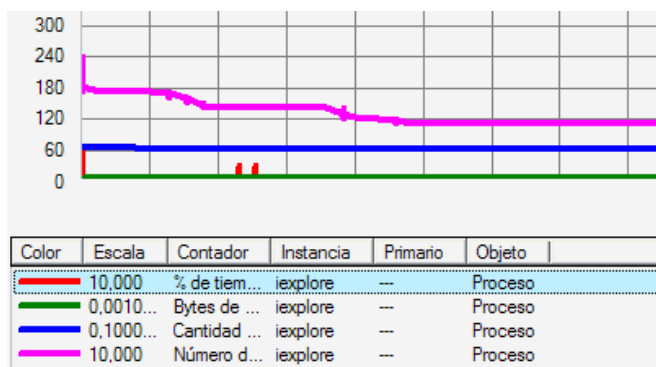


Figura 8. Comportamiento de iexplore.exe sin intervención del usuario

Para ilustrar gráficamente una aproximación de la actividad de red y de CPU de un malware que utiliza Internet Explorer (iexplore.exe), vamos a poner el ejemplo de una sesión de Windows Update que verifica las actualizaciones que tenemos ya instaladas en el sistema, envía esos datos a algún servidor de Windows Update y nos devuelve una lista de las actualizaciones que nos faltan por cargar (si las hay). En este ejemplo, la intervención del usuario está restringida sólo a acceder a la web de Windows Update y seguir las instrucciones. El resultado no dura más de dos minutos y lo podemos ver en la Figura 9.

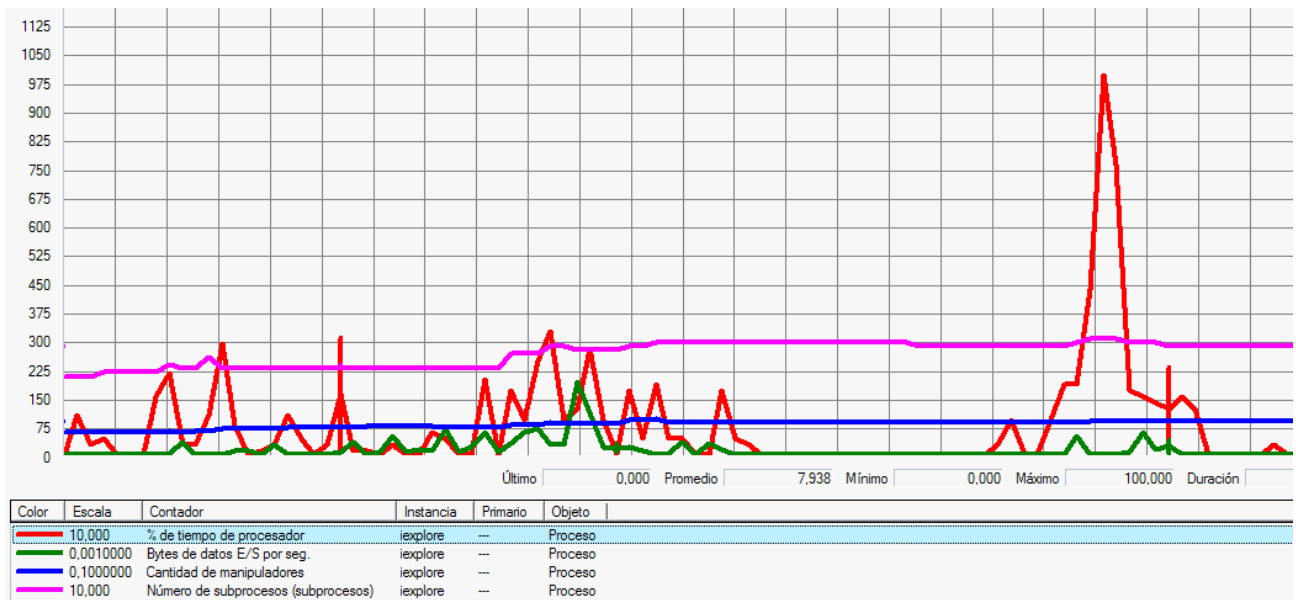


Figura 9. Comportamiento de iexplore.exe en una sesión de Windows Update

Como podemos ver en la Figura 9, la actividad de envío/recepción de datos es pequeña pero significativa, acompañada de % de tiempo de procesador igual de significativos. La cantidad de manipuladores o threads casi no varían durante toda la sesión y los subprocesos realizan una subida significativa junto con la subida de las tasas de envío/recepción de datos y % de tiempo de procesador. Es interesante ver que después de la actividad hay un llano y después un pequeña actividad de envío/recepción de datos y un gran pico de % de tiempo de procesador.

Este comportamiento puede interpretarse en tres partes: la primera es la recolección y envío de datos, la segunda es el llano que corresponde al procesamiento de los datos en el servidor remoto y después tendríamos la tercera parte que es la recepción de resultados. Encontrarse algo así en un Internet Explorer sin intervención del usuario es demasiado sospechoso, exceptuando el ejemplo expuesto de Windows Update.

Verificación de archivos no firmados digitalmente

Como vimos anteriormente, Process Explorer tiene la opción “Verify” que sirve para comprobar si el archivo de un proceso en ejecución está o no firmado digitalmente por su fabricante. La firma digital garantiza la integridad del archivo. “Verify” en sí es bastante fiable porque se pone en contacto vía Internet con el fabricante del proceso para realizar la comprobación de la firma.

Pero, ¿que ocurre con los archivos de procesos que no están firmados digitalmente y no están en ejecución actualmente?. Conviene comprobar la firma digital de archivos, ya sean de Microsoft o de otros fabricantes, aunque no estén activos debido a que podría existir malware que se inicie en algún momento, ya sea invocado en el inicio de alguna aplicación, algún evento del sistema, etc.

Para comprobar la firma digital de los archivos de sistema de Windows tenemos la aplicación de Microsoft “sigverif” que podemos abrir desde Inicio, Ejecutar. Iniciado “sigverif” pulsamos en “Iniciar” y realizará la comprobación. Después tendremos un listado con el nombre de los archivos no firmados, la ubicación, la fecha de modificación, el tipo de archivo y la versión. Todos estos archivos son sospechosos, y los que no tienen versión lo son más todavía. Con las opciones avanzadas de “sigverif” podemos comprobar la firma digital de cualquier archivo en cualquier ubicación (conveniente comprobar todos los archivos de C:\Windows\system32\drivers y C:\Windows\system32).

Si localizamos archivos no firmados sospechosos (sin versión, nombre extraño, etc.), podemos analizar su contenido con la utilidad gratuita “FileAlyzer” de Patrick M. Kolla (también creador de Spybot) y comprobar más firmas y las cadenas de texto. Si después de esto deducimos más sospechas, podemos mover los archivos sospechosos a otra carpeta (o renombrarlos) y reiniciar la máquina “a ver qué pasa”.

Para ilustrar la importancia de la verificación de la firma digital de archivos, vamos a emplear la utilidad “sigcheck” (análoga a sigverif pero en versión línea de comandos) de Sysinternals para analizar a C:\Windows\system32\ntoskrnl.exe en un caso real.

La Figura 10 muestra el resultado de la comprobación de la firma digital de ntoskrnl.exe. En este caso el archivo está firmado indicando el resto de propiedades como la versión, y además tenemos con el comando DIR la fecha, la hora de creación y el tamaño. Por otro lado, la Figura 11 muestra el resultado de la misma comprobación en el mismo ordenador en un momento anterior. En esta ocasión el archivo no está firmado y también difiere el tamaño (2.293.760 bytes) respecto al archivo firmado (2.138.624 bytes). El resto de propiedades son idénticas, como la versión (5.1.2600.3093), el nombre (ntoskrnl.exe), la ubicación (C:\Windows\system32) y la fecha/hora.

```
C:\WINDOWS\system32>sigcheck -i ntoskrnl.exe

Sigcheck v1.41
Copyright (C) 2004-2007 Mark Russinovich
Sysinternals - www.sysinternals.com

C:\WINDOWS\system32>dir ntoskrnl.exe:
Signers:
  Microsoft Windows Component Publisher
  Microsoft Windows Verification Intermediate PCA
  Microsoft Root Authority
Signing date: 17:22 28/02/2007
Publisher: Microsoft Corporation
Description: Sistema y núcleo de Windows NT
Product: Sistema operativo Microsoft« Windows«
Version: 5.1.2600.3093
File version: 5.1.2600.3093 (xpsp_sp2_gdr.070227-2254)

C:\WINDOWS\system32>dir ntoskrnl.exe
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 8C40-0DF4

Directorio de C:\WINDOWS\system32
28/02/2007 17:02 2.138.624 ntoskrnl.exe
1 archivos 2.138.624 bytes
```

Figura 10. *ntoskrnl.exe* firmado digitalmente

```
C:\WINDOWS\system32>sigcheck -i ntoskrnl.exe

Sigcheck v1.41
Copyright (C) 2004-2007 Mark Russinovich
Sysinternals - www.sysinternals.com

C:\WINDOWS\system32>dir NTOSKRNL.EXE:
Verified: Unsigned
File date: 17:02 28/02/2007
Publisher: Microsoft Corporation
Description: Sistema y núcleo de Windows NT
Product: Sistema operativo Microsoft« Windows«
Version: 5.1.2600.3093
File version: 5.1.2600.3093 (xpsp_sp2_gdr.070227-2254)

C:\WINDOWS\system32>dir ntoskrnl.exe
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 8C40-0DF4

Directorio de C:\WINDOWS\system32
28/02/2007 17:02 2.293.760 NTOSKRNL.EXE
1 archivos 2.293.760 bytes
```

Figura 11. *ntoskrnl.exe* no firmado digitalmente (muy sospechoso)

La modificación de este archivo o de cualquier otro archivo del sistema de Windows es un comportamiento típico de malware. Aunque Windows XP tiene un sistema de protección de archivos de sistema (WPF) de forma que, cuando un archivo importante del sistema cambia, automáticamente se reemplaza por una versión nueva que proviene de “C:\Windows\Driver Cache\i386”, el malware también puede modificar la versión cacheada. Existe malware que puede modificar archivos del sistema (como hemos visto) igual que lo puede hacer Windows Update, la instalación de parches de Windows y hotfixes, sólo necesita hacer los cambios adecuados y esperar un reinicio del sistema. Con esto, el malware integrado en un archivo de sistema y en su caché, asegura su persistencia en el ordenador evitando a WPF. Este tipo de malware también suele cambiar los archivos de los antivirus más utilizados y otras aplicaciones de seguridad para que no detecten nada.

En cualquier caso, es muy conveniente utilizar también la herramienta de Windows “SFC” para comprobar la firma digital de los archivos protegidos de sistema de Windows y reemplazar de forma automática las versiones de archivos encontrados no firmados por sus versiones firmadas desde el CD de Windows XP. El procedimiento adecuado sería utilizar primero “SFC /PURGECACHE” para eliminar la caché local de estos archivos y empezar de forma limpia, y después utilizaríamos “SFC /SCANONCE”, para comprobar las firmas digitales durante el próximo reinicio.

Inicialización automática de malware

Cuando el malware tiene la forma de un archivo ejecutable en algún lugar del disco duro, normalmente se inicia sin la intervención del usuario, invocado de forma automática, como un driver, servicio o como cualquiera de los innumerables procesos que Windows XP inicia automáticamente.

Existen varias formas de inicialización automática de procesos que conviene revisar. Las más accesibles son las tareas programadas y la carpeta de inicio de todos los usuarios que es “C:\Documents and Settings\All Users\Menú Inicio\Programas\Inicio”.

La parte más compleja respecto a la inicialización automática de procesos es el registro de Windows. La herramienta Autoruns de Sysinternals (Figura 12) sintetiza espectacularmente el registro para analizar las ubicaciones críticas donde se inician los procesos de forma automática.

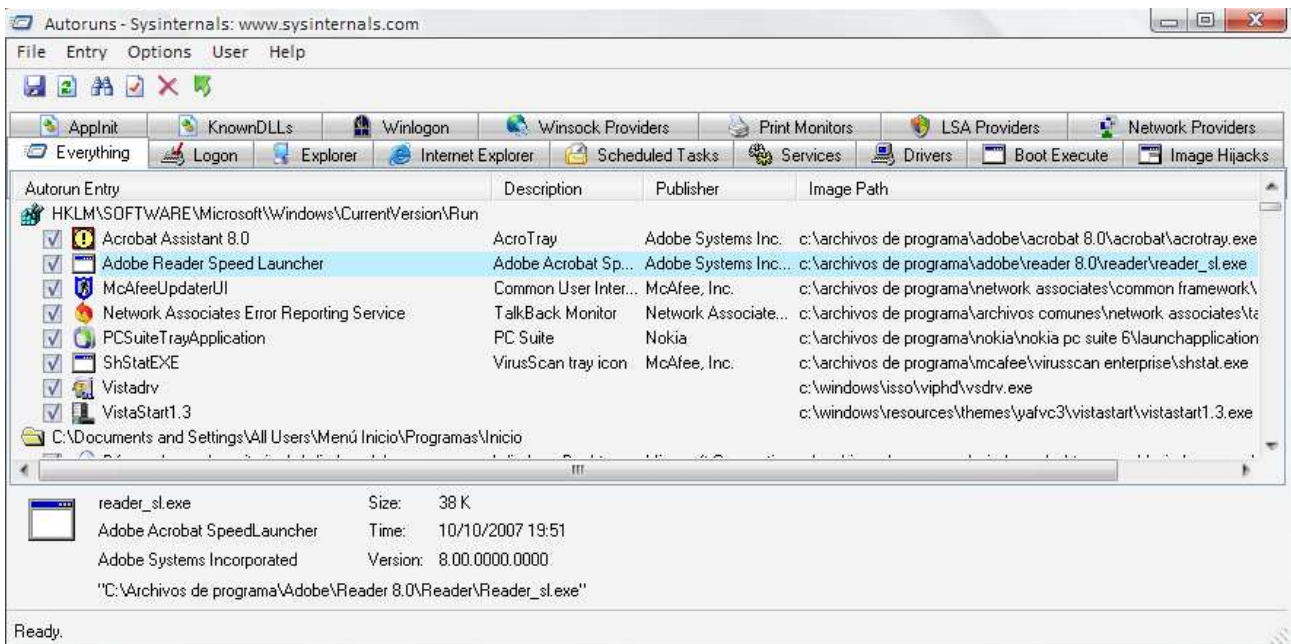


Figura 12. Sección “Everything” de Autoruns

Como vemos en la Figura 12, al iniciar Autoruns nos muestra en la pestaña “Everything” todas las entradas del registro que contiene procesos que se inician de forma automática. Las entradas típicas suelen ser:

- “HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run”
- “HKCU\Software\Microsoft\Windows\CurrentVersion\Run”
- “HKLM\System\CurrentControlSet\Services”, donde se alojan los servicios actuales en ejecución del sistema, así como los drivers.

La pestaña “Everything” suele tener una ingente cantidad de líneas. Para facilitar la tarea de análisis tenemos un par de opciones: ocultar las entradas de Microsoft marcando dicha opción en Options, Hide Microsoft Entries, y analizar pestaña por pestaña.

Como cada pestaña tiene un contexto distinto de inicio, vamos a explicarlas un poco:

- Logon: contiene los procesos que se inician justo después que el usuario introduce su nombre de usuario y contraseña para iniciar sesión.
- Explorer: componentes, manipuladores, extensiones, interfaces, protocolos y más elementos que carga Explorer al iniciar el escritorio de Windows.
- Internet Explorer: plugins, barras de herramientas, complementos, etc. Algunos se cargan al entrar en el escritorio (por su integración con Explorer) y otros se inician al abrir Internet Explorer.
- Scheduled Tasks: son las tareas programadas que hay en el sistema. Conviene revisar todas y en especial las que se inician cuando el usuario inicia sesión.
- Services: son los servicios de sistema de Windows y de otros fabricantes. Aquí están incluidos los servicios que pertenecen a los grupos de svchost. Los servicios de sistema se cargan durante el arranque de Windows antes de que el usuario inicie sesión.
- Drivers: los controladores que carga Windows ya sean de Microsoft o de otros fabricantes. También se cargan antes de que el usuario inicie sesión.
- Boot Execute: aquí se ejecuta autochk.exe que se encarga de realizar una comprobación de consistencia entre volúmenes de disco y letras de unidad. Evidentemente se carga antes de que el usuario inicie sesión. Es muy sospechoso que aquí haya algo más aparte del autochk.exe.
- Image Hijacks: esta entrada se utiliza con fines de depuración de archivos. Su peligro estriba en que puede suplantar procesos legítimos por malware. Si no tenemos conocimiento de esta opción de Windows a nivel de depuración y si hay alguna entrada con algún archivo ejecutable que no sea ntsd.exe, que es un depurador de Windows, puede ser muy sospechoso.
- AppInit: es una característica antigua de Windows NT para inicializar DLL's durante el arranque. En Windows XP es extraño que se utilice. Si aparece alguna entrada indicando algún archivo DLL puede ser sospechoso y conviene analizar dicha DLL.

- KnowDLLs: son varias DLL's de Windows XP que ofrecen API's a las aplicaciones de usuario. Suelen residir en C:\Windows\system32
- Winlogon: en el inicio de sesión del usuario de Windows, inicia *logonui.exe* que se encarga de recibir y procesar las credenciales de usuario. También se inician algunas DLL's que, si es necesario, realizan notificaciones de dicho proceso, así como para el inicio de sesión secundario. Aquí también se inician algunos archivos .scr (salvapantallas).
- Winsock Providers: carga los protocolos Winsock que ofrecen funciones TCP/IP para las aplicaciones de usuario (C:\Windows\system32\mswsock.dll). También carga C:\Windows\system32\Rsvpsp.dll que se encarga de controlar la calidad vía RPC del tráfico TCP y UDP.
- Print Monitors: aquí se inician varias DLL's de la cola de impresión.
- LSA Providers: carga las DLL's de servicios de seguridad como Kerberos, TLS/SSL y otros sistemas de autenticación y cifrado de Windows.
- Network Providers: son los proveedores de red que dan servicios como Terminal Server, el Cliente Web (necesario para navegar), la red de Microsoft Windows, etc.

En el apartado de **Eliminación del malware** hablaremos de cómo utilizar Autoruns para eliminar/desactivar entradas del registro sospechosas de una forma fiable.

Rootkits

Un tipo de tecnología de ocultación aplicada al malware es la denominada "Rootkit". Rootkit viene del mundo Unix (con finalidad en la administración de sistemas) y poco a poco ha ido incorporándose a otros sistemas operativos. Básicamente, un Rootkit es un proceso que permite ocultar procesos activos en memoria, archivos y en el caso de Windows hasta entradas en el registro.

Existen varios tipos de Rootkits, y podemos clasificarlos en:

- Rootkits de firmware: se inician mediante la memoria firmware de algún dispositivo electrónico conectado al bus de direcciones o de datos de la placa base (BIOS, CMOS, Chipsets, tarjetas PCI, firmware de discos duros, etc.). Como estos dispositivos se inician antes del arranque del sistema operativo, pueden tomar control sobre éste. Recientemente se han dado casos en discos duros, y por otro lado, John Heasman tiene un documento muy interesante sobre los Rootkits en dispositivos PCI (ver referencias al final del artículo).
- Rootkits de kernel: se inician en el mismo sistema operativo y toman el control sobre el kernel que puede, incluso, modificarlo en disco para permitir la persistencia en el sistema.
- Rootkits a nivel usuario: se inician en el mismo sistema operativo y toma el control sobre DLL's del sistema o sobre alguna aplicación de usuario.
- Rootkits de máquinas virtuales: toman control sobre máquinas virtuales y en consecuencia sobre los sistemas operativos de la máquina virtual.

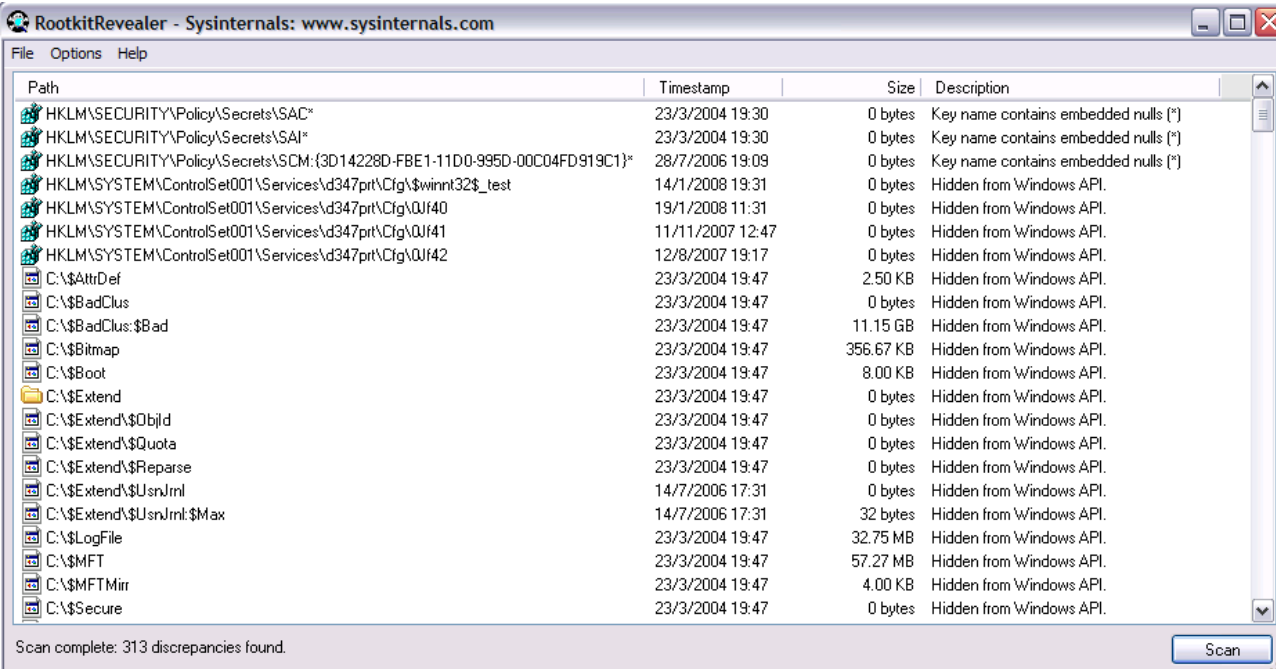
Los Rootkits de firmware y de kernel son muy difíciles de detectar una vez iniciado el sistema operativo afectado.

Los Rootkits a nivel de usuario y de máquinas virtuales son más sencillos de detectar porque suelen modificar algún proceso de alguna DLL u otro ejecutable de sistema a nivel de usuario directamente en memoria (hooks). Para permitir su persistencia pueden modificar también algún archivo DLL para obtener el mismo resultado (patch). Con estos procedimientos se integran en tales archivos creando llamadas de sistema ilícitas que permiten su ocultamiento.

La detección de Rootkits se basa en la comprobación de las firmas digitales de las DLL's, de archivos de sistema, controladores, y cualquier otro archivo en disco que se cargue o pueda ser cargado. La tecnología de detección avanza de igual manera que el mismo Rootkit.

Las herramientas que existen actualmente para detectar Rootkits son numerosas y de distintas tecnologías. Las más conocidas para Windows son Rootkit Revealer de Sysinternals y F-Secure Blacklight.

Utilizando Rootkit Revealer podemos ver las entradas de registro y archivos del sistema de archivos de todas las unidades ocultos a Windows (Figura 13).



The screenshot shows the RootkitRevealer application window. The title bar reads "RootkitRevealer - Sysinternals: www.sysinternals.com". The menu bar includes "File", "Options", and "Help". The main area displays a table with the following columns: Path, Timestamp, Size, and Description. The table lists various registry keys and files, many of which are marked as "Hidden from Windows API". At the bottom of the window, it states "Scan complete: 313 discrepancies found." and has a "Scan" button.

Path	Timestamp	Size	Description
HKLM\SECURITY\Policy\Secrets\SAC*	23/3/2004 19:30	0 bytes	Key name contains embedded nulls (*)
HKLM\SECURITY\Policy\Secrets\SAI*	23/3/2004 19:30	0 bytes	Key name contains embedded nulls (*)
HKLM\SECURITY\Policy\Secrets\SCM:{3D14228D-FBE1-11D0-995D-00C04FD919C1}*	28/7/2006 19:09	0 bytes	Key name contains embedded nulls (*)
HKLM\SYSTEM\ControlSet001\Services\d347prt\Cfg\\$\winn32\$_test	14/1/2008 19:31	0 bytes	Hidden from Windows API.
HKLM\SYSTEM\ControlSet001\Services\d347prt\Cfg\0Jf40	19/1/2008 11:31	0 bytes	Hidden from Windows API.
HKLM\SYSTEM\ControlSet001\Services\d347prt\Cfg\0Jf41	11/11/2007 12:47	0 bytes	Hidden from Windows API.
HKLM\SYSTEM\ControlSet001\Services\d347prt\Cfg\0Jf42	12/8/2007 19:17	0 bytes	Hidden from Windows API.
C:\\$AttrDef	23/3/2004 19:47	2.50 KB	Hidden from Windows API.
C:\\$BadClus	23/3/2004 19:47	0 bytes	Hidden from Windows API.
C:\\$BadClus:\$Bad	23/3/2004 19:47	11.15 GB	Hidden from Windows API.
C:\\$Bitmap	23/3/2004 19:47	356.67 KB	Hidden from Windows API.
C:\\$Boot	23/3/2004 19:47	8.00 KB	Hidden from Windows API.
C:\\$Extend	23/3/2004 19:47	0 bytes	Hidden from Windows API.
C:\\$Extend\\$\\$bld	23/3/2004 19:47	0 bytes	Hidden from Windows API.
C:\\$Extend\\$\\$quota	23/3/2004 19:47	0 bytes	Hidden from Windows API.
C:\\$Extend\\$\\$repairse	23/3/2004 19:47	0 bytes	Hidden from Windows API.
C:\\$Extend\\$\\$snjml	14/7/2006 17:31	0 bytes	Hidden from Windows API.
C:\\$Extend\\$\\$snjml:\$Max	14/7/2006 17:31	32 bytes	Hidden from Windows API.
C:\\$LogFile	23/3/2004 19:47	32.75 MB	Hidden from Windows API.
C:\\$MFT	23/3/2004 19:47	57.27 MB	Hidden from Windows API.
C:\\$MFTMirr	23/3/2004 19:47	4.00 KB	Hidden from Windows API.
C:\\$Secure	23/3/2004 19:47	0 bytes	Hidden from Windows API.

Figura 13. Resultados de RootkitRevealer (entradas de registro de Windows y archivos ocultos)

Windows XP, de por sí oculta entradas en el registro como las primeras que vemos en la figura 13 (SAC* y SAI*). Estas entradas tienen fines de seguridad. Más sospechoso es encontrar entradas como las que vemos a continuación que hacen referencia a caracteres poco habituales como "0Jf40" dentro de la inicialización del servicio "d347prt", pero parece que según el foro de Sysinternals (http://forum.sysinternals.com/forum_posts.asp?TID=5795) pertenecen al servicio de DeamonTools. Respecto a los siguientes resultados que hacen referencia a archivos ocultos en C:, son los metadatos NTFS que Windows utiliza en su sistema de archivos. Es posible encontrar en los resultados entradas en el registro y archivos de algún antivirus u otro software de seguridad (incluido también el mismo RootkitRevealer).

Si encontramos alguna entrada sospechosa, tomamos nota y buscaremos más información en Internet.

Otra forma más contundente de comprobar la existencia de Rootkits consiste en iniciar la máquina sospechosa con otro sistema operativo (desde otro disco duro, Live CD's, pendrives, o cualquier otro método de arranque). En este caso, los archivos del disco duro sospechoso quedan al descubierto y pueden cotejarse con los archivos del mismo disco duro cuando inicia en condiciones normales.

La tecnología Rootkit y la tecnología anti-Rootkit (que trata de desvelar a los Rootkits) es prácticamente la misma, y según Mark Russinovich es un juego donde ninguno gana.

La documentación que existe acerca de los Rootkits es muy amplia y un buen punto de partida es www.rootkit.com.

Para ver en acción a un Rootkit, veremos el Rootkit FU (creado por fuzen_op) y disponible en: https://www.rootkit.com/vault/fuzen_op/FU_Rootkit.zip. Antes de descargarlo conviene desactivar el antivirus del sistema.

El contenido de FU_Rootkit.zip lo podemos descomprimir en C: y tendremos una carpeta llamada FU_Rootkit, y dentro de ella, en la carpeta EXE tenemos al archivo ejecutable del Rootkit en cuestión (fu.exe).

Para ocultar, por ejemplo al proceso activo de Microsoft Paint (mspaint.exe), abrimos Process Explorer y nos fijamos en su PID. En este caso es 516 (Figura 14 apartado 1).

Después abrimos una consola shell (cmd.exe) y nos dirigimos a la carpeta donde se encuentra fu.exe y tecleamos "fu -ph 516". El efecto de este comando lo tenemos en la Figura 14 apartado 2 donde fu.exe aparece y desaparece (color rojo). Por último, en la Figura 14 apartado 3 ya no aparecen ni fu.exe ni mspaint.exe, aunque los dos están en funcionamiento.

1		
ctfmon.exe	260	
soffice.exe	3620	
soffice.bin	3628	
mspaint.exe	516	
Photoshp.exe	1412	
ieexplore.exe	3292	
cmd.exe	2540	
procexp.exe	556	1

2		
ctfmon.exe	260	
soffice.exe	3620	
soffice.bin	3628	
mspaint.exe	516	
Photoshp.exe	1412	
ieexplore.exe	3292	
cmd.exe	2540	
fu.exe	3240	1
procexp.exe	556	4

3		
ctfmon.exe	260	
soffice.exe	3620	
soffice.bin	3628	
Photoshp.exe	1412	
ieexplore.exe	3292	
cmd.exe	2540	
procexp.exe	556	1

Figura 14. *Fu.exe* ocultando a *mspaint.exe*

La versión de Fu que hemos utilizado en este ejemplo es la versión actual disponible (enero 2008) y está muy mejorada. Funciona perfectamente en un Windows XP Pro SP2 con todas las actualizaciones de Windows Update cargadas a la fecha.

Eliminación de malware

El proceso de eliminación de malware suele requerir varios pasos y supone realizar bastantes comprobaciones. Actualmente los antivirus y otras herramientas anti-malware como Spybot facilitan la eliminación de archivos y entradas creadas por el malware en el registro de Windows. Debido a que la mayoría de estos programas también son atacados por el malware, los resultados pueden diferir.

Lo primero que tenemos que saber es el alcance del malware sospechoso localizado en base a los procedimientos de los apartados anteriores. Con esto sabemos lo que podemos eliminar: procesos, threads (hilos), manipuladores (handles), archivos y entradas de registro sospechosos.

La eliminación de malware activo en memoria puede ser:

- De procesos: seleccionar el proceso con el botón derecho y pinchar en “Kill Process”. Si queremos matar también a todos los procesos hijos elegiremos “Kill Process Tree”. Si eliminamos algún proceso crítico del sistema como Winlogon o LSASS, Windows apagará el sistema. Pueden existir procesos que no podamos eliminar dando un error de acceso denegado.

Puede que al eliminar un proceso, vuelva a iniciarse de nuevo. En estos casos tendríamos que comprobar con la herramienta “Procmon” de Sysinternals cuál es el proceso que inicia al proceso eliminado. Dependiendo del proceso padre, podremos eliminarlo también, o no.

Respecto a los servicios de Windows, podemos detener y/o deshabilitar los que resulten sospechosos con el administrador de servicios “services.msc”. Como muestra la Figura 15, al realizar doble click con el botón izquierdo en un servicio, aparece entre otras informaciones la elección del tipo de inicio. Si vemos algún servicio sospechoso podemos detenerlo y deshabilitarlo desde aquí para asegurarnos de que no vuelva a iniciarse.



Figura 15. Tipo de inicio de un servicio

Como vimos en el apartado de **Inicialización automática del malware**, los servicios se inician con la clave de registro “HKLM\System\CurrentControlSet\Services” que hace referencia a “CurrentControlSet”. Este apartado y sus hermanos “ControlSet001” y “ControlSet002” (Figura 16) se utilizan en la forma que “CurrentControlSet” es la configuración de servicios que está actualmente en ejecución en el sistema, mientras que ControlSet001 y ControlSet002 se utilizan como respaldo (uno será la última configuración buena conocida y el otro la configuración por defecto). Es conveniente revisar todos los servicios de todos los ControlSet.

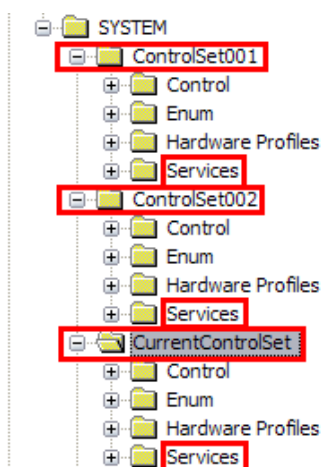


Figura 16. Registro de Windows: ControlSets con los “Services” correspondientes

- De Threads: la eliminación de un thread dentro de un proceso se realiza abriendo el proceso y en la pestaña “Threads” seleccionamos al thread en cuestión y pinchamos en el botón “Kill”.
- De manipuladores: se realiza seleccionando dicho proceso y localizando en el panel inferior de Process Explorer (en modo handles) el manipulador en cuestión. Después seleccionamos el manipulador con el botón derecho y pinchamos “Close Handle” para eliminarlo.

La eliminación de archivos de procesos malware del disco duro

Se puede realizar directamente con el explorador de Windows, seleccionando el archivo y pulsando en “Supr” o “Mayus”+”Supr” si queremos que se elimine directamente sin pasar por la papelera de reciclaje.

Como indiqué anteriormente, una alternativa a la eliminación del archivo es renombrarlo. De esta forma y, después de un reinicio el proceso del archivo no se activará en memoria y podemos analizarlo.

Existen varios casos donde la eliminación del archivo en disco no es posible. Pueden ser:

- No tenemos privilegios suficientes: necesitamos iniciar sesión en Windows como administrador local para eliminar el archivo.
- El archivo tiene el atributo de sólo lectura activado: con el comando “attrib -r nombre_archivo” le quitamos el atributo y podremos eliminarlo.
- El archivo está abierto: con Process Explorer pulsamos CTRL+F o en el menú principal vamos a “Find”, “Find handle or DLL”. En el campo de búsqueda introducimos el nombre de archivo que no podemos eliminar y nos devolverá el proceso que abrió el archivo, su PID, su tipo (que será “handle”) y la ubicación del “handle”. En este punto, si hacemos doble click en el resultado mencionado, automáticamente nos dejará en el handle en cuestión en el panel inferior. Entonces pincharemos con el botón derecho en el handle y podemos cerrarlo con “Close Handle”. De esta forma el archivo quedará liberado y podremos eliminarlo. Es una forma muy sutil de matar a una función (handle) de un proceso sin matar al proceso en sí.
- En cualquier otro caso: si el archivo se resiste a la eliminación podemos utilizar la utilidad “movefile” incluida en el paquete “PendMoves and MoveFile” de Sysinternals. Movefile marcará el archivo que le indiquemos y lo eliminará durante el próximo reinicio del sistema.

En la eliminación de entradas malware del registro de Windows

Podemos utilizar “regedit” de Windows (donde tendremos disponible todo el registro de Windows) o Autoruns para las entradas que inician automáticamente procesos.

Con Autoruns, podemos pinchar con el botón derecho en la entrada que consideremos oportuna y elegir “Delete”. Otra opción interesante es desactivar la entrada desmarcando la casilla que tiene justo a su izquierda. De esta forma, dicha clave se ignora y al no eliminarla, podemos tenerla en cuenta para seguir investigando.

Cuando eliminamos o desactivamos alguna clave del registro con Autoruns, podemos pulsar F5 para actualizar la vista y comprobar si vuelve a aparecer y/o activarse. Si ocurre esto último quiere decir que existe un proceso activo que comprueba la eliminación y/o desactivación de la clave y la vuelve regenerar y/o activar. En estos casos tendremos que rastrear este tipo de eventos, con la utilidad "Procmon" (indicada anteriormente) para averiguar cual es el proceso responsable de la reactivación, localizarlo, eliminarlo y volver a desactivar o eliminar la entrada de registro sospechosa con éxito.

Es posible que alguna entrada del registro no se pueda eliminar debido a que dicha entrada contenga algún carácter NULL. En estos casos podemos utilizar la herramienta "RegDelNull" de Sysinternals para sustituir el carácter NULL por un asterisco y así poder eliminar la entrada en cuestión.

Eliminando Rootkits

Debido a la naturaleza del Rootkit, su eliminación del disco duro normalmente es muy complicada y en la mayoría de los casos es necesaria una reinstalación de Windows XP previo formateo.

Una alternativa antes del formateo, si hemos detectado donde está alojado el Rootkit y otros archivos ocultos por él, es utilizar algún sistema operativo que pueda arrancar de forma externa y que permita escribir en el sistema de archivos NTFS (el CD de Windows XP en modo consola, otro disco duro con otro Windows XP, ERDCommander de Sysinternals, etc.). De esta manera, el Rootkit no se inicia y quedará a la vista igual que los archivos que hubiera podido ocultar. Una vez localizados, procedemos a eliminarlos.

En el caso de Rootkits a nivel de firmware, la eliminación puede llegar a ser más complicada todavía y, lo más fácil sería prescindir del componente electrónico afectado.

Eliminar, reiniciar, comprobar...

Una vez eliminado todo lo considerado malware, necesitamos reiniciar la máquina y volver a realizar las comprobaciones oportunas para asegurarnos de que el malware está eliminado completamente.

Conclusión

La continua evolución del malware hace que la carrera malware-antimalware no termine nunca. Las herramientas disponibles que eliminan malware automáticamente (antivirus, antisпам, etc.) deben actualizarse de forma constante con los nuevos cambios detectados en el malware y aún así, nada se puede hacer con el nuevo malware que no ha sido localizado y etiquetado (0-day).

En estas condiciones que no van a cambiar, la debida utilización de las herramientas que nos permitan profundizar todo lo que podamos en un sistema operativo son imprescindibles junto con la experiencia y la intuición.

Un sistema operativo no deja de ser una compleja estructura donde hay sitios olvidados o poco accesibles. El debido entendimiento de lo que ocurre en la integridad de un sistema operativo es crucial para localizar tanto elementos legítimos como ilegítimos.

En el fondo siempre estará la máquina y siempre tendremos otras máquinas con las que compararlas y sacar conclusiones más exactas.

Y por último y no menos importante: la copia de seguridad evita disgustos, y un buen formateo evita perder el tiempo.

Referencias

- Lista de componentes de Windows:
http://en.wikipedia.org/wiki/List_of_Microsoft_Windows_components#Configuration_and_maintenance
- Proceso de inicio de Windows NT: http://en.wikipedia.org/wiki/Windows_NT_Startup_Process
- Protección de archivos de Windows (WFP): <http://www.fermu.com/content/view/382/2/lang,es/>
- Análisis de archivos (FileAlyzer): <http://www.safer-networking.org/es/filealyzer/index.html>
- Foro de Sysinternals: <http://forum.sysinternals.com/>
- Utilidades de Sysinternals (Process Explorer, Autoruns, Sigcheck, RegDelNull, PendMoves, ProcMon, RootkitRevealer): <http://technet.microsoft.com/en-us/sysinternals/default.aspx>
- Webcast de Mark Russinovich acerca de la limpieza avanzada de malware:
<http://www.microsoft.com/emea/itsshowtime/sessionh.aspx?videoid=359>
- Castlecops: <http://www.castlecops.com/>
- Procesos especiales de Windows XP: <http://tapatuculo.blogspot.com/2007/02/process-explorer-procesos-especiales-de.html>
- Microsoft Windows Internals, Fourth Edition: Microsoft Windows Server 2003, Windows XP, and Windows 2000 by Mark E. Russinovich, David A. Solomon. ISBN 0735619174
- Rootkits: www.rootkit.com
- PCI Rootkits:
http://www.nextgenss.com/research/papers/Implementing_And_Detecting_A_PCI_Rootkit.pdf
- Video de la prueba de concepto del Rootkit FU: <http://www.echogod.com/video/fu.swf>

- Dedicado a los que empezaron jugando y todavía no se han aburrido -