UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DISEÑO Y DESARROLLO DE UN ANTIVIRUS

TESIS PROFESIONAL

Para Obtener El Título De:

INGENIERO EN COMPUTACIÓN

Presenta:

Aldo Jiménez Arteaga

Directora de Tesis:

M. en C. María Jaquelina López Barrientos

México D.F., 2007.

ÍNDICE

INTRODUCCIÓN

Uno de los cambios más sorprendentes del mundo es la evolución de los sistemas de información; modernos sistemas permiten que el flujo de conocimientos sea independiente del lugar físico en que nos encontremos.

El escenario electrónico actual es unir redes internas a la Internet, la que crece a razón de más de un 10% mensual; al unir una red a la Internet se tiene acceso a otras redes. De este universo de computadoras interconectadas, no es difícil pensar que haya personas con perversas intenciones para utilizar la información circulante en las redes.

Diariamente se reciben reportes de los ataques a redes informáticas, los cuales se han vuelto cada vez más siniestros: los archivos son alterados subrepticiamente, las computadoras no operan correctamente, se ha copiado información confidencial sin autorización, se ha reemplazado el software para agregar “puertas traseras” de entrada, y miles de contraseñas han sido robadas. Los administradores de sistemas deben gastar horas y a veces días enteros volviendo a cargar o reconfigurando sistemas comprometidos, con el objeto de recuperar la confianza en la integridad del sistema.

En muchas ocasiones, estas acciones no son llevadas a cabo por los intrusos directamente, sino por programas creados por ellos mismos; estos programas son conocidos como **virus informáticos**. Son llamados virus porque presentan características similares a los virus biológicos, las cuales son:

- ✓ Son agentes parásitos intracelulares; es decir, necesitan una célula viva para replicarse.
- ✓ Su principal función es replicarse.
- ✓ Puede propagarse de una célula a otra o de un huésped a otro.

Los virus informáticos son uno de los principales riesgos de seguridad para los sistemas de información, ya sea que estemos hablando de un usuario hogareño que utiliza su máquina para trabajar y conectarse a Internet o de una empresa con un sistema informático complejo.

Un virus se valdrá de diversas técnicas para lograr su cometido:

- ✓ *Ocultamiento*. Esconde los posibles signos de infección del sistema.
- ✓ *Polimorfismo*. Es una técnica en la que el virus es capaz de cifrarse cada vez que infecta un archivo, ocultando de esta forma cualquier posible indicio que pueda facilitar su búsqueda; el virus debe poseer una rutina de descifrado.
- ✓ *Añadidura o empalme.* El código del virus se agrega al final del archivo ejecutable, modificando las estructuras de la cabecera, de manera que el control del programa pase primero al virus cuando se quiera ejecutar el archivo.
- ✓ *Inserción*. Los virus que utilizan el método de inserción buscan alojarse en zonas de código no utilizadas o en segmentos de datos dentro de los archivos que contagian.
- ✓ *Reorientación*. Este método es una variante del anterior. Bajo este esquema se introducen centrales víricas (los códigos principales del virus) en zonas del disco duro marcadas como defectuosas o en archivos ocultos del sistema. Estos códigos virales implantan pequeños trozos de código en los archivos ejecutables que infectan, que luego actúan como puentes a las centrales víricas.
- ✓ *Sustitución*. El método de sustitución, usado por los Caballos de Troya, es quizás el método más primitivo. Consiste en sustituir el código completo del archivo original por el código del virus; al ejecutar el programa infectado el único que actúa es el virus.

Así entonces, encontraremos virus muy simples que sólo se dedican a jugar bromas y algunos otros mucho más complejos que intentan dañar la información.

Un virus es un programa diseñado para dañar sistemas informáticos, alterando su forma de trabajar o dañando información; además, puede replicarse a sí mismo y propagarse a otras computadoras, sin el conocimiento o permiso del afectado. Los virus son diseñados para realizar una acción concreta en los sistemas informáticos; se programa en lenguaje ensamblador y por lo tanto, requiere algunos conocimientos del funcionamiento interno de la computadora.

Un virus tiene tres características primarias:

- ✓ *Es dañino*. Un virus informático siempre causa daños en el sistema que infecta.
- ✓ *Se replica a sí mismo*. La característica más importante de este tipo de programas es la de crear copias de sí mismo, cosa que ningún otro programa convencional hace.
- ✓ *Es subrepticio*. Esto significa que utilizará varias técnicas para evitar que el usuario se de cuenta de su presencia.

La verdadera peligrosidad de un virus no está dada por su arsenal de instrucciones maliciosas, sino por lo crítico del sistema que está infectando; es decir, mientras más importante sea el sistema infectado, más dañina se volverá la invasión del virus. Tomemos como ejemplo un virus conejo cuya principal función, como si de un conejo se tratase, es la de reproducirse infinitamente, copiándose a sí mismo hasta que ocupa toda la memoria libre o el disco del ordenador, dejándolo bloqueado. Si éste infectara una computadora hogareña, el equipo puede permanecer inactivo durante varios días; si el usuario tiene conocimientos sobre infecciones puede reiniciar la máquina infectada con un disquete de arranque limpio y con un antivirus se elimina el virus. Si afectara a un servidor de una empresa, posiblemente el sistema dejaría de funcionar por algún tiempo, significando una pérdida de horas máquina y de dinero. Pero si este virus infectara una máquina que gobierna una grúa robótica o algún dispositivo quirúrgico, posiblemente se perderían vidas humanas.

Los virus informáticos no pueden causar un daño directo sobre el hardware; en su defecto, un virus puede ejecutar operaciones que reduzcan la vida útil de los dispositivos. Esta compleja acción es posible, aunque muy poco probable y por lo general los virus prefieren atacar la parte lógica.

Existen varias clasificaciones de los virus. Éstas atienden a características muy diversas y específicas:

- ✓ *Objetivo de infección*. Se conocen tres divisiones de virus: aquéllos que infectan archivos, los que infectan los sectores de arranque y los que infectan a ambos.

- ✓ *Objetivo de modificación*. Las categorías de esta clasificación son: virus de archivo (modifican archivos) y virus de sistema operativo (infectan archivos que gobiernan el ordenador).
- ✓ *Comportamiento*. En función de su comportamiento, todos los virus anteriores pueden a su vez clasificarse en otros subgrupos (silenciosos, de macro, polimórficos, etc.); existen más clasificaciones según su comportamiento.

El daño que un virus puede causar a un sistema informático es muy variado; puede ser la modificación de programas para que se produzca un funcionamiento incorrecto, modificación de datos, eliminación de programas y/o datos, saturación del medio de almacenamiento, saturación de información procesada, robo de información confidencial, etc. Además se tienen los daños subsecuentes: pérdida de tiempo, capital, empleo y en un caso extremo de vidas humanas. Todo esto depende de las intenciones del programador del virus y de su "creatividad".

La forma de detectar un virus es muy variada; algunos síntomas de la infección de un virus son: los programas comienzan a ocupar más espacio de lo habitual, aparecen o desaparecen archivos, cambia el tamaño de un programa o un objeto, aparecen mensajes u objetos extraños en la pantalla, el disco trabaja más de lo necesario, la cantidad de espacio libre del disco disminuye sin ningún tipo de explicación, se modifican sin razón aparente el nombre de los archivos, no se puede acceder al disco duro, y muchos otros muy variados.

Actualmente existen aplicaciones llamadas **antivirus** que combaten a los virus. El software antivirus es un programa más de computadora, y como tal, debe ser adecuado para el sistema: debe estar correctamente configurado según la arquitectura del hardware que se tenga; si se trabaja en un lugar que posee conexión a redes es necesario que el programa antivirus tenga la capacidad de detectar virus de redes. Los antivirus reducen sensiblemente los riesgos de infección, pero cabe reconocer que no serán completamente eficaces y su utilización debería estar acompañada con otras formas de prevención como: abstenerse de abrir o ejecutar archivos o correo electrónico de procedencia desconocida, instalación de un *firewall* o el respaldo de la información crítica.

Los antivirus son programas creados para prevenir o evitar la activación de los virus, así como su propagación y contagio; cuentan con rutinas de detección, eliminación y reconstrucción de los archivos y las áreas infectadas del sistema. Un antivirus tiene tres principales funciones y componentes:

- ✓ *Protector*: Se activa y permanece en la memoria; actúa como filtro de los archivos que son movidos para ser leídos o copiados. Esta función se realiza en tiempo real.
- ✓ *Detector*: Examina todos los archivos existentes en el disco duro; tiene instrucciones de control y reconocimiento de códigos que detectan e identifican a los virus.
- ✓ *Eliminador*: Procede a eliminar y desactivar los virus detectados; repara los archivos y áreas afectados.

La función básica de detección e identificación se realiza mediante dos técnicas básicas: el rastreo de cadenas, y la búsqueda heurística; la protección del sistema se realiza mediante el monitoreo de actividad maliciosa.

Es muy difícil prever la propagación de los virus y qué máquina intentarán infectar; de ahí la importancia de saber cómo funcionan típicamente y tomar en cuenta los métodos de protección adecuados para evitarlos.

A medida que la tecnología evoluciona, van apareciendo nuevos estándares y acuerdos entre compañías que pretenden compatibilizar los distintos productos en el mercado. Con el tiempo, esto permitirá que un virus pueda contaminar cualquier sistema en cualquier plataforma, incrementando aún más los potenciales focos de infección.

Así, el presente trabajo de tesis se desarrolla persiguiendo diversos objetivos. En primera instancia, se sentará una base teórica para un futuro estudio sobre virus informáticos, además de analizar lo que es este tipo de programas, cómo trabajan y qué efectos producen; el siguiente objetivo es estudiar algunos métodos de defensa y construir un antivirus que permita poner en práctica los conocimientos adquiridos en este campo; finalmente, se pretende que este documento

sirva de base a otros estudiosos de la seguridad informática para que se mantengan a la vanguardia en esta área.

De manera que en el capítulo I se presenta un panorama del campo referente a la seguridad de la información (definiciones, contexto, amenazas, etc.). Esta base permite avanzar hacia el conocimiento particular de los virus, en el capítulo II; para después, en el capítulo III, estudiar su relación con los sistemas operativos más populares actualmente. El trabajo no estaría completo si no se dedicara espacio para hablar sobre las herramientas y los métodos que existen para defender a la información de los códigos víricos; este conocimiento está condensado en el capítulo IV. Finalmente, en los capítulos V y VI se presenta la construcción (el diseño y desarrollo) y el mantenimiento de un antivirus, respectivamente; en estos capítulos se aplica el conocimiento adquirido a lo largo del documento.

El estudio de esta asignatura no es sencillo; pero con dedicación, esfuerzo y conciencia se puede lograr una mayor eficiencia en la protección contra los códigos maliciosos. El presente trabajo de tesis colaborará a esta causa, fomentando que más y más personas se involucren en el apasionante mundo de la seguridad informática.

CAPÍTULO I

LOS SISTEMAS INFORMÁTICOS Y SU IMPORTANCIA

I.1. Los Sistemas Informáticos

El ser humano siempre ha buscado técnicas y procedimientos para manipular, preservar y transmitir *información*. Las culturas antiguas crearon complejos sistemas de escritura, los cuales se basaban en signos y símbolos que tallaban o pintaban en madera, piedra o tela. Al comienzo de la Edad Media los trovadores y juglares llevaban la información de pueblo en pueblo por medio de cánticos que ellos mismos componían y adornaban según la región que visitaban; los estudiosos de esa época escribían y transcribían libros con su propia mano. Con el nacimiento de la imprenta el proceso de transmisión y preservación de la información se volvió automatizado: podían crearse cientos de ejemplares de periódicos, revistas y libros en un tiempo más reducido. Desde principios y mediados del siglo XX la radio y la televisión han sido los medios más populares para la transmisión de información en tiempo real. En todos estos casos el procesamiento de la información lo lleva a cabo quien crea el libro, el grabado o el cántico.

Conforme ha evolucionado, científica y tecnológicamente, el ser humano ha encontrado mejores métodos para transmitir y preservar la información: ha inventado aparatos como la imprenta, la radio y la televisión. Sin embargo, desde muchos siglos atrás, las tareas de manipulación y procesamiento siempre han recaído en gran parte en el ser humano; ésto ha llevado a buscar métodos rápidos y eficientes para procesar la información. Debido a esta necesidad y al requerimiento de una transmisión más precisa, confiable y rápida, poco a poco se han creado mejores herramientas que han facilitado una solución. Así nacen los ordenadores.

El *ordenador* es la herramienta que actualmente nos permite el tratamiento automático de la información; nos facilita en gran medida su organización, proceso, transmisión y almacenamiento. Es una máquina compuesta de elementos físicos, denominados *hardware*, y un conjunto de instrucciones y datos, llamado *software*, capaz de realizar una gran variedad de trabajos a gran velocidad y con gran precisión.

El tratamiento de la información por medio de ordenadores ha llevado a la creación de una nueva disciplina llamada *Informática*. Se puede decir que la Informática, Ciencia o Ingeniería en Computación es el campo del conocimiento que abarca todos los aspectos del diseño y uso de los

ordenadores. Como disciplina, utiliza los métodos y procedimientos de los desarrollos teóricos, experimentales y de diseño; por ello se le considera tanto una ciencia como una ingeniería. La Informática es el conjunto de conocimientos que trata del diseño, análisis, implementación, eficiencia y aplicación de procesos que transforman la información. La herramienta principal de ésta disciplina es el ordenador; entonces, se puede afirmar que éste es un sistema informático.

I.1.1. Organización de los Sistemas Informáticos

Un ordenador es una máquina electrónica capaz de realizar cálculos con rapidez, obedeciendo instrucciones muy específicas y elementales que reflejan su estructura funcional y organizacional.

Se puede definir conceptualmente a una computadora como una máquina que consta de elementos de entrada, elementos de salida, una unidad de control y una memoria. Véase la figura 1.1.

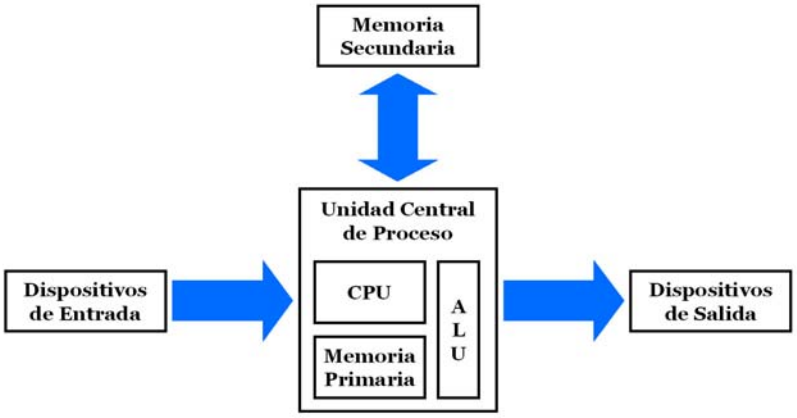


Figura 1.1. Esquema básico de un ordenador.

El esquema de la figura 1.1 muestra la relación que guardan los elementos del ordenador; todos ellos interconectados para lograr un fin común. Estas características demuestran porque se dice que el ordenador es un **sistema informático**.

Concepto de Sistema

El término sistema se utiliza de muchas maneras. Para los usuarios de ordenadores, un sistema es un conjunto de partes que están integradas con el propósito de lograr un objetivo. Las siguientes tres características son fundamentales:

- ✓ *Un conjunto de partes*. Un sistema tiene más de un elemento.
- ✓ *Partes integradas*. Debe existir una relación lógica entre las partes de un sistema.
- ✓ *El propósito de lograr un objetivo en común*. El sistema se diseña para alcanzar uno o más objetivos. Todos los elementos del sistema deben estar ligados y controlados de manera que se logre llegar al objetivo.

Dado que un ordenador es un grupo de partes integradas, que tiene el objetivo de llevar a cabo las operaciones que indica el programa en ejecución, entra dentro de la definición de sistema.

Como ya se dijo, un sistema es un conjunto de partes ligadas entre sí; por lo tanto, necesitará de un cierto número de componentes para realizar las acciones que lo lleven a cumplir con su objetivo. Se puede definir tres secciones fundamentales que se deben observar al momento de diseñar un sistema, sea cual sea su objetivo final:

- ✓ *Entrada al sistema*. Es la parte donde se introducen los elementos necesarios para que el sistema trabaje.
- ✓ *Cuerpo del sistema*. Es el lugar donde se realizan las acciones encaminadas a alcanzar el objetivo del sistema.
- ✓ *Salida del sistema*. Es donde se presentan los resultados obtenidos y se verifica si se alcanzó el objetivo planteado.

Estos segmentos definidos de un sistema se pueden observar en la figura 1.1.

Componentes Físicos de un Sistema Informático

El hardware, como se ha mencionado, es la parte física del computador. Tuvo especial importancia en las primeras generaciones de ordenadores debido a que no se utilizaban los medios electrónicos miniaturizados que se usan en la actualidad. Como todo sistema, la computadora se compone de unidades funcionales; éstas se definen a continuación.

Unidad de entrada. Se conoce como dispositivos de entrada. Es el conjunto de terminales por donde se introducen al computador los datos e instrucciones. En esta unidad se transforma los datos en señales binarias de naturaleza eléctrica. Como se observa en la figura 1.2, algunos de estos dispositivos son: a) teclado, b) ratón, c) escáner, d) cámara Web, e) lápiz óptico, etc.

Figura 1.2. Dispositivos de entrada.

Unidad de salida. También conocida como dispositivos de salida. Es el conjunto de terminales que muestran los resultados obtenidos por los programas ejecutados en el ordenador. La mayor parte de los componentes de esta unidad transforman las señales eléctricas binarias en caracteres, imágenes, sonido, etc. Dichos dispositivos se pueden observar en la figura 1.3: a) monitor o pantalla, b) impresora, c) altavoces, d) trazador de gráficos, etc.

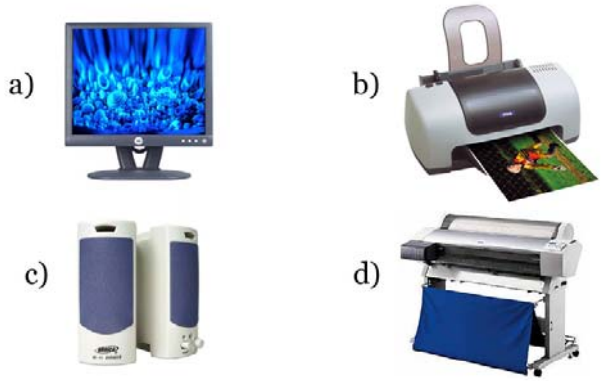


Figura 1.3. Dispositivos de salida.

A los diversos componentes de las unidades de entrada y salida se les conoce comúnmente como dispositivos periféricos. Existen también dispositivos que permiten entrada y salida de datos simultáneamente; por ejemplo, entre estos dispositivos encontramos a las tarjetas de red, pantallas táctiles y los módems.

Unidad central de proceso (UCP). Detecta y gobierna las señales de estado procedentes de distintas unidades, indicando su situación o condición de funcionamiento; capta las instrucciones del programa y genera señales de control dirigidas a todas las unidades. Consta de la *unidad aritmético-lógica (ALU)*, de la *unidad de control de proceso (CPU)* y de la *memoria principal*.

Almacenamiento secundario. Es un medio de almacenamiento que, a diferencia de la memoria principal, puede guardar una gran cantidad de información por tiempo prolongado, recuperarla antes de ser enviada a la memoria principal y almacenarla nuevamente. Se distinguen tres tipos de este almacenamiento: magnético (discos duros, discos flexibles, cintas), óptico (CD y DVD) y electrónico (*memoria USB, Flash, Memory Stick*).

Componentes Lógicos de un Sistema Informático

Para que el ordenador funcione, necesita información con la cual trabajar: órdenes y *datos*. La información que maneja una computadora puede ser de diferentes tipos, dependiendo del tratamiento que se le dé. Se puede afirmar que los datos se clasifican en:

- ✓ *Datos de entrada*: son los que se suministran al ordenador desde los periféricos de entrada.
- ✓ *Datos intermedios*: son aquellos datos que se encuentran en el proceso de tratamiento informático.
- ✓ *Datos de salida*: son llamados resultados y se muestran a través de los dispositivos de salida. Estos datos procesados reciben el nombre de información.

Las instrucciones u órdenes también reciben una clasificación; y en conjunto se llaman software. Sin el software, la computadora sería un conjunto de medios electrónicos inútiles; al cargar las instrucciones (también llamadas programas) en el ordenador éste comenzará a operar.

El software es un conjunto de programas, documentos, procedimientos y rutinas asociados con la operación de un sistema de cómputo; éste asegura que el sistema informático cumpla por completo con sus objetivos, opere con eficiencia, esté adecuadamente documentado y sea suficientemente sencillo de operar. Se puede clasificar en tres grandes categorías:

- ✓ *Sistemas Operativos*. Es el gestor y organizador de todas las actividades que realiza el computador; por lo tanto, el sistema operativo, debe ser cargado en la memoria principal antes que ninguna otra información. Sus principales funciones son: proporcionar una comunicación entre el usuario y el ordenador (interfase), administrar los dispositivos de hardware, administrar y mantener los sistemas de archivos de almacenamiento secundario y apoyar a otros programas durante su ejecución.
- ✓ *Lenguajes de Programación*. Es un lenguaje artificial que se utiliza para definir una secuencia de instrucciones que se procesarán en un ordenador. Existen dos grupos de lenguajes de programación: lenguajes de bajo nivel y lenguajes de alto nivel.
- ✓ *Software de Aplicación*. Está diseñado y escrito para realizar tareas específicas personales, empresariales o científicas. Todas estas aplicaciones procesan datos y generan información para el usuario. Se puede encontrar diferentes tipos de aplicaciones como: hojas de cálculo, procesadores de texto, manejadores de base de datos, antivirus, reproductores multimedia, etc.

I.1.2. Comunicación de Datos

El tratamiento de la información no estaría completo sin su transmisión a diversos sitios. Para lograr ese objetivo los actuales ordenadores no sólo interactúan con usuarios, también lo hacen con otros ordenadores. Al intercambio de información entre computadoras se llama comunicación de datos.

La comunicación entre computadoras siempre implica la transferencia de datos en bloques, en lugar de secuencias continuas. Esto se traduce en que no hace falta una conexión permanente entre dos ordenadores para intercambiar datos. La información no puede transmitirse de manera aleatoria; necesita regirse por diferentes normas para que tanto el ordenador emisor y el ordenador receptor lleven a cabo una comunicación exitosa. Estas normas se llaman protocolos.

Los protocolos son la base fundamental de la comunicación entre ordenadores y hacen posible que estos se conecten no sólo de uno a uno, también realizan la conexión de varios ordenadores conectados entre sí. Estos arreglos de computadoras se llaman redes de computadoras.

Redes de Computadoras

Las *redes* de computadoras son sistemas de comunicaciones, ya que permite comunicarse con otros usuarios y compartir archivos y periféricos; es decir, es un sistema de comunicaciones que conecta a varias unidades y que les permite intercambiar información.

Las redes consisten en compartir recursos, y uno de sus objetivos es hacer que todos los programas, datos y equipo estén disponibles para cualquiera de la red que así lo solicite, sin importar la localización física del recurso y del usuario. Un segundo objetivo consiste en proporcionar una alta fiabilidad, al contar con fuentes alternativas de suministro; por ejemplo, todos los archivos podrían duplicarse en dos o tres máquinas, de tal manera que si una de ellas no se encuentra disponible, podría utilizarse una de las otras copias. Además, la presencia de

múltiples CPU significa que si una de ellas deja de funcionar, las otras pueden ser capaces de encargarse de su trabajo, aunque se tenga un rendimiento global menor.

Las redes pueden clasificarse dependiendo de su *topología* o su alcance. Con respecto a su topología se clasifican en redes de bus, árbol, anillo, estrella, malla, híbrida, etc. (véase figura 1.4). Por su alcance se clasifican como red de área local (LAN, *Local Area Network*), red de área metropolitana (MAN, *Metropolitan Area Network*) y red de alcance mundial (WAN, *Wide Area Network*), como se muestra en la figura 1.5.

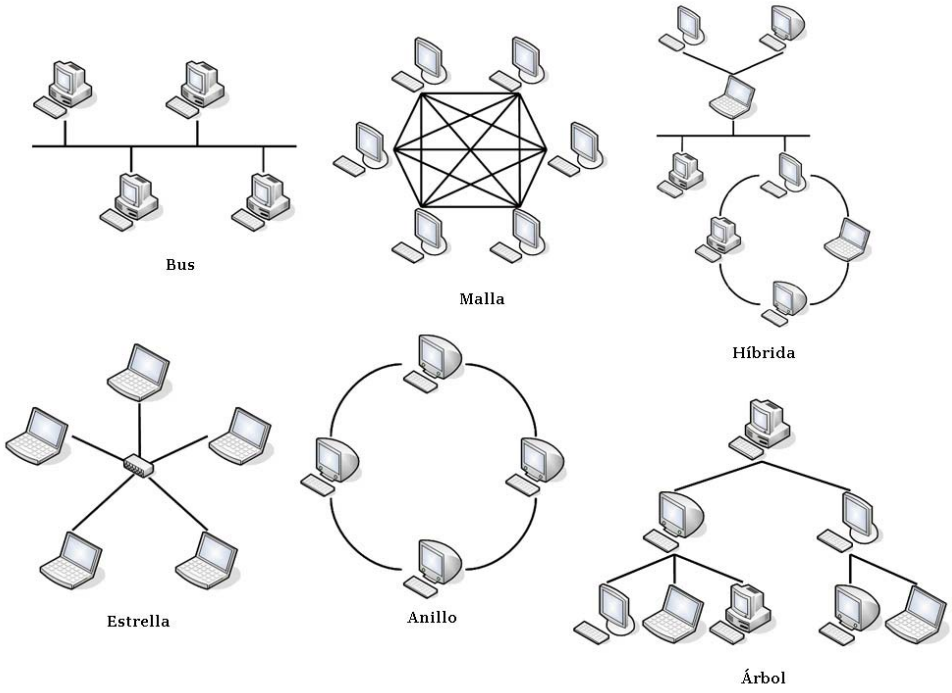


Figura 1.4. Topología de las redes de computadoras.

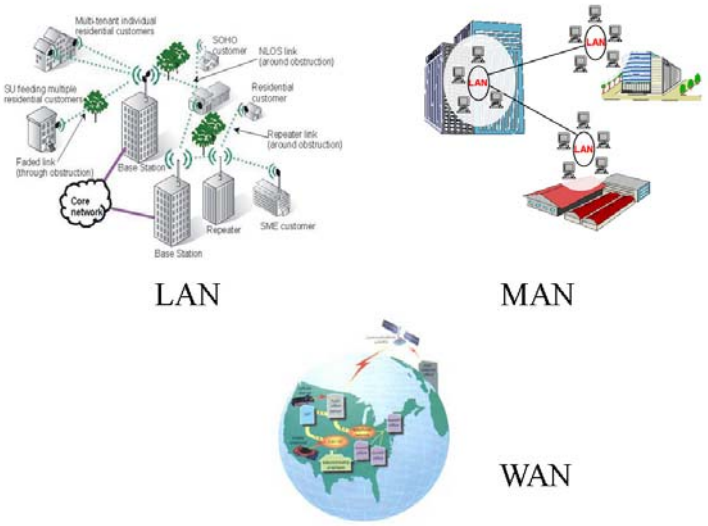


Figura 1.5. Alcance de las redes de computadoras.

El alcance de las redes le permite a millones de personas en el mundo compartir, buscar, procesar y transmitir información en fracciones de segundo. Sin embargo, estos beneficios aportados por las redes han originado que la información no siempre sea procesada para fines honestos. Para que la información no sea utilizada de manera ventajosa o perjudicial se ha desarrollado una disciplina importante en el mundo de la informática: la seguridad informática.

I.2. Importancia de la Seguridad Informática

I.2.1. Importancia de la Información

Cuando se habla de la Informática, generalmente se menciona tecnología nueva: nuevas aplicaciones, nuevos dispositivos de hardware, nuevas formas de elaborar información más consistente, etc. Sin embargo, se suele pasar por alto o se tiene muy implícita la base que hace posible la existencia de los anteriores elementos: la información.

Es muy importante conocer el significado de la información dentro de la función informática; de forma esencial cuando su manejo está basado en tecnología moderna. Para ésto se debe reconocer que la información:

- ✓ Se almacena y procesa en computadoras
- ✓ Puede ser confidencial para algunas personas
- ✓ Puede ser mal utilizada o divulgada
- ✓ Puede estar sujeta a robos, sabotaje o fraudes

Los primeros dos puntos muestran que la información está centralizada y que puede tener un alto valor para un grupo de personas; los últimos dos nos muestran que se puede provocar la destrucción total o parcial de la información, que incurre directamente en su disponibilidad e integridad y que puede causar retrasos de alto costo.

Es necesario tener presente que el lugar donde se centraliza la información, con frecuencia un sistema informático, puede ser el activo más valioso y al mismo tiempo el más vulnerable. Ésto convierte al sistema informático en el bien más importante de una persona o un grupo de personas; por lo tanto, es imprescindible que los propietarios destinen dinero, tiempo y esfuerzo para conservar sin alteraciones sus bienes (ordenadores e información).

I.2.2. Seguridad Informática

Antaño la seguridad no era mayor problema: la información se procesaba en centros de cómputo con una sola máquina y muy pocas terminales, el esquema de procesamiento estaba centralizado, había un único sistema operativo, las redes eran pequeñas y locales (pocos nodos y todos internos). Al no existir necesidad de seguridad no había razón para la misma.

Actualmente la seguridad se considera un gran dilema: los sistemas informáticos están dispersos alrededor del mundo, se dispone de numerosas máquinas de distinto tipo que se conectan en red y éstas, a su vez, se conectan a otras redes. Ello obliga a poner mayor atención a la seguridad, para la protección y prevención de daños a los bienes informáticos.

Contexto de la Seguridad Informática

El término seguridad de la información se refiere a la prevención y a la protección, a través de ciertos mecanismos, para evitar que ocurra de manera accidental o intencional, la transferencia, modificación o destrucción no autorizada de la información.

Seguridad informática es el nombre genérico dado a una colección de herramientas diseñadas para proteger datos y detener a los perpetradores; es decir, es la protección de los sistemas de cómputo para evitar amenazas de confidencialidad, integridad o disponibilidad. En la figura 1.6 se muestra el contexto de la seguridad informática, con lo cual se hace referencia a las condiciones que proponen las amenazas y que conforman el desarrollo y el uso de escudos.

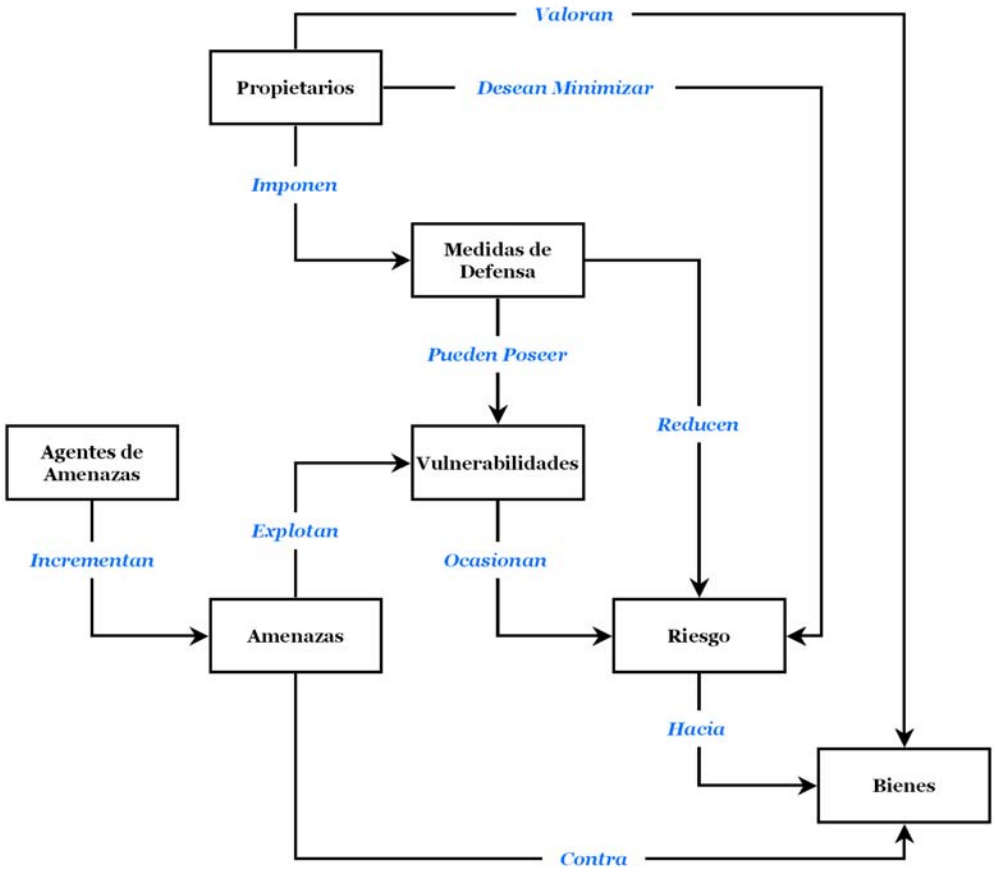


Figura 1.6. Contexto de la seguridad informática.

Ataques de Seguridad

En la figura 1.6, que muestra el contexto de la seguridad informática, aparece la palabra amenazas. El objetivo principal de la seguridad informática es reducir, lo más posible, el número de amenazas y de ataques propiciados por éstas hacia nuestros bienes.

Una amenaza es una condición del entorno que se representa a través de una persona, una circunstancia, un fenómeno o una idea maliciosa y que podría producir un daño si hay una violación de seguridad; es decir, una amenaza es todo aquello que intenta o pretende destruir. Por su parte, un ataque es la ejecución de una amenaza. Cuando se presenta una oportunidad de realizar la violación, automáticamente se está llevando a cabo un ataque.

Las amenazas de seguridad provienen de diversas fuentes descritas a continuación.

Amenazas humanas. La amenaza surge por ignorancia en el manejo de la información, descuido, negligencia, inconformidad. Se manifiesta por medio de robo, adulteración, modificación, revelación, pérdida, sabotaje o destrucción de la información o del sistema informático.

Amenazas tecnológicas. Estas amenazas se presentan cuando:

- ✓ Existen fallas físicas (de hardware) que presente cualquiera de los dispositivos que conforman al sistema informático.
- ✓ Hay fallas en el flujo de los canales de comunicación (de red) por donde circula la información.
- ✓ Se presentan fallas en el diseño o implementación de los programas (de software) que procesan la información.

Este tipo de amenazas se reconoce cuando hay interrupción de servicios, sobrecargas del sistema, pérdida de datos, interrupciones eléctricas, errores en programas, saturación y caída de la red.

Amenazas por desastres ambientales. Las amenazas de este tipo surgen por factores naturales como aire, tierra, agua, o fuego; repercuten en el funcionamiento físico de las computadoras, redes, instalaciones, líneas de comunicación. Se manifiestan en las inundaciones, incendios, terremotos, huracanes, derrumbes, etc.

Un diseñador de sistemas de seguridad debe identificar, en las normas o *políticas de seguridad* y en el *análisis de riesgos*, las amenazas que han de ser contrarrestadas, especificando los mecanismos de seguridad y servicios necesarios.

Clasificación General de Amenazas

Si modelamos un sistema informático como un flujo de información desde un origen hasta un destino, en este flujo no debe existir ningún tipo de obstáculo para que la información llegue al destinatario; ésto se muestra con claridad en la figura 1.7.

Figura 1.7. Flujo normal de la información.

Con base en lo anterior, encontraremos que existen cuatro categorías generales de amenazas (todo aquello que intenta o pretende destruir) o ataques (la ejecución de una amenaza).

Interrupción. Es un ataque contra la disponibilidad de la información en donde el recurso del sistema es destruido o no está disponible. Véase figura 1.8.

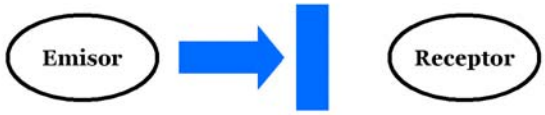


Figura 1.8. Interrupción.

Intercepción. Este ataque atenta contra la confidencialidad de la información. Consiste en que una entidad que no está autorizada consigue acceso al flujo de información. La intercepción se divide en: intercepción de datos, cuando se copian de forma ilícita programas, archivos o datos; y la intercepción de identidades, cuando se leen cabeceras de paquetes para revelar la identidad de uno o más emisores o receptores de información. Véase figura 1.9.

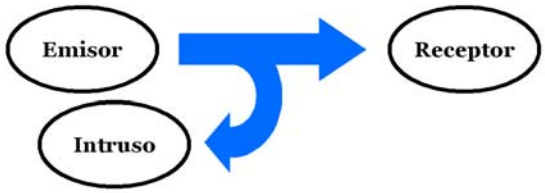


Figura 1.9. Intercepción.

Modificación. Es un ataque contra la integridad de la información. Una entidad que no esta autorizada no sólo consigue acceder a un recurso, sino que es capaz de manipularlo. Véase figura 1.10.

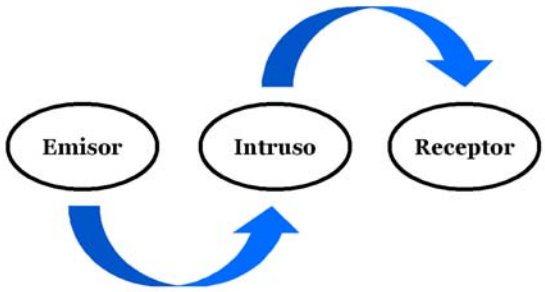


Figura 1.10. Modificación.

Suplantación. Una entidad que no está autorizada agrega objetos e información falsos en el sistema. Éste es un ataque contra la autenticidad. Véase figura 1.11.

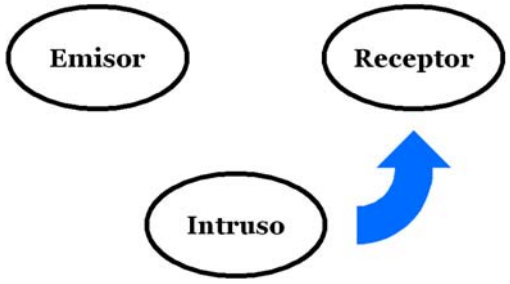


Figura 1.11 Suplantación.

Estos ataques pueden clasificarse asimismo en ataques pasivos y activos; de esta forma, se ordena a los ataques dependiendo si han o no alterado la información.

Ataques pasivos. Son aquéllos en donde no existe modificación alguna de la información; el atacante sólo monitoriza, observa, escucha u obtiene la información mientras está siendo transmitida. Los ataques pasivos son muy difíciles de detectar precisamente porque no se altera la información transmitida; sin embargo, son fácilmente evitables mediante el uso de distintos mecanismos de seguridad como lo es el cifrado de datos.

Ataques activos. Se entiende por ataque activo como la modificación del flujo de información; ya sea por adición, modificación o eliminación de datos en el flujo o por la creación de un flujo de datos falso.

Servicios de Seguridad

Para hacer frente a las amenazas y ataques a la seguridad del sistema se definen una serie de servicios para proteger los sistemas de proceso de datos y de transferencia de información; es el segundo aspecto que se considera en la seguridad de la información después del ataque de seguridad. Estos servicios hacen uso de uno o varios mecanismos de seguridad para proveer el servicio. Los *servicios de seguridad* se especifican en las siguientes líneas.

Confidencialidad. Requiere que la información sea accesible únicamente por las entidades autorizadas. La confidencialidad de datos se aplica a todos los datos intercambiados por las entidades autorizadas o tal vez a sólo porciones o segmentos seleccionados de los datos, por

ejemplo mediante cifrado. La confidencialidad de flujo de tráfico protege la identidad del origen y destino del mensaje, así como el volumen y el momento de tráfico intercambiado, de forma que sean indistinguibles para un intruso. La desventaja de estos métodos es que incrementan drásticamente el volumen de tráfico intercambiado, repercutiendo negativamente en la disponibilidad del ancho de banda bajo demanda.

Autenticación. Requiere una identificación correcta del origen de la petición, asegurando que la entidad no es falsa. Se distinguen dos tipos: de entidad, que asegura la identidad de las entidades participantes en el acceso, mediante biométrica, tarjetas de banda magnética, contraseñas, o procedimientos similares; y de origen de información, que asegura que una unidad de información proviene de cierta entidad.

Integridad. Permite que la información sólo pueda ser modificada por las entidades autorizadas; la modificación incluye escritura, cambio, borrado, creación y reactuación de los mensajes transmitidos. La integridad de datos asegura que los datos recibidos no han sido modificados de ninguna manera, mientras que la integridad de secuencia de datos asegura que la secuencia de bloques o unidades de datos recibida no ha sido alterada y que no hay unidades repetidas o perdidas.

No-repudio. Ofrece protección de un usuario frente a otro que niegue, posteriormente, que en realidad se realizó cierto acceso. Esta protección se efectúa por medio de una colección de evidencias irrefutables que permitirán la resolución de cualquier disputa. El no-repudio de origen protege al receptor de que el emisor niegue haber enviado una petición, mientras que el no-repudio de destino protege al emisor de que el receptor niegue haber recibido la petición.

Control de acceso. Requiere que el acceso a los recursos (información, capacidad de cálculo, nodos de comunicaciones, entidades físicas, etc.) sea controlado y limitado por el sistema destino, mediante el uso de contraseñas o llaves de hardware, protegiéndolos frente a accesos o usos no autorizados o manipulación.

Disponibilidad. Requiere que los recursos del sistema informático estén disponibles para todas las entidades autorizadas cuando alguna de ellas haga alguna petición de acceso o uso; ésto es, que los recursos puedan ser utilizados cada que se requieran y cuantas veces sea necesario.

I.2.3. Los Códigos Maliciosos

Las amenazas que atentan en contra de la información son muchas y muy variadas. Algunos de los factores que favorecen a las amenazas son:

- ✓ Desastres naturales,
- ✓ Fallas de hardware,
- ✓ Fallas de software,
- ✓ Códigos Maliciosos.

Existe una variedad muy extendida de programas que se dedican a modificar, robar y destruir información o también a espiar las actividades de usuarios. Este tipo de programas reciben, en conjunto, el nombre de códigos maliciosos o *malware*. Dentro de la clasificación de factores antes detallada, los códigos maliciosos son la amenaza creada por el ser humano más diseminada y difícil de controlar fuera de nuestro sistema informático.

La palabra *malware* proviene de la contracción de las palabras ***mal****icious soft****ware***. Estos programas o archivos ejecutables, que son dañinos para el ordenador, pueden ser virus, gusanos, troyanos o *spyware*, que intentarán conseguir el objetivo de dañar o robar la información contenida en el ordenador.

A lo largo de este trabajo se dedicará toda la atención a los virus; a continuación se describen otros códigos maliciosos.

Gusano. Son programas que no dependen de archivos portadores para poder contaminar otros sistemas. Cuando es ejecutado, modifican el sistema operativo, las aplicaciones y la

información con el fin de subsistir en red por medio de la replicación. Pueden propagarse por sí solos.

Caballo de Troya. Es una pieza de software dañino disfrazado de software legítimo. Es capaz de alojarse en computadoras y permitir el acceso a usuarios externos a través de una red, con el fin de recabar información o controlar remotamente la máquina "huésped". Los caballos de Troya no son capaces de replicarse por sí mismos y pueden ser adjuntados con cualquier tipo de software.

Puerta trasera (backdoor). Es un programa que permite el acceso al sistema de la computadora ignorando los procedimientos normales de autenticación. Pueden trabajar como los caballos de Troya o los gusanos.

Spyware. Son aplicaciones que se dedican únicamente a recopilar información sobre una persona u organización sin su conocimiento. Normalmente los programas espía infectan de manera parecida a los troyanos.

Exploit. Es aquel software que ataca una vulnerabilidad particular de un sistema operativo. Los *exploit* no son necesariamente maliciosos, algunos son generalmente creados por investigadores de seguridad informática para demostrar que existe una vulnerabilidad. Esta acción de los *exploit* les permite ser componentes comunes de otros programas maliciosos como los gusanos informáticos.

Rootkit. Son programas que son insertados en un ordenador después de que algún atacante ha ganado el control de un sistema. Incluyen funciones para ocultar los rastros del ataque; también pueden incluir puertas traseras (que permitirán al atacante obtener un nuevo acceso al sistema) o *exploits* (para atacar otros sistemas).

Actualmente la información es considerada uno de los bienes más relevantes: influye en economía, educación, sociedad, política y muchos otros aspectos de la vida del ser humano. Es por ello que los sistemas informáticos han evolucionado y se han vuelto una herramienta importantísima en el desarrollo de la humanidad. Son capaces de almacenar, procesar, interpretar, comunicar y presentar grandes volúmenes de información en un tiempo muy corto. Así pues, es necesario proteger al sistema informático y a la información que contiene, ya que un uso indebido de la misma puede ocasionar conflictos que desemboquen en pérdidas irrecuperables para el usuario.

CAPÍTULO II

VIRUS INFORMÁTICOS

II.1. Definición y Características de los Virus Informáticos

Virus es una palabra de origen latino cuyo significado es veneno. El término virus comenzó a utilizarse en la última década del siglo XIX para describir a seres microscópicos, más pequeños que las bacterias, compuestos tan sólo de material genético rodeado por una envoltura de proteínas y que son agentes causantes de enfermedades infecciosas.

Los virus son parásitos intracelulares obligados; sólo se replican en células con metabolismo activo, y fuera de ellas son sólo partículas inertes. Al infectar una célula huésped, el virus inserta su ácido nucleico en las cadenas del ácido de la célula para tomar el control de ésta última y así crear copias exactas o mutadas de sí mismo (replicarse), provocando la muerte celular. Las características de estos microorganismos (ser parásitos, dañinos y que se replican a sí mismos) fueron determinantes para trasladar el vocablo virus a la informática.

II.1.1. ¿Qué es un Virus Informático?

Definición

Muchos autores definen a los virus de diversas formas; cada uno de acuerdo a su propia concepción, conocimiento e investigación del tema.

Ralph Burger menciona en su libro Lo Que Debe Saber Acerca de los Virus Informáticos (*What You Should Know About Computer Virus*) que “un virus es un programa que puede insertar copias ejecutables de sí mismo en otros programas”.

Alberto Rojas en su artículo ¿Ya Vacunó su PC?, publicado en la revista mexicana PC/TIPS, define a los virus como “todo aquel código que al ser ejecutado altera la estructura del software del sistema y destruye programas o datos sin autorización y conocimiento del operador”.

Symantec Corporation define a los virus como “un pequeño programa creado para alterar la forma en que funciona un equipo sin el permiso o el conocimiento del usuario. Presentan dos características: es capaz de ejecutarse y de replicarse”.

Por su parte Fred Cohen, uno de los más reconocidos investigadores de virus informáticos, define a los virus como “programas que pueden infectar a otros programas modificándolos para incluir una, posiblemente evolucionada, versión de sí mismo”.

De las tres definiciones anteriores se observan características que son relevantes en la búsqueda de la definición exacta de virus informático: es un programa, crea copias de sí mismo, modifica archivos, programas o datos y sus actividades no son detectadas por el usuario.

Con base en lo anterior, se puede definir a los virus de la siguiente manera: *un virus informático es un programa que modifica otros programas para realizar copias de sí mismo y producir daños al ordenador al cual infecta, ocultando su presencia para que no sea detectado.* Esta definición hace hincapié en que un virus informático tiene tres características principales:

- ✓ *Es dañino*. Un virus informático siempre causa daños en el sistema que infecta. Es importante aclarar que el hacer daño no significa forzosamente que debe destruir; pueden presentarse efectos negativos para la computadora como consumo de memoria principal o tiempo de procesador; o para el usuario, como pérdida de tiempo, dinero y esfuerzo.
- ✓ *Se replica a sí mismo*. La característica más importante de este tipo de programas es la de crear copias de sí mismo, algo que ningún otro programa convencional realiza. La replicación no persigue otro fin más que el de propagación a otros archivos u ordenadores.
- ✓ *Es subrepticio*. Esto significa que utilizará técnicas para evitar que el usuario se de cuenta de su presencia. Los virus realizan sus actividades de manera clandestina; por lo tanto, deben ocultarse para impedir que sean detectados y detenidos en su cometido.

Tal como los virus biológicos, los virus informáticos se esparcen rápidamente, dificultando su erradicación. Pueden insertarse en casi cualquier tipo de archivo, siempre y cuando éste sea ejecutable o haga referencia a entidades ejecutables, y diseminarse mientras los archivos se copian, distribuyen y entregan entre usuarios.

Para que un virus cause algún daño a un sistema, debe ejecutarse; no puede realizar ninguna acción si sólo se ha copiado dentro del ordenador. Para dañar, realizan acciones tan variadas como mostrar imágenes o mensajes molestos, destruir archivos, modificar la tabla de particiones del disco duro, modificar la estructura del *BIOS*, reducir el espacio de almacenamiento y memoria o decrecer en general la capacidad de procesamiento del ordenador. Sin embargo, no pueden causar un daño directo sobre el hardware; no hay instrucciones que destruyan o derritan los componentes mecánicos y electrónicos del ordenador. En su defecto, un virus puede modificar los archivos controladores para ejecutar operaciones que reduzcan la vida útil de los dispositivos de hardware.

Un aspecto importante es que los virus no pueden infectar archivos de datos, de texto, de *HTML* o *PHP* puro (que no contengan una aplicación ejecutable), imágenes, vídeo, audio o documentos que no contengan macros; ya que este tipo de archivos no contienen instrucciones que el ordenador ejecute. Sin embargo, sí pueden causarle daños a estos archivos.

II.1.2. ¿Quién Hace los Virus?

Los ataques por virus cada vez son más frecuentes y graves. El incremento de estos incidentes se atribuye al rápido crecimiento y expansión de las redes, particularmente la Internet y las intranets.

Los virus informáticos están hechos por personas con conocimientos de programación, pero que no son necesariamente unos grandes expertos en computación; tienen conocimientos de lenguaje ensamblador y de cómo funciona internamente la computadora. Resulta bastante más difícil crear un programa, como un sistema de facturación, que un simple virus; que aunque esté mal programado, sería suficiente para molestar al usuario.

Actualmente, los virus son producidos en cantidades extraordinarias por gente en cualquier punto del planeta. Algunos de ellos dicen hacerlo por diversión, otros quizás para probar sus habilidades. La motivación de los autores de virus para llevar a cabo su obra es diversa. Algunos de los posibles factores que propician la creación de virus son:

- ✓ Algunos de los programadores de virus, sostienen que su interés por el tema es puramente científico; desean investigar todo lo relacionado con los virus y sus usos.
- ✓ Las agrupaciones de programadores de virus están abiertas a cualquiera que se interese en ellas; ofrecen consejos y pocas limitaciones.
- ✓ La escritura de programas virales da al programador cierta fuerza coercitiva: lo pone fuera de las reglas convencionales de comportamiento. El sentimiento de pertenencia es algo necesario para todo ser humano, y es probado que dicho sentimiento pareciera verse reforzado en situaciones marginales; en este caso la creación de programas dañinos.
- ✓ Muchas veces los programadores colocan sus creaciones al alcance de mucha gente (vía Internet, BBS especializadas, etc.), haciendo la salvedad de que el material es peligroso, por lo cual el usuario debería tomar las precauciones necesarias.
- ✓ Existen programadores que alegan que sus programas son creados para hacer notoria la falta de protección de que sufren la mayoría de los usuarios de computadoras.

La anterior es sólo una lista pequeña de factores que influyen en la creación de un virus. Sin embargo, estos factores son la fuente de muchos otros, como venganza, ambición o estrategia, que no necesariamente son motivos de un programador, sino de cualquiera que necesite los “servicios” de un virus, ya sea un solo individuo o un grupo.

II.2. Antecedentes Históricos y Clasificación

II.2.1. Historia

Los virus informáticos fueron creados casi al mismo tiempo que las computadoras. No existen fuentes fidedignas que establezcan, de manera precisa y puntual, la historia de los virus y de los contagios virales. La causa probable es que los organismos gubernamentales, científicos,

militares o empresariales ocultaron hechos sobre ataques virales, ya que no reconocían las vulnerabilidades en sus sistemas de seguridad a los que consideraban infalibles.

Sin embargo, debido a hechos y testimonios aislados, que se han divulgado en diversos medios especializados, es posible construir una pequeña visión del desarrollo de la virología informática. A continuación se expone una breve síntesis de la evolución de los virus informáticos, desde sus inicios hasta nuestros días.

El Inicio de los Virus

En 1939, el matemático de origen húngaro John Louis Von Neumann escribió un artículo, publicado en una revista científica de Nueva York, exponiendo su Teoría y Organización de *Autómatas* Complejos (*Theory and Organization of Complicated Automata*), donde demostraba la posibilidad de desarrollar pequeños programas que pudiesen tomar el control de otros de similar estructura. Es el primer documento donde se expone la idea del código auto replicable.

Core Wars

Se reconoce como el primer antecedente de los virus actuales un juego creado por programadores de la empresa AT & T, que desarrollaron la primera versión del sistema operativo *UNIX*, a principios de la década de 1960. En los laboratorios de la Bell Computer, subsidiaria de la AT & T, tres jóvenes programadores: Robert Thomas Morris, Douglas McIlory y Victor Vysottsky, a manera de entretenimiento y como parte de investigaciones, crearon un juego al que denominaron *Core Wars* (Guerras en el Núcleo), inspirados en la teoría de John Von Neumann.

Core Wars es un juego entre programas hecho en lenguaje ensamblador. En este juego, dos programadores creaban “organismos” de software que luchaban entre sí por un espacio de memoria y cuyo vencedor era el que conseguía más memoria o el que aniquilaba al programa contrario. Cada jugador debía “matar” al programa oponente sobrescribiéndolo con su propio código; los programas debían sobrevivir utilizando técnicas de ataque, ocultamiento y reproducción similares a las de los actuales virus informáticos.

Conscientes de lo peligroso del juego, decidieron mantenerlo en secreto, y no hablar más del tema; no se sabe si esta decisión fue por iniciativa propia, o por órdenes superiores. Por lo tanto, pese a su popularidad entre investigadores, no se supo nada de *Core Wars* hasta 1983: Ken Thompson da a conocer *Core Wars* y anima a la experimentación con esas pequeñas "criaturas lógicas". La revista *Scientific American* difundió a *Core Wars* con una explicación detallada del funcionamiento del programa, lo que provocó que muchos de sus lectores experimentaran con ella. Así aparecen los primeros virus experimentales.

La Epidemia Actual

Los antecedentes de los virus actuales después de la creación del *Core Wars* (década de los 60) y hasta principios de los 80 son inciertos e imprecisos. Un único antecedente confirmado se produce en 1974: la Xerox Corporation presentó un programa que ya contenía un código duplicador de sí mismo.

1980. Se creó un virus para la Apple II, diseñado para propagarse por medio de disquetes de arranque. El virus se llamó Elk Cloner y al ejecutarse desplegaba la siguiente rima:

It will get on all your disks
It will infiltrate your chips
Yes it's Cloner!
It will stick to you like glue
It will modify RAM too
Send in the Cloner!

1983. En la Universidad del Sur de California, Fred Cohen presentó un virus residente en una PC y demostró que algunos programas para computadora podían replicarse a si mismos, introducirse en otros programas y alterar el funcionamiento de las computadoras. En 1984 publicó su libro Virus Informáticos: Teoría y Experimentos (*Computer Viruses: Theory and Experiments*) donde define lo qué son los virus informáticos y describe los experimentos que

realizó para comprobar sus hipótesis sobre los códigos replicables. Estas investigaciones lo convirtieron en el primer investigador oficial de los virus.

1986. Aparece el primer virus dañino y destructivo. Este virus, llamado BRAIN, fue creado en Pakistán por una compañía que vendía copias ilegales de programas comerciales. En Estados Unidos las copias de BRAIN fueron modificadas, dando lugar a diferentes cepas, más peligrosas que la original. BRAIN infectaba los sectores de arranque de disquete, impidiendo leer su contenido; colocaba en la etiqueta de volumen del disquete ©BRAIN.

1987. Se creó en la IBM un programa para desinfectar el correo interno, pues estaba contaminado con un virus que hacía aparecer un mensaje junto a un árbol navideño. Al escribir la palabra *CHRISTMAS* el virus se esparcía por la red.

1988. Aldus Corporation lanza al mercado el programa Frenad para *Macintosh*; éste estaba infectado por el virus Macintosh Peace. El virus mostraba un mensaje de paz en la pantalla para celebrar el aniversario del lanzamiento de la Macintosh II.

Se identificó el virus Jerusalén, mejor conocido como Viernes 13, en la Universidad Hebrea de dicha ciudad. Según algunas versiones, fue creado por la Organización para la Liberación de Palestina con motivo de la celebración del cuarenta aniversario del último día en que Palestina existió como nación.

Un virus invade computadoras basadas en UNIX en universidades e instalaciones de investigación militares, donde las velocidades de procesamiento fueron reducidas y en otros casos detenidas. El programa se difundió a través de un corrector de errores para correo electrónico, que se movió principalmente en *Internet* (en ese entonces la *ARPANET*) y contaminó miles de computadoras en todo el mundo, incluyendo la NASA que vio interferidas sus actividades durante el lanzamiento del Transbordador Espacial *Atlantis*. El culpable de la infección es Robert Morris, hijo de Robert Thomas Morris uno de los creadores de *Core Wars*, estudiante de 23 años, que declaró haber cometido un error al propagar el gusano con el cual experimentaba en su propio ordenador.

1989. Aparece el primer antivirus heurístico, y los primeros virus con nuevas técnicas de ocultamiento. El antivirus era capaz de detectar, no sólo los virus que ya eran conocidos, sino también aquéllos que surgieran en el futuro y que reprodujesen patrones sospechosos.

1991. Surgieron los primeros *kits* para construcción de virus, lo que facilitó y aumentó, en gran medida, su velocidad de creación. El primero fue el VCL (*Virus Creation Laboratory*), creado por *Nowhere Man*, y más tarde apareció el *Phalcon/Skism Mass-Produced Code Generator*, de *Dark Angel*.

1992. El virus Michelangelo atacó por primera vez; es el virus que más publicidad ha recibido, gracias a ello se tuvo más conciencia sobre lo que provocan los virus.

1993. AVISPA se convirtió en epidemia al infectar archivos .EXE cada vez que se ejecutaba. Este virus siempre se cifra con una clave distinta (polimórfico), para dificultar su detección por medio de antivirus heurísticos. CAMOUFLAGE II infecta el sector de arranque de los disquetes y la tabla de partición de los discos rígidos; es bastante simple y fácil de ser detectado. LEPROSO (creado en Rosario, provincia de Santa Fe), se activa el día 12 de Enero (cumpleaños del autor), y hace aparecer un mensaje que dice: “Felicitaciones, su máquina está infectada por el virus leproso creado por J. P. Hoy es mi cumpleaños y lo voy a festejar formateando su disco rígido. Bye...”.

1994. El virus NATAS inició una epidemia por todo México; se cree que es un virus de origen mexicano.

1997. Apareció un virus de macro sin características especiales ni efectos dañinos destacables, pero tuvo una amplia repercusión en la prensa, debido a que tiene el anecdótico interés de haber sido escrito en memoria de la princesa Diana Spencer. El efecto del virus es la reproducción en pantalla de la letra de la canción *Candle In The Wind*.

1998. Aparecen los primeros virus de macro para Excel y Access. El virus Laroux fue el primer virus de macro para Excel y ha sido ampliamente copiado e imitado; se sitúa entre los

virus que provocan más infecciones. El primer virus de Access fue el AccessiV; éste reemplaza el macro autoexec y copia módulos adicionales a la base de datos. El virus Strange Brew es el primer virus conocido de *Java*, capaz de replicarse únicamente en caso de que se le permita acceso a los archivos del disco; es decir, que sólo funcionará bajo ciertas condiciones como aplicación Java, y no como *applet*. No es en absoluto peligroso e infectarse por medio de él es muy difícil. En definitiva, las posibilidades de extensión de este primer virus de Java son muy limitadas.

1999. Aparecen los virus que están diseñados para explotar los recursos de la red, y así llevar a cabo una rápida propagación. Ejemplares como Melissa, Happy99 o Explore.Zip son sólo ejemplos de lo que puede ser un virus de nueva generación.

2000. Love Letter se convirtió en el gusano más rápido en propagarse al colapsar ordenadores desde el correo electrónico. El gusano de VB Script Timofonica trató de infectar por medio de la red telefónica española. Se desarrolló el primer Troyano para Palm PDA llamado Liberty. Pirus, descubierto el 9 de noviembre de 2000, se adhería por sí solo a archivos HTML o PHP.

2001. Gnuman fue el primer virus oculto dentro del sistema para compartir archivos de Gnutella (el primero en atacar un sistema de comunicación punto a punto) y pretendía ser un archivo MP3 para descargar. Se propagó un virus que podía infectar las plataformas Windows y Linux: Winux (o Lindows) fue creado en la República Checa. Se encontró el primer gusano Apple Script que usaba Outlook Express (o Entourage en Macintosh) para diseminarse por correo electrónico.

2002. Donut fue el primer gusano que atacaba los servicios .NET. Le siguió Sharp-A un virus hecho en C++. Apareció después SQLSpider que atacaba equipos donde está instalado Microsoft SQL Server (y programas que usen SQL Server). Benjamin utiliza la red punto a punto de KaZaa para diseminarse.

2003. Blaster (también conocido como Lovsan y MSBlast), se diseminaba rápidamente a través de una vulnerabilidad en *Windows Distributed Component Object Model* (DCOM) y la interfase *Remote Procedure Call* (RPC); fue la mayor amenaza vista en 2003.

2004 – 2006. Estos años se destacaron por la siguiente oleada de gusanos y troyanos que invadieron Internet, donde destacan Trojan.Xombe, Randex, Witty, Troj/Stinx-E y otros más. En el nuevo sistema operativo de Apple, Mac OS X, se detecta el código OSX/Leap.A. A la fecha, los nuevos códigos maliciosos replicables siguen surgiendo y atacando nuevas vulnerabilidades de los sistemas operativos más populares. La guerra contra estos programas continúa.

II.2.2. Clasificación de los Virus

La clasificación correcta de los virus siempre resulta variada según los criterios que se consulten; se puede agruparlos por la entidad que parasitan (sectores de arranque o archivos ejecutables), por su grado de dispersión a nivel mundial, por su comportamiento, por su agresividad, por sus técnicas de ataque o por cómo se ocultan. A continuación se expone dos clasificaciones, las cuales desde este punto de vista, son las más sobresalientes: por objetivo de infección y por comportamiento.

Clasificación por Objetivo de Infección

Esta clasificación obedece al lugar donde se aloja un código viral cuando entra en un ordenador. Las categorías de esta clasificación son virus de sectores de disco y virus de archivos.

Virus de Sectores de disco. Los sectores del sistema son áreas especiales de los discos de almacenamiento donde se guardan programas que se ejecutan cada vez que el ordenador es encendido. Cada disco tiene un sistema de sectores que es invisible para las aplicaciones, pero vital para el correcto funcionamiento del ordenador. Son un blanco muy común para el ataque de un virus.

Virus de sector de arranque. Estos virus infectan el área de sistema de un disco; es decir, el registro de arranque de los disquetes y los discos duros. Los virus del sector de arranque se copian en esta parte del disco y se activan cuando el usuario intenta iniciar el sistema desde el disco infectado. Estos virus están residentes en memoria por naturaleza. La mayoría se crearon para *DOS*, pero todos los equipos, independientemente del sistema operativo, son objetivos potenciales para este tipo de virus. Para que se produzca la infección basta con intentar iniciar el equipo con un disquete infectado; posteriormente, mientras el virus permanezca en memoria (mientras no se apague el ordenador), todos los disquetes que no estén protegidos contra escritura quedarán infectados al acceder a ellos. Algunos ejemplos de virus del sector de arranque son Form, Disk Killer, Michelangelo y Stoned.

Virus de sector de arranque maestro. Estos virus están residentes en memoria e infectan los discos de la misma forma que los virus del sector de arranque. La diferencia entre ambos tipos de virus es el lugar en que se encuentra el código viral; los virus del sector de arranque maestro normalmente guardan una copia legítima del sector de arranque maestro en otra ubicación. Algunos ejemplos de virus del sector de arranque maestro son NYB, AntiExe y Unashamed.

Virus de archivos. Estos virus atacan a los archivos de programa, no importando si se trata de archivos del sistema operativo o de otras aplicaciones. Normalmente infectan el código ejecutable contenido en archivos con extensiones .COM y .EXE, por ejemplo. También pueden infectar otros archivos cuando se ejecuta un programa infectado desde un disquete, una unidad de disco duro o una red. Muchos de estos virus están residentes en memoria; una vez que la memoria se infecta, cualquier archivo ejecutable que no esté infectado pasará a estarlo. Algunos ejemplos conocidos de virus de este tipo son Jerusalem y Cascade.

Virus múltiples. Son virus que infectan tanto los registros de arranque como los archivos de programa. Son especialmente difíciles de eliminar: si se limpia el área de arranque, pero no los archivos, el área de arranque volverá a infectarse; ocurre lo mismo a la inversa, si el virus no se elimina del área de arranque, los archivos que hayan sido limpiados volverán a infectarse. Algunos ejemplos de virus múltiples son One_Half, Emperor, Anthrax y Tequilla.

Clasificación por Comportamiento

El criterio para esta clasificación se refiere a la forma en que el virus trabaja cuando se encuentra dentro de un ordenador. Un virus catalogado de esta manera puede entrar dentro de una o más categorías.

Virus polimórficos. Los virus polimórficos son quizás los más complejos y difíciles de detectar y en consecuencia de eliminar. Poseen la capacidad de cifrarse para que no sean detectados fácilmente; la rutina y la clave que se encargarán de descifrar el virus quedan en claro. Una vez descifrado el virus intentará alojarse en algún archivo de la computadora.

Ahora se tiene un virus que presenta una forma distinta a la original, en la que puede infectar libremente. Para que el virus presente su característica de cambio de formas debe poseer una rutina especial. Si mantuviera siempre su estructura, esté cifrado o no, cualquier antivirus podría reconocer ese patrón; para eso incluye un generador de claves al que se conoce como motor de mutación (*mutation engine*). Este motor utiliza un generador numérico aleatorio que, combinado con un algoritmo de cifrado, modifica la firma del virus. Gracias al motor de mutación el virus creará varias, pero diferentes copias de sí mismo, manteniendo operativo su código viral. El virus siempre se propaga con la misma rutina de descifrado, lo único que cambia es la clave de cifrado; por esta razón, el cuerpo cifrado del virus cambiará de una infección a otra (véase figura 2.1).

Figura 2.1. Comportamiento del virus polimórfico.

Los métodos básicos de detección no pueden dar con este tipo de virus. Muchas veces para virus polimórficos particulares existen programas que se dedican especialmente a localizarlos y

eliminarlos. Las últimas versiones de los programas antivirus ya cuentan con detectores de este tipo de virus.

Virus sigilosos. El virus sigiloso posee un módulo de defensa bastante sofisticado. Éste intentará permanecer oculto tapando todas las modificaciones hechas a los archivos o al sector de arranque infectados; anulan efectos tales como el tamaño de los archivos, los cambios de la fecha, hora y atributo. Cuando uno de estos virus está activo en memoria, cualquier cambio pasará desapercibido, ya que el virus habrá tomado el control. Para producir estos efectos los virus utilizan las *interrupciones* asociadas al manejo de archivos.

Cuando un virus se adiciona a un archivo el tamaño de éste aumenta; ésta es una clara señal de que un virus lo infectó. La técnica de ocultamiento (*stealth*) captura las interrupciones del sistema operativo que solicitan ver los atributos del archivo; el virus le devuelve la información que poseía el archivo antes de ser infectado y no las reales. Algo similar pasa con la técnica de ocultamiento de lectura: cuando el sistema operativo solicita leer una posición del archivo, el virus devuelve los valores que debería tener ahí y no los que tiene actualmente. En el caso del sector de arranque del disco, la técnica consiste en desviar el arranque original a otros sectores del disco que se marcan como erróneos.

Virus rápidos y virus lentos. Los virus rápidos están diseñados para infectar la mayor cantidad de archivos posibles; por lo tanto, un virus rápido puede contaminar cada archivo al que se tiene acceso. La desventaja de este método radica en que mientras más archivos sean infectados, más fácil será la detección, ya que el virus puede realizar muchas acciones sospechosas que pueden ser detectadas y notificadas por el antivirus.

Por otro lado, los virus lentos están diseñados para infectar archivos de manera más discreta: los virus lentos sólo atacan a los archivos en la medida que éstos son ejecutados, modificados o creados; simplemente aprovechan cada una de los programas que se ejecutan. Están diseñados para evadir su detección al limitar su actividad.

Normalmente ambos tipos de virus llevan una bitácora de los archivos que ya han infectado para no modificarlos nuevamente.

Retrovirus o virus antivirus. Un retrovirus intenta, como método de defensa, atacar directamente al programa antivirus incluido en la computadora. Para los programadores de virus ésta no es una información difícil de obtener; ya que pueden conseguir cualquier copia de antivirus que hay en el mercado, descubrir cuáles son los puntos débiles del programa y buscar una buena forma de aprovecharse de ello.

Generalmente, los retrovirus buscan el archivo de definiciones de virus y lo eliminan, imposibilitando al antivirus la identificación de sus enemigos. Suelen hacer lo mismo con el registro del comprobador de integridad; pueden contener instrucciones que borran o infectan directamente a los archivos de las vacunas.

Otros retrovirus detectan al programa antivirus en memoria y tratan de ocultarse o inician una rutina destructiva antes de que el antivirus logre encontrarlos. Algunos incluso modifican el entorno para afectar el funcionamiento del antivirus.

Virus voraces. Estos virus alteran el contenido de los archivos de forma indiscriminada. Generalmente uno de estos virus sustituirá el programa ejecutable por su propio código. Son muy peligrosos porque se dedican a destruir completamente los datos que puedan encontrar.

Bombas lógicas. Este tipo de virus contienen instrucciones que activan su código cuando se da una condición lógica en el sistema; por ejemplo, una combinación de letras, que se ejecute cierto archivo, alguna orden del sistema operativo, una invocación a una interrupción, una hora o una fecha. Un caso especial de estos códigos es cuando utilizan la fecha como detonador; se les conoce como bombas de tiempo. En algunos casos, poco comunes, se utiliza alguna hora del día.

Conejo. También son conocidos como peste. Cuando se ejecuta, el código viral se reproduce a toda velocidad y de forma infinita. El programa se coloca en la cola de procesos y cuando llega su turno se ejecuta haciendo una copia de sí mismo, agregándola también en la cola de espera; los procesos al ser ejecutados van multiplicándose hasta consumir toda la memoria de la computadora – primero la memoria primaria y después la secundaria –, interrumpiendo todos los procesos.

Virus de macro. Estos virus infectan los archivos de documentos. Utilizan el lenguaje de programación interno de otro programa o aplicación, creado para permitir a los usuarios automatizar ciertas tareas dentro de ese mismo programa. Debido a la facilidad con que se pueden crear estos virus, existen actualmente miles de ellos en circulación; con la llegada de Visual Basic en Microsoft Office 97, se puede crear un virus de macro que no sólo infecte los archivos de documentos, sino también otros archivos. Los virus de macro infectan archivos de Microsoft Office: Word, Excel, PowerPoint y Access. Actualmente están surgiendo también nuevos derivados en otros programas.

Virus falsos. También son conocidos como virus hoax. Los virus falsos son simplemente mensajes que circulan por correo electrónico que advierten sobre algún virus inexistente. Estos virus falsos no infectan el sistema, sólo son falsas alarmas, que se multiplican y se mandan por Internet con una gran velocidad; no tienen ningún código oculto ni instrucciones para ejecutar. La forma de replicarse es la siguiente: un usuario recibe un correo con la advertencia de algún virus peligroso, el usuario lo reenvía a otros para advertirles, entonces se genera en la red un tráfico sobre una amenaza inexistente.

II.3. Funcionamiento y Formas de Infección

II.3.1. Ciclo de vida de un Virus

Las epidemias virales ocurren cuando un ordenador o una red se encuentra abrumado por varias copias de algún virus, causando diversos daños (según sea el virus que contaminó el

sistema). Como todo programa de computación, un virus tiene un ciclo de vida; éste comienza con su creación y termina cuando es erradicado.

ETAPA 1: Creación. Hace unos años, crear un virus requería conocimientos sobre computación y programación; hoy en día, cualquiera puede programar un virus, teniendo o no conocimientos sobre programación.

ETAPA 2: Propagación. Los virus se replican por naturaleza; crea copias de sí mismo para infectar otros programas o sistemas. Un virus bien diseñado se replicará por un período largo e indefinido antes de comenzar con sus actividades dañinas; esto permite un mayor alcance en su diseminación.

ETAPA 3: Activación. Las rutinas dañinas de un virus se activan dentro de ciertas condiciones; por ejemplo, en alguna fecha o cuando el usuario ejecuta cierta actividad en el ordenador. Los virus sin rutinas dañinas causan malestar al ocupar espacio de almacenamiento. En estas rutinas también se encuentra incluida la replicación.

ETAPA 4: Descubrimiento. Cuando un virus es detectado y aislado, es llevado a la Asociación Internacional de Seguridad Informática (*International Computer Security Association*, http://www.icsa.net) en Washington, D.C., para ser documentado y entregado a los desarrolladores de antivirus. Algunas veces el descubrimiento de un virus acelera su activación.

ETAPA 5: Asimilación. En esta etapa, los desarrolladores de antivirus modifican su software para que detecten al nuevo virus. Ésto puede llevar desde un día hasta meses, dependiendo del desarrollador y del tipo de virus. La actualización se basa en la documentación creada en la etapa del descubrimiento.

ETAPA 6: Erradicación. Si un número suficiente de usuarios instala adecuadamente la actualización del antivirus, los nuevos virus pueden ser eliminados. Desde luego ningún virus ha desaparecido completamente, pero muchos han dejado de ser una peligrosa amenaza. Desafortunadamente, la erradicación de un virus es el inicio de la creación de otro.

En la figura 2.2 se muestra un esquema del ciclo de vida de un virus.

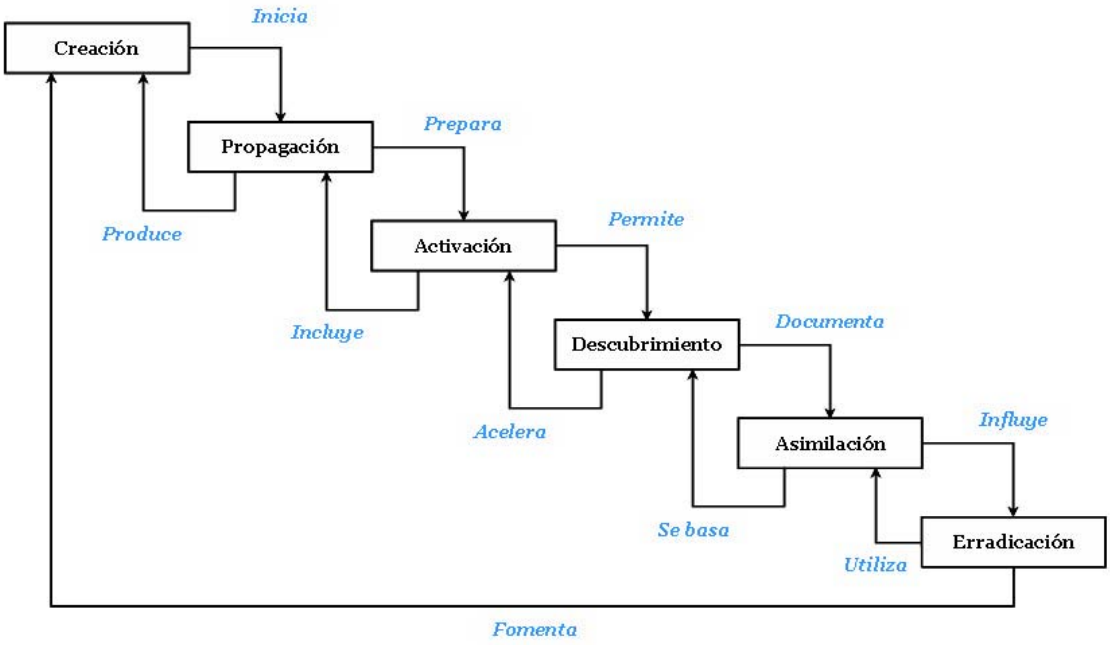


Figura 2.2. Ciclo de vida de un virus informático.

II.3.2. Configuración de un Virus

Para cumplir un cometido en específico, los programas computacionales están divididos en subrutinas o módulos. Esta característica permite aumentar la eficiencia de un programa ya que cada subrutina realiza una sola actividad. Como todo programa, un virus también puede ser programado en subrutinas, las cuales varían de uno a otro, dependiendo del talento y la habilidad del programador. En su forma más general, un virus puede contener tres módulos principales:

- ✓ *Módulo de replicación*. Es el encargado de manejar las rutinas para infectar entidades ejecutables que asegurarán la subsistencia del virus; cuando toma el control del sistema infecta otras entidades ejecutables. Cuando estas entidades sean trasladadas a otras computadoras se asegura la dispersión del virus.

- ✓ *Módulo de ataque*. Contiene las rutinas de daño adicional o implícito. El módulo puede ser disparado por distintos eventos del sistema: una fecha, hora, el encontrar un archivo específico, el encontrar un sector específico, una determinada cantidad de arranques desde que ingresó al sistema, o cualquier otra cosa a la que el programador quisiera tomar como detonador del ataque.
- ✓ *Módulo de defensa*. Su principal objetivo es proteger el cuerpo del virus; incluirá rutinas que intentarán que el virus permanezca invisible a los ojos del usuario y del antivirus, disminuyendo los síntomas que delaten su presencia. Las técnicas incluidas en este módulo resultan ser muy sofisticadas; pueden dar información falsa al sistema operativo – y en consecuencia al usuario – o localizándose en lugares poco comunes para el registro de los antivirus, como la memoria Flash-ROM.

II.3.3. ¿Cómo Funciona un Virus?

La forma en que un virus actúa es tan diversa y propia como su tipo, su objetivo de infección y la forma de su programación. A continuación se expone, de manera general, la secuencia que un virus debe recorrer para replicarse y causarle daño a un sistema informático.

Puntos de Entrada al Sistema

Se distingue dos vías que un virus utiliza como entrada al ordenador: puede hacerlo por un puerto de comunicaciones o bien por un dispositivo de almacenamiento secundario extraíble (disquetes, CD, DVD, etc.).

Cuando un virus entra por un puerto de comunicación puede traspasar cualquier tipo de control de acceso o de protocolos de intercambio de información; esta técnica es utilizada comúnmente por los caballos de Troya y sobre todo por los gusanos. En muchas otras ocasiones el virus entra por medio del correo electrónico o directamente de la mano del usuario del equipo, al permitir que se ejecuten o descarguen programas desde Internet.

El método clásico es la infección desde un disco, mediante un programa infectado. Un virus que infecta desde Internet puede infectar desde disco y viceversa. Se debe aclarar que el simple hecho de llevar el virus en un disco (duro, flexible u óptico) o de introducirlo dentro del ordenador no produce la infección ni mucho menos su propagación; ésta se produce exclusivamente cuando se ejecuta el programa contaminado.

Replicación

Una vez que el virus ha entrado en el ordenador, sólo es cuestión de tiempo para que comience la infección. La llegada del intruso no tiene ningún efecto mientras su secuencia de código no sea ejecutada. Como el virus es ante todo un conjunto de instrucciones, no se diferencia de otros programas.

Para que el virus comience una infección tendrá que sembrar una copia de sí mismo en algún lugar de la memoria, ya sea primaria o secundaria. En el caso de la memoria primaria (RAM) resultará infectado todo archivo que sea ejecutado, y que por lo tanto, pasa por esta memoria. Para la memoria secundaria un buen lugar para dejar la copia es un archivo; por ejemplo, el COMMAND.COM de MS-DOS. Un virus puede ser capaz de replicarse desde cualquier tipo de memoria en que se encuentre.

Un virus intentará infectar el máximo de archivos para así iniciar una propagación, con forma de progresión geométrica, por todo el ordenador. El objetivo de la infección es garantizar que el código se ejecute más veces, para replicarse, propagarse o para realizar sus actividades dañinas. Ésta es la razón por la cual el virus elija contaminar programas que se sabe de antemano el sistema ejecutará con frecuencia; no se trata de una regla fija, pero es la más común. Para otros códigos virales tiene más interés la propagación a otros equipos, por lo que se anidan en programas susceptibles de ser compartidos.

Desde estos archivos el virus ha obtenido el control del ordenador. Debe tener especial cuidado en no reinfectar o destruir el archivo portador original, ya que un daño en el código causaría un mal funcionamiento del propio virus. Simplemente debe añadir copias de sí mismo

mientras el programa ejecuta sus funciones normales, y de ser posible ejecutarse antes que el código verdadero.

Cuando un virus infecta un archivo puede realizar una de dos acciones: añadir su código al programa, o sustituir un fragmento del programa por el código viral. Esta última técnica se conoce como *overwrite* y su principal desventaja es que la detección es inmediata ya que el archivo huésped queda dañado y se pueden producir errores de ejecución.

La figura 2.3 muestra un esquema general de cómo se replica un virus dentro del ordenador.

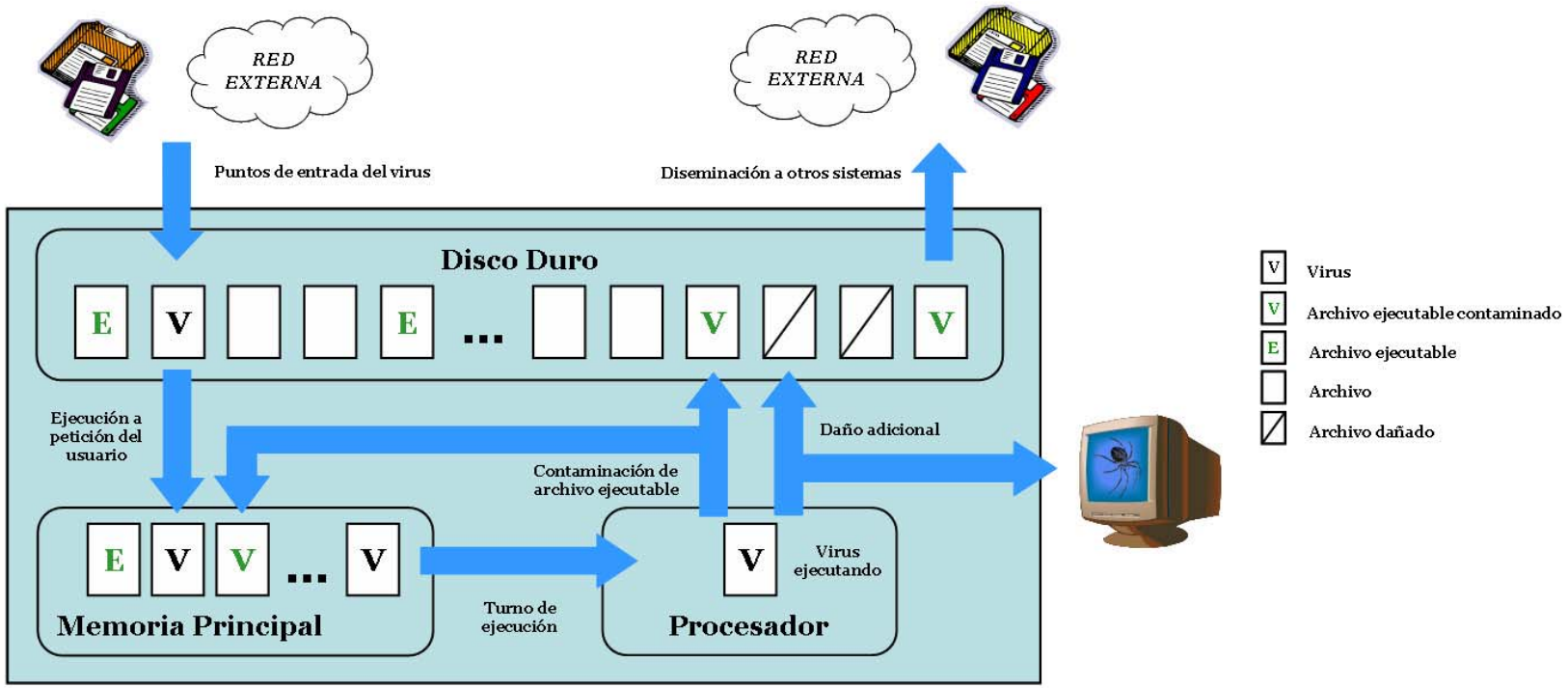


Figura 2.3. Replicación general de un virus.

La capacidad de replicación de un virus no es siempre la misma; puede variar desde una única copia (la necesaria para su instalación permanente en el sistema) a una replicación en la que cada copia realiza copias de sí misma, con lo cual se reduce la capacidad de almacenamiento y satura al ordenador.

La replicación persigue dos fines: su propagación a otros sistemas y tener más probabilidades de éxito en la consecución de algún otro objetivo específico.

La replicación se puede enumerar de la siguiente manera:

i. Búsqueda de un huésped dónde instalarse.
ii. Comprobación de la existencia de la copia realizada, para evitar contaminar varias veces el mismo programa.
iii. Recomposición de las partes del programa esparcidas por el disco, si es que el código se dividió para ocultarse.
iv. Copia en el programa huésped para garantizar su ejecución en otro sistema.

Ocultamiento

El virus evitará ser descubierto, al menos de manera inmediata, pues cuanto más tiempo permanezca oculto más daño causará al ordenador y la infección se extenderá mucho más. Muchos virus llegan a permanecer ocultos por un período bastante largo (horas, días o meses) y en el momento en que se activan logran esparcirse por miles de computadoras en un período relativamente corto (minutos u horas). Esto puede conseguirse mediante técnicas de ocultamiento complicadas y muy difundidas, como la técnica *stealth*, muy utilizada por los virus sigilosos.

Las técnicas de ocultamiento pueden ser muy variadas dependiendo del conocimiento y de la inventiva del programador del virus. Algunas técnicas de ocultamiento son:

- ✓ Ocultar el código viral a los métodos típicos de observación (editores o desensambladores).

- ✓ Ocultar el accionar del virus, como en el caso de los virus lentos, al reducir al mínimo sus actividades.
- ✓ El cifrado del cuerpo del virus cada vez que se replica, como lo hacen los virus polimórficos.
- ✓ Fragmentar su código, colocando la parte inicial de en el programa huésped y el resto en otro lugar del disco.
- ✓ También pueden ejecutarse antes que el código huésped, tomar el control del sistema a intervalos de reloj o simplemente marcar el archivo infectado con la propiedad de archivo oculto.

Toma del Control del Sistema

Para realizar sus actividades, el virus debe tomar el control del ordenador, de preferencia de manera momentánea. Dependiendo de cada virus, la forma en que ejerce el control puede cambiar; lo puede tomar directamente antes de la ejecución de un programa de usuario o a intervalos cortos por medio de las interrupciones.

Si el código del virus es ejecutado antes que el programa huésped, es conveniente para él comenzar por el programa de uso más frecuente; por ejemplo el COMMAND.COM. Así se asegura que se ejecutará un número elevado de veces, e infectará a otros archivos ejecutables (véase figura 2.4).

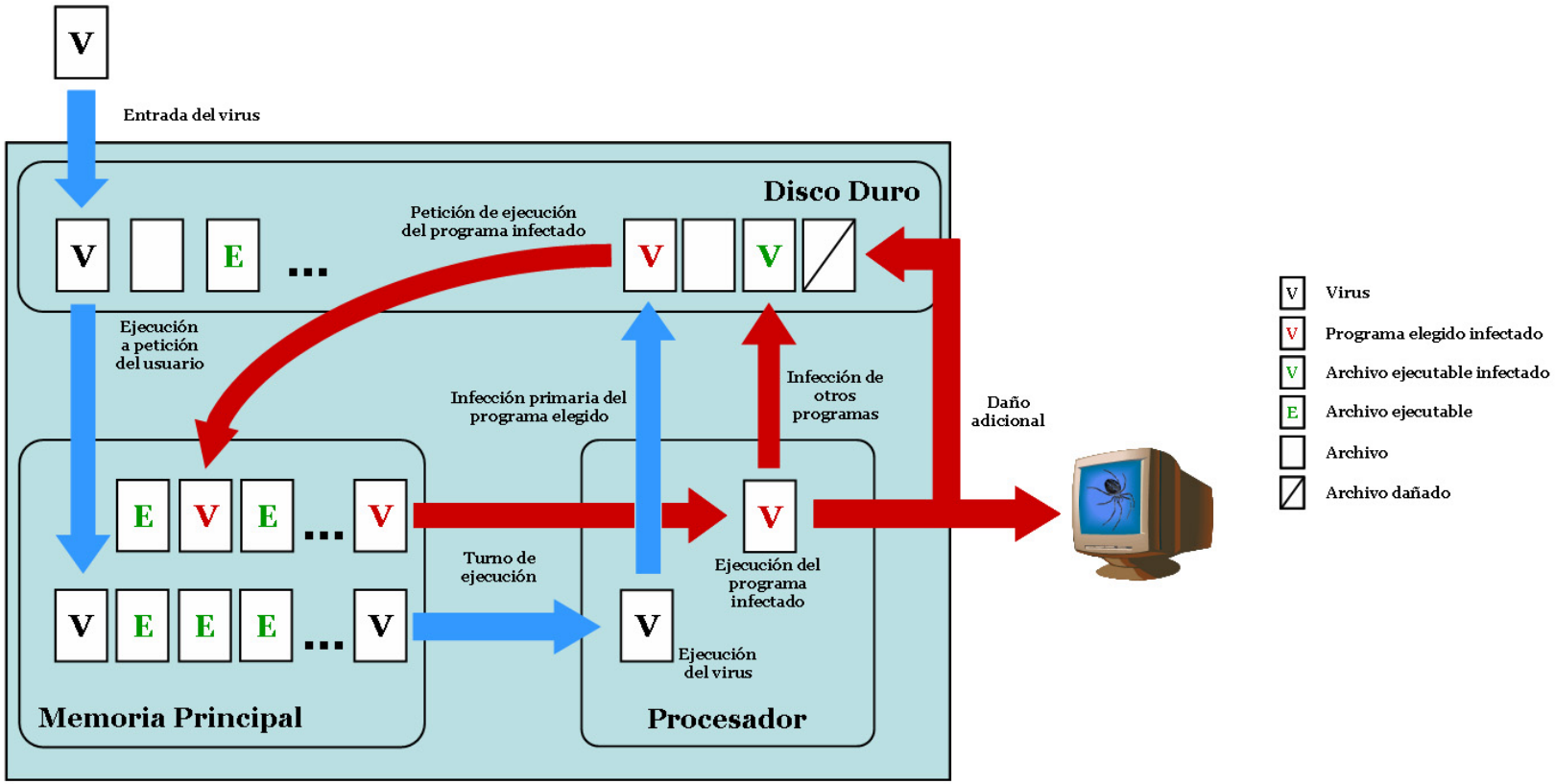


Figura 2.4. Toma del control por medio de un programa elegido.

El virus puede instalarse en memoria principal y quedar residente, tomando completo control del sistema operativo y así infectar todo archivo que pase por ella.

Se dice que un virus es residente en memoria cuando termina su ejecución pero permanece cargado en la memoria. Esto es posible debido a interrupciones (27h y 21h, función 31h) que permiten la terminación de un programa y evitan que sea borrado al cargar algún otro programa.

Cuando el virus está residente en memoria necesita ser llamado en algún momento durante la ejecución de otro programa. Lo que se hace es aprovechar la interrupción de reloj (o alguna otra que el programador del virus desee utilizar), que se ejecuta a intervalos regulares. La interrupción de reloj es utilizada por el ordenador para actualizar la hora del sistema. Dicha interrupción se activa varias veces por segundo para procesar las instrucciones que componen la rutina de atención.

En la parte baja de la memoria se encuentra una tabla que indica la dirección de memoria donde inicia cada rutina de interrupción. Esta tabla es leída por el procesador cada vez que se produce una interrupción. El virus modifica dicha tabla sustituyendo la dirección de comienzo de la rutina de reloj, por la dirección de su propio código. De esta forma, cuando el procesador lee la tabla de interrupciones, extrae la dirección necesitada y pasa el control al virus. Así, éste se asegura una constante actividad, que aprovechará para contaminar otros archivos del sistema. Para evitar ser descubierto, da un salto a la rutina de interrupción auténtica para que ésta efectúe las tareas predeterminadas (véase figura 2.5).

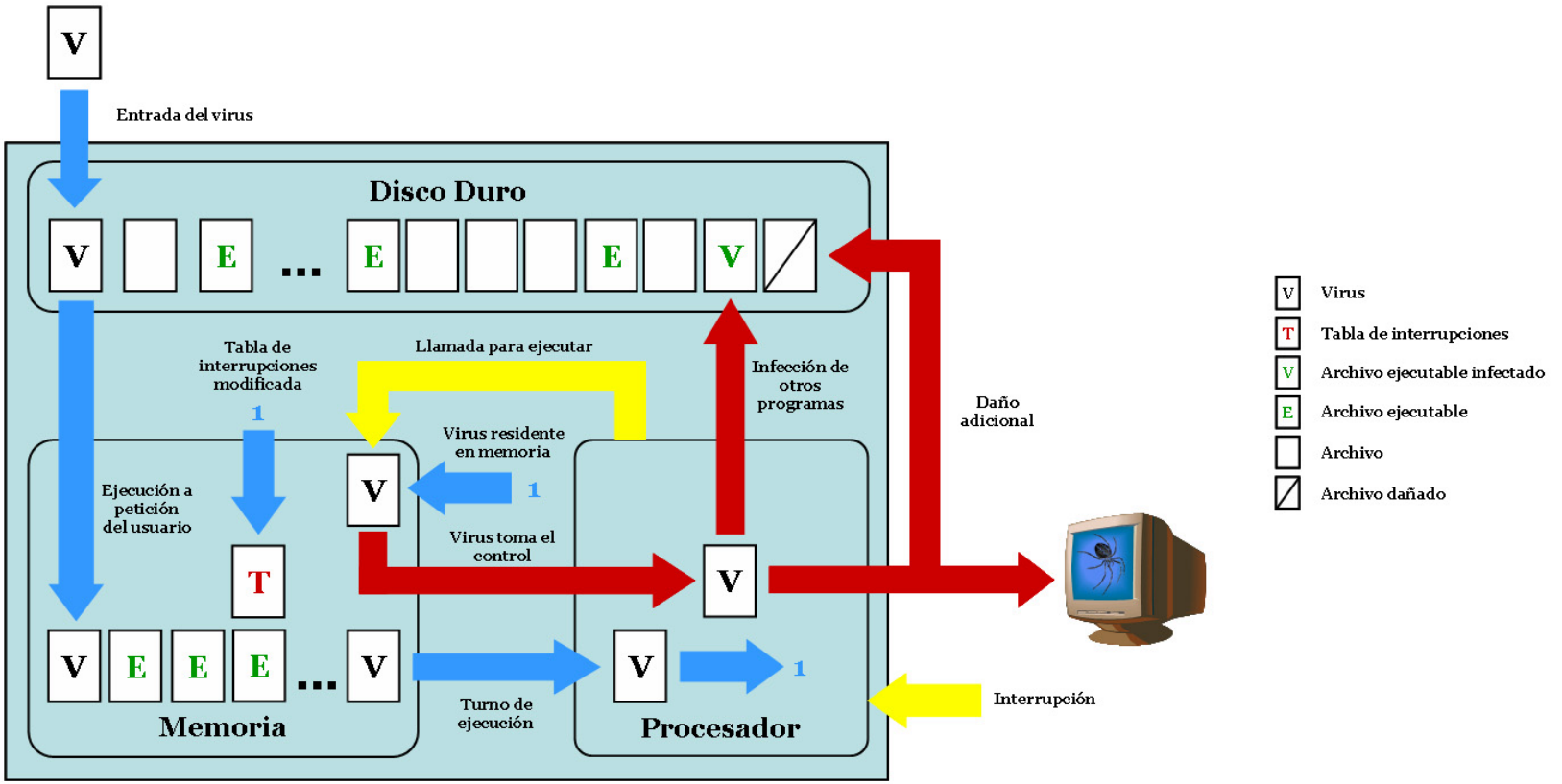


Figura 2.5. Toma del control por medio de una interrupción.

Acciones Dañinas

Ésta es la última fase del programa viral. En ella manifiesta su presencia dentro del ordenador, unas veces abiertamente y otras secretamente para provocar el mayor daño posible. La clasificación de la forma en que se manifiesta tiene dos vertientes.

En el primer grupo se encuentran los virus que hacen aparecer de forma periódica mensajes o dibujos en la pantalla o solicitan la ejecución de determinada acción como pulsar una tecla, escribir un texto, etc. Su principal objetivo es darse a conocer por medio de estas molestas acciones que desconciertan al usuario. Al segundo grupo pertenecen los virus que se ocultan lo más posible para causar daño considerable a la información dentro del ordenador. Las acciones habituales pueden ser el formateo del disco (destruyendo toda la información contenida en él), el borrado de un directorio o de algún archivo específico, la utilización exagerada de algún dispositivo de hardware, etc. El objetivo de estos virus es destruir información, mediante cualquier método. En la tabla 2.1 se muestra una clasificación de los niveles de daño que pueden ocasionar los virus.

Tipo de daño	*Descripción*
Trivial	En este tipo de daño la integridad de la información no corre ningún riesgo. La mayoría de las veces sólo aparecen letreros, imágenes o descomposición de la pantalla.
Menor	En este caso el daño generalmente se centra en la contaminación de archivos ejecutables.
Moderado	Parte de la información se pierde por el ataque viral. Este ataque se produce cuando el virus borra o modifica algún tipo de archivo específico o hace decrecer la capacidad de procesamiento del ordenador.
Mayor	La información es destruida completamente; se puede reducir la vida útil de los dispositivos de hardware. Hasta este nivel el usuario puede darse cuenta de las actividades del virus.
Severo	En este caso el virus actúa lenta, progresiva y sigilosamente, destruyendo toda la información disponible en el ordenador. Se crean vulnerabilidades en la seguridad del sistema. El usuario no puede detectar el daño inmediatamente.

Tabla 2.1. Niveles de daños ocasionados por virus.

Sintomatología de una Infección

Realmente no se puede determinar a ciencia cierta qué síntomas muestra el ordenador cuando está infectado, ya que, como se ha visto, los virus son muy variados y sus formas de comportamiento también; además de que los virus “bien programados” contienen un módulo de ocultamiento bastante sofisticado que no permitirá que se reconozca al virus tan fácilmente. Sin embargo, algunos indicios que delatarían la presencia de un virus son los siguientes:

- ✓ Los comandos o acciones que se hace ejecutar por la computadora aparentan ser más lentos.
- ✓ Los programas comienzan a ocupar más espacio de lo habitual.
- ✓ Aparecen o desaparecen archivos.
- ✓ Cambia el tamaño de un programa o un objeto.
- ✓ Aparecen mensajes u objetos extraños en la pantalla.
- ✓ El disco trabaja más de lo necesario.
- ✓ Los objetos que se encuentran en la pantalla aparecen ligeramente distorsionados.
- ✓ La cantidad de espacio libre del disco disminuye sin ningún tipo de explicación.
- ✓ Se modifican sin razón aparente el nombre de los ficheros.
- ✓ No se puede acceder al disco duro.
- ✓ Programas o procesos en memoria que son desconocidos.

Existen otras manifestaciones que se confunden con síntomas cuando en realidad no lo son. Algunas de las acciones pueden ser provocadas por conflictos en la configuración o instalación de las aplicaciones, del sistema o de los dispositivos de hardware. En muchas ocasiones se debe a la mala programación del sistema operativo instalado.

Las acciones que realmente delatan la presencia de un virus son gráficos poco comunes que aparecen en la pantalla, mensajes nunca antes vistos, letras que se caen o rebotan en el fondo de la pantalla, etc. Estos virus fueron programados para ese tipo de acciones y no son consecuencias secundarias en el sistema debido a su presencia.

II.3.4. Métodos de Infección y Ocultamiento

La forma en que un virus infecta y se oculta dentro de un sistema es bastante variada. Los programadores de virus pueden desarrollar alguna nueva técnica con base en los conocimientos que tengan sobre el funcionamiento del sistema operativo o de los antivirus; es decir, actualizan sus creaciones a la par con las actualizaciones de los sistemas informáticos. A continuación se describen algunos métodos conocidos que un virus utiliza para infectar archivos.

Métodos de Infección

Añadidura o empalme. Un virus usa esta técnica cuando su código se agrega al final de los archivos ejecutables, modificando las estructuras del archivo anfitrión de manera que el control del programa pase primero al virus cuando se quiera ejecutar el programa. Este cambio de secuencia permite que el código vírico añadido se active, realice sus tareas específicas y luego devuelva el control al programa anfitrión para que éste se ejecute normalmente. La principal desventaja de este método es que el tamaño del archivo infectado es mayor al original, lo que permite su rápida detección.

Inserción. El virus que utiliza este método copia su código directamente dentro de archivos ejecutables, en vez de añadirse al final de los archivos anfitriones. Busca alojarse directamente en zonas de código no utilizadas o en segmentos de datos dentro de los archivos ejecutables que contagian; de esta manera la longitud total del archivo infectado no varía. Este método exige mayores técnicas de programación para poder detectar las zonas posibles de contagio dentro de un archivo ejecutable, por lo que generalmente no es muy utilizada por los programadores de virus informáticos.

Reorientación. Este método es una variante de la inserción. Bajo este esquema se copia el código vírico en zonas del disco duro que se marcan como defectuosas o en archivos ocultos del sistema. Estos virus, al ejecutarse, implantan pequeños trozos de código en los archivos ejecutables que infectan, que luego actúan como puente hacia las zonas “dañadas” donde se encuentra el resto del código viral. Por este método, el virus tiene la ventaja de que su cuerpo, al no estar inserto en el archivo infectado sino en otro sitio oculto, puede tener un tamaño bastante grande, aumentando así su complejidad. Sin embargo, al borrar archivos ocultos sospechosos o reescribir las zonas del disco marcadas como defectuosas se puede eliminar al virus fácilmente. Esta técnica es utilizada por virus que contaminan archivos con extensión .COM.

Sustitución. El método de sustitución consiste en sustituir el código completo del archivo original por el código del virus. Al ejecutar el programa infectado el único que actúa es el virus, que cumple con su tarea de contagiar otros archivos y luego termina la ejecución del programa.

Ésta permite que en cada infección se eliminen archivos de programas válidos, los cuales son reemplazados por nuevas copias del virus. Los virus del sector de arranque utilizan esta técnica, ya que sustituyen el programa de carga del sistema operativo.

Métodos de Ocultamiento

Stealth. La técnica *stealth* se describió anteriormente en el tema de virus sigilosos.

Tunnneling. La técnica de *tunneling* (o del túnel) es utilizada por los virus para evadir la vigilancia de los programas antivirus que monitorean todas las actividades que acontecen dentro del ordenador.

Cuando las actividades de un ordenador son vigiladas por un antivirus, cada petición de interrupción es interceptada por el programa de vigilancia, analizada y, en caso de no ser sospechosa, invocada por el antivirus y devuelta hacia la fuente original de la petición. Para evitar ser detectado, el virus busca las direcciones originales de las interrupciones del DOS y del BIOS (que se encuentran en la tabla de vectores de interrupción, en la memoria primaria). Cuando obtiene dichas direcciones invoca directamente las rutinas de servicio de interrupción, saltándose cualquier programa antivirus que pueda estar vigilando las actividades del sistema.

Los virus tienen diferentes formas de replicarse, ocultarse y atacar. Replicarse es el principal objetivo de un virus y, para ello, utilizará alguna de las técnicas de ocultación, replicación e infección mencionadas para lograrlo. Las medidas de defensa desarrolladas no son completamente eficaces contra estos pequeños fragmentos de código; debido, en parte, a la diversidad de tipos de virus y de técnicas que utilizan para replicarse, o de objetivos de ataque.

Sin embargo, la principal causa de fallas en las defensas antivirus, es la inexperiencia de los usuarios. Para que un virus se replique y destruya información debe ser activado por éstos, de lo contrario no podrá realizar ningún daño al sistema informático; conocer al enemigo es el primer paso para evitar que haga daño. Por ello, es importante saber cómo funcionan, cómo causan daño y cómo se pueden contrarrestar sus efectos. Así pues, la principal forma de protección en contra de los virus informáticos (y de los códigos maliciosos en general) es conocerlos; sólo así se tendrá la certeza de que la información se mantendrá lo más segura posible.

CAPÍTULO III

SISTEMAS OPERATIVOS Y VIRUS

III.1. Introducción a los Sistemas Operativos

En los capítulos anteriores se ha mencionado la forma en que un sistema informático está compuesto, su importancia como herramienta para procesar la información y las amenazas que atentan contra su seguridad. También se ha dado una descripción del funcionamiento, clasificación y peligrosidad de una de estas amenazas: los virus.

Sin embargo, para el correcto funcionamiento del ordenador, y también del virus, es necesario un elemento fundamental: el **sistema operativo**.

El sistema operativo proporciona la administración de recursos e información, además de servir como enlace entre el usuario y el ordenador. Para el virus, el sistema operativo representa una herramienta para manipular al ordenador y una amenaza que le impida realizar sus actividades maliciosas.

III.1.1. ¿Qué es un Sistema Operativo?

Para operar, las computadoras digitales utilizan un sistema de codificación de instrucciones en sistema de numeración binaria; ésto se debe a los requerimientos de hardware, ya que éste opera con base en la presencia o ausencia de voltajes.

Al inicio de la era informática no existían los sistemas operativos; por lo que las computadoras eran herramientas muy complicadas de usar y se requerían conocimientos técnicos muy elevados para manejarlas.

Esto motivó a crear un medio para que el usuario manejara el ordenador con un entorno, lenguaje y operación bien definidos para hacer un uso óptimo de éste. Así surge el sistema operativo.

El sistema operativo es el conjunto de programas que se encarga de coordinar, controlar y administrar la ejecución de programas de aplicación y de dispositivos de hardware; además,

brinda al usuario una forma amigable y sencilla de operar y emitir las indicaciones al ordenador para que éste realice los procesos necesarios para completar una tarea informática determinada.

III.1.2. Funciones de un Sistema Operativo

Un sistema informático puede organizarse de manera jerárquica (figura 3.1). Esta organización tiene como objetivo el proporcionar las aplicaciones necesarias para que un usuario trabaje su información por medio del ordenador.

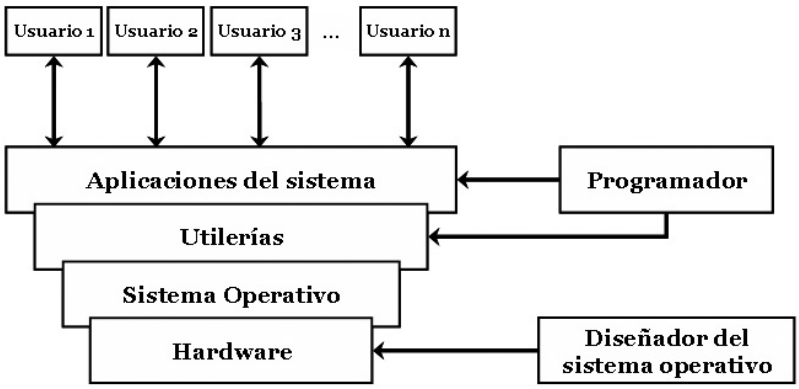


Figura 3.1. Capas de un sistema informático.

El usuario utiliza un ordenador por medio de programas responsables del control del hardware; para ésto se cuenta con utilerías, las cuales implementan funciones y que ayudan en la creación del programa, la administración de archivos y el control de dispositivos de entrada y salida. De todo el software, el más importante de estos programa es el sistema operativo.

El sistema operativo realiza las siguientes funciones:

- ✓ *Creación de programa*. Proporciona facilidades y servicios, como editores y depuradores, para asistir al usuario en la creación de programas.
- ✓ *Ejecución de programa*. Maneja las tareas destinadas para la ejecución de programas: carga de datos e instrucciones en memoria principal, inicializa los dispositivos de hardware y otros recursos que necesite el programa en ejecución.

- ✓ *Administración de dispositivos de hardware*. Se encarga de manejar las instrucciones y señales que controlan a los dispositivos de hardware para su correcta operación. Una de sus tareas más importantes es la gestión del espacio en memoria para los múltiples procesos que se ejecuten.
- ✓ *Acceso controlado a archivos*. Maneja información que incluye una comprensión de la naturaleza del dispositivo donde se encuentra el archivo, así como del formato de archivo en el medio de almacenamiento.
- ✓ *Acceso al sistema*. Controla el acceso a todo el sistema y a recursos específicos del mismo; proporciona protección a los recursos y datos de usuarios que no estén autorizados. Además, resuelve los conflictos ocasionados en el caso de disputas por recursos.
- ✓ *Detección de errores*. Durante el funcionamiento de un ordenador pueden ocurrir errores; el sistema operativo se encarga de detectarlos y emitir una respuesta que corrija la condición de error, reduciendo el impacto que se pueda causar a los procesos en ejecución.
- ✓ *Contabilidad*. Recopila estadísticas del uso de diferentes recursos y vigila todos y cada uno de los parámetros de ejecución. Esta función se utiliza para anticipar la necesidad de mejoras futuras y para ajustes en la mejora del rendimiento del sistema.

III.1.3. Categorías de Sistemas Operativos

Todo tipo de software tiene su clasificación: las aplicaciones se dividen en hojas de cálculo, procesadores de texto, etc.; incluso los virus, como se ha visto, tienen su propia clasificación. La clasificación de los sistemas operativos obedece, como todo programa, a la tarea para la que se diseña y las limitaciones que impone el hardware en donde se instalará. Con base en esto, una posible clasificación es la que se expone inmediatamente.

Sistema operativo multitarea. El modo de funcionamiento se basa en que un ordenador procesa varias tareas al mismo tiempo. Existen tres tipos de estos sistemas:

- ✓ *Conmutación de contextos (context switching)*. Dos o más aplicaciones están cargadas al mismo tiempo, pero sólo se esta procesando la aplicación que se encuentra en primer plano (la que ve el usuario).
- ✓ *Multitarea cooperativa*. Las tareas en segundo plano reciben tiempo de procesado durante los tiempos muertos de la tarea que se encuentra en primer plano, siempre que esta aplicación lo permita.
- ✓ *Multitarea de tiempo compartido*. Cada tarea recibe la atención del microprocesador durante cierto tiempo. Para mantener el sistema en orden, cada tarea recibe un nivel de prioridad o se procesa en forma secuencial.

Sistema operativo monotarea. Estos sistemas pueden manejar uno y sólo un proceso en cada momento de ejecución. Si se está realizando una tarea, no se puede iniciar un nuevo proceso hasta que termine el anterior.

Sistema operativo multiusuario. En esta categoría se encuentran los sistemas que cumplen simultáneamente las necesidades de dos o más usuarios que comparten los mismos recursos. La manera de operar es por medio de la división del tiempo de procesamiento. Este tipo de sistemas se emplean especialmente en las redes.

Sistema operativo monousuario. Son aquéllos que sólo pueden atender a un solo usuario a la vez, debido a las limitaciones del hardware, los programas o la aplicación que se esté ejecutando.

Sistema operativo de secuencia por lotes. La forma de trabajar de estos sistemas es por medio de la ejecución de listas de trabajos (llamadas lotes) que son procesados de manera secuencial bajo el control de una pieza de software llamada monitor. El usuario no tiene acceso directo al ordenador.

Sistema operativo en tiempo real. Este tipo de sistemas procesa las instrucciones recibidas al instante, y una vez procesadas muestra los resultados. Existe un solo operador y no es necesario compartir el procesador entre varias solicitudes.

Sistema operativo en tiempo compartido. Consiste en ejecutar programas separados de forma concurrente, intercambiando porciones de tiempo asignadas a cada programa. Está pensado para el uso de un sistema por más de un usuario al mismo tiempo.

III.1.4. Seguridad en los Sistemas Operativos

Como se ha explicado en el capítulo I, la seguridad es uno de los aspectos más importantes que deben considerarse en el momento de diseñar y desarrollar sistemas informáticos. El sistema operativo, al ser parte esencial en el funcionamiento del ordenador, se ha convertido en materia de gran preocupación en el medio informático. Todos los sistemas operativos presentan vulnerabilidades; por esta razón, esta gran pieza de software constituye la base para la implementación de medidas de seguridad. Posteriormente, se robustece la seguridad con diversas herramientas que apoyan a la seguridad que ofrece el sistema operativo.

Gran parte de la seguridad y protección de los sistemas operativos se agrupa en tres categorías:

- ✓ *Control de acceso.* Regula el acceso de los usuarios al sistema, subsistemas y datos, así como al acceso a procesos de varios recursos y objetos dentro del sistema.
- ✓ *Control de flujo de información*. Regula el flujo de datos dentro, hacia y desde el sistema y su entrega a los diversos usuarios.
- ✓ *Certificación*. Se relaciona con probar que los mecanismos de acceso y control de flujo se desempeñen de acuerdo con sus especificaciones y que hagan cumplir la protección y las políticas de seguridad deseadas.

Para cada sistema operativo las técnicas y niveles de seguridad son diferentes, dependiendo del sistema operativo utilizado, la arquitectura del ordenador, las necesidades del usuario, el número de usuarios que acceden al sistema, etc.

Las tres categorías de seguridad antes mencionadas están implícitas en las características generales de cualquier sistema operativo. Al contar con funciones de acceso controlado a archivos y al sistema, detección de errores, asistencia en ejecución de programas y contabilidad, el sistema operativo cumple con políticas que llevan a resguardar la integridad, confidencialidad y disponibilidad de los recursos y la información dentro del ordenador.

Otro punto importante es que la seguridad de un sistema operativo se considera en la categoría a la que pertenece; es decir, que la seguridad será diferente para cada sistema operativo. Por ejemplo, si un sistema operativo es multiusuario, éste deberá contemplar políticas de seguridad más estrictas que un sistema monousuario, debido a que muchas más entidades pueden acceder a la información contenida en el ordenador.

Seguridad Contra Virus

Como se ha estudiado, el virus informático no es otra cosa que un programa que intentará realizar una tarea específica para llevar a cabo algún daño. Al ser un programa, podrá solicitar todos los servicios que pueda requerir (lectura y escritura de archivos, manipulación de dispositivos, residencia en memoria, etc.).

Desde un punto de vista personal, para que un sistema operativo pueda defenderse de este tipo de códigos maliciosos se debe hacer hincapié en los siguientes aspectos:

- ✓ *Gestión de la memoria*. Un virus puede utilizar la memoria principal para quedar residente, alterar los datos y programas que estén cargados, alterar zonas exclusivas del sistema o consumir un espacio excesivo. Por esta razón, un sistema operativo debe contar con una administración de memoria capaz de prohibir la alteración indiscriminada

de datos e instrucciones, la ejecución indistinta de procesos, reservar una zona específica para escritura y protección del área a la cual sólo el sistema tiene acceso.

- ✓ *Escritura de archivos*. Para proliferar en el ambiente computacional, un virus deberá replicarse; lo hará al alterar los programas ejecutables contenidos en el ordenador. Esto implica que se debe proteger a los archivos ejecutables, de manera que no se permita una manipulación fácil y completa por parte del programa. El sistema operativo debe encargarse de esta tarea, ya que es el administrador de los archivos que se encuentran dentro del ordenador; o a la espera de entrar o salir (por medio de disquetes, CD, DVD, red, etc.) de él.
- ✓ *Monitoreo de procesos*. Una característica importante del sistema operativo es que debe administrar los procesos que son ejecutados para evitar conflictos entre procesos y conflictos de hardware. Para llevar a cabo su cometido, el virus deberá moverse de manera discreta, evitando que sea descubierto. Esto hace que la vigilancia del sistema operativo sobre cada proceso sea trascendental, para evitar que un se realicen acciones inadecuadas (intercepción de interrupciones, modificación de archivos, modificación de datos en memoria, etc.) que puedan corromper la integridad de la información o la estabilidad del sistema.
- ✓ *Protección frente al usuario*. El sistema operativo debe defenderse contra las acciones que pueda cometer un usuario, accidentales o no. Esto quiere decir, que debe denegar a ciertos usuarios la capacidad de configurar y manipular los archivos del sistema. Un virus actuará conforme a los privilegios que tenga el usuario que lo ejecutó. Si el usuario no posee la libertad necesaria para que sus procesos modifiquen el funcionamiento del sistema, el virus simplemente no podrá realizar sus acciones.

Las características antes mencionadas forman una base para evitar que un código viral realice sus tres tareas fundamentales: replicarse, ocultarse y dañar información. Obviamente, contar con estas características dentro del sistema operativo no lo hace invulnerable contra los códigos virales; en ocasiones se necesita una serie de herramientas que complementen la seguridad ya establecida.

III.2. Virus en los Sistemas Operativos más Populares

Los tres sistemas operativos más habituales son Windows, Mac OS y Linux (ver figura 3.2). Windows funciona en los ordenadores con procesadores de Intel y AMD, Mac OS lo hace únicamente en los ordenadores de Apple y Linux funciona en ambas plataformas, aunque está mejor preparado para la primera. Al momento de elegir el sistema operativo hay que tomar muy en cuenta el tipo de ordenador que se tiene o que se adquirirá, el propósito al que se le destinará y las características de seguridad que se requiere.

Figura 3.2. Los sistemas operativos más populares.

Estos aspectos son tomados en cuenta no sólo por usuarios hogareños, empresariales o desarrolladores de software, también para los programadores de códigos maliciosos. Las características de cada sistema operativo son esenciales para encontrar puntos débiles en su seguridad y explotarlos.

En el caso de los virus, existen sistemas operativos más vulnerables que otros. Al inicio de la epidemia, los objetivos favoritos eran los sistemas MS-DOS, Macintosh, Atari, Amiga y las primeras versiones de Windows. Otros sistemas como UNIX y OS/2 eran un blanco de pocos virus – aunque los pocos códigos maliciosos existentes eran muy peligrosos, como el gusano para UNIX de 1988 –. La principal razón de esta disparidad es la importancia que los desarrolladores de sistemas operativos le dan a la seguridad.

El panorama actual no es muy diferente. Mac OS X y todas las versiones de Linux son sistemas que presentan un ambiente muy hostil para la proliferación de virus, debido a que forman parte de la familia de sistemas operativos UNIX; en cambio, Windows sigue siendo un fértil campo de cultivo para los códigos virales. Nuevamente, las políticas de seguridad contempladas en el diseño de cada sistema operativo son el factor principal para la proliferación de los virus.

III.2.1. Microsoft Windows XP

Windows XP es la siguiente versión de Microsoft Windows después de Windows 2000 y Windows Millennium. Es la fusión de las dos ramas de sistemas operativos Windows (una era dirigida a los ordenadores domésticos y la otra para el mercado empresarial), ya que integra a las características de Windows 2000 – seguridad, administración y confiabilidad en redes – los mejores elementos de Windows Me – tecnología *Plug and Play*, recursos multimedia e interfase de usuario – e innovadores servicios de soporte técnico.

Windows XP fue diseñado con base en el código de Windows 2000; tiene dos versiones: una orientada a usuarios hogareños (Windows XP Home Edition) y la otra al ámbito empresarial (Windows XP Professional).

Windows XP posee las siguientes características técnicas:

- ✓ *Multitarea*. Dos o mas procesos pueden ejecutar y ser administrados de forma simultánea. Windows controla dichos procesos por medio del Administrador de Tareas.
- ✓ *Entorno*. Windows XP emplea un entorno gráfico llamado Luna, el cual se asemeja mucho al Aqua de Mac OS X. El clásico sistema de ventanas e íconos permite una navegación rápida por el escritorio y las aplicaciones.
- ✓ *Facilidad de uso*. Windows es un sistema que permite una sencilla configuración y personalización sin necesidad de conocimientos avanzados de computación.
- ✓ *Redes*. Se puede tener acceso a archivos compartidos, navegación por la red y acceso a equipos conectados a una intranet. Permite además soporte para servidores.

Características Generales de Seguridad

Windows XP proporciona un conjunto de componentes del sistema y de herramientas que ayudan a identificar cualquier acceso a recursos protegidos. A continuación se presentan las características que forman parte de su infraestructura de seguridad.

Cuentas de usuario. La cuenta de usuario proporciona una identidad única a cada persona que tiene acceso al ordenador. Existen varios tipos de cuentas de usuario:

- ✓ *Administrador*. Dispone de todos los derechos posibles sobre todo el ordenador; tiene acceso y control sobre todos los programas y archivos del sistema, administración de cuentas de usuario, instalación y ejecución de programas, concesión de privilegios, administración de dispositivos de hardware. Puede existir más de un administrador por equipo.
- ✓ *Limitada*. Estos usuarios puede ejecutar programas, guardar documentos, configurar el entorno de trabajo y lectura de documentos compartidos.
- ✓ *Invitado*. Es una cuenta para usuarios ocasionales con privilegios predeterminados limitados. Bajo este usuario sólo se puede ejecutar programas y guardar documentos.
- ✓ *Desconocido*. Es una cuenta que no pertenece a ninguno de los tipos anteriores y que aparece cuando se actualiza a Windows XP desde una versión anterior de Windows (son los usuarios existentes de la versión anterior).

Cuando varias cuentas de usuario son de un mismo tipo, se dice que forman un grupo de seguridad. Estos grupos se dividen en:

- ✓ *Administradores*. Las cuentas de tipo Administrador pertenecen a este grupo.
- ✓ *Usuarios*. Este grupo engloba a las cuentas de tipo Limitado.
- ✓ *Invitados*. A este grupo corresponde todas las cuentas de tipo Invitado.

Control de acceso. Después del inicio de sesión exitoso de un usuario, Windows utiliza la información de seguridad asociada a la cuenta (tipo de cuenta y privilegios) para determinar que

recursos están disponibles para el usuario. El control de acceso establece permisos para manipulación de archivos, carpetas, dispositivos y recursos compartidos de red.

Cifrado y autenticación. Windows XP permite el cifrado de archivos y claves por medio del Sistema de Cifrado de Archivos (*Encrypting File System*, EFS) que utiliza los algoritmos *DES* y *RSA*. La autenticación de una persona, organización, equipo o servicio es proporcionada por las herramientas de Certificados y la Consola de Administración de Microsoft, las cuales gestionan *certificados digitales*.

Control de acceso a red. Windows XP impide los accesos no permitidos por medio de un *firewall* incluido en el sistema. En caso de que se logre una intrusión, asigna privilegios de Invitado a cualquier entidad que intente acceder al sistema desde una red. Por medio de Kerberos 5, Windows brinda un servicio de autenticación más eficiente entre servidores y clientes, incluso en las más grandes y complejas redes.

Políticas restrictivas de software. Este servicio provee mecanismos para identificar programas en ejecución. Por medio de esta característica, un administrador puede prevenir la ejecución de aplicaciones no solicitadas; los programas pueden o no ejecutar dependiendo de la zona de Internet de donde procede, la ruta de acceso del programa, el certificado digital que firma el programa y la huella que identifica a un programa o archivo (Hash).

Seguridad Contra Virus en Windows XP

Desafortunadamente, la familia de sistema operativos Windows ha sido, a lo largo de 20 años, el blanco predilecto de los virus informáticos. Muchas veces se piensa que se debe a su gran difusión, su popularidad y a que este sistema sirve como introducción a la computación. Esto es falso, la verdadera razón es la falta de atención que Microsoft ha puesto en los códigos maliciosos y en la seguridad en general.

Debido a esta desventaja que presenta Windows, existen varios virus que pueden corromper la integridad y disponibilidad de los recursos del sistema. Para remediar este problema, Windows

XP recibe ayuda de diversos programas antivirus como lo son los antivirus de Norton, McAfee y Panda entre otros desarrolladores.

Las formas en que un virus ataca a un sistema Windows, son muy variadas (como se vio en el capítulo II): pueden infectar archivos ejecutables, destruir documentos, consumir los recursos del sistema o contaminar sectores de disco. Cuando un virus infecta un sistema Windows, puede modificar los valores del registro, reemplazar archivos de sistema e invadir programas de correo electrónico.

El principal escudo que Microsoft desarrolló para evitar que algún procedimiento extraño ejecutara en Windows XP fueron las políticas de restricción de software. La utilización de este tipo de políticas se enfoca en la necesidad de regular el software desconocido o poco confiable. Con el aumento en la utilización de redes de computadores como Internet y el correo electrónico como herramientas de negocios e investigación, el usuario se encuentra expuesto a la instalación de software nuevo de varias formas.

El usuario constantemente debe tomar decisiones de ejecutar o no instalaciones de software desconocido (con frecuencia virus o troyanos); por lo que resulta difícil tomar una decisión acertada. Mediante el uso de políticas se pueden proteger los equipos de cómputo de software poco confiable y especificar que programa está autorizado para ser ejecutado en una computadora. Se pueden definir niveles de seguridad por defecto, que van desde no restringido hasta deshabilitado, para grupos de usuarios, de tal forma que el software será permitido o no permitido con base en estos privilegios; también se pueden hacer excepciones para el nivel de seguridad mediante la creación de políticas para software especifico. Con las políticas de restricción de software se puede:

- ✓ Controlar el software que se ejecuta en el sistema.
- ✓ Permitir al usuario correr aplicaciones específicas en el ordenador.
- ✓ Decidir los permisos del usuario para los equipos de cómputo.
- ✓ Especificar las políticas para cada usuario.

Existen dos formas de usar las políticas restrictivas de software: si los administradores han identificado una lista de programas permitidos para ejecutar, puede utilizar las políticas para permitir la ejecución de esa lista definida de aplicaciones confiables; por el contrario, si los administradores no conocen todas las aplicaciones que utilizan los demás usuarios, deberán restringir la ejecución de aplicaciones sospechosas.

Las políticas de restricción de software se aplican en dos escenarios distintos:

- ✓ *Sólo se ejecuta código conocido*. Ningún código se ejecutará, a menos que sea identificado como software confiable; por ejemplo, un administrador puede crear una política donde se permita ejecutar aplicaciones de Microsoft Office y por el contrario, impida ejecutar programas descargados de Internet o desde algún disquete porque no se reconoce como software confiable.
- ✓ *Evitar ejecutar código no solicitado*. En algunos casos un administrador no puede predecir una lista entera de los programas que se necesita ejecutar. En este caso sólo se puede identificar el código indeseable encontrado y evitar que se utilice; por ejemplo, un administrador puede crear una regla que prevenga la instalación de algún programa que se sabe causará conflictos a otras aplicaciones.

Para identificar el software confiable, se utilizan las reglas siguientes:

- ✓ *Reglas Hash*. Permiten a un administrador buscar e identificar un archivo o programa por medio de su firma digital.
- ✓ *Reglas de ruta*. Identifican programas por medio de la ruta de ejecución, localizando el directorio o subdirectorio donde se encuentra el programa.
- ✓ *Reglas de certificado*. Permiten que se configure certificados de instalación de programas firmados por una entidad reconocida como confiable.
- ✓ *Reglas de zona*. Identifican programas que provengan de Internet, de una intranet local, de sitios confiables o de sitios restringidos.

Con esta medida, Microsoft aseguró que la existencia de virus llegaría a su fin; sin embargo, el sistema Windows XP sigue siendo vulnerable a virus y otros códigos maliciosos. A criterio personal, esto se debe a dos causas: la primera es que esta característica sólo se encuentra presente en Windows XP Professional Edition, y la segunda es debida a errores en el diseño y la programación del sistema operativo.

Con respecto a la primera causa, Windows XP Home Edition es la versión instalada en ordenadores caseros, los cuales son frecuentemente utilizados por personas con pocos conocimientos de computación. Al no contar con políticas restrictivas de software en esta versión, los usuarios hogareños quedan más expuestos al ataque de virus u otros códigos maliciosos; es decir, no existe una primera barrera que impida la ejecución de programas no solicitados.

La segunda causa es más peligrosa aún. Un código malicioso siempre atacará una vulnerabilidad del sistema para lograr su cometido. Windows XP no es más que un heredero de los sistemas operativos Windows anteriores (Win3.1, Win95, Win98, WinME, WinNT, Win2000), que a su vez tienen su origen en el MS-DOS, un sistema que nació cuando los primeros virus dañinos no eran considerados una amenaza. Lo anterior es una gran desventaja, ya que Windows XP no sólo incluye las mejores características de cada sistema anterior, sino también sus vulnerabilidades.

Una segunda medida de seguridad, la cual es muy popular, es el uso de antivirus. Un antivirus es un programa que rastreará, detectará e impedirá las actividades de los códigos maliciosos; además, trabajará conjuntamente con el sistema operativo para evitar que actividades ilícitas se lleven a cabo dentro del ordenador.

Actualmente las compañías desarrolladoras de antivirus, no sólo incluyen vacunas y detectores dentro de sus programas, sino también herramientas que permiten rastrear el flujo de información, bloquear intentos de intrusión al ordenador y casi cualquier actividad que comprometa la integridad y disponibilidad de los recursos del sistema.

En el capítulo IV se analizará y describirá el funcionamiento de las herramientas de detección de virus informáticos. La mayoría de estos programas antivirus están destinados a salvaguardar la seguridad en la familia de sistemas operativos Windows.

Virus en Windows XP

Existe una gran variedad de virus que pueden provocar alteraciones en el funcionamiento de los sistemas Windows XP; muchos de ellos causan daños moderados a los ordenadores. Los virus más destacados de Windows XP se describen en seguida.

Melissa. Es un virus de macro típico con una inusual forma de manifestación: cuando un usuario abre un documento infectado, el virus intentará mandar por correo electrónico una copia del documento utilizando MS Outlook. Puede infectar los documentos de MS Word 97 y MS Word 2000 al añadir un módulo de macro llamado Melissa; además, intentará infectar otros documentos del usuario, incluso si el ordenador no cuenta con MS Outlook para el envío del archivo.

Cuando un usuario abre o cierra un documento infectado, el virus revisa si ya se ha hecho el envío masivo por correo, al verificar la siguiente línea de registro: HKEY_CURRENT_USER\Software\Microsoft\Office\ y como valor “Melissa?”. Si la línea contiene el valor “...by Kwyjibo” en lugar de “Melissa?”, significa que el envío masivo ya se ha realizado desde la máquina infectada. El virus no intentará mandar un segundo envío si ya lo ha realizado anteriormente.

Para ocultarse, el virus desactivará el menú: Herramientas>Macro de MS Word 97. Al desactivar estos comandos, el virus previene que cualquier usuario verifique el listado de macros y encuentre señales de la infección. Además también desactivará el menú Macro>Security de MS Word 2000, para evitar que un usuario cambie le nivel de seguridad de macro en MS Word 2000.

Winux. Este es un virus que no es residente en memoria. Infecta archivos ejecutables tanto en Windows (archivos PE) como en Linux (archivos ELF). Está escrito en ensamblador y tiene un tamaño de 2.5 KB.

El virus está constituido en dos bloques de código: el primero ejecuta bajo un ambiente Windows y el segundo bajo un ambiente Linux. Para infectar busca todos los archivos ejecutables en el directorio donde se encuentra el archivo infectado y en un nivel hacia arriba. Ambos bloques de código pueden infectar tanto archivos ejecutables PE de Windows como ELF de Linux (distingue entre ambos tipos de ejecutables al revisar el formato del archivo).

Existen muchos más virus que atacan a Windows XP. Estos programas no tienen mucha atención de los medios informativos, ya que actualmente son los gusanos como Blaster, LovLetter o Kamasutra quienes ocasionan más ataques. Debido a que se ha hecho la diferencia entre virus y gusanos, no se dará explicación de cómo trabajan estos códigos maliciosos.

III.2.2. Linux Fedora Core

Linux es el nombre de un núcleo, pero se suele denominar con este nombre a un sistema operativo tipo UNIX de libre distribución; donde el código fuente está disponible públicamente y cualquier persona, con los conocimientos informáticos adecuados, puede libremente estudiarlo, usarlo, modificarlo y redistribuirlo. El núcleo fue creado por Linus Torvalds en la Universidad de Helsinki, Finlandia, basado en una pequeña implementación de UNIX para PC denominada Minix. Apareció por primera vez a finales de 1991. A partir de entonces, Linux ha sido rediseñado por diversas compañías, dando lugar a una gran gama de versiones como Red Hat Linux, Mandriva, Fedora Core, Turbo Linux, Linux Debian y otros.

Fedora Core (también conocido como Fedora Linux) es una distribución Linux desarrollada por la comunidad Fedora y promovida por la compañía estadounidense Red Hat. El objetivo del proyecto Fedora es conseguir un sistema operativo de propósito general, basado exclusivamente en software libre, con el apoyo de la comunidad Linux. Red Hat continúa participando en la construcción y desarrollo de este proyecto.

Originalmente, Red Hat Linux fue desarrollado exclusivamente dentro de Red Hat, pero contaba con la ayuda de informes de usuarios que detectaban fallos y hacían contribuciones a los paquetes de software incluidos, aunque no había contribuciones al desarrollo y distribución del sistema operativo como tal. Esto cambió en septiembre de 2003, cuando Red Hat Linux se derivó dando origen el Proyecto Fedora, que está orientado a la comunidad de usuarios y así mismo, dando una base para que el sistema Red Hat Enterprise Linux se desarrolle con más efectividad y adopte las nuevas características que se añaden en el Proyecto Fedora.

Las características generales de Fedora Core son aplicables a cualquier versión de este sistema operativo:

- ✓ *Multitarea*. Puede administrar dos o más procesos de forma simultánea. El control de las tareas depende de la versión del núcleo, la cantidad de memoria libre, velocidad del procesador y la capacidad y velocidad del disco duro.
- ✓ *Multiusuario*. Es capaz de responder a solicitudes de varios usuarios que emplean un mismo ordenador simultáneamente, pero que tienen necesidades diferentes.
- ✓ *Multiplataforma*. Fedora Core fue diseñado para ordenadores con procesadores Intel o AMD; sin embargo, Linux también es soportado por ordenadores con procesadores Amiga, Atari, Alpha y Power PC.
- ✓ *Sistema de archivos*. Tiene la capacidad de operar con diversos sistemas de archivos como FAT de DOS, VFAT de Windows 9x, OS2/FS o la ISO9660.
- ✓ *Red*. Linux ha sido desarrollado como un sistema operativo para trabajo en red. Su protocolo principal es TCP/IP.
- ✓ *Entorno*. Linux puede trabajar tanto en modo texto como en un entorno gráfico similar a Windows. Algunos de estos modos gráficos son FWVM, GNOME, KDE, CDE, Enlightenment o Nextlevel.
- ✓ *Compatibilidad con el estándar IEEE POSIX.1*. Gracias a esta compatibilidad, Linux soporta muchos estándares establecidos para la familia de sistemas UNIX.
- ✓ *Código fuente no propietario*. El kernel de Linux no utiliza código de fuente propietaria. Compañías comerciales, el proyecto GNU y programadores de todo el mundo han desarrollado software para Linux.

- ✓ *Soporte mediante software GNU*. Linux puede ejecutar una amplia variedad de software, desde desarrollo de aplicaciones hasta la administración del sistema, disponible gracias al proyecto GNU.

Características Generales de Seguridad

El sistema operativo Linux es considerado como uno de los sistemas más robustos y seguros que se han desarrollado. En su diseño se implementan políticas de seguridad restrictivas, que no sólo protegen al sistema de los usuarios que tienen acceso a él, sino también de intrusos que intenten acceder por medios ilícitos. Estas políticas se presentan por medio de características que se configuran al momento de instalar Linux.

Cuentas de usuario. Cada cuenta de usuario es una identidad independiente con privilegios de acceso independientes. Cada usuario recibe un directorio principal y un espacio en el disco duro, que es independiente de los archivos del sistema y de las áreas que ocupan los otros usuarios. Los usuarios se dividen en tres tipos:

- ✓ *Usuario root*. Tiene el control absoluto sobre la totalidad de componentes del ordenador; sólo el núcleo del sistema puede establecerle restricciones.
- ✓ *Usuario normal*. Puede conectarse al sistema y realizar tareas básicas como navegar por Internet, leer correo, crear documentos, etc., pero tiene restringido el acceso a archivos y directorios del sistema y no puede realizar ninguna tarea a nivel de sistema.
- ✓ *Usuario de sistema*. Es una cuenta que se utiliza para propósitos específicos del sistema que no pertenecen a una persona en particular.

Control de acceso discrecional. Linux puede controlar el grado de acceso que tiene cada usuario a los archivos y directorios; es decir, para cada archivo, se especifica quien puede leerlo, quien puede escribir en él o quien puede ejecutarlo.

Control de acceso a la red. Linux tiene la capacidad para permitir a determinados usuarios y servidores conectarse entre sí mediante el uso de reglas de acceso previamente establecidas.

Cifrado. Linux posee diversos mecanismos de cifrado que permiten proteger las contraseñas y los datos que circulan por la red. DES es el algoritmo de cifrado utilizado.

Registro, auditoría y control de red. La capacidad de registro es fundamental para la seguridad de los sistemas, ya que proporciona la única evidencia real de que se ha producido un ataque. Linux guarda registros a nivel de red, de servidor y de usuario para confirmar las actividades que se han llevado a cabo.

Detección de intrusiones. Examinando el tráfico en un segmento de red, Linux tiene la capacidad para detectar intentos de intrusión haciendo uso de reglas de comparación y emparejamiento de patrones. De esta manera se avisa al administrador del sistema ante cualquier ataque.

Seguridad Contra Virus en Linux

Una pregunta muy común entre usuarios inexpertos de Linux es: ¿existen virus que afecten a los sistemas Linux? La respuesta es muy sencilla: sí, aunque aún no representan un peligro tan grande como en sistemas Windows. La principal razón es la importancia que se le da a la seguridad en cada sistema operativo: mientras que Windows se destaca por mantener su poca seguridad con base en herramientas externas al sistema, Linux lo hace con base en políticas restrictivas implementadas en su diseño.

Linux es mucho más vulnerable a otro tipo de códigos maliciosos como gusanos, troyanos, *exploits* o *rootkits* que a virus. La razón por la cual Linux no se ve afectado por los virus radica en la gestión de memoria y la asignación de permisos, los cuales hacen casi imposible que un programa no autorizado se ejecute y propague. Sin embargo, esto no significa que este sistema operativo sea invulnerable: los programas y el mismo *kernel* (núcleo) poseen fallas que al ser explotadas permiten que, en casos extremos, un extraño tome control del equipo. En cuyo caso, cualquier código malicioso puede ser potencialmente peligroso para el sistema.

Como se mencionó en el apartado anterior, Linux basa parte de su seguridad en una jerarquía de usuarios con permisos y restricciones, un constante monitoreo y registro de los procesos que son ejecutados y por quien son ejecutados. Como un virus es un proceso en ejecución tiene las mismas restricciones que cualquier otro proceso de usuario.

Para que un virus infecte entidades ejecutables en Linux, el usuario que activa el virus debe poseer los permisos necesarios para escribir en archivos ejecutables. Esto se debe a que los archivos ejecutables son administrados únicamente por el usuario *root*. Mientras menos experimentado sea un usuario, más baja es la probabilidad de que obtenga el privilegio de manipular programas ejecutables.

El virus además encuentra el obstáculo de la gestión de memoria. El *kernel* de Linux no permitirá fácilmente a ningún programa en ejecución escribir o alterar las áreas de memoria destinadas al sistema; no se tendrá acceso a zonas de memoria que no sean propias de la cuenta del usuario activo, como lo es la tabla de vectores de interrupción. Por esta razón ningún virus en Linux, hasta ahora, es residente en memoria.

Linux y el software del sistema son casi completamente de código abierto. Esta disponibilidad del código fuente tiene dos efectos sobre los virus: el código abierto es un lugar poco favorable para que un virus se oculte, ya que permite su rápida identificación; y una instalación nueva previamente compilada desactiva el código viral al rescribir el programa original sobre el código infectado. Cada uno de estos obstáculos representa un impedimento significativo para el éxito de la propagación de un virus.

Como se mencionó en el capítulo II, un virus tiene un ciclo de vida que se inicia en la creación y que sigue con la replicación. Los virus se replican para subsistir. Cada uno de los obstáculos antes mencionados reduce drásticamente el rango de esparcimiento de los virus de sistemas Linux. Si la replicación no cumple con el propósito de diseminación, el virus es condenado desde el principio – incluso antes de que se den a conocer informes sobre su existencia – y termina siendo un programa sin éxito.

La razón del por qué no se ha visto una epidemia verdadera de virus en Linux es simplemente que ninguno de los virus existentes puede prosperar exitosamente en el ambiente que les proporciona. Los virus que hoy existen para este sistema son sólo curiosidades de programación. La realidad es que aún no existen virus viables que ataquen a sistemas Linux; por supuesto, esto no significa que nunca habrá una epidemia. El máximo reto para los programadores de virus en Linux es la forma de que sus "creaciones" obtengan todos los privilegios con los que cuenta el usuario *root*.

Aun así, existen diversos antivirus para sistemas Linux, pero están enfocados para mantener a los servidores de correo libres de códigos virales. Esta medida es tomada no porque los virus que llegan en los mensajes electrónicos pongan en riesgo al sistema Linux, sino para prevenir que dichos mensajes lleguen a otros servidores u ordenadores que corren bajo otro sistema operativo (Windows XP o Windows 2003 Server, por ejemplo) y causen el daño que en Linux no pueden provocar.

Virus Destacados de Linux

Pocos son los virus que han logrado realizar su cometido en el ambiente Linux. Sin embargo, no han podido comenzar una verdadera epidemia; y en las versiones actuales de Linux no son contemplados como una amenaza.

Bliss. Se refiere a dos cepas de virus diseñados en lenguaje GNU C que no son residentes en memoria. El virus busca los archivos .ELF y los infecta, grabándose al principio del archivo huésped. En la cepa A la longitud del virus es de 17892 Bytes y sólo infecta ejecutables no contaminados; por su parte, la cepa B tiene una longitud de 18604 Bytes e infecta todos los archivos ejecutables.

Después de infectar, devuelve el control al sistema. Sólo puede infectar archivos que tengan permiso de escritura para el usuario activo; o sea, que podrá infectar todos los archivos ejecutables del sistema cuando el usuario tenga permiso de administrador (*root*).

VIT.4096. Es un virus que no es residente en memoria y es el segundo que se reconoce que afecte a sistemas Linux. Sólo podrá infectar a archivos ejecutables .ELF que tengan permiso de escritura; trabajará a pleno rendimiento si el usuario posee permiso de administrador. Al infectar un archivo, se insertará directamente en medio del cuerpo del ejecutable, haciendo un agujero en la cabecera del archivo y ahí graba 4096 Bytes de código.

El virus evita infectar dos veces un mismo archivo. Mientras infecta, utiliza el archivo temporal VI324.TMP.

Winux. Mencionado en el tema de Virus en Windows XP.

III.2.3. Macintosh OS X

Macintosh OS es el nombre del primer sistema operativo de Apple para los ordenadores Macintosh. El Mac OS original fue el primer sistema operativo con una interfaz gráfica de usuario en tener éxito.

Mac OS X fue comercializado por primera vez en el año 2001 y es la última versión de Mac OS; está basado en un entorno de trabajo tipo UNIX. Su núcleo se llama Darwin y es código abierto, con lo que cualquier persona puede aportar contribuciones encaminadas a mejorar la plataforma siempre y cuando las notificaciones públicas se produzcan después de hacerlas a Apple.

Las características de este sistema operativo son comunes a las de la familia UNIX y, obviamente, a versiones anteriores del Mac OS; éstas son:

- ✓ *Multitarea*. Puede ejecutar varios procesos de forma concurrente, utilizando la técnica de multitarea cooperativa.
- ✓ *Multiusuario*. Más de un usuario puede acceder a la información y los recursos del sistema al mismo tiempo.

- ✓ *Entorno*. Mac OS X dispone de una interfase gráfica de usuario (GUI) denominada Aqua, desarrollada por Apple. Es considerada una de las interfases más avanzadas de la industria, debido a su facilidad de uso, coherencia y armonía estética.
- ✓ *Código fuente no propietario*. El núcleo Darwin es de código abierto y se encuentra registrado por OpenDarwin y GnuDarwin (licencia GNU), ambas versiones perfectamente compatibles entre sí.
- ✓ *Aplicaciones*. Las aplicaciones están divididas en tres familias: Cocoa para aplicaciones propias de Mac OS X, Carbon para aplicaciones modificadas a Mac OS X desde los anteriores sistemas de Macintosh, y Classic, que ejecuta las aplicaciones de los anteriores sistemas directamente en Mac OS X mediante un emulador de Mac OS 9.
- ✓ *Red*. Ofrece amplia compatibilidad con los servicios de archivos en red para trabajar desde prácticamente cualquier entorno de red conocido. Mac OS X soporta conexión con servidores NFS UNIX, enlaza de forma dinámica servicios en red – como servidores de archivos –, además de conexión a una red remota.

Características Generales de Seguridad

Al igual que Linux, Mac OS X está basado en la familia de sistemas operativos UNIX, la cual se ha desarrollado y actualizado desde hace 40 años. Estas características le dan a Mac OS X gran estabilidad, potencia y seguridad para la manipulación de datos e información.

Mac OS X incorpora las características de seguridad presentes en cualquier sistema descendiente de UNIX.

Cuentas de usuario. Mac OS X cuenta con tres diferentes tipos de cuentas que permiten o restringen el acceso completo al sistema:

- ✓ *Root*. Tiene permisos completos para acceder a todos los recursos en el sistema; esto significa que puede ejecutar, leer, modificar y eliminar cualquier archivo y directorio. Además, es quien se encarga de configurar los dispositivos y los servicios del sistema.

- ✓ *Administrador*. El administrador puede realizar la mayoría de operaciones normalmente asociadas con el usuario *root*. Lo único que el administrador no puede hacer es agregar, modificar o eliminar archivos en el dominio del sistema.
- ✓ *Usuario*. La cuenta de usuario es la cuenta con menos privilegios en el sistema Mac OS X; sólo puede modificar configuraciones para su cuenta y no puede impactar todo el sistema.

A diferencia del UNIX tradicional, Mac OS X desactiva la cuenta *root* por defecto. Este método previene que virus o usuarios no autorizados hagan cambios dañinos en el sistema operativo. Adicionalmente a las clases principales de cuentas de usuario, existen servicios del sistema y software que requieren acceso especializado a ciertos componentes del sistema, pero que no requieren acceso autenticado. Mac OS X usa cuentas del sistema menos privilegiadas para ejecutar estas funciones.

Autenticación. Mac OS X soporta autenticación local o basada en red para asegurar que sólo los usuarios con credenciales de autenticación válidas puedan acceder a los datos, aplicaciones y servicios de red de la computadora. Kerberos es la herramienta que permite autenticación única a todos los sistemas y servicios autorizados.

Confidencialidad de los datos. Mac OS X protege la confidencialidad de datos, ya sea que estén almacenados en el directorio inicial, transmitiéndose por Internet o siendo compartidos localmente en red. FileVault mantiene seguros los documentos, incluso cuando la computadora es extraviada o robada, almacenándolos en forma cifrada en el directorio inicial. Además permite crear imágenes de disco cifradas para almacenarla en el sistema local o un servidor de archivos en red. El cifrado se realiza por medio del algoritmo AES.

Seguridad en red. Mac OS X integra protocolos de red altamente seguros que están basados en estándares abiertos, tales como OpenSSL y OpenSSH. Así se confirma la seguridad de los datos mientras atraviesan redes de área local o Internet. Mac OS X vigila y protege la red de un acceso sin autorización mediante su *firewall*, basado en tecnología FreeBSD, que protege las computadoras UNIX más críticas en Internet.

Certificados digitales. El uso de certificados digitales permite a Mac OS X soportar comunicaciones seguras. Los certificados digitales posibilitan importantes servicios de seguridad como autenticación, integridad de los datos, cifrado y no-repudio. Para transacciones seguras en la Web, el navegador de Internet Safari utiliza certificados digitales X.509 para validar usuarios y *hosts*, así como para cifrar la comunicación en Internet.

Seguridad Contra Virus en Macintosh OS X

Los mecanismos de seguridad de Mac OS X en contra de los virus son comunes a los utilizados por otros miembros de la familia de sistemas UNIX.

El primer paso para evitar que un virus cause daño al sistema es la desactivación de la cuenta de *root*; ya que, al ocultar esta cuenta, el sistema impide al acceso de usuarios inexpertos o sin autorización a la configuración del sistema. Esta primera barrera impide que cualquier virus obtenga los privilegios necesarios para tomar el control del sistema.

La siguiente barrera se encuentra en la protección de la memoria; es decir, el área de memoria reservada para el sistema no puede ser invadida por algún proceso de usuario. Al igual que en Linux, las zonas de sistema están restringidas única y exclusivamente para los procesos controlados por el núcleo.

La siguiente barrera la pone los privilegios de los usuarios. Si algún usuario hiciera la petición de ejecutar un programa contaminado, el virus solo podría infectar archivos si la cuenta contase con permisos de escritura. Los únicos archivos vulnerables serían los del directorio del usuario, pero el resto del sistema continuaría resguardado.

Puesto que el usuario *root* es el único usuario con permiso de escritura, el virus no podría infectar archivos del sistema o el sector de arranque de disco. Si los permisos son abiertos de par en par de modo que cualquiera pueda escribir, entonces el virus podría esparcirse en todo el ordenador.

Con estas características Mac OS X permanece un tanto invulnerable al ataque de virus, como toda la familia de sistemas operativos UNIX. Sin embargo, como en el caso de Linux, no quiere decir que no se presentará nunca una epidemia de virus en esta plataforma.

Virus Destacados en Macintosh OS X

Hasta ahora sólo se reconoce la existencia de un código malicioso que ataca al Mac OS X: el gusano Leap.A, descubierto por la compañía británica de antivirus Sophos. Este gusano se esparce por medio del sistema de mensajería instantánea iChat de Apple.

Al momento de escribir este trabajo, aún no se tiene noticia alguna de ataques a sistemas Mac OS X por parte de virus informáticos. Esto no quiere decir que no se desarrollarán, en un futuro cercano, virus contra la nueva tecnología de software de Apple. Por el momento, este sistema operativo se encuentra libre de amenazas virales.

Los temas expuestos, en este capítulo y en el anterior, demuestran cómo un virus utilizará el funcionamiento del sistema operativo para lograr su objetivo; las vulnerabilidades que éste posea serán un factor determinante para la realización de dicho fin. Por lo tanto, el esquema de seguridad que utilice un sistema operativo debe ser lo suficientemente amplio y completo para evitar que un código malicioso obtenga beneficios al manipularlo. Sin embargo, las tareas encaminadas a resguardar la seguridad del ordenador no pueden recaer completamente en el sistema operativo; éste debe ser una base sólida para garantizar la seguridad, no la estructura completa.

Para evitar el daño producido por virus, se cuenta con programas antivirus que complementan la seguridad del sistema operativo. Los antivirus deben poseer dos características fundamentales: excelente acoplamiento a la estructura de seguridad del sistema operativo y un eficiente modelo de detección y protección contra códigos maliciosos. Con estas características se tendrá la certeza de que ningún virus tendrá un camino fácil para provocar una infección.

CAPÍTULO IV

ANTIVIRUS

IV.1. Historia de los Antivirus

Muchas de las enfermedades infecciosas que afectan a la humanidad son provocadas por virus. A finales del siglo XVIII se descubrió la primera vacuna de la historia: la vacuna contra la viruela. Desde ese momento, la humanidad ha apoyado y financiado el desarrollo de diversas vacunas que permitan evitar el contagio de muchas enfermedades infecciosas conocidas (varias de ellas provocadas por virus), para alcanzar el objetivo de la erradicación total de dichos males.

En el campo de la Informática, también se han desarrollado herramientas para contrarrestar los efectos y los daños que los virus pueden causar; dichas herramientas se conocen con el nombre de antivirus y son de los programas más conocidos por los diversos usuarios de la computación.

Actualmente un antivirus no sólo representa el conjunto de programas encaminadas a detectar, desactivar y eliminar los códigos virales, sino también el impedir que otros tipos de códigos maliciosos puedan causar daños a la información o comprometer la estabilidad del sistema informático; además, también se incluyen algunas otras herramientas como *firewalls*, bloqueo de ventanas emergentes, controles de privacidad, eliminadores de *spyware*, detectores de intrusiones, etc.

En este trabajo sólo se discutirá el desarrollo de una herramienta que pueda impedir el correcto funcionamiento de virus informáticos, sentando así una base para el futuro desarrollo de un complemento que pueda combatir a los diversos códigos maliciosos existentes en el ámbito computacional.

IV.1.1. Primeros Antivirus

El primer antecedente de un “antivirus” se encuentra en la década de los setenta. Robert Thomas Morris, uno de los creadores de *Core Wars*, introduce en la ARPANet un programa llamado CREEPER que emitía un mensaje en pantalla: “I’m the creeper... catch me if you can”

(Soy la enredadera... atrápame si puedes). Para resolver el problema se desarrolló el REAPER, un programa que se dedicaba a eliminar al CREEPER.

En la misma década de los setenta, en el Centro de Investigación de Xerox, en California, John F Scoch y John Hupp crearon un programa llamado WORM; el propósito del programa era conseguir un mayor rendimiento para una red de ordenadores. Sería como un supervisor que aumentaría el rendimiento de la red realizando tareas de mantenimiento automático a los más de cien equipos del centro. Se suponía que crearía una copia de sí mismo en sólo seis equipos, pero se salió de control e "infectó" a todos los equipos del centro. El problema fue que había demasiadas copias de WORM, provocando que varios ordenadores estuviesen bloqueados por la falta de memoria. Para resolver el dilema se creó un programa que buscaba, desactivaba y eliminaba cada copia de WORM; éste fue el primer antivirus, como tal, de la historia.

IV.1.2. Los Antivirus Actuales

En 1988 comenzaron a aparecer los fabricantes de antivirus, creando una moda de lo que en principio sólo era un problema potencial. Los vendedores de software antivirus eran pequeñas compañías, que ofrecían sus productos a muy bajo precio, en algunos casos gratuitamente. Fue en este año cuando la compañía IBM se dio cuenta de que tenía que tomarse el asunto de los virus completamente en serio; y desde ese momento, el *High Integrity Laboratory* de IBM fue el encargado del área sobre virus.

El software antivirus existente era utilizado para detectar virus de sector de arranque; solamente fue escrito un programa antivirus, de manera excepcional, para afrontar los brotes de Cascada y Jerusalem.

Aparece en el mercado uno de los primeros antivirus denominado FLU SHOT, creado por Ross Greenberg. John McAfee crea el primer programa antivirus en contra del virus paquistaní BRAIN y además fue el fundador y programador de uno de los antivirus más conocidos de la historia: VIRUSCAN. A partir de este momento surgen diferentes compañías como McAfee, Symantec, Panda Software, Kaspersky Labs o F-Secure que se encargan de combatir a los virus y

a los nuevos códigos maliciosos como gusanos, troyanos *rootkits*, etc.; además de que incluyen ahora herramientas como *firewalls* para evitar intrusiones de usuarios a ordenadores ajenos. Algunos de dichos productos se muestran en la figura 4.1: a)BitDefender 9 Standard, b)McAfee Viruscan 2006, c)F-Secure Anti-Virus 2006, d)Symantec Norton Antivirus 2006, e)Panda Titanium 2006 Antivirus, entre otros.

Figura 4.1. Antivirus comerciales más populares.

Desde entonces, los cambios históricos en el desarrollo de defensas contra virus no han variado mucho. Compañías han aparecido y desaparecido; sin embargo, el enemigo sigue vigente y cambiante a cada momento. Los antivirus deben evolucionar al mismo paso que lo hacen los programas maliciosos que combaten, para continuar la guerra que aún continua.

IV.2. Definición y Tipos de Antivirus

IV.2.1. Definición de Antivirus

Como se dijo anteriormente, un virus se caracteriza por realizar las siguientes funciones: replicarse, ocultarse y dañar. La esencia del antivirus es evitar que los virus lleven a cabo las acciones anteriormente mencionadas; para ello, se realizarán actividades de detección, prevención y corrección de procesos virales. Éste es el aspecto más importante de un antivirus, puesto que el hecho de detectar la posible presencia de un virus, detener su ejecución y tomar las

medidas necesarias, es suficiente para acotar un buen porcentaje de los daños posibles. Con base en lo anterior, la definición de antivirus, a diferencia de la de virus, es mucho más uniforme y más rápida de asimilar. Se puede definir a un antivirus como: *el conjunto de herramientas y aplicaciones que están destinadas a prevenir y detectar la actividad de virus informáticos para evitar su propagación, así como para reparar los daños que hayan causado*.

Así pues, un antivirus es simplemente una barrera que se encargará de bloquear cualquier intento de ataque por parte de virus informáticos en contra de los ordenadores. Una versión robusta de estos programas es una colección de herramientas y aplicaciones que mantienen al ordenador, lo más posible, a salvo de las infecciones y ataques producidos por virus y, en algunos casos, por otros códigos maliciosos. Esta extensión ha sido necesaria debido al gran número de gusanos, caballos de Troya y otros programas malignos que han surgido y se han sumado a los virus en la tarea de robar y modificar la información electrónica.

El antivirus es un programa que debe ser adecuado para el sistema en el cual se desee instalar; y además, estar correctamente configurado según los dispositivos de hardware que se tengan disponibles. Por ejemplo, si se trabaja en un lugar que posee conexión a redes es necesaria la instalación de un programa antivirus que tenga la capacidad de detectar virus de redes. Los antivirus reducen sensiblemente los riesgos de infección, pero cabe reconocer que no serán completamente eficaces y su utilización debe estar sujeta a las políticas de seguridad informática vigentes en el lugar de trabajo.

IV.2.2. Tipos de Antivirus

La clasificación de antivirus puede ser variada; estos programas de defensa se dividen en diversas categorías, atendiendo a varios criterios: por el método de detección que se utiliza, por el instante en que se activa el antivirus, por objeto de desinfección y por las acciones que se realizan cuando se detecta un virus.

La clasificación más adecuada de antivirus es por medio de la función que realiza en contra de un ataque viral; las acciones que puede realizar el antivirus son: detección, protección,

vacunación y reparación. Así pues, estas acciones nombran a cada una de las cuatro categorías de antivirus existentes: detector, protector, vacuna, y reparador.

1. Antivirus Protector

Se les conoce como sistemas de prevención. Este tipo de antivirus intentará detener las acciones del virus en el mismo momento en que se produzca el ataque. Estos programas permanecen residentes en memoria y vigilan las operaciones que realizan los programas que se están ejecutando.

Los sistemas de prevención intentan detener el avance del software viral en tiempo real, vigilando constantemente los posibles puntos de entrada viral al sistema informático e intentan detectar algún tipo de funcionamiento anormal del mismo. Para actuar eficientemente, este tipo de programas se instalan como residentes en la memoria del sistema e interceptan todas las llamadas especiales del mismo; como por ejemplo, abrir programa, escribir en el disco, borrar archivo, etc. Siempre en línea y operativos, confían en el control constante de las interrupciones para detectar e interpretar solicitudes de órdenes gobernadas por software. Cuando se encuentren actividades dudosas (por ejemplo, cuando un programa se carga y pide permiso para escribir sobre los sectores de arranque), los sistemas de prevención entran en acción: interceptan las llamadas al sistema operativo e informan a los usuarios sobre los acontecimientos bloqueados. Es posible que después los usuarios sean consultados sobre si se debe permitir que las acciones interceptadas continúen. Por supuesto, en el caso de solicitudes de escribir en el sector de arranque del disco, las acciones deben ser impedidas (a menos que los propios usuarios estén inicializando los discos protegidos).

Pese a que este método es bueno en la práctica adolece de algunas desventajas. En primer lugar, consume dos recursos importantes del sistema: memoria y tiempo de procesador, ya que los algoritmos de prevención pueden ocupar varios KB de memoria primaria. En segundo término, están constantemente analizando cualquier pedido de acceso al sistema de archivos de todos los discos del computador; interrumpen el trabajo habitual en desarrollo del usuario con advertencias (en pantalla) de intentos de cargas de programas, lecturas y escrituras de discos,

pidiendo la confirmación para proseguir con dichas acciones. Éstas son actividades normales y constantes de uso informático diario y los programas antivirus de prevención no tienen forma de saber qué actividades son iniciadas por los usuarios y cuáles por virus. Además, el mero hecho de permanecer residente en la memoria y de interceptar prácticamente todas las actividades de lectura y escritura en disco es seguro que causará algún conflicto (bloqueo, disminución en velocidad de proceso, etc.) con el software de otros programas.

2. Antivirus Detector.

Éste es el tipo más conocido. Los sistemas antivirus de detección comprueban el código del programa antes de que se ejecute; el usuario podrá ser avisado de los posibles peligros del programa que se ejecutará.

Los sistemas de detección se cargan, se ejecutan y existen de la misma manera que otros programas de aplicaciones. Al contrario que los sistemas de prevención, no retienen permanentemente grandes fragmentos de memoria ni interceptan o interrumpen de algún modo el funcionamiento de otros programas; analizan los archivos ejecutables antes de que se ejecuten, buscando en ellos indicios específicos que hagan suponer la presencia de algún tipo de infección viral. Para realizar esta tarea utilizan algoritmos que permiten aislar “trazas características” – llamadas firmas virales – de varios virus conocidos que se encuentran almacenadas en la lista de definición y que son los códigos que puede detectar el antivirus.

Los antivirus más especializados buscan también la presencia de indicios tales como mensajes característicos – que muchos códigos maliciosos utilizan para darse a conocer – o instrucciones que hagan suponer la presencia de algún tipo de virus no conocido por la versión del programa antivirus. Al detectar una posible infección, se alerta al usuario y es éste quien decide los pasos a seguir: usar igualmente el programa contaminado, descartarlo, analizarlo mas profundamente, eliminarlo, etc.

Los programas de detección antivirus se clasifican en dos grupos: detectores específicos de programas (llamados comúnmente exploradores de virus) y detectores genéricos.

- ✓ *Detectores específicos de programas*. Los detectores específicos de programas buscan un número limitado de virus conocidos. Sondean los archivos de destino en busca de características de los virus para los que han sido programados que detecten; requieren actualizaciones frecuentes cuando se descubren virus nuevos. Son extremadamente vulnerables a los virus polimórficos.
- ✓ *Detectores genéricos*. Estos programas son el tipo más seguro de antivirus. Atacan la principal debilidad que todos los virus informáticos comparten: necesitan modificar los archivos ejecutables para sobrevivir. Operan bajo la suposición de que los cambios no autorizados que ocurran en archivos de programas son indicativos de actividad vírica. Sus archivos de datos normalmente consumen importantes cantidades de espacio de disco; a veces generan falsas alarmas y están expuestos a ataques víricos.

3. Antivirus Vacuna

La vacuna es el tipo más antiguo de productos antivirus que salió a la luz cuando apareció el espectro de los virus informáticos. La respuesta inicial del usuario final a estos productos fue positiva; pero, con el paso del tiempo las desventajas de las vacunas fueron cada vez más evidentes. Prácticamente ya no existen vacunas en el mercado actual de antivirus.

Las vacunas de software agregan pequeños programas y datos de suma total a los archivos ejecutables. Estos archivos son modificados para que, cuando se ejecuten, se pase primero el control a los programas antivirus agregados que comparan las sumas totales activadas con sus datos de suma total agregada. Cuando la comparación cuadra, el control se devuelve a los archives ejecutables, los cuales continúan sus operaciones normales; en caso contrario, se alerta a los usuarios para tomar las medidas adecuadas.

Las principales deficiencias y complicaciones asociadas con las vacunas son las siguientes:

- ✓ Los programas vacunados tardan más tiempo en cargarse debido al proceso de comparación de sumas de datos.

- ✓ Se consume un mayor espacio en disco al aumentar el tamaño de los archivos ejecutables.
- ✓ La mayoría de las vacunas no pueden proteger a los módulos de gestión de periféricos o datos ejecutables.
- ✓ La conducta de las vacunas puede causar conflictos con otros sistemas de defensas víricas.
- ✓ Los virus pueden detectar la presencia del código de la vacuna y borrarlo o modificarlo para saltar la rutina de suma total del tiempo de carga.

Las vacunas operan de un modo fundamentalmente semejante al de los virus informáticos: se agregan a los programas y se ejecutan antes que éstos, aunque no se reproducen sin autorización ni dañan a los archivos intencionalmente. Hay alternativas disponibles más seguras y menos drásticas que insertar código ajeno en los programas del ordenador.

4. Antivirus Reparador

Los reparadores (también conocidos como desinfectores, erradicadores o antídotos) aparecieron en el mercado poco después de la introducción de las vacunas de software. Incluso hoy día, cuando surge una nueva tendencia vírica, se desarrolla inmediatamente un nuevo antídoto vírico. En un principio, algunos antídotos se integraron en programas de vacunas; ahora la mayoría se venden como sistemas autónomos y especializados.

La mayoría de los sistemas de software antivirus ofrecen clasificaciones dentro de sus géneros respectivos y los antídotos víricos no son la excepción. La forma más sencilla de clasificar a los antídotos es separarlos en antídotos de desastre y antídotos de infección.

- ✓ *Antídotos de desastre*. Denominados programas de recuperación de formato, se diseñan para devolver a los sistemas a un estado de funcionamiento después de que haya ocurrido algún acontecimiento destructivo. Se trata simplemente de copias de seguridad de información crítica del disco: sectores de arranque, procesadores de órdenes, tablas de asignación de archivos, directorios de discos y datos de particiones.

- ✓ *Antídotos de infección*. Buscan y eliminan los virus conocidos por medio de una base "archivo por archivo". Los antídotos de infección pueden quitar solamente un conjunto limitado de virus de un reducido grupo de tipos de programas conocidos. La mayoría de los antídotos de infección están dedicados a eliminar una sola clase de virus. Como aparecen nuevos virus continuamente, los antídotos de infección se vuelven anticuados e ineficaces.

La red ideal de seguridad en contra de los virus consta de una combinación inteligente, bien probada y equilibrada, de políticas de seguridad y de herramientas de prevención y detección viral. Las políticas se implantan para que los usuarios eviten riesgos y protejan su información; el software de prevención establece barreras que impiden a los programas maliciosos acceder a los datos del usuario, y las herramientas de detección identifican a los códigos maliciosos después de que se hayan introducido entre los escudos de prevención.

IV.3. Funcionamiento y Efectividad de los Antivirus

Desde los primeros virus, creados como experimentos en los años 80, hasta los últimos, una de las mayores preocupaciones de cualquier usuario de ordenador ha sido la entrada de códigos malignos en su sistema.

Para evitar que los virus realicen su cometido dentro del ordenador, los especialistas en informática han propuesto dos soluciones. La primera es conocida como la "burbuja", que significa aislar completamente al ordenador; es decir, desconectar el equipo de la red y prescindir de cualquier lector de disquetes, CD-ROM o unidades extraíbles. Con esta acción se tendrá la absoluta seguridad de que no entrará ningún virus. Sin embargo, este "método" es un tanto radical, ya que tampoco entrará ningún dato que no sea por el teclado, lo que haría del ordenador una máquina, completamente alejada de lo que es la informática: el tratamiento automático de la información; si no hay información que entre, no se podrá tratar.

La segunda solución es la instalación de un programa antivirus. Con ellos se tendrá la seguridad de que cuando algún código maligno pretenda alterar nuestro sistema, no lo logrará tan

fácilmente. Pero ¿cómo hacen esto los programas antivirus?, ¿por qué un antivirus permite que se instale un juego y no permite que se copie un virus? El funcionamiento de dichas herramientas se expone a continuación.

IV.3.1. Estructura del Antivirus

Al ser uno más de los diferentes tipos de programas existentes, los antivirus cuentan con módulos que le permiten realizar sus tareas de detección y protección del ordenador de una manera más rápida y eficiente. A continuación se describe, de manera general, los módulos que componen a un antivirus.

- ✓ *Módulo de control y monitoreo*. Este módulo registra todos los cambios que se realicen en los programas ejecutables o en áreas críticas del disco duro; el objetivo que persigue es la verificación de la integridad de la información. De esta forma se controla la información en el disco duro, para que sólo sea modificada de acuerdo con los requerimientos del usuario. Además, se encarga de monitorear las rutinas de interrupción que pidan acceso a otros recursos del equipo (hardware y software). Así, se restringe en un gran margen el campo de acción de determinado programa que ha sido identificado previamente como potencialmente nocivo: limitando su uso de recursos; es posible, por ejemplo, que se impida escribir o dar formato a ciertas áreas críticas del disco duro. Este módulo es la implementación del sistema de prevención.
- ✓ *Módulo de identificación*. En este módulo se realiza la identificación del virus, para lo cual existen técnicas como el rastreo (*scanning*) y los algoritmos heurísticos. La identificación de código nocivo encuentra líneas de código incluidas en programas que tengan como objetivo alterar la información del disco duro de manera ilícita; para ello es necesario desensamblar el programa y revisar el código línea por línea. El sistema de detección es implementado en este módulo.
- ✓ *Módulo de respuesta y reparación*. El antivirus reacciona con una alarma que consiste en un aviso en la pantalla ante la presencia de un programa sospechoso. Este módulo también informa al usuario de las posibles alternativas que se tienen para evitar la propagación del virus.

Cada módulo realiza su función en colaboración con otro. El módulo de monitoreo registra todas las actividades que se realicen durante el tiempo de trabajo del ordenador, al acecho de acciones sospechosas que pongan en riesgo al ordenador y a su información. En el momento en que dicho módulo detecta alguna actividad que pueda comprometer al sistema, se activa el módulo de identificación, para determinar si la actividad es provocada por algún virus – u otro código malicioso –; en caso afirmativo se emite la alerta, por medio del módulo de respuesta, y es cuando el usuario debe determinar el siguiente paso para evitar las acciones maliciosas que puedan llevarse a cabo en el ordenador.

IV.3.2. Funcionamiento

Un programa antivirus no es más que un sistema que analiza archivos, y en caso de que se encuentren corrompidos por algún código malicioso, procede a su desinfección. El análisis se produce dependiendo del tipo – o módulo – de antivirus que esté activo. Las figuras 4.2.y y 4.3 muestran, respectivamente, como actúa un antivirus protector y uno detector.

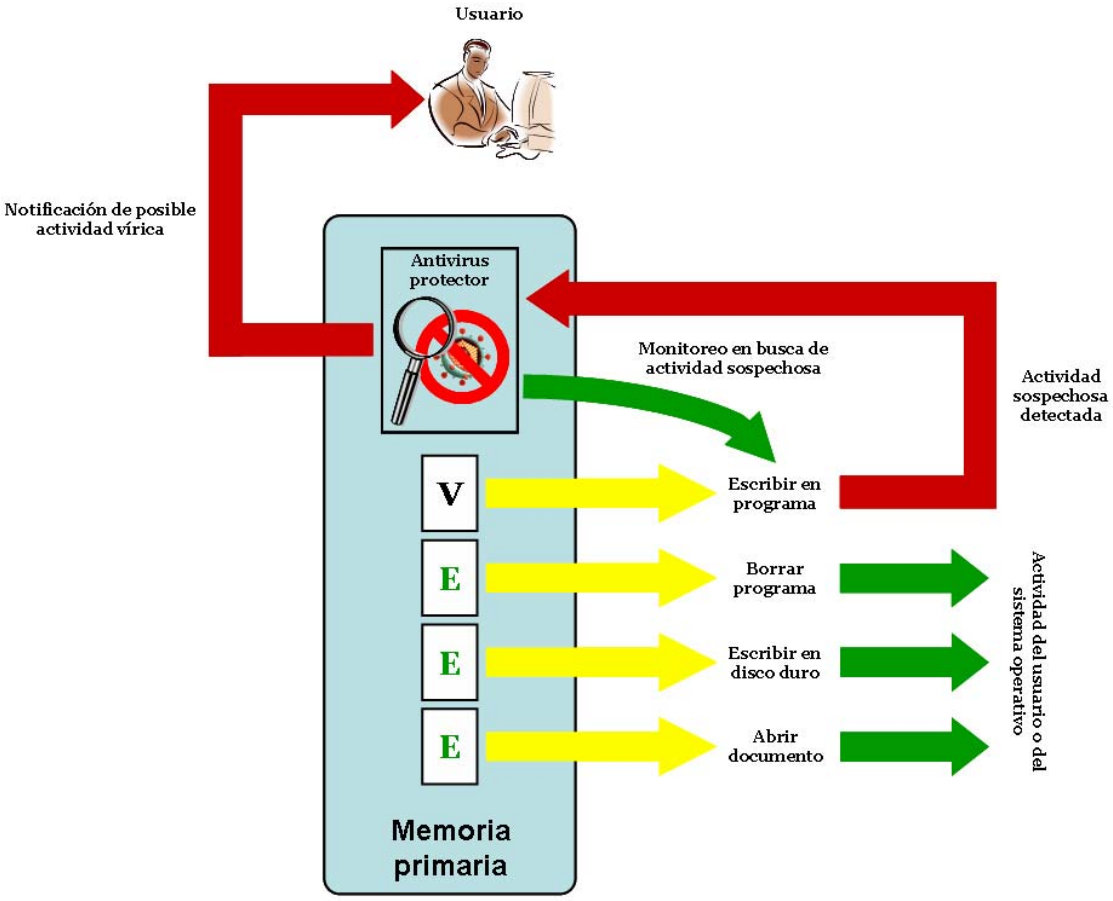


Figura 4.2. Funcionamiento de un antivirus protector.

Al instalar un antivirus protector, este entrará en acción cada vez que el ordenador inicie actividades; quedará residente y mantendrá vigilancia sobre las actividades que realice cada programa que se ejecute en el sistema: apertura, escritura, borrado de archivos, escritura a disco, etc. En el momento en que al antivirus detecte una acción encaminada a la alteración ilegal de la información o de la corrupción del sistema, se emite una alerta que indica al usuario lo que está sucediendo. En este momento, el usuario debe decidir – de entre las opciones que presente el programa antivirus – que medidas deben tomarse para evitar un posible ataque.

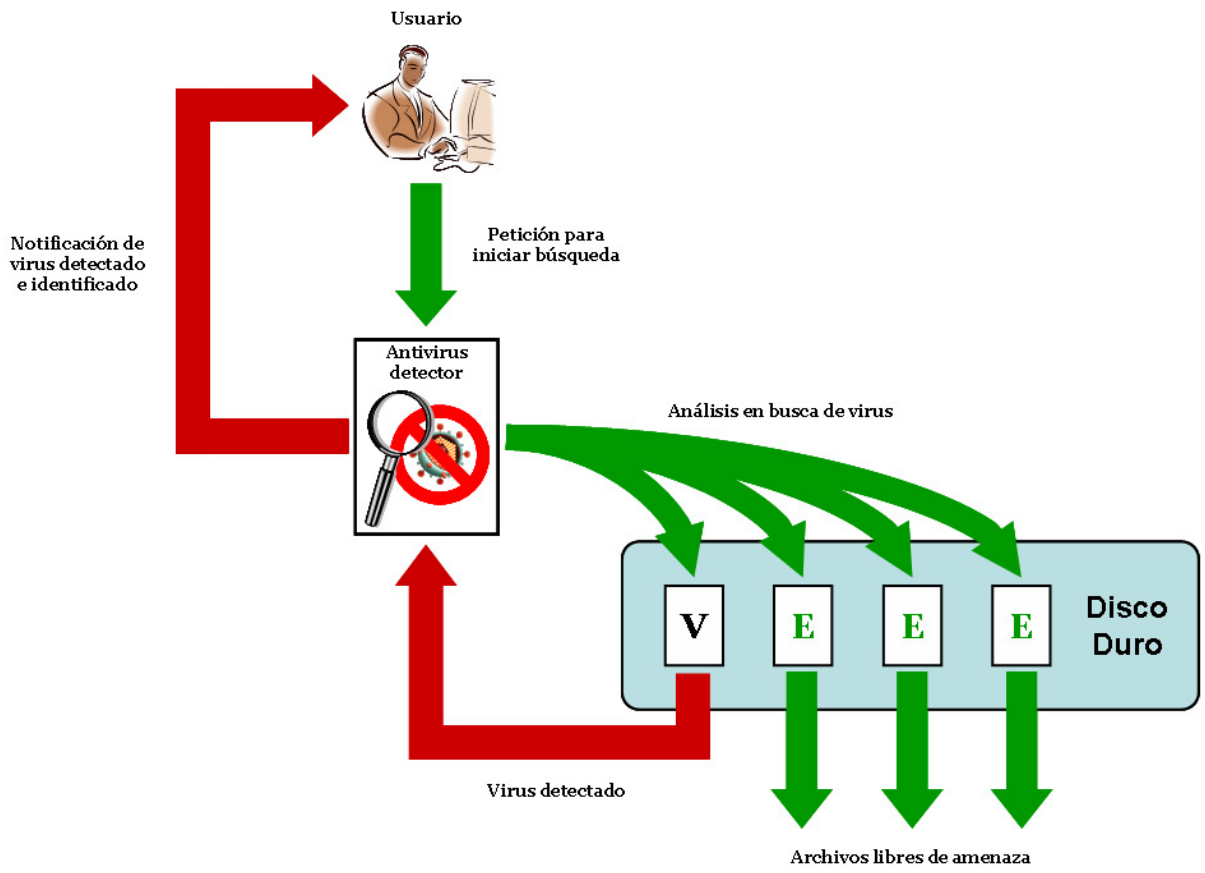


Figura 4.3. Funcionamiento de un antivirus detector.

El antivirus detector permanecerá inactivo hasta que un usuario, el sistema o el antivirus protector haga una petición de búsqueda. A partir de ese momento, el detector analizará todos y cada uno de los archivos o programas que se encuentran en el medio de almacenamiento o el directorio donde se realiza la búsqueda; también puede especificársele que registre los sectores de arranque de discos magnéticos. Si no encontrase evidencia de código malicioso en los archivos que registra, simplemente notificará que dichos archivos están libres de amenaza; por el contrario, al detectar algún indicio de código dañino en el archivo procederá a identificarlo y emitirá una alerta al usuario para que éste indique la medida correspondiente para evitar un ataque.

Una vez analizada la información por el antivirus se lleva a cabo una de dos posibles acciones:

- ✓ Devolver la información al sistema para que siga su curso de procesamiento o ejecución de manera normal, en el caso del protector; o se reporta al archivo como libre de agentes maliciosos, en el caso del detector.
- ✓ Emitir una alarma a la interfase del usuario. Esta interfase es muy diversa, ya que puede tratarse de un mensaje mostrado por pantalla, un mensaje de correo electrónico, un mensaje a la red interna, una entrada en un informe de actividad o una comunicación de algún tipo a la herramienta de gestión del antivirus.

IV.3.3. Técnicas Para la Detección De Virus Informáticos

El modelo más primario de las funciones de un programa antivirus es la detección de su presencia y, en lo posible, su identificación. Para ello, desde el inicio del desarrollo de antivirus, se han implementado técnicas como el rastreo o las búsquedas heurísticas.

Rastreo (S*canning*)

Fue la primera técnica que se popularizó para la detección de virus informáticos, y que todavía se utiliza – aunque cada vez con menos eficiencia –. Consiste en revisar el código de todos los archivos ubicados en la unidad de almacenamiento – fundamentalmente los archivos ejecutables – en busca de pequeñas porciones de código que puedan pertenecer a un virus informático. Este procedimiento se realiza empleando una base de datos que contiene fragmentos de código representativos de cada virus conocido.

La técnica de rastreo fue bastante eficaz en los primeros tiempos de los virus informáticos, cuando había pocos y su producción era pequeña. Este relativamente pequeño volumen de virus permitía que los desarrolladores de antivirus por rastreo tuvieran tiempo de analizar el virus, extraer el fragmento de código usado para la identificación y agregarlo a la base de datos del programa para lanzar una nueva versión. Sin embargo, la obsolescencia de este mecanismo de identificación como una solución antivirus completa se encontró en su mismo modelo.

La gran debilidad de este sistema radica en que al detectarse un nuevo virus, éste debe aislarse por el usuario y enviarse al fabricante de antivirus, la solución siempre será a posteriori: es necesario que un virus informático se disperse considerablemente para que se envíe (por usuarios capacitados, especialistas o distribuidores del producto) a los desarrolladores de antivirus. Éstos lo analizarán, extraerán el trozo de código necesario que lo identifica y lo incluirán en la próxima versión de su programa antivirus; el proceso puede demorar meses a partir del momento en que el virus comienza a tener una gran dispersión, lapso en el que puede causar graves daños sin que pueda identificarse. La consecuencia inmediata de esta limitante es que los programas antivirus deben actualizarse periódicamente debido a la aparición de nuevos virus.

En síntesis, la técnica de rastreo es altamente ineficiente hoy en día, pero se sigue utilizando debido a que permite identificar rápidamente la presencia de los virus más conocidos.

Comprobación de Suma (*Check Sum*)

Es otro método de detección de virus. Mediante una operación matemática que abarca a cada byte del archivo, se genera un número llamado suma de comparación (*check sum*) para cada archivo. Una vez obtenido este número, las posibilidades de que una modificación del archivo alcance el mismo número son muy pocas; si se detectan programas infectados, simplemente no se permite que se ejecuten. Por eso, es un método tradicionalmente muy utilizado por los sistemas antivirus. Los programas que realizan esta tarea se denominan monitores de integridad.

Para realizar las comprobaciones el antivirus debe tener una imagen del contenido de la unidad de almacenamiento desinfectada. Se crea entonces un registro con las características de los archivos, como puede ser su nombre, tamaño, fecha de creación o modificación utilizando la suma de comparación. Si un virus inyectara parte de su código en el archivo la nueva comprobación de la suma sería distinta a la que se guardó en el registro y el antivirus alertaría de la modificación. En el caso de los sectores de arranque algunos antivirus pueden guardar directamente una copia de cada unidad física (registro de arranque maestro) y cada unidad lógica

(registro de arranque) en un archivo y luego compararlos contra los que se encuentran en las posiciones originales.

Una vez que el antivirus conforma un registro de cada uno de los archivos en la unidad podrá realizar las comprobaciones de integridad. Cuando el comprobador es puesto en funcionamiento cada uno de los archivos serán registrados; nuevamente se aplica la función *checksum* y se obtiene un valor que es comparado contra el que se guardó en el registro. Si ambos valores son iguales el archivo no sufrió modificaciones durante el período comprendido entre el registro anterior y la comprobación reciente; por el otro lado, si los valores no concuerdan significa que el archivo fue alterado y en ciertos casos el antivirus pregunta al usuario si quiere restaurar las modificaciones. Lo más indicado en estos casos sería que un usuario con conocimientos sobre su sistema avale que se trata realmente de una modificación no autorizada –y por lo tanto atribuible a un virus –, elimine el archivo y lo restaure desde la copia de respaldo.

La comprobación de integridad en los sectores de arranque es similar: el monitor verificará que la copia que está en uso sea igual a la que fue guardada con anterioridad. Si se detectara una modificación en cualquiera de estos sectores, se preguntará al usuario por la posibilidad de reconstruirlos utilizando las copias guardadas. Cuando se detecta una operación de escritura en uno de los sectores de arranque, el programa alerta al usuario indicándole sobre qué es lo que está sucediendo. Por lo general, el antivirus ofrece sobre como proceder, como evitar la modificación, dejarla continuar o no tomar ninguna medida.

La gran desventaja es que los programas de comprobación de suma sólo pueden detectar una infección después de que se produzca. Además, los virus más modernos, para ocultarse, buscan los archivos que generan los programas antivirus con estos cálculos de tamaño; una vez encontrados, los borran o modifican su información.

Búsqueda Heurística

En virtud de la pronta obsolescencia de la técnica de rastreo, los desarrolladores de programas antivirus han dotado a sus creaciones con métodos de búsquedas que no identifican

específicamente al virus sino a algunas de sus características generales y comportamientos universales. La alternativa para poder neutralizarlos, sin haber programado antes el antivirus para su reconocimiento, es la denominada *búsqueda heurística*, que nace de la necesidad de una detección genérica de los virus informáticos. Como se vio anteriormente, la detección genérica da la posibilidad de detectar cualquier virus, aún sin haberlo analizado antes y sin estar en la base de datos del antivirus que se esté considerando. A través de ella, el programa antivirus analiza el código de los programas buscando instrucciones, acciones sospechosas o indicios que delaten la presencia de virus en el sistema, de acuerdo a los patrones habituales empleados por estos códigos maliciosos. Es una tecnología de programación que dentro de sus rutinas de detección de especies virales, incluye las cadenas clásicas que son similares o afines a virus auténticos.

El funcionamiento de dichas búsquedas es sencillo. En el primer paso se analiza cada programa sospechoso sin ejecutar las instrucciones; es decir, se desensambla el código de máquina para deducir que haría el programa si se ejecutara. Cuando se analizan las primeras instrucciones de cualquier archivo, se verán instrucciones para detectar los parámetros de la línea de comandos, borrar la pantalla, llamar a alguna función, ejecutar alguna macro, etc.; no obstante, si se tratase de un virus, suelen ser otras instrucciones muy diferentes las que se presentan como activar o descifrar el código viral o buscar más archivos para intentar insertarles su código. El segundo paso es concluir si dichas líneas de código presentan el comportamiento de un programa viral y avisar al usuario del peligro potencial que representa el programa.

El principal problema de las búsquedas heurísticas ha sido los llamados errores tipo I y tipo II. Cuando sucede que el antivirus confunde a un programa benigno con un código viral, se dice que ha cometido un error de tipo I o “falso positivo”, ya que alerta sobre la presencia de un virus inexistente; en cambio cuando el antivirus no reconoce a un virus se dice que ha incurrido en un error de tipo II o “falso negativo”, debido a que no detectó la presencia de un código viral existente.

A pesar de que se ha mejorado mucho para corregir estos errores, en los últimos años se sigue sin conseguir demasiada efectividad. El problema, más que en la calidad de la rutina heurística, está en la interpretación que el usuario realice del aviso heurístico; esto hace necesario

que el usuario conozca un poco acerca del funcionamiento del ordenador para poder identificar las falsas alarmas generadas por el algoritmo heurístico. Además, se tiene la vulnerabilidad de que los virus nuevos evitan directamente la búsqueda heurística modificando los algoritmos, hasta que el antivirus no es capaz de identificarlos.

La técnica de búsqueda heurística es una forma eficiente de detectar a especies virales que pertenecen a una misma familia, aunque no es un método absolutamente exacto o eficiente. Entendiendo a la heurística como un indicador de probabilidad de contagio, se lleva a considerarla como un sistema de detección mejorada, que al incluirla en los antivirus permite establecer un sistema de detección ante la aparición de mutaciones de virus o incluso de nuevos virus.

IV.3.4. Efectividad

Al igual que otros programas, los antivirus muestran debilidades que reducen su efectividad para combatir códigos maliciosos. Esta reducción no sólo depende de las debilidades propias de las técnicas de protección, detección y reparación; también depende de los errores en la programación, los errores al instalar el programa o incluso errores del usuario al tomar decisiones frente a una contingencia viral.

Políticas de Seguridad Contra Virus

Las políticas de seguridad, como se mencionó en el capítulo I, son un conjunto de leyes que regulan la manera de proteger los recursos y alcanzar los objetivos de seguridad. Estas políticas deben contemplar amenazas, recomendaciones, planes de contingencia, normas de seguridad, herramientas y cualquier otro aspecto que influya directamente en la seguridad tanto física como lógica de la información, los equipos de cómputo y los usuarios.

Las políticas antivirus son un apartado dentro de las políticas de seguridad. En este apartado se debe considerar el acceso y control de la información de los usuarios, la instalación y

configuración de los programas antivirus, la capacitación del personal para el manejo adecuado del antivirus y la administración de los sistemas para evitar la ejecución de programas maliciosos.

El buen funcionamiento del antivirus está sujeto a las políticas antivirus. Si estas políticas no se siguen correctamente, el antivirus de poco servirá como protección del ordenador. Una mala instalación o configuración del antivirus, la irresponsabilidad o poca experiencia del usuario al manejar medios informáticos o incluso un mal diseño de políticas antivirus pueden permitir que un código malicioso (ya sea virus o cualquier otro tipo) comprometa la información y al ordenador y hará del antivirus un programa inútil.

Por esta razón debe instruirse a los usuarios de ordenadores del peligro que representan los virus informáticos, cómo funcionan y qué hacen, de las herramientas que existen para combatirlos y de cómo usar esas herramientas.

El antivirus por si sólo no protegerá completamente al ordenador, pero usándose correctamente e informándose sobre las amenazas de las que trata de proteger, será una fuerte barrera que protegerá los bienes informáticos de los usuarios.

Errores de Programación y de Instalación

Los errores de programación – también llamados *bugs* – son muy comunes e inherentes a todos los programas. Son debidos a errores en el diseño, al escribir código o en los algoritmos del programa.

En el caso de los antivirus se puede presentar errores en la programación que limiten su efectividad al proteger el sistema informático; esto último puede crear vulnerabilidades que sean explotadas por los códigos maliciosos. En todo caso, la única forma de evitar los errores de programación es mediante una revisión exhaustiva y constante del diseño del antivirus, experiencia en la programación de los algoritmos y de la colaboración de los usuarios informáticos que informen sobre diversos *bugs* que presente el antivirus.

Con respecto a los errores de instalación, éstos pueden presentarse principalmente en dos casos: 1) los requerimientos de hardware no son adecuados y 2) existe conflicto entre el sistema operativo y el antivirus.

En los requerimientos de hardware es necesario conocer la arquitectura del procesador, la memoria con la que se cuenta, el espacio en disco duro y los dispositivos habilitados del ordenador. Teniendo esta información se da el primer paso en la exitosa instalación de la herramienta antivirus seleccionada. El desarrollador de antivirus creará su programa con base en el objetivo de proteger la información, pero dotará a su creación de diferentes características que le permitan realizar su tarea en diferentes arquitecturas.

Una vez realizada la comprobación de que el hardware es apto para ejecutar el antivirus se debe tomar en cuanta la compatibilidad con el software ya instalado en el ordenador. Como se vio en el capítulo anterior, cada sistema operativo trabaja y administra los dispositivos y programas de diferente manera. Por lo tanto, es indispensable que el antivirus que se desea instalar sea compatible con la versión del sistema operativo instalada en el ordenador; no será lo mismo instalar un antivirus para Microsoft Windows XP que uno para Linux Fedora Core o uno para servidores de correo de Macintosh.

Efectividad de los Algoritmos de Búsqueda y Pruebas de Antivirus

Los algoritmos de búsqueda utilizados por los antivirus cada día son más robustos y eficaces al realizar su tarea en la detección de virus informáticos. Las técnicas antes mencionadas han evolucionado junto con los nuevos virus, convirtiéndose en una guerra en la cual se debe pensar como el enemigo antes de que este ataque.

Para que un antivirus sea considerado efectivo en la protección de un sistema informático y capaz de combatir a los códigos maliciosos, se analizan las ventajas y desventajas de los métodos de detección utilizados, la calidad en la programación, pero sobre todo, se les somete a rigurosas pruebas encaminadas a mostrar que tan capaces son para la detección, protección y eliminación de los códigos virales.

Para los antivirus comerciales existen varias organizaciones que se dedican a realizar las pruebas de capacidad. A continuación se describen brevemente algunas de las pruebas de antivirus mas aceptadas en el mundo.

- ✓ *Certificación ICSA*. La ICSA (*Internacional Computer Security Association*) realiza pruebas de antivirus desde 1992; muchos productos populares son sometidos a sus varios esquemas de certificación. Los criterios considerados en esta prueba son detección al momento del acceso y por solicitud, remoción de virus y falsos positivos. Las pruebas de detección primaria se dividen en dos: la detección de virus en *the Wild*, y la detección de virus en el *zoo*. En la primera prueba los productos deben detectar al menos el 90% de la lista *zoo* de ICSA. En las pruebas de los virus en *the Wild* los productos deben detectar el 100%, usando la versión de *The WildList* que haya sido lanzada un mes antes de la fecha de la prueba.
- ✓ *Westcoast Labs Checkmark. Westcoast Publishing* se estableció como líder mundial en la prueba y certificación de antivirus a mediados de los noventa con su introducción del *Westcoast Labs Checkmark*. Los criterios de prueba dependen del nivel de certificación para el que se aplique. El nivel uno mide la habilidad del producto evaluado para detectar todos los virus en *The Wild*, usando muestras basadas en la edición de *The WildList* de no menos de dos meses antes del lanzamiento del producto. En el nivel dos los productos deben desinfectar los virus del nivel uno y además los de la versión de la *WildList* que haya sido publicada un mes antes del lanzamiento del producto. Ambas pruebas son llevadas a cabo usando virus replicados por *West Coast*, midiendo así la habilidad de los productos para detectar virus que constituyen la amenaza real. Los productos certificados son anunciados regularmente.
- ✓ *Prueba VTC Malware de la Universidad de Hamburg*o. Los estudiantes del VTC (*Virus Test Centre*) de la Universidad de Hamburgo han estado diseñando y realizando pruebas de antivirus desde 1994. Los resultados de estos proyectos se distribuyen gratuitamente al público en general. Esas pruebas han evolucionado desde simples pruebas de detección de virus de arranque de sistema y de archivos hasta las actuales pruebas de virus. En adición a su gran colección *zoo*, a principios de 1999, el VTC empezó a usar

ejemplos copiados de la colección de virus en *The Wild* de la *WildList Organization* en sus pruebas, asegurando de esta manera una representación acertada de la habilidad del producto para enfrentar una amenaza real de virus.

- ✓ *Universidad de Magdeberg*. Las pruebas de antivirus para el ATC (*Antivirus Test Centre*) de la Universidad de Magdeberg, hechos en cooperación con *GECA Software & Medienservice*, son relativamente nuevas en el área de la evaluación de antivirus. Las pruebas, patrocinadas por compañías de antivirus, proveen resultados a revistas como CHIP, FreeX, Network World, PC Shopping y PC welt; los resultados de estas pruebas han sido incluidos en las reseñas que han publicado. La mayoría de los criterios de prueba han sido escogidos por administradores de redes, usuarios, revistas, Compañías Antivirus y la Universidad; estos criterios incluyen detección de virus en *The Wild* y desinfección. Los resultados son publicados en inglés y en alemán.
- ✓ *Virus Bulletin*. El Virus Bulletin (VB) ha probado productos antivirus desde que la publicación comenzó en 1989. Los productos para las varias plataformas son revisados regularmente en las Reseñas Comparativas del VB, las cuales prueban los productos con una colección *zoo* (Infecciones de archivos estándar de DOS y Windows, virus macro y virus polimórficos) así como también con un grupo de virus en *The Wild*. El grupo de virus en *The Wild* del Virus Bulletin está basado en una versión de la *WildList* anunciada aproximadamente dos semanas antes de la fecha límite de entrega del producto.

Estas pruebas son sólo algunas de muchas otras que se practican a los programas antivirus. Se mencionan debido a que son las más destacadas y que son las que dan certificación de que un programa antivirus cumple su función adecuada y eficientemente.

Al igual que los virus, los antivirus cuentan con algoritmos bastante sofisticados para realizar sus funciones. Ambos tipos de programas son piezas de software; como tales, sus respectivos programadores saben de la necesidad de una mayor eficiencia en sus rutinas. Es por ello que, conforme se descubren nuevas herramientas, técnicas y algoritmos de programación, la evolución de ambos debe ser un proceso metódico y bien estructurado.

El antivirus no es más que una herramienta; y como tal, se debe utilizar en conjunto con otras utilidades que permitan resguardar la seguridad de la información. Existen muchos tipos, marcas, características de rastreo, etc. El antivirus invulnerable no existe; sin embargo, su correcta utilización (que va desde la elección del tipo de antivirus hasta su correcta manipulación) mantendrá a salvo de ataques virales o de otras amenazas latentes, lo más posible, al sistema informático.

CAPÍTULO V

DISEÑO DEL ANTIVIRUS

V.1. Análisis Previo al Diseño del Antivirus

Este capítulo se destinará a explicar el diseño de un antivirus. Para ello se tomará en cuenta todo el conocimiento explicado en los capítulos anteriores, con el fin de obtener un antivirus que cumpla con los objetivos propuestos al inicio de este documento de investigación.

Antes de comenzar con el diseño del antivirus hay varias preguntas que se deben contestar: ¿qué problema se piensa resolver con la pieza de software a desarrollar?, ¿qué tipo de programa se diseñará?, ¿para qué sistema operativo se diseñará?, ¿cuáles son los requerimientos de dicho sistema operativo?, ¿qué herramientas son las más adecuadas para el desarrollo de la pieza software?, ¿qué evolución se espera del programa?, etc. Estas preguntas se deben plantear antes de comenzar el diseño de cualquier pieza de software, independientemente de las tareas y funciones que realizará.

V.1.1. Planteamiento del Problema

Para iniciar el desarrollo de cualquier pieza de software se debe hacer esta pregunta: ¿para qué se diseñará el programa? Esta pregunta permite conocer cual es el problema a enfrentar y presentar sus posibles soluciones. La respuesta concreta a la pregunta anterior se encontrará siempre en los objetivos que se planteen posteriormente a la formulación de la pregunta.

La principal problemática que presentan los virus informáticos es el caos que causan dentro de un ordenador: destruyen información, consumen recursos y reducen la vida útil de los dispositivos. La pérdida de información, a su vez, repercute en otros aspectos fuera del contexto informático como el económico, el educativo, el científico y tecnológico, el biológico y en general en toda la sociedad.

Los virus son una de las armas que se utilizan con el fin de destruir información. Para evitarlo, es necesario recurrir a los dos objetivos primordiales de la Seguridad Informática: prevenir y proteger la información. Prevenir indica que se realizarán acciones encaminadas a reconocer, analizar y evitar todas las amenazas que pueden dañar los bienes de una persona o un

grupo de personas (en este caso la información). Proteger quiere decir que, cuando la prevención no ha sido suficiente, se tienen barreras necesarias para impedir que las amenazas dañen los bienes.

Una de las barreras que protegen la información son los antivirus. Para este caso particular, el diseño de un antivirus está basado no sólo en la necesidad de contar con una herramienta que impida el accionar de estos códigos maliciosos, también está justificado con base en los siguientes objetivos:

- ✓ Que se fomente el desarrollo de herramientas de seguridad informática dentro de la Facultad de Ingeniería.
- ✓ Que se tenga a la mano un documento que forme una base para el estudio de virus informáticos, sus consecuencias y como combatirlos.

En ambos enunciados se encuentra implícito el primer objetivo de un antivirus: impedir a los virus informáticos su tarea de destrucción de la información.

Así pues, se puede responder que se diseñará el antivirus con la intención de presentar una herramienta que combata a los virus informáticos y que sirva de base para futuros proyectos sobre seguridad informática dentro de la Facultad de Ingeniería.

V.1.2. Plataforma de Desarrollo

En el capítulo III se describió las características técnicas y de seguridad de los tres sistemas operativos más populares: Microsoft Windows XP, Linux y Macintosh OS X. Se señaló que de estos tres sistemas, el más popular y desafortunadamente el más vulnerable es Windows XP.

Los ataques de códigos virales a lo largo de 20 años han hecho de los sistemas operativos de Microsoft un blanco fácil, debido a las pocas opciones de seguridad que estos sistemas implementan. La más clara muestra es Windows XP en ambas versiones: Home y Professional Edition. La característica que Microsoft ha destacado, con respecto a seguridad, es la opción de

políticas de restricción de software: se enfocan en la necesidad de regular el software desconocido o poco confiable, para evitar que códigos malintencionados se ejecuten y comprometan la seguridad del sistema. Aún con las medidas que se han tomado, Windows XP sigue siendo el blanco predilecto de virus.

¿Pero que sucede en la versión Home? Como se mencionó anteriormente las políticas de restricción de software están disponibles únicamente en Windows XP Professional. Esto último indica que los ordenadores hogareños (que en su mayoría cuentan con la versión Windows XP Home) están virtualmente desprotegidos contra ataques de virus. Por lo tanto, es necesario que el usuario adquiera una versión completa de algún antivirus para evitar ataques al ordenador. Esta necesidad provoca que se desarrollen herramientas de fácil uso para usuarios fuera del ámbito computacional.

Este antivirus será desarrollado para el ambiente de Windows XP Home Edition, atendiendo la necesidad de protección de un usuario hogareño, además de brindar un diseño básico y simple para su fácil comprensión y posterior evolución en complejidad.

V.1.3. Definición del Tipo de Antivirus

Uno de los puntos principales del análisis es determinar que tipo de antivirus será el más apropiado desarrollar, tomando como base las características de la plataforma en la que se desarrollará, de cada tipo de antivirus, de las necesidades del usuario final y de las características de las herramientas de desarrollo.

Como se vio en el capítulo IV, los diferentes tipos de antivirus tienen ventajas y desventajas que permiten entablar una discusión sobre cual es el más adecuado de diseñar, desarrollar, implementar y utilizar. Dichas características se concentran en la tabla 5.1.

Antivirus	*Ventajas*	*Desventajas*
Protector	✓ Vigila constantemente el sistema. ✓ Mantiene un estricto control de las actividades dentro del ordenador.	✓ Consume recursos del sistema todo el tiempo. ✓ Interrumpe el trabajo habitual del usuario. ✓ Retrasa los procesos que se ejecutan o ejecutarán. ✓ Puede generar conflictos con otros programas.
Detector	✓ Comprueba el código antes de que se ejecute un programa. ✓ Consume recursos sólo en el momento de operación. ✓ No interrumpe las actividades de otros programas o del usuario.	✓ No puede detener un ataque viral en proceso. ✓ Puede generar falsa alarmas. ✓ Necesita de actualizaciones constantes.
Vacuna	✓ Es eficaz contra un tipo específico de virus. ✓ Comprueba el código del programa antes de que se ejecute.	✓ No puede detener un ataque viral en proceso. ✓ Los programas vacunados tardan más tiempo en cargarse. ✓ Consume un mayor espacio en disco. ✓ No protege a controladores de periféricos. ✓ Puede causar conflictos con otros antivirus.
Reparador	✓ Puede devolver a los sistemas a un estado de funcionamiento después de un ataque. ✓ Realiza copias de seguridad de la información.	✓ Consume espacio en el disco duro. ✓ Solamente puede eliminar una sola clase de virus. ✓ No puede detener un ataque viral en proceso.

Tabla 5.1. Ventajas y desventajas de los tipos de antivirus.

Con base en el cuadro anterior, desde una visión particular, el antivirus más conveniente de diseñar es el detector, ya que presenta ventajas más bien pasivas – no impide un ataque en el momento en que se está realizando –, fomentan la prevención por medio de la búsqueda de amenazas potenciales y, sobre todo, brindan comodidad al usuario al no interferir durante todo el tiempo que el ordenador esté trabajando (consume recursos sólo en el momento de su activación y no retrasa los procesos rutinarios al monitorear el sistema).

V.1.4. Herramientas Para el Diseño del Antivirus

Uno de los momentos más importantes en los pasos previos es la visualización de las herramientas de programación que se utilizarán.

Existe actualmente una gran variedad de herramientas de programación que presentan diferentes características, dependiendo de la plataforma, sistema operativo y utilidad que se le dará al programa.

Debido a que el sistema operativo es Windows XP, las herramientas de desarrollo más utilizadas y populares en esta plataforma son *C/C++*, *Delphi*, Java y *Visual Basic*. Cada lenguaje tiene características que están orientadas a diferentes aspectos de desarrollo de software.

La familia del lenguaje C/C++ es considerada como la más completa, compleja y potente para el desarrollo de sistemas y, en muchas ocasiones, de aplicaciones, ya que aporta un diseño de tipos y estructuras de datos que consiguen claridad y eficacia en su manejo. Es un lenguaje que permite realizar tres tipos de programación: estructurada, orientada a objetos y genérica. Esta flexibilidad permite economizar las expresiones, con abundancia de operadores y tipos de datos (aunque los básicos sean pocos); codificación en alto y bajo nivel simultáneamente, reemplazando ventajosamente la programación en ensamblador y permitiendo una utilización natural de las funciones primitivas del sistema; clases y funciones virtuales brindan un auténtico motor de objetos con herencia múltiple; no está orientado a ningún área en especial y sobre todo consigue un código objeto altamente optimizado que repercute en que los programas se ejecutan rápidamente y otorgan el control absoluto de la aplicación al programador; además es un código

sumamente portátil: un programa realizado en una plataforma predeterminada puede ser compilado nuevamente en otra plataforma completamente diferente. Las razones anteriores hacen que C/C++ sea la opción más viable para el desarrollo de este proyecto.

V.1.5. Futuro del Antivirus

El propósito del antivirus que se diseñará es establecer una base para el futuro desarrollo de herramientas de seguridad informática y fomentar el estudio, dentro de la Facultad de Ingeniería, de los virus y posteriormente de otros códigos maliciosos.

Para evitar que el tiempo ponga en desuso esta pieza de software, se debe mantener una regular atención en la actualización de cada módulo del programa. El mantenimiento, la corrección y la actualización son los procedimientos más adecuados para que este programa antivirus contribuya al combate de los virus. Además, es posible que se desarrollen extensiones del programa encaminadas a la protección contra otros códigos maliciosos (no sólo virus), otras amenazas específicas de seguridad (*monitorización*, *spoofing*, etc.) o incluso a defender a los usuarios de nuevas formas de explotar las vulnerabilidades de un sistema, que tal vez en este momento no han sido descubiertas.

El futuro de esta herramienta está pensado a largo plazo. Si bien, no será posible que se cuente dentro de las herramientas más modernas y sofisticadas en un futuro próximo, pero con un mayor tiempo de desarrollo, se puede convertir en uno de los antivirus más eficaces y confiables. El camino es largo, pero con la colaboración de alumnos, profesores o cualquier otra persona interesada en el tema, se podrá concretar y obtener una herramienta lo suficientemente eficaz para evitar la modificación malintencionada de la información.

V.2. Diseño del Antivirus

Para comenzar con el diseño del antivirus se debe contestar las siguientes preguntas: ¿qué tareas realizará el programa?, ¿cómo se realizarán dichas tareas?

El antivirus que se diseñará tiene los propósitos de detectar e identificar código viral dentro de un archivo ejecutable. Para realizar estas acciones se debe manipular el archivo que se analizará con funciones básicas como abrirlo, leerlo y comparar su código con patrones de virus.

Como se vio en el capítulo IV, los antivirus cuentan generalmente con tres módulos: módulo de monitoreo, módulo de identificación y módulo de respuesta. En el antivirus detector, el módulo de monitoreo será aquel que realice la lectura y el análisis del código ejecutable; el módulo de identificación se encargará de especificar el nombre del código que se ha encontrado en el archivo (si fuese el caso); el módulo de respuesta, con base en la información proporcionada por el módulo de identificación, informará al usuario de lo encontrado e intentará corregir el problema. La figura 5.1 muestra un esquema de bloques del antivirus con sus módulos y las acciones que intervienen en el análisis.

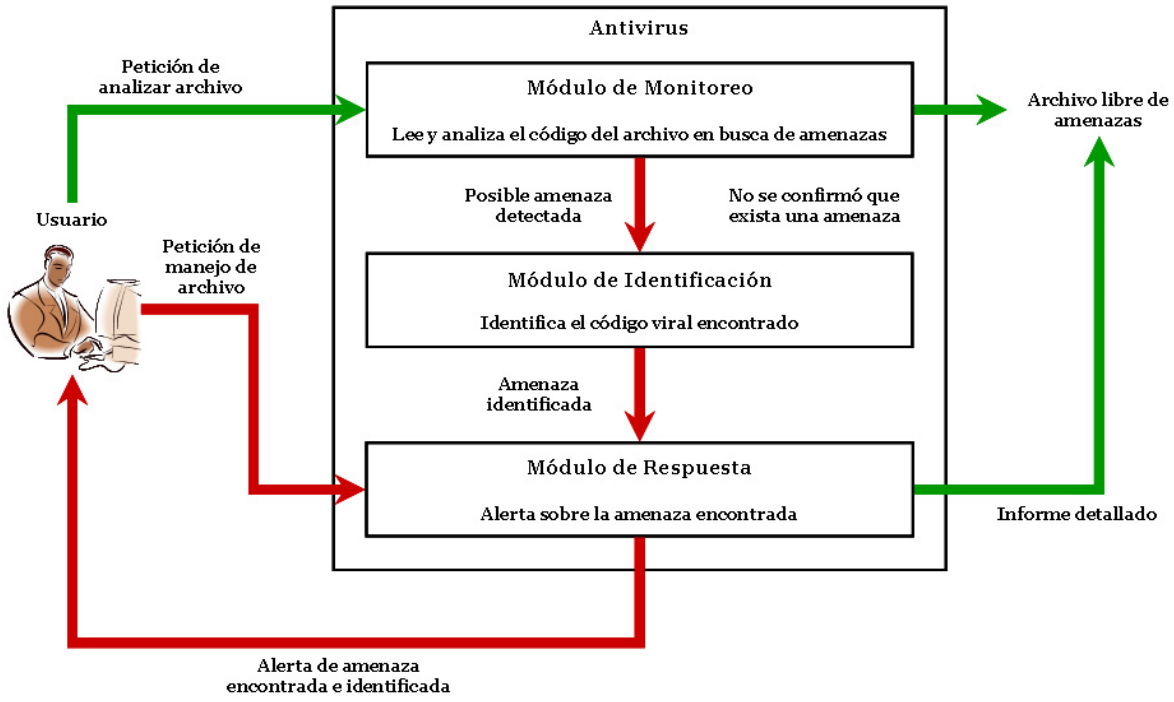


Figura 5.1. Diagrama de bloques del antivirus.

El diagrama de bloques muestra la dirección que tomará cada acción que se realice dentro del análisis de un archivo. Para explicar detalladamente lo que cada bloque realizará, se muestra en la figura 5.2 el *diagrama de flujo* correspondiente.

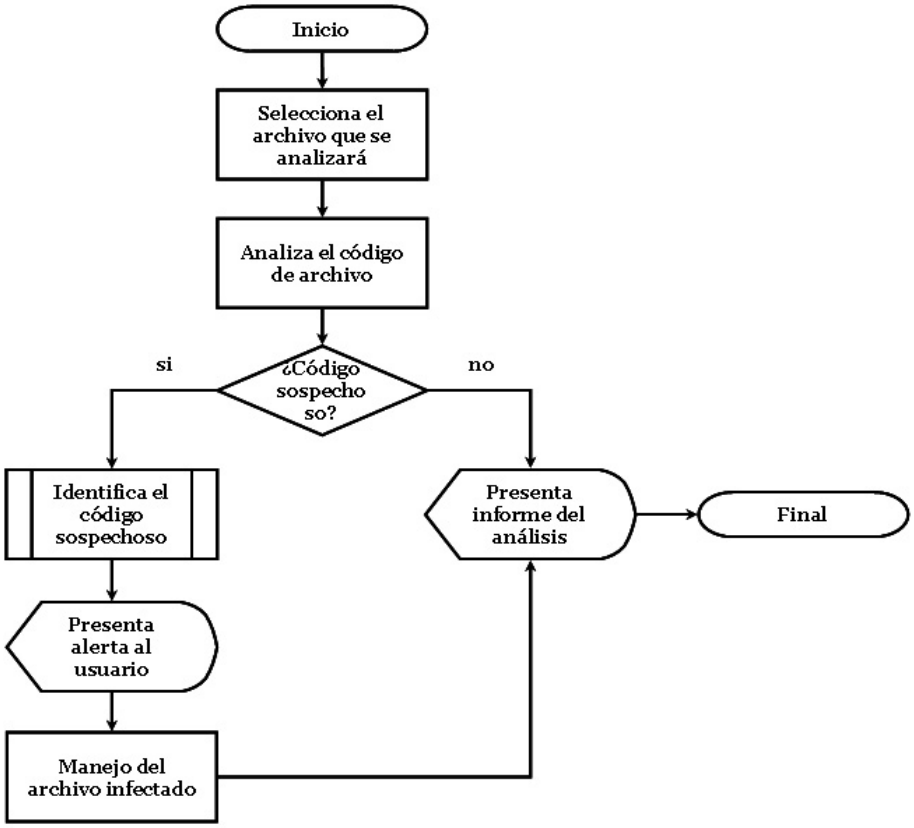


Figura 5.2. Diagrama de flujo del antivirus.

El diagrama muestra todos los procesos que sucederán durante el análisis de un archivo ejecutable; se muestra además, los momentos en los cuales se informará al usuario de lo que sucedió durante el análisis por medio de un informe.

Los primeros dos rectángulos de proceso describen las acciones primordiales durante el análisis: la selección del programa y la búsqueda de código viral. Estos dos rectángulos forman al módulo de detección del antivirus; éste termina por medio de una decisión que evalúa si ha encontrado dentro del archivo el rastro de una infección. Si el resultado del análisis es positivo, el módulo de identificación determina el agente que ha contaminado al programa que se está analizando; en caso contrario se presenta el informe y se termina el análisis.

La parte final del diagrama corresponde al módulo de respuesta, donde se informará de los resultados arrojados por la detección e identificación, independientemente del estado del archivo analizado; los tres rectángulos de proceso representan las acciones posibles de manejo del

archivo: reparación o eliminación del archivo o mantenerlo intacto. Finalmente, se presenta el informe con los hechos encontrados y las acciones realizadas.

Los siguientes apartados presentan de manera más detallada los procesos que se llevan acabo en cada módulo del antivirus; al final de capítulo se muestra el código fuente del programa.

V.2.1. Módulo de Monitoreo e Identificación

Este fragmento del programa incluye los módulos de detección e identificación que en conjunto buscará código malicioso dentro de un archivo ejecutable. Como se explicó en el capítulo IV, existen diferentes técnicas para la detección de virus: detección de cadenas víricas, suma redundante o búsqueda heurística. El diseño de este módulo estará basado en el método de detección de cadenas víricas.

En la figura 5.3 se muestra el diagrama de flujo correspondiente al módulo de detección e identificación. El método utilizado en este antivirus permite agrupar en un solo módulo la detección e identificación de código viral.

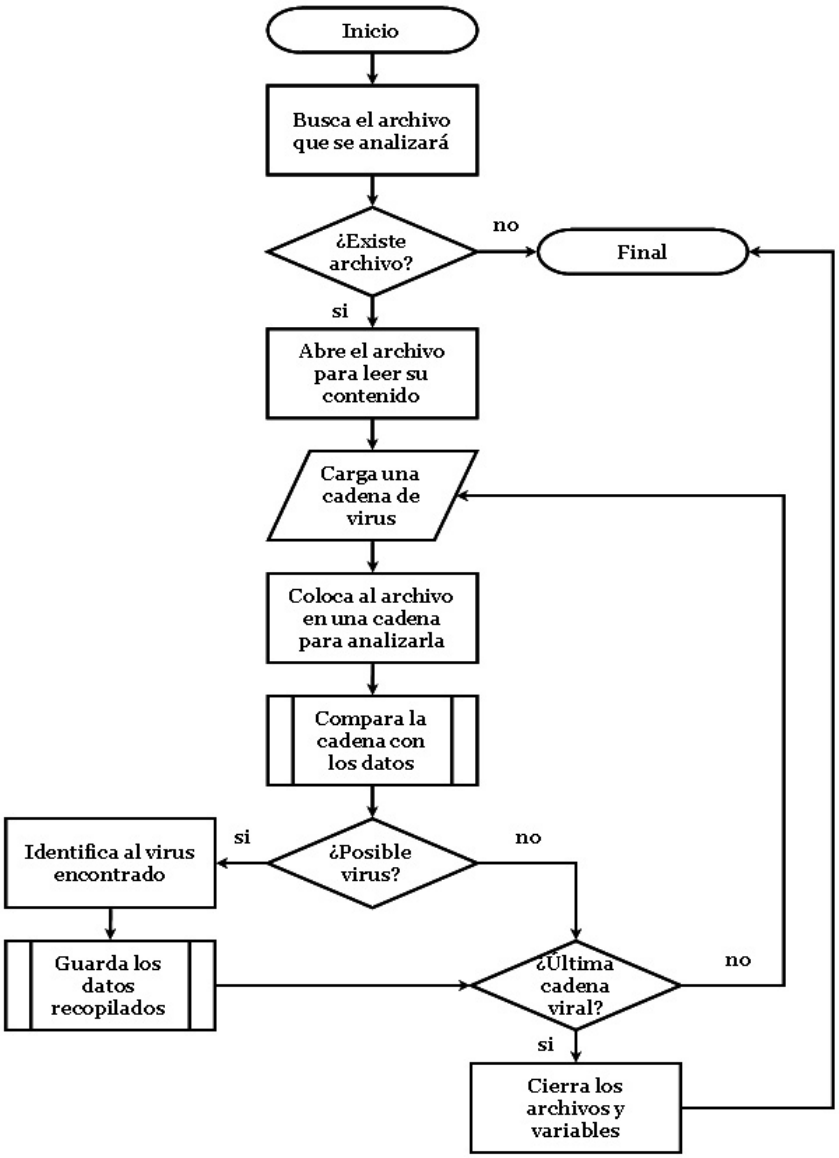


Figura 5.3. Diagrama de flujo del módulo de monitoreo e identificación.

El primer paso que se realizará es la indicación del archivo que se requiere analizar, seguido de su apertura para leer el código; al final de estos procesos se cargarán las definiciones de virus; ambos archivos se abrirán como archivos binarios, de esta forma no se leerán ni compararán caracteres o cantidades, sino Bytes puros.

El antivirus detectará e identificará, en principio, los virus listados en la tabla 5.2. Si se necesitará identificar un rango mayor de códigos virales simplemente serían agregados al archivo de definiciones de cadenas de virus; esto se discutirá con mayor detalle en el capítulo IV.

Virus	*Cadena*	*Efectos*
Cascada 1701	74 0F 8D B7 4D 01 BC 82 06 31 34 31 24 46 4C 75 F8	✓ Los archivos infectados crecen 1701 Bytes. ✓ Las letras caen a la parte inferior de la pantalla. ✓ Se produce un reinicio de manera aleatoria.
BRAIN	A0 06 7C A2 09 7C 8B 0E 07 7C 89 0E 0A 7C E8 57 00	✓ Infecta el sector de arranque de los disquetes. ✓ Cambia la etiqueta de volumen por ©BRAIN. ✓ Es residente en memoria.
Cascada-B	FA 8B EC E8 00 00 5B 81 EB 01 31 2E F6 87 01 2A 01	✓ Los archivos infectados crecen 1704 Bytes. ✓ Se produce un reinicio de manera aleatoria.
Dark Avenger	9D 73 48 2E 3B 1E 08 07 75 3A 85 DB 74 36 E8 AB 02 9D E8 83 00 72 34	✓ Infecta archivos .EXE, .COM y .OVL. ✓ Sobrescribe un sector del disco duro con su código. ✓ Es residente en memoria.
DatacrimeB	2E 8A 07 2E C6 05 22 32 C2 D0 CA 2E 88 07 43 2E	✓ Los programas .EXE Y .COM crecen 1280 Bytes.. ✓ Muestra el mensaje "*DATACRIME VIRUS RELEASED*". ✓ Realiza un formato de disco a bajo nivel.
Flip	FB B8 03 00 E8 1F 00 06 B8 42 00 50 B8 C0 07	✓ Los programas .EXE y .COM son infectados. ✓ Altera el sector de arranque y la tabla de particiones. ✓ Causa daños a archivos y al disco duro.
Telecom	FA 33 C0 8E D0 BC 7C 00 16 1F A1 04 13 48 A3 04 13	✓ Residente en memoria. ✓ Infecta archivos .COM. ✓ Da formato al disco duro.

Tabla 5.2. Virus detectados por el antivirus en desarrollo.

Como se dijo en el capítulo IV, las cadenas de definiciones representan Bytes característicos de virus específicos.

Una vez realizada la apertura de archivos (de análisis y de datos), el siguiente paso será comparar cada Byte del archivo de análisis con los Bytes de la cadena de definiciones, con el fin de verificar si un virus se ha insertado en el código del programa. Esta tarea se llevará a cabo por medio de dos cadenas de Bytes (una correspondiente al archivo y otra a la definición de virus) que se compararán elemento a elemento; en el caso de que se encuentre la cadena de definición insertada en el código del programa, se procederá a identificar el código insertado. Esta tarea es sencilla, debido a que la cadena debe pertenecer exclusivamente a un único virus.

El último proceso de este módulo se encargará de hacer una llamada al módulo de respuesta para que informe al usuario del resultado del análisis.

V.2.2. Módulo de Respuesta

El módulo de respuesta es la rutina más compleja, ya que no sólo engloba las acciones de alertar al usuario sobre los resultados del análisis, también realiza las tareas de reparación o eliminación de archivos en caso de detectarse una infección.

La remoción de código viral de un archivo ejecutable varía dependiendo del virus que se haya detectado; por ejemplo, las medidas de desinfección para el *BRAIN* serán completamente diferentes de las del virus *Darth Vader*; o la desinfección de archivos PE (*portable executable*) diferirán de la de documentos que contengan macros o de ejecutables con extensión .COM. En el caso de este antivirus, se incluye una rutina de desinfección de dos virus pertenecientes a la familia Cascada.

A continuación se muestra en la figura 5.4 el diagrama de flujo del módulo de respuesta.

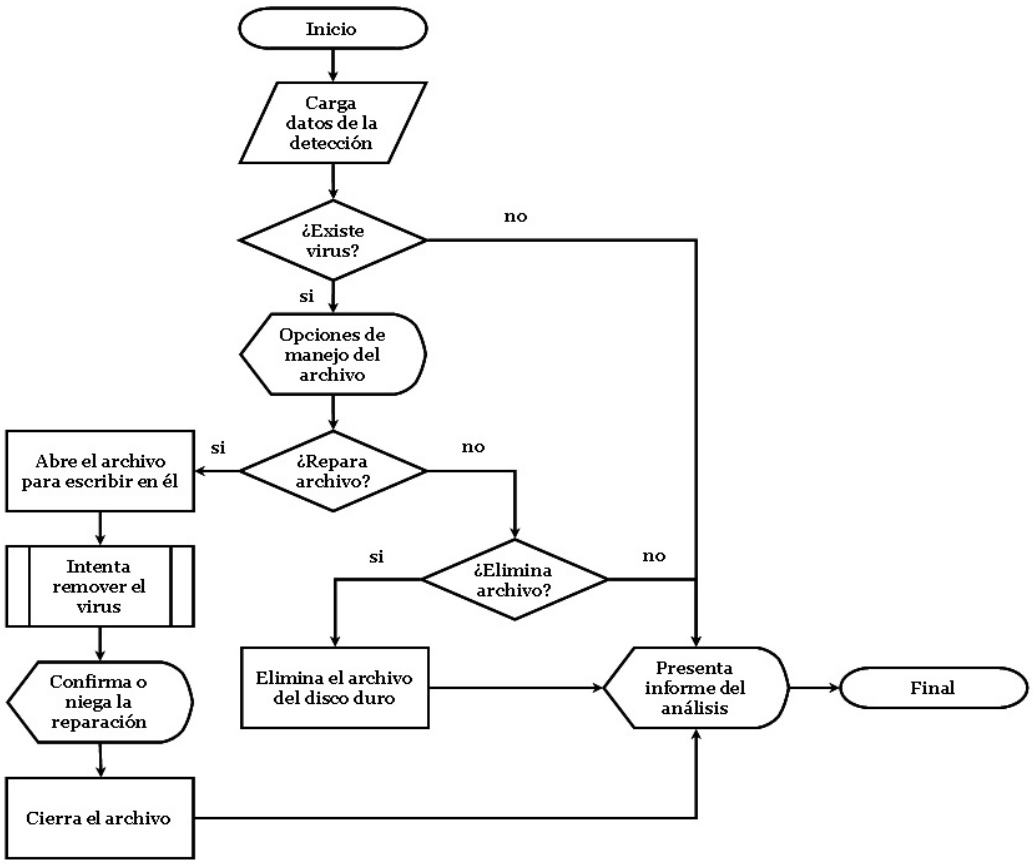


Figura 5.4. Diagrama de flujo del módulo de respuesta.

La primera acción que se ejecuta dentro del módulo es la carga de datos arrojados por el módulo de detección; independientemente de lo que se haya detectado en el módulo anterior, siempre se deberá mostrar un informe al usuario sobre las acciones realizadas durante el análisis.

En el caso de que el informe presente muestras de una infección, se presentarán al usuario opciones para el manejo del archivo infectado: reparar el archivo infectado, eliminarlo o dejarlo intacto; en el primer caso sólo se repararán los archivos que se encuentren infectados con los virus Cascada 1701 y Cascada-B, en el segundo simplemente se eliminará el archivo del directorio donde se encuentre y en el tercero no se hará modificación alguna al archivo.

Cuando se escoja la opción de reparación, el antivirus validará si se puede remover el virus exitosamente; en el caso contrario simplemente mostrará las razones por las cuales la desinfección ha fracasado.

El último paso de este módulo será la presentación del informe con los resultados arrojados por el análisis.

V.3. Código Fuente

En este apartado se mostrará el listado de instrucciones que componen las funciones del antivirus. Para una mejor comprensión se mantendrá el estilo con sangrías y comentarios originales. Al final de cada función se hará una descripción general de las instrucciones utilizadas. El código de las funciones del antivirus se encuentra en el apéndice A.

Existen variables que no están definidas dentro de las funciones de detección y respuesta; éstas variables se han declarado como globales, ya que son utilizadas por otras funciones a parte de las que componen al antivirus. Dichas variables, su tipo y su utilidad dentro del programa se exponen a continuación:

char NomArchivo[100]: Nombre del archivo que se analizará.
unsigned char *StrAnalisis: Puntero que señala a la cadena de bytes del archivo.
bool infeccion: Bandera que indica si el archivo está infectado.
char *virus: Nombre del virus.

V.3.1. Función que Contiene al Módulo de Detección

Esta función corresponde al módulo de detección e identificación. La función comienza con las variables locales que se utilizarán durante el proceso de rastreo de cadenas víricas.

La apertura del archivo que se desea analizar se realiza por medio *streams*. Esto facilita la lectura de archivos binarios, ya que es necesaria la manipulación de bytes para la comparación con las cadenas víricas; si la lectura se hiciese con base en archivos que no son binarios, la comparación sería de caracteres y se llegaría a un error dentro del algoritmo de rastreo.

Una vez abierto el archivo se calcula su tamaño y se reserva memoria suficiente para albergarlo; se realiza una asignación dinámica de memoria para evitar que se desperdicie al momento de analizar archivos con diferentes tamaños. Para obtener una cadena de definición de virus se realiza un procedimiento análogo al de apertura y lectura de archivos; se consumirá un ciclo por cada cadena disponible en el archivo Virus.dat (véase tabla 5.2).

Una vez que las definiciones de virus y el archivo has sido cargados, se procede a compararlos byte a byte. El proceso se realiza por medio de una condición que evalúa si los bytes de la cadena coinciden con algún conjunto de bytes del archivo. En el caso de que se detecte una cadena de definición en un archivo, se activará la bandera correspondiente y se guardarán los nombres del archivo infectado y del virus identificado en una bitácora; ésta le servirá al módulo de respuesta para tomar las medidas necesarias.

El módulo concluirá su trabajo cuando se haya comparado la última cadena de definición. Finalmente, regresará el control a la posición del programa donde fue llamado.

V.3.2. Función que Contiene al Módulo de Respuesta.

En el módulo de respuesta se tiene una variable local que contendrá la acción que el usuario indique que debe realizarse en el caso de una infección.

La primera acción que se realizará será la apertura de la bitácora como un archivo normal. En este caso la bitácora contendrá los nombres del archivo infectado y del virus invasor; al tratarse exclusivamente de texto no será necesario abrirla como un archivo binario para leer los bytes.

Una vez que se haya extraído la información resultante del rastreo, el módulo preguntará al usuario que acción se debe tomar para enfrentar la contingencia. Las opciones son:

- ✓ Desinfectar el archivo. En este caso sólo se llevará a acabo esta acción siempre y cuando el virus detectado sea Cascada 1701 o Cascada B.

- ✓ Borrar el archivo. Eliminará definitivamente al archivo infectado de su ubicación original.
- ✓ Ignorar el archivo. No realizará acción alguna.

Para finalizar, se mostrará al usuario la información necesaria de lo que ha acontecido en el análisis; una vez presentada esta pantalla se regresará al menú inicial.

El diseño de una pieza de software constituye una de las tareas más complicadas que hay en el ámbito computacional. Se necesitan realizar estudios sobre las necesidades naturales del usuario que utilizará el producto final. Se podría pensar que un antivirus tiene un solo objetivo: erradicar los virus; esto es completamente erróneo.

Para diseñar un antivirus se necesitan conocimientos claros acerca de los que los virus hacen; así, se podrá tener una visión más clara de lo que se necesita que el antivirus haga. Además de estudiar las características primordiales en la construcción de una pieza de software, los antivirus necesitan que sus creadores dispongan de una rica noción acerca de los virus y de sus creadores. No hay mejor forma de vencer a un oponente que conociéndolo constantemente; los antivirus modernos deben realizar esa tarea, si no quieren sufrir la misma suerte de sus antepasados.

CAPÍTULO VI

PRUEBAS Y MANTENIMIENTO

VI.1. Pruebas

Durante el desarrollo de cualquier pieza de software se debe considerar un lapso para realizar pruebas. En estas pruebas se verifica si el producto realmente cumple con las especificaciones dadas en el análisis previo, con las expectativas del usuario y también para eliminar cualquier falla que pueda tener antes de lanzar el producto final.

En primer lugar, antes de las pruebas, se hará una especificación pequeña sobre la presentación del producto desarrollado en el capítulo V. A continuación de la presentación se muestran las pruebas realizadas al antivirus; éstas abarcan dos aspectos importantes: detección de código viral y respuesta ante una contingencia viral.

VI.1.1. Presentación del Antivirus

El programa antivirus desarrollado se ha bautizado con el nombre de **Antivirus Interferón** en su versión 1.0. En Medicina, el interferón es una sustancia que una célula segrega cuando es atacada por un virus; ésta es la principal razón para nombrar de este modo al antivirus, ya que hace referencia a la analogía entre virus informáticos y biológicos, mencionada en el capítulo II de este documento.

Al iniciar el programa antivirus se mostrará una pantalla como la que se muestra en la figura 6.1; inmediatamente después aparecerá el menú principal del programa, mostrado en la figura 6.2.

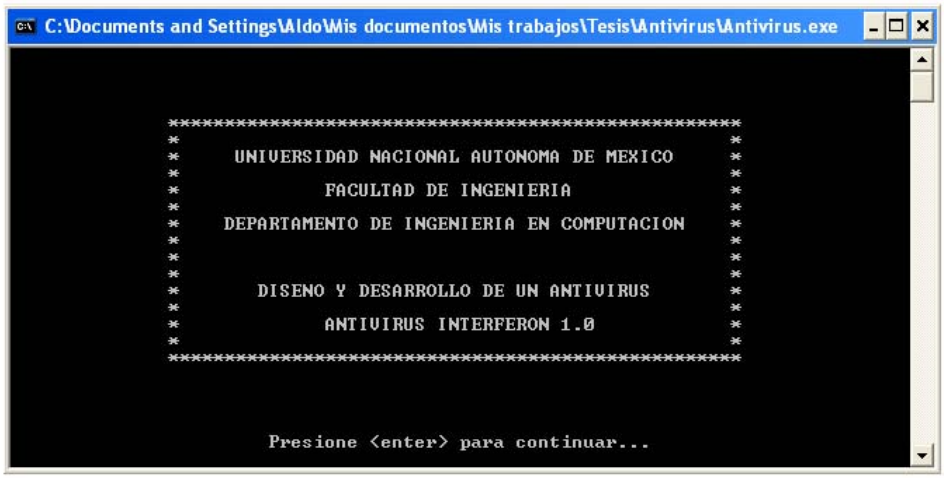


Figura 6.1. Pantalla de presentación de Interferón.

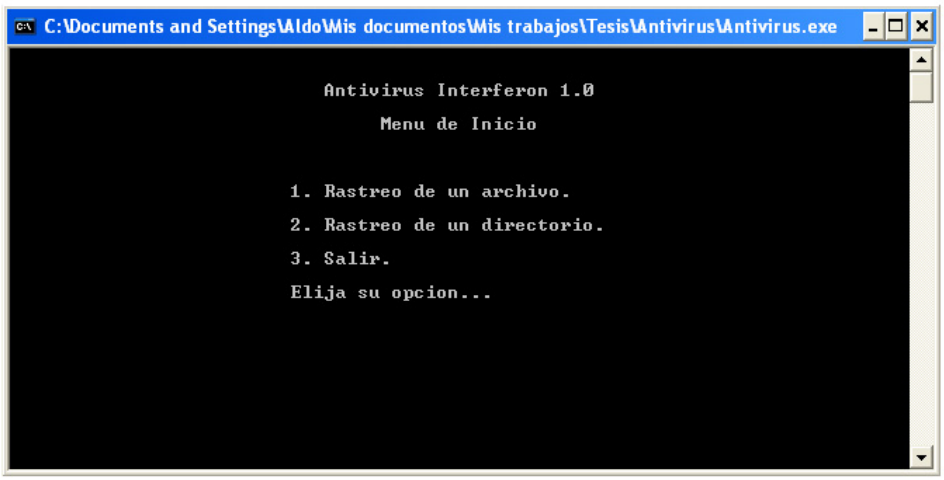


Figura 6.2. Menú principal de Interferón.

Interferón proporciona dos formas de análisis: análisis de un archivo y análisis de una carpeta. La primera opción rastreará código viral en el archivo que le sea especificado desde la línea de comandos, sin importar el tipo de archivo que se le especifique (véase figura 6.3); la segunda opción analizará únicamente los archivos con extensión .EXE de la carpeta especificada, sin inspeccionar sus subcarpetas (figura 6.4).

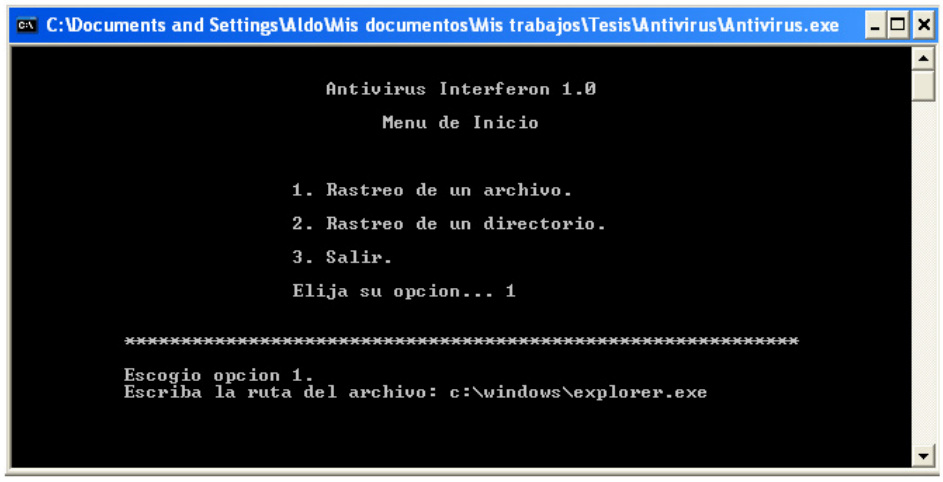


Figura 6.3. Opción 1 Rastreo de un archivo.

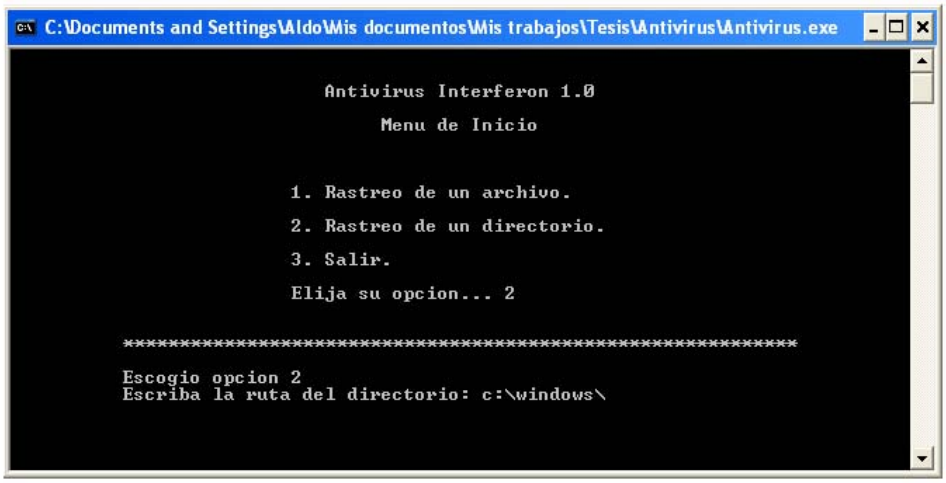
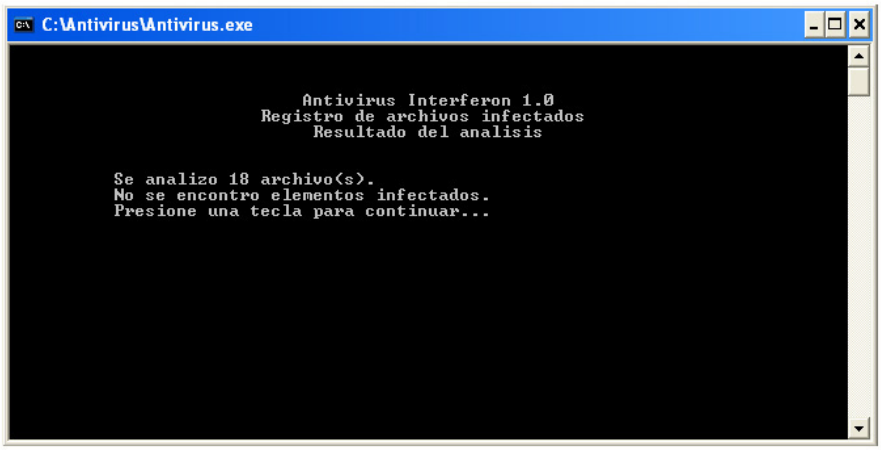


Figura 6.4. Opción 2: rastreo de un directorio.

Al final del análisis (no importando la opción elegida) se mostrará la pantalla de informes, donde se condensará el resultado del rastreo con el número de archivos analizados y, si fuese el caso, el número de archivos contaminados con el nombre del agente contaminante. La figura 6.5 muestra la pantalla a) sin rastro de infección y b) con archivos infectados encontrados.

a)



b)

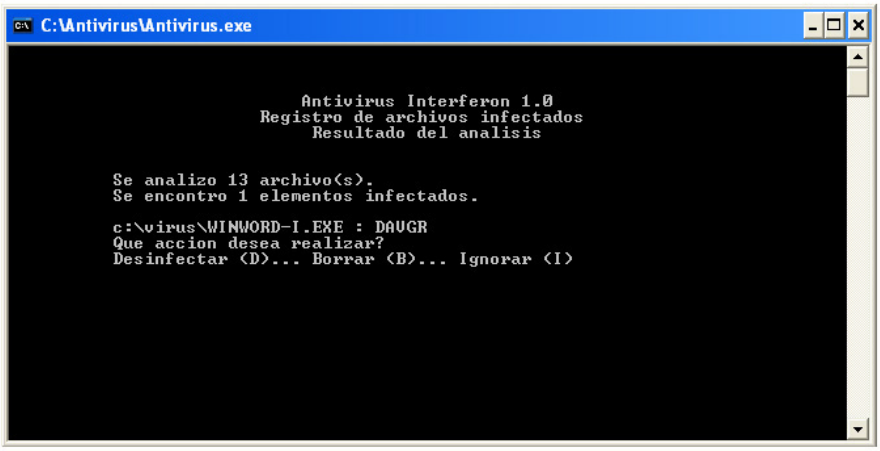


Figura 6.5. Pantalla de resultados del análisis.

Finalmente, se vuelve a la pantalla de menú, ya sea para realizar otro análisis o para salir del programa; la opción 3 termina con la ejecución de Interferón.

VI.1.2. Pruebas Realizadas

Las pruebas que se realizaron para comprobar el buen funcionamiento de Antivirus Interferón 1.0 persiguieron los siguientes objetivos:

- ✓ Comprobar que se realiza el rastreo de cada archivo especificado.
- ✓ Verificar que detecta los códigos maliciosos disponibles en el archivo de definiciones de virus.
- ✓ Reconocer los posibles errores de tipo I y II que puedan presentarse.

- ✓ Comprobar la capacidad de respuesta frente a archivos infectados por los virus Cascada-1701 y Cascada-B.

Estos objetivos fueron fijados para evaluar el comportamiento del antivirus en dos escenarios: un análisis de un ordenador completamente libre de virus y de un ordenador que contiene archivos infectados.

Con base en los objetivos antes mencionados, las pruebas realizadas a Interferón 1.0 se guiaron por las siguientes pautas:

- ✓ Las pruebas se realizaron en una carpeta temporal de experimentación denominada C:\Virus\.
- ✓ La carpeta C:\Virus\ contiene cinco archivos de programas con licencia, cinco archivos de procedencia desconocida y siete archivos infectados con los virus mencionados en la tabla 5.2 del capítulo V.
- ✓ En cada prueba se utilizan archivos diferentes.
- ✓ Las pruebas se dividieron en tres grupos: pruebas sin archivos señuelo, pruebas con archivos señuelo y pruebas de respuesta. Se añadió una prueba extra para comprobar errores de tipo I.
- ✓ El rastreo de archivos se realizó primero archivo por archivo; en segunda instancia se inspeccionó la carpeta completa en una sola ocasión.
- ✓ Después de los dos rastreos mencionados en el punto anterior se desinfectó los archivos contaminados (si pudiesen desinfectarse).
- ✓ Los resultados obtenidos se condensan en una tabla para una mejor observación y comparación.

Bajo los puntos anteriores se procedió a realizar las pruebas en el siguiente orden: pruebas de detección sin archivos señuelo (parcialmente contaminados), pruebas de detección con archivos señuelo y pruebas de respuesta; al finalizar dichas pruebas, se añadió una adicional para demostrar que pueden existir falsos positivos (errores tipo I).

Pruebas sin Archivos Señuelo.

Estas pruebas tienen como fin el asegurar que Interferón cumple con la tarea de examinar los archivos indicados. En esta primera parte de las pruebas se colocó en la carpeta de pruebas cinco programas con licencia y cinco de procedencia desconocida; los diez archivos fueron escogidos al azar. No se tomó en cuenta los archivos contaminados debido a que sólo se requiere comprobar que el antivirus realiza correctamente la inspección de archivos. La tabla 6.1 muestra los archivos analizados; la columna que lleva por título Primera Pasada corresponde al rastreo archivo por archivo; la columna con el encabezado Segunda Pasada contiene los resultados del rastreo de carpeta (no se examinan subcarpetas).

Archivo	*Procedencia*	*1ª Pasada*	*2ª Pasada*
WINWORD.EXE	Microsoft Corporation	Sin infección	Sin infección
WINZIP32.EXE	WinZip Computing	Sin infección	Sin infección
Photopnt.exe	Corel Corporation	Sin infección	Sin infección
Install_Messenger.exe	Microsoft Corporation	Sin infección	Sin infección
QuickTimePlayer.exe	Apple Computer Inc.	Sin infección	Sin infección
dap5.exe	Desconocida	Sin infección	Sin infección
Unvise32qt	Desconocida	Sin infección	Sin infección
Setupmp3towav.exe	Desconocida	Sin infección	Sin infección
3c509x1.exe	Desconocida	Sin infección	Sin infección
Chomp.exe	Desconocida	Sin infección	Sin infección

Tabla 6.1. Análisis de archivos ejecutables limpios.

Como era de esperarse, los cinco archivos provenientes de desarrolladores conocidos no son portadores de virus; mientras que, a los programas desconocidos se les encontró libres de infección, ya que no se detectó alguno de los virus reconocibles por Interferón.

Pruebas con Archivos Señuelo

La siguiente prueba consistió en agregar a la carpeta de pruebas siete archivos infectados. Dentro de estos archivos se encuentra parte de códigos virales; por motivos de seguridad para el equipo en el cual se realizaron las pruebas, los archivos contaminados no contenían el código viral completo. Al igual que en la primera prueba, se realizaron dos análisis: el primero verificó el código archivo por archivo y el segundo arrojó los resultados de examinar la carpeta completa; en las columnas primera y segunda pasada se escribe el nombre del virus encontrado. La tabla 6.2 muestra lo que sucedió durante la prueba.

Archivo	*Procedencia*	*1ª Pasada*	*2ª Pasada*
WINWORD.EXE	Microsoft Corporation	Sin infección	Sin infección
WINZIP32.EXE	WinZip Computing	Sin infección	Sin infección
Photopnt.exe	Corel Corporation	Sin infección	Sin infección
Install_Messenger.exe	Microsoft Corporation	Sin infección	Sin infección
QuickTimePlayer.exe	Apple Computer Inc.	Sin infección	Sin infección
dap5.exe	Desconocida	Sin infección	Sin infección
Unvise32qt	Desconocida	Sin infección	Sin infección
Setupmp3towav.exe	Desconocida	Sin infección	Sin infección
3c509x1.exe	Desconocida	Sin infección	Sin infección
Chomp.exe	Desconocida	Sin infección	Sin infección
Goldflip.exe	Desconocida	Flip	Flip
Minucleo.exe	Desconocida	Brain	Brain
setup.exe	Desconocida	Telecom	Telecom
Chomp-I.exe	Desconocida	Cascada-1701	Cascada-1701
POWERPNT-I.EXE	Microsoft Corporation	Cascada-B	Cascada-B
WINWORD-I.EXE	Microsoft Corporation	Dark Avenger	Dark Avenger
EXCEL.EXE	Microsoft Corporation	DatacrimeB	DatacrimeB

Tabla 6.1. Análisis de archivos ejecutables limpios.

Los siete archivos infectados fueron detectados por Interferón; por lo tanto, significa que se detecta los archivos contaminados con cualquiera de los virus disponibles en el archivo de definición de cadenas.

Pruebas de Respuesta.

Esta evaluación consistió en verificar que Interferón responde correctamente ante una contingencia viral; es decir, que presenta el informe correspondiente con todos los archivos encontrados como contaminados, cuando el archivo puede desinfectarse lo hace correctamente, y cuando no puede eliminarlo o simplemente no lo toma en cuenta. La figura 6.6 muestra el informe que Interferón presentó en la segunda parte del análisis de la prueba con archivos señuelo.

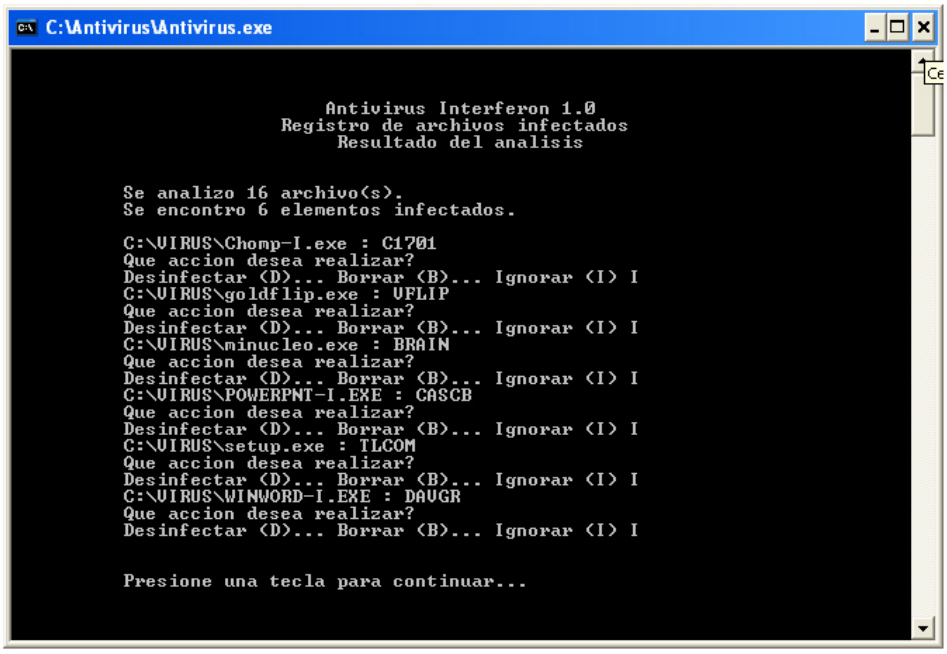


Figura. 6.6. Resultado del análisis de Interferón.

Dentro de la misma figura se leen las acciones que se le indicó a Interferón 1.0 que realizara con los archivos infectados. Después de cada operación se corroboró en la carpeta de pruebas que los archivos fueron manipulados correctamente; se encontró que los que se le indicó eliminar fueron borrados correctamente y los ignorados no fueron tocados. A los archivos que se

le pidió a Interferón desinfectar, primero verificó que el virus contaminante perteneciera a la familia Cascada; al validar esto, se verificó en C:\Virus\Virus.dat que la descontaminación fuera exitosa. El resultado fue satisfactorio.

Errores de tipo I y II.

Como se explicó en el capítulo IV, la técnica de detección heurísticas pueden presentar errores en el momento de rastrear un virus; dichos errores son conocidos como falsos positivo y negativo. En el método de rastreo es muy difícil que puedan presentarse estos errores; sin embargo, puede llegar a darse el caso.

El rastreo se basa en la búsqueda de segmentos de código viral dentro de archivos ejecutables; dichos fragmentos son propios de los virus y, en teoría, ningún otro programa debería contenerlos – a menos que esté infectado –. Sin embargo, no es imposible que algún archivo limpio (ejecutable o no) pueda poseer la cadena característica de un virus. El ejemplo inmediato, y más claro, de esta situación es el archivo de definiciones de virus; dentro de este archivo se encuentran todas las cadenas características de los códigos malignos conocidos que el antivirus puede detectar. Esto quiere decir que si se somete a este archivo a un rastreo de virus se obtendrá un resultado de búsqueda positivo (si es que no se le ha especificado al antivirus que éste es benigno).

Para realizar esta prueba se le indicó a Interferón 1.0 la ruta del archivo de definiciones de virus por medio de la opción 1 del menú principal. La figura 6.6 muestra el resultado obtenido de proporcionarle al antivirus la ruta C:\Antivirus\Virus.dat.

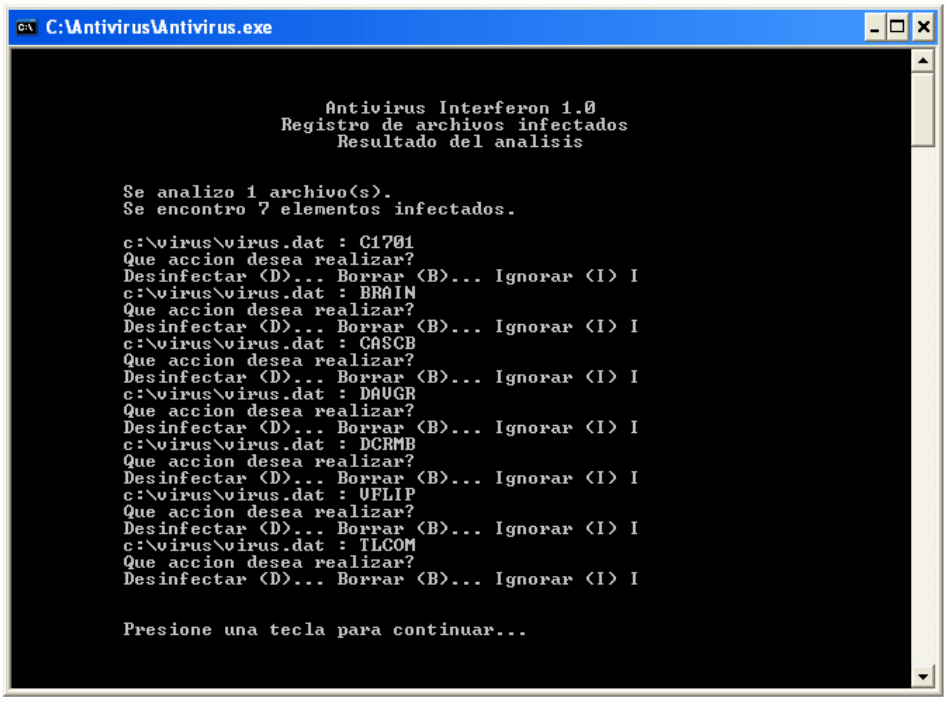


Figura 6.6. Falso positivo producido por el análisis del archivo Virus.dat.

Como se observa, Virus.dat da un resultado positivo para los siete virus que Interferón puede detectar; esto es coherente, ya que el archivo contiene las cadenas de definición de dichos códigos maliciosos.

En el método de rastreo, el falso negativo sólo podrá presentarse si en el archivo de definiciones no se ha incluido una muestra del virus que no se detectó. Esto se debe a que el antivirus siempre buscará cada cadena disponible; sencillamente, si no encuentra la firma de un virus dentro de un archivo es porque este no está contaminado.

VI.2. Mantenimiento

Todo aquello creado por el ser humano necesita atención constante para que siga cumpliendo la tarea a la que fue destinado. El mantenimiento se encarga de corregir defectos iniciales, mejorar el diseño, la eficiencia del producto, entre otras cosas; así, se prolonga su vida útil.

El software no es la excepción. Para que un programa siga cumpliendo las tareas para las que fue creado necesita mantenimiento constante, basado en actualizaciones (*updates*) y promociones (*upgrades*), máxime si se trata de un antivirus. La firma de seguridad alemana Test dice que diariamente se descubren entre 70 y 100 amenazas nuevas, muchas de ellas variantes de las ya conocidas. Esto muestra la importante necesidad de mantener al día el sistema de protección del ordenador para evitar ataques de seguridad.

Dentro ese mantenimiento que se le debe dar a un antivirus (y en general, a cualquier pieza de software) se debe considerar las nuevas especificaciones dadas por los usuarios que hacen uso del programa, por los desarrolladores de sistemas operativos y por los desarrolladores de nueva tecnología informática. Los detalles anteriores, además de la constante evolución de las amenazas, hacen indispensable la periódica actualización del antivirus, para evitar que quede reducido a un obsoleto programa de museo.

VI.2.1. Actualizaciones (*Updates*).

Optimización de la Programación del Algoritmo de Búsqueda

Dentro de las primeras actualizaciones que deben realizarse se encuentra la optimización del código. En este momento, el antivirus realiza las tareas encomendadas de manera correcta; sin embargo, la eficiencia y rapidez con la cual lo desarrolla no es la óptima.

Debido a los pocos virus contenidos en las definiciones, la manera en que Interferón examina los archivos es considerablemente adecuada; pero, en el momento que se incremente el número de virus que se pueden reconocer, el análisis de cada archivo tardará más. Este es uno de los inconvenientes del diseño inicial del antivirus, ya que el procesamiento de la información debe ser lo más rápido posible. La recomendación sugerida por alumnos de la Facultad de Ingeniería es la programación de hilos para el análisis; es decir, que se realice el rastreo con todas las cadenas virales en un solo ciclo (carga en paralelo), en lugar de realizarlo una por cada ciclo, como está implementado (carga en serie).

Actualización del Archivo de Definiciones

Como todos los antivirus, se necesita que Interferón reconozca todos los códigos virales que se vayan descubriendo. Esta es una tarea difícil, ya que se necesitan los códigos completos de los virus que se requieran agregar, analizarlos e incorporar la solución al diseño del antivirus. Esto hace que el punto anterior sea importantísimo, ya que al incrementarse el número de códigos virales detectados se incrementa el tiempo de rastreo.

VI.2.2. Promoción (*Upgrade*).

Cambio de Interfase

Interferón 1.0 tiene una interfase basada en aplicación de consola MS-DOS. Para mayor comodidad del usuario se tiene pensado realizar la primera promoción al incorporar una interfase gráfica, que sea acorde con el ambiente intuitivo y fácil de manejar que presentan los sistemas operativos actuales. Así pues, cualquier tipo de usuario puede hacer uso del antivirus.

Expansión del Rango de Búsqueda

Para que un antivirus detector cumpla completamente con sus funciones, es necesario que tenga la capacidad de examinar no sólo una carpeta, sino un sistema completo. Interferón realiza sus tareas únicamente dentro de una capeta, excluyendo todos los subdirectorios contenidos en ella; esto presenta una deficiencia en su capacidad de análisis, ya que se necesitaría examinar carpeta por carpeta para verificar que el sistema está libre de virus.

Otro de los puntos importantes del rastreo es verificar otras unidades de almacenamiento – no sólo la unidad de disco actual – junto con sus sectores de arranque; esto permitirá que el antivirus tenga un margen más amplio para resguardar la seguridad del equipo informático al cual protege.

Implementación de un Complemento Protector

Como se dijo en el capítulo IV, es recomendable que un antivirus esté compuesto por una combinación de protección y detección; de esta manera se colocará una barrera fortalecida entre los códigos maliciosos y la información que pretenden destruir.

Uno de los pasos importantes que debe dar Interferón en su evolución es realizar la tarea de protección. Esto implica no sólo conocer perfectamente la arquitectura del procesador sobre el cual trabajará, sino también el funcionamiento completo del sistema operativo que actuará como base del programa antivirus. Esta es una promoción ambiciosa, más no inalcanzable.

Uno de los aspectos más importantes dentro de la seguridad informática es la constante certificación y actualización de los sistemas de seguridad; no se puede pensar que basta con establecer las políticas y herramientas adecuadas. Se debe tomar en cuenta que día con día surgen nuevas y más peligrosas amenazas; un sistema que no ha sido, probado, certificado y actualizado adecuadamente corre el riesgo de ser blanco de ataques.

Los antivirus, como parte de las herramientas actuales de seguridad, necesitan someterse a pruebas, actualizarse y certificarse cada determinado tiempo; la principal razón de llevar a cabo estas acciones es la constante aparición de nuevos códigos maliciosos, ya sea actualizaciones de los ya existentes o programas completamente nuevos. Interferón tiene un camino largo que recorrer; dentro de esa ruta, deberá detenerse por momentos y cubrir con todos los requerimientos que se exigen a los antivirus más reconocidos y eficaces del mundo; así, podrá continuar la tarea para la cual fue creado: combatir a los virus informáticos.

CONCLUSIONES

El tema principal de este documento es conocer una de las amenazas más comunes que atentan contra el bienestar de la información: los virus. Y con él se espera que este trabajo colabore para evitar la creación y propagación de los códigos maliciosos.

Existe cierta complicación en la creación de herramientas encaminadas a la seguridad de la información, ya que, como se ha dicho a lo largo del trabajo, se debe comenzar con un análisis exhaustivo de las amenazas que pretenden destruir la información. Los conocimientos adquiridos en el desarrollo de este trabajo permiten reconocer una de las amenazas más importantes en la Informática: los virus; se destaca su origen, evolución, modo de operación y efectos que causan. Además, se hace una reseña de las herramientas utilizadas para contrarrestar sus efectos y las posibles deficiencias que éstas tengan. Todo esto se encaminó a la creación de una herramienta para combatir a los virus: el antivirus Interferón.

La investigación sobre los virus informáticos es muy amplia; desafortunadamente, en México, se tiene poca bibliografía disponible para el público en general. La principal causa de las afecciones por virus informáticos es el desconocimiento; este documento pone a disposición de los usuarios los conocimientos básicos sobre el operar característico de los virus informáticos, así como una visión general de lo que un antivirus hace y cuales son las deficiencias más frecuentes en la utilización de estas herramientas. La información contenida en este trabajo de tesis permitirá a estudiantes, profesores, profesionales de la informática y al público en general adentrarse dentro del pequeño mundo que es la virología informática y sobre todo tener presente la amenaza que puede representar un pequeño fragmento de código.

La magnitud del ataque de un virus no se cuantifica por la destrucción que cause, sino por su propagación a través de los ordenadores; esto quiere decir, que ni el medio de transmisión (red, disquete, CD, DVD, etc.) ni la plataforma de trabajo son un impedimento para crear y transmitir virus.

Esto hace que las herramientas que se diseñen para el combate de los virus, presenten los menos errores posibles y puedan ser modificadas conforme evolucionan los algoritmos y las técnicas de programación. Se debe tomar en cuenta que los antivirus son una herramienta de ayuda, no una solución; de nada sirve mantener un antivirus eficiente, si las políticas de seguridad en contra de los virus no son respetadas o si la herramienta es utilizada incorrectamente. A lo largo de este trabajo se exponen los diversos tipos de técnicas de detección y protección que utilizan los antivirus modernos; la intención es documentar de las virtudes, los defectos, las ventajas y desventajas que son inherentes a la programación del software antivirus.

La construcción de un antivirus no es sencilla; la forma en la que se deben crear estas piezas de software es bastante compleja y no es posible abarcar todos los aspectos necesarios para la elaboración de dicho software. Se considera que el antivirus creado en el desarrollo de este trabajo cumple con las expectativas propuestas al inicio del proyecto; con un esquema sencillo de funcionamiento se pretende que este antivirus ejemplifique las funciones básicas de búsqueda y eliminación de virus, así como una explicación sencilla de la técnica de rastreo, la más empleada en varios productos contra códigos maliciosos. Sin embargo, el trabajo aún no se puede dar por concluido; solamente se tiene la base para la construcción de una herramienta. Con el tiempo, la creación de varios complementos permitirán a Interferón estar a la vanguardia en lo que se refiere a las herramientas de seguridad; y este documento permitirá también un crecimiento en el desarrollo de investigadores de seguridad y virología informáticas.

El primer paso para resguardar la información es conocer las amenazas que atentan contra ella; si se tiene conocimiento sobre lo que se está combatiendo, será más fácil prevenir sus acciones y contrarrestar sus efectos. No existe enemigo pequeño, y un virus dista mucho de serlo; la guerra contra los códigos malignos no terminará en un futuro próximo, pero sí se acerca rápidamente porque el adversario ya ha sido reconocido.

APÉNDICE A

CÓDIGO FUENTE

```
//Función que contiene al módulo de detección e identificación.
void deteccion(void) {
    char virus[5] = ""; //Nombre del virus.
    unsigned char StrDatos[23] = "";  //Puntero que señala a la cadena de definiciones cargada.
    long tamaniochar = 0;  //Tamaño de la cadena vírica.
    long cont2 = 0;    //Variable local de control.
    unsigned char aux;   //Variable auxiliar.
    FILE *Bitacora; //Puntero al archivo de registro.

    infeccion = false;

    //PROCESO PARA ABRIR EL ARCHIVO QUE SE ANALIZARÁ.
    //Búsqueda y apertura del archivo.
    ifstream fanalisis(NomArchivo, ios::in | ios::binary);

    //Valida si el archivo existe; en caso contrario, regresa al menú principal.
    if(fanalisis) {
        NumArchivos++;

        //PROCESO PARA LEER EL ARCHIVO.
        //Calcula el tamaño del archivo.
        fanalisis.seekg(0, ios::end);
        tamanio = fanalisis.tellg();
        fanalisis.seekg(0, ios::beg);

        //Reserva el espacio suficiente en memoria que albergará a una cadena de Bytes
        //del tamaño del archivo.
        StrAnalisis = new unsigned char[tamanio];

        //Copia en StrAnalisis cada byte del archivo.
        while(cont < tamanio) {
            fanalisis.read(reinterpret_cast<char *>(StrAnalisis + cont), sizeof(char));
            cont++;
        }
```

```
//PROCESO PARA ABRIR Y LEER LAS CADENAS DE DEFINICIÓN DE VIRUS.
//Abre el archivo de definiciones.
ifstream fdatos("Virus.dat", ios::in | ios::binary);

//Valida si el archivo de definiciones existe.
if(fdatos) {
    for(;;) {

        for(cont = 0; cont <= tamaniochar; cont++) {
            StrDatos[cont] = '\0';
            fdatos.read(reinterpret_cast<char *>(&aux), sizeof(char));
            if(cont < 5) virus[cont] = aux;
            else {
                if(aux == 0x20) break;
                StrDatos[cont - 5] = aux;
            }
            tamaniochar++;
        }
        if(fdatos.eof()) break;
        cont = 0;
        cont2 = 0;

        //PROCESO DE RASTREO BASADO EN LA COMPARACIÓN DE BYTES.
        //Ciclo donde se compara, byte a byte, el archivo y la cadena de definición.
        while(cont < tamanio) {

            //Compara el byte n-ésimo del archivo con el byte m-ésimo de la cadena.
            if(*(StrAnalisis + cont) == StrDatos[cont2]) {

                //Evalua si se ha llegado al final de la cadena.
                if(cont2 == tamaniochar - 6) {

                    //Si el archivo está contaminado se activa la bandera que señala
                    //la infección.
                    infeccion = true;

                    //Añade a la bitácora los nombres del archivo infectado y del
                    //virus encontrado en él.
                    Bitacora = fopen("Bitacora.log", "a");
                    fprintf(Bitacora, "\n%s\n", NomArchivo);
                    for(cont = 0; cont < 5; cont++)
```

```
                            fprintf(Bitacora, "%c", virus[cont]);
                        fclose(Bitacora);
                        NumInfectados++;
                        break;
                    }
                    cont2++;
                }
                else cont2 = 0;
                cont++;
            }

            //Se liberan las variables para cargar la siguiente cadena de definición.
            tamaniochar = 0;
        }
        fdatos.close();
    }
    else
        cout << "          El archivo de definiciones no existe." << endl;
    delete[] StrAnalisis;
}
else
    cout << "          No se encontro ningun archivo." << endl;
return;
}

//Función que contiene al módulo de respuesta.
void respuesta(void) {
    char CadenaV[5];
    char opcionresp;    //Opción de manejo de archivo.

    system("cls");
    cout << endl << endl << endl;
    cont = 0;
    cout << "                            Antivirus Interferon 1.0" << endl;
    cout << "                         Registro de archivos infectados" << endl;
    cout << "                              Resultado del analisis" << endl << endl << endl;
    cout << "          Se analizo " << NumArchivos << " archivo(s).";

    //Apertura de la bitácora de resultados en caso de una infección
    if(infeccion) {
        ifstream fbitacora("Bitacora.log");
```

```
cout << endl << "          Se encontro " << NumInfectados << " elementos infectados."
     << endl << endl;

//PROCESO PARA ESCRIBIR EN EL ARCHIVO.
//Obtiene los nombres de los archivos infectados y de los virus contaminantes.
fbitacora.getline(CadenaV, 10);
do {
    cont = 0;
    strset(NomArchivo, '\0');
    strset(CadenaV, '\0');
    fbitacora.getline(NomArchivo, 200);
    fbitacora.getline(CadenaV, 10);
    cout << "          " << NomArchivo << " : " << CadenaV << endl;

    //Presenta las opciones para manejar un archivo infectado.
    cout << "          Que accion desea realizar?"
         << endl << "          Desinfectar (D)... Borrar (B)... Ignorar (I) ";
    cin >> opcionresp;
    switch(opcionresp) {
        case 'd':
        case 'D':

            //PROCESO PARA ELIMINAR EL VIRUS.
            //En caso de que el agente sea el virus Cascada-1701 o
            //Cascada-B, se procederá a la descontaminar el archivo.
            if(!strcmp(CadenaV, "C1701") || !strcmp(CadenaV, "CASCB")) {

                //Se lee el archivo contaminado y se calcula su tamaño.
                ifstream finfectado(NomArchivo, ios::in | ios::binary);
                finfectado.seekg(0, ios::end);
                tamanio = finfectado.tellg();
                finfectado.seekg(0, ios::beg);
                StrAnalisis = (unsigned char *) malloc(tamanio*sizeof(unsigned char));
                cont = 0;
                while(cont < tamanio - 1701) {
                    finfectado.read(reinterpret_cast<char *>(StrAnalisis + cont), sizeof(char));
                    cont++;
                }
                finfectado.close();

                //Se abre un nuevoa rchivo que contendrá el código
```

```
            //desinfectado del archivo.
            ofstream freparado(NomArchivo, ios::out | ios::binary);
            cont = 0;
            if(!strcmp(CadenaV, "C1701"))
                while(cont < tamanio - 1701) {
                    freparado.write(reinterpret_cast<char *>(StrAnalisis + cont), sizeof(char));
                    cont++;
                }
            else
                while(cont < tamanio - 1704) {
                    freparado.write(reinterpret_cast<char *>(StrAnalisis + cont), sizeof(char));
                    cont++;
                }
            freparado.close();
            free(StrAnalisis);

            //Anuncia la desinfección exitosa.
            cout << "          Archivo desinfectado." << endl;
        }
        else
            cout << "          Imposible desinfectar." << endl;
        break;
    case 'b':
    case 'B':

        //PROCESO PARA ELIMINAR EL ARCHIVO INFECTADO.
        //Elimina el archivo infectado.
        unlink(NomArchivo);
        cout << "          Archivo eliminado." << endl;
        break;
    case 'i':
    case 'I':

        //No realiza ninguna acción y termina el análisis en curso.
        break;
    default:
        cout << "          Accion invalida." << endl;
        break;
}
cin.get();
strset(NomArchivo, '\0');
```

```
        } while(!fbitacora.eof());
        fbitacora.close();
        cout << endl << endl;
    }
    else
        cout << endl << "          No se encontro elementos infectados." << endl;
    NumArchivos = 0;
    NumInfectados = 0;
    return;
}
```

APÉNDICE B

ARTÍCULOS PERIODÍSTICOS

Martes, 24 de enero de 2006 – 11:55 GMT

Cumpleaños infeliz: 20 años de virus

Este mes se cumplen 20 años de la aparición del primer virus informático.

Fue durante las primeras semanas del año 1986 cuando se descubrió que el primer virus, llamado Brain, se estaba diseminando libremente.

Aunque obtuvo fama por ser el primer virus de su tipo, Brain no llegó a alcanzar una gran expansión, ya que sólo se podía transmitir a través de un *floppy disk* o disquete "contagiado", utilizado por varios usuarios.

El virus se contagiaba cada vez que el *floppy disk* era utilizado en una computadora.

Ahora, 20 años después de su aparición, más de 150.000 programas malignos amenazan la salud de los ordenadores.

Propagación lenta

No hay acuerdo sobre los orígenes del virus Brain; algunos creen que fue creado por una compañía informática paquistaní para ayudar a proteger los programas que ellos mismos creaban y vendían.

Lo descubrieron en enero de 1986, pero probablemente llevaba cierto tiempo rondando las computadoras, ya que se propagaba mediante un método relativamente lento.

Brain, ahora ya extinguido, fue clasificado como un virus de tipo *boot-sector*, por la manera en que infectaba y se instalaba en el sector de arranque (o *boot sector*) de un disquete.

Al esconderse en esta región, el virus se contagiaba cada vez que el *floppy disk* era utilizado en una computadora.

En la actualidad, más de 150.000 códigos malignos amenazan a los ordenadores.

Aunque Brain fue el primer virus que afectaba a las computadoras, no fue el primer programa informático maligno. Ese honor le correspondió al llamado Elk Cloner, producido por Richard Skrenta, que afectaba a los ordenadores Apple II.

Evolución

El término virus informático data de 1984, y se le atribuye al científico estadounidense Fred Cohen.

Desde la aparición de Brain, el número de virus y otros programas malignos en circulación ha crecido enormemente.

El sistema operativo Windows de Microsoft es el objetivo favorito de los creadores de virus.

El crecimiento de las redes informáticas locales, el correo electrónico e Internet, ha alimentado este aumento, y ahora en sólo dos horas un nuevo virus puede infectar a miles de usuarios.

“El cambio más significativo es la evolución desde la gente que creaba virus como pasatiempo, hacia la formación de grupos más organizados de delincuencia que tienen por objetivo beneficios financieros.”

Mikko Hypponen

Además, hay muchos tipos diferentes de “infecciones”, que utilizan técnicas de contagio distintas para afectar a las computadoras.

Las razones que llevan a ciertos usuarios a crear virus también han cambiado.

“El cambio más significativo es la evolución desde la gente que creaba virus como pasatiempo, hacia la formación de grupos más organizados de delincuencia que tienen por objetivo beneficios financieros”, explicó Mikko Hypponen, jefe de investigaciones en la compañía finlandesa de antivirus F-Secure.

Según cifras publicadas por el FBI esta semana, el 84% de las empresas estadounidenses fueron atacadas por virus, *spy-ware* y otros programas malignos durante el 2005.

Como promedio, las compañías se gastan unos US$ 24.000 en protegerse o recuperarse de estos ataques, según los datos del FBI.

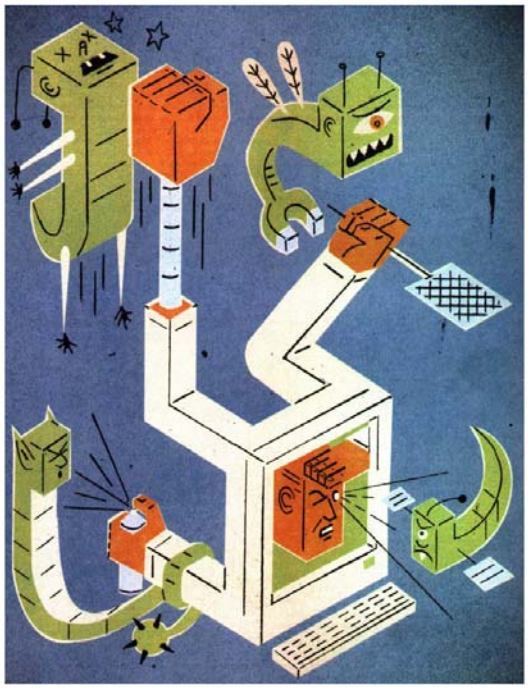
Condensado de BBCMUNDO.com. www.bbc.co.uk/home/i/. Enero de 2006.

Los programas antivirus de hoy no tienen problemas para detener a los intrusos conocidos, pero ¿estará usted protegido contra las amenazas que no se conocen? Las pruebas que realizamos a diez contrincantes revelan una nueva Mejor Compra.

Las nuevas armas contra LOS VIRUS

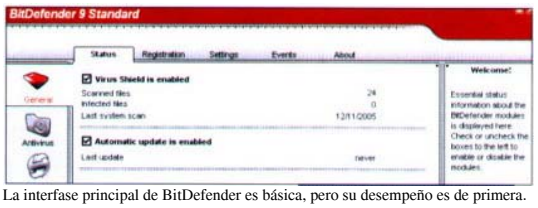
POR TONY BRADLEY. ILUSTRACIONES DAVID PLUNKERT.

Tenemos buenas y malas noticias acerca de la guerra que se libra contra los virus informáticos. La buena es que todos los productos antivirus que probamos para este número fueron 100 por ciento eficaces en identificar y bloquear las amenazas de seguridad reconocidas. La mala noticia es que estas herramientas todavía no le protegen completamente de las nuevas amenazas; y éstas abundan. Test (find.pcworld.com/51168), la firma de seguridad alemana con la que PC World colaboró para esta historia, dice que todos los días se descubren entre 70 y 100 amenazas nuevas. Aunque muchas de ellas son variantes de amenazas existentes, si hay que esperar aunque sea unas horas para que el fabricante de su programa antivirus publique soluciones para ellas; mientras tanto, su computadora quedará expuesta a la infección. Además, los virus no son el único problema. Los creadores de códigos maliciosos también envían gusanos – que no necesitan un archivo anfitrión para multiplicarse – y otros programas destructivos, como los caballos de Troya, en archivos adjuntos al correo electrónico. “Al autor del Bagle le gusta mucho hacer eso”, dice el investigador de seguridad Joe Stewart de LURHQ, una compañía que ofrece servicios de consultoría y administración de seguridad. Debido a estos peligros, es importante que la aplicación antivirus que usted tenga sea capaz de reconocer y eliminar no solamente los virus sino también otros tipos de amenazas.



LAS HERRAMIENTAS ANTIVIRUS CONTRAATACAN

Las compañías de software antivirus adaptan y mejoran sus productos de diversas maneras. Ahora casi siempre incluyen aplicaciones antivirus tradicionales con otros componentes de seguridad, como herramientas contra programas espías y *firewalls*, para brindarle una protección más completa; en algunos casos esta funcionalidad adicional forma parte del propio producto antivirus. Las compañías también están acortando el tiempo que tardan en publicar actualizaciones de las firmas, que las herramientas antivirus individuales descargan y utilizan para reconocer y destruir las amenazas recién identificadas.



La interfase principal de BitDefender es básica, pero su desempeño es de primera.

Además, los vendedores están refinando la heurística de sus productos, es decir, los algoritmos matemáticos que pueden reconocer las nuevas amenazas a la seguridad basándose en su similitud con segmentos de código que han sido identificados anteriormente como dañinos. "El análisis heurístico de los motores de software antivirus ha madurado en los últimos años, con mejor detección y menos falsas alarmas", dice Douglas Schweitzer, autor del libro *Securing the Network From Malicious Code: A Complete Guide to Defending Against Viruses, Worms, and Troyans* (Protegiendo la red contra el código malicioso: Una guía completa para defenderse de virus, gusanos y caballos de Troya). En las falsas alarmas (o en los positivos falsos) una aplicación identifica erróneamente a un archivo como un programa malicioso. En el mejor de los casos, esta equivocación haría perder el tiempo a los usuarios, pero en el peor de los casos le llevaría a borrar archivos benignos.

Las compañías también están usando la detección basada en el comportamiento para combatir nuevas amenazas que sus productos todavía no pueden reconocer con las actualizaciones de firmas. Esta tecnología vigila las partes de su sistema que un archivo malicioso podría atacar, marca el comportamiento sospechoso y lo detiene. La desventaja que tiene este enfoque es que el programa malicioso debe estar ya activo en su computadora para poder detectar su comportamiento. Por esta razón, la detección basada en el comportamiento funciona mejor como una capa de protección suplementaria detrás del motor de detección de virus, que idealmente elimina la amenaza antes de que ésta se materialice.

APLICACIONES AUTÓNOMAS, CONJUNTOS Y HERRAMIENTAS GRATUITAS

Con estas tendencias en mente, PC World se dispuso a ver cuál de los productos antivirus actuales le protege mejor de los programas maliciosos conocidos y desconocidos. Probamos diez productos cuyo precio abarca desde gratuitos hasta US$75 Para que se compitiera en condiciones parejas, probamos aplicaciones antivirus autónomas si estaban disponibles, y sólo los componentes antivirus de los conjuntos integrados o suites que ofrecen otras funciones, como la protección contra programas espías y los *firewalls*. Si hubiéramos probado esos conjuntos manteniendo activos los componentes que no están dirigidos a los virus, les habríamos dado una ventaja injusta sobre los programas antivirus autónomos, a los cuales se les pueden agregar (y le recomendamos que lo haga) herramientas de *firewall* y de contraespionaje.

En nuestro grupo de prueba, el Avast Home Edition 4.6 de Alwil Sofhvare, el AntiVir PersonalEdition Classic 6.32 y el AVG Free Edition 7.1 de Grisoft son programas autónomos que no cuestan nada. Anti-Virus 2006 de F-Secure, Kaspersky Anti-Virus Personal 5.0 de Kaspersky Labs, VirusScan 2006 de McAfee y BitDefender 9 Standard son aplicaciones autónomas pagadas. Panda Titanium 2006 Antivirus + Antispyware de Panda Software y Norton AntiVirus 2006 de Symantec incluyen herramientas contra programas espías. Trend Micro vende su herramienta antivirus sólo como parte del PC-cillin Internet Security Suite 2006.

PC-cillin de Trend Micro pone mucha información en una pantalla bien diseñada.

Un producto que no calificamos fue el ZoneAlarm Antivirus de Zone Labs, ganador del premio Clase Mundial de nuestra edición estadounidense para esa categoría en el 2005. El programa combina el motor Vet Antivirus de Computer Associates con el *firewall* para redes de Zone Labs y OSFirewall, una tecnología de prevención, basada en el comportamiento, que detecta cualquier actividad sospechosa en el sistema. AV-Test sí evaluó el motor analizador de Computer Associates, que se desempeñó pobremente y fue el que más se demoró en publicar actualizaciones de firmas para las nuevas amenazas. Sin embargo, para este artículo AV-Test no pudo evaluar la eficacia en la prevención de programas maliciosos basada en el comportamiento que ofrece Zone Labs. Ponerlo a prueba con la colección de programas dañinos de AV-Test hubiera tomado meses, ya que cada archive debe estar activo en el sistema de prueba. Como OSFirewall forma parte integral del producto de Zone Alarm, excluimos el producto completo (el producto de Panda, que sí calificamos, también usa la detección basada en el comportamiento).

CÓMO REALIZAMOS LAS PRUEBAS

En general, AV-Test realizó cinco pruebas. Primero, determinó si los productos podían detectar 1518 muestras "salvajes" de programa maliciosos, una lista publicada de virus y otras amenazas que han sido identificadas como activas en circulación pública por la *WildList Organization*.

Segundo, probó la capacidad de los programas para detectar las amenazas que no están incluidas en la *WildList* usando su propia colección (o zoológico) de 136.250 programas clandestinos, caballos de Troya y robots (también conocidos como zombis). El zoológico incluye programas maliciosos recopilados de clientes, revistas de computación y potes de miel, que son servidores conectados a la Internet que los investigadores preparan para atraer a los programas maliciosos.

Debido a la forma en que la *WildList* se publica (frecuentemente no está al día e intencionalmente excluye las amenazas que no se replican a sí mismos, como los caballos de Troya y el software clandestino), el zoológico de programas maliciosos de AV-Test representa un buen complemento.

COMPARACIÓN DE CARACTERÍSTICAS

Centro de PRUEBAS

La detección de programas maliciosos de BitDefender **ES INMEJORABLE**

Esta herramienta antivirus de bajo costo fue la que se desempeñó mejor en nuestras pruebas heurísticas y detectó la mayor variedad de programas maliciosos

	Software Antivirus	*Clasif. PCW*	*Rendimiento*	*Virus en la WildList*	*Pruebas de zoo de AV-Test*	*Detección heurística con firmas de un mes de creada*	*Detección heurística con firmas de dos meses de creada*	*Facilidad de uso*	*Conclusión*
1	BitDefender 9 Standard $480 con IVA find.pcworld.com/51130	**92** Superior	Superior	100%	95%	56%	38%	Muy Bueno	Un producto barato que recibió puntuaciones óptimas en nuestras pruebas de desempeño, aunque su velocidad de exploración fue baja.
2	McAfee VirusScan 2006 $499 find.pcworld.com/51132	**87** Muy Bueno	Superior	100%	89%	53%	34%	Muy Bueno	Su heurística relativamente buena contribuye a la sólida protección que ofrece VirusScan. La asistencia por teléfono cuesta US$3 por minuto.
3	Kaspersky Lab Kaspersky Anti-Virus Personal 5.0 US$60 find.pcworld.com/51134	**85** Muy Bueno	Superior	100%	100%	51%	26%	Bueno	Este programa tuvo la respuesta más rápida a los nuevos brotes de programas maliciosos. La interfase es limpia, pero no excepcionalmente buena.
4	F-Secure Anti-Virus 2006 US$40 find.pcworld.com/51136	**83** Muy Bueno	Superior	100%	97%	52%	27%	Bueno	Programa de sólido desempeño que respondió rápidamente en las pruebas y que provee noticias de última hora sobre los brotes de programa maliciosos.
5	Symantec Norton AntiVirus 2006 $700 find.pcworld.com/51290	**80** Muy Bueno	Bueno	100%	97%	22%	8%	Muy Bueno	Esta herramienta veterana ofrece una sólida detección de amenazas y una bonita interfase. La asistencia por teléfono cuesta US$30 por incidente.
6	Panda Software Panda Titanium 2006 Antivirus + Antispyware US$75 find.pcworld.com/51142	**79** Bueno	Muy Bueno	100%	86%	21%	16%	Bueno	Esta antigua Mejor Compra se desempeñó bien, aunque no extraordinariamente, en nuestras pruebas más nuevas no relacionadas con los programas de espías.
7	AntiVir PersonalEdition Classic 6.32 Gratis find.pcworld.com/51140	**78** Bueno	Bueno	100%	95%	11%	6%	Bueno	AntiVir tuvo el mejor desempeño de todos los programas gratuitos, aunque no logró limpiar varios virus de macro más viejos.
8	Alwil Software Avast Home Edition 4.6 Gratis find.pcworld.com/51138	**77** Bueno	Regular	100%	86%	9%	5%	Muy Bueno	Este producto gratuito tiene una vistosa interfase al estilo de un reproductor de medios que oculta algunas funciones. LA velocidad de exploración fue baja en las pruebas.
9	Trend Micro PC-cillin Internet Security Suite 2006 US$50 find.pcworld.com/51144	**77** Bueno	Regular	100%	76%	6%	3%	Superior	Otra antigua Mejor Compra, se desempeñó mal en las pruebas heurísticas y de zoológico pero tiene la mejor interfase del grupo.
10	Grisoft AVG Free Edition 7.1 Gratis find.pcworld.com/51146	**73** Bueno	Regular	100%	80%	8%	4%	Regular	Programa gratuito con heurística inferior y una de las interfases más complicadas de todos los productos que probamos.

NOTAS DE LA TABLA: Los precios comerciales eran válidos a partir del 15/03/05. La calificación PCW es una puntuación total basada en el desempeño, el precio, las especificaciones y el diseño del producto

Un *firewall* para redes detectará las aplicaciones clandestinas, los robots y caballos de Troya, pero como ocurre con la detección basada en el comportamiento, el *firewall* sólo le notificará que hay un problema cuando la amenaza ya existe en su PC. "Los *firewalls* detienen el tráfico de la red", dice Stewart de LURHQ. "Pueden evitar que un caballo de Troya haga contacto con el exterior, pero no pueden evitar que ese programa ejecute [en su PC]", afirma.

Tercero, AV-Test evaluó las posibilidades heurísticas de cada producto. Para hacerlo, la compañía probó la habilidad de las versiones de los programas disponibles uno o dos meses atrás, que no tenían instaladas las firmas de virus más recientes, para reconocer los programas maliciosos que han surgido desde entonces.

Así, AV-Test determinó la capacidad de los programas para detectar gusanos y software clandestino, sin el beneficio de las actualizaciones de firmas. Probar los gusanos y las aplicaciones clandestinas fue un método apropiado; porque esas fueron las amenazas más comunes y peligrosas durante el período de prueba, y los virus totalmente nuevos son difíciles de encontrar, según AV-Test.

Cuarto, AV-Test examinó la capacidad de cada producto para limpiar 110 virus de macros que atacan a los programas de Microsoft Office. Y quinto, recopiló datos sobre el tiempo de respuesta promedio de cada compañía de software antivirus a 16 brotes durante un período de ocho meses en 2006, lo que da una indicación de la velocidad con que la compañía publica actualizaciones de firmas después de identificar un nuevo programa malicioso.

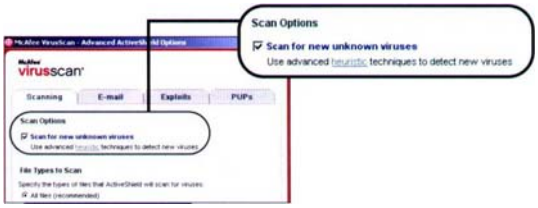
Para completar nuestra prueba, PC World cronometró la velocidad de los diversos productos al efectuar las exploraciones de virus y luego evaluó la facilidad de uso de cada producto, las características y la política de asistencia técnica.

NUESTRAS SELECCIONES DE PROGRAMAS ANTIVIRUS

Cuando el polvo finalmente se disipó, BitDefender 9 Standard surgió como nuestra Mejor Compra. Figuró entre los cuatro mejores en todas las medidas de desempeño y sólo cuesta $480. El McAfee VirusScan 2006 ($499) – con su desempeño heurístico relativamente bueno y su interfaz intuitiva – terminó segundo.

PC-cillin Internet Security Suite 2006 de Trend Micro, un descendiente de nuestra Mejor Compra de junio de 2004, terminó noveno entre los diez productos. Tuvo un desempeño pésimo en las pruebas de zoológico y heurísticas, y es relativamente caro porque solamente está disponible como un conjunto de seguridad integral. En el lado positive, tuvo tiempos de respuesta muy cortos a los brotes y ofrece una interfaz de usuario estelar.

Los tres programas gratuitos también se quedaron cortos: AntiVir terminó séptimo, Avast quedó en octava posición y AVG en la décima. Por supuesto, para los que no tienen presupuesto para un programa antivirus, cualquiera de estos productos es mejor que nada.



McAfee VirusScan quedó segundo en ambas pruebas heurísticas.

ELIMINAR LAS AMENAZAS QUE CONOCEMOS

En sus configuraciones predeterminadas y con las últimas definiciones de virus instaladas, todos los productos evaluados por AV-Test fueron 100 por ciento eficaces en detectar los virus de la *WildList* en tiempo real y a petición, esas ocasiones en que un usuario realiza un análisis manual o programado de la computadora.

Los programas detectaron y quitaron exitosamente los virus de macro, con algunas excepciones. Avast no pudo limpiar diez virus, entre ellos dos que atacan a los archivos de PowerPoint desde la versión 97 a la 2003, y cuatro virus que afectan a los archivos de Word 6. Panda no limpió totalmente dos de los virus de PowerPoint, aunque los archivos todavía se podían usar. AntiVir no logró limpiar diez virus de Word 6, entre otros, y BitDefender no detectó dos virus que infectan a los archivos de Word desde las versiones 97 a la 2003. Estos virus no son nuevos, por eso los productos actuales deberían ser capaces de detectarlos.

La capacidad para detectar los virus de *WildList* es esencial, ya que son ampliamente conocidos; detectar las alimañas del zoológico de AV-Test, sin embargo, es algo muy diferente.

Kaspersky Anti-Virus Personal 5.0 fue el único programa evaluado que detectó exitosamente los tres tipos de amenazas del zoológico el 100 por ciento de las veces. F-Secure y Symantec acertaron el 97 por ciento de las veces; una puntuación excelente.

Norton Antivirus explica claramente los elementos de la interfase y las opciones que tiene el usuario.

En el otro extreme del espectro, PC-cillin produjo uno de los peores resultados, ya que detectó solamente un 76 por ciento de las amenazas del zoológico (esta puntuación incluye el 85 por ciento de los robots, el 82 por ciento del software clandestino y el 69 por ciento de los caballos de Troya). Trend Micro dice que decidió no gastar recursos desarrollando archivos de firmas para los programas maliciosos que forman parte del zoológico de AV-Test, porque esas amenazas nunca han afectado a sus clientes, No podemos decir a ciencia cierta si todas las amenazas del zoológico son pertinentes, pero preferiríamos un producto que detecte el 100 por ciento de las especies comprendidas en esa colección.

ELIMINAR LAS ALIMAÑAS QUE NO CONOCEMOS

Ninguno de los productos se desempeñó excepcionalmente bien en nuestras pruebas heurísticas, lo cual demuestra que hay oportunidades para mejorar la identificación de nuevas amenazas. En nuestras pruebas de aplicaciones realizadas con firmas de hace un mes, BitDefender se desempeñó mejor que los demás, ya que detectó el 41 por ciento de los gusanos y el 57 por ciento de los programas clandestinos.

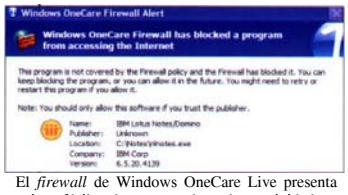
La heurística de Panda demostró ser sólo regular.

McAfee le sigue de cerca en segundo lugar, pues detectó el 41 por ciento de los gusanos y el 55 por ciento del software clandestino. F-Secure y Kaspersky quedaron a poca distancia, al acertar el 32 por ciento de las veces con los gusanos y detectar el 53 por ciento de los programas clandestinos (AV-Test dice que un grado de detección del 50 por ciento es muy bueno). En las pruebas de aplicaciones con firmas de dos meses de antigüedad, todos los programas estuvieron más desacertados.

UNA VISTA AL PRODUCTO BETA

Alternativa antivirus: MICROSOFT ONECARE LIVE

Microsoft pronto se contará entre las compañías que ofrecen una protección de seguridad integral para los consumidores. Echamos un vistazo a la edición pública beta de Windows OneCare Live, un nuevo paquete de protección para la PC basado en suscripciones. Es uno de los varios servicios basados en la Internet disponibles para bajarlos desde las páginas Windows Live Ideas (find.pcworld.com/51178). OneCare Live es una colección de utilidades y herramientas de seguridad que usted puede administrar desde una sola interfase. Los componentes de seguridad actualmente consisten en software antivirus y un *firewall*; Microsoft espera agregar una aplicación contra programas espías en una versión beta subsiguiente. Las otras herramientas que integran el conjunto incluyen una aplicación de copias de seguridad y una rutina de afinamiento que automatiza las tareas como la desfragmentación y la limpieza de discos.

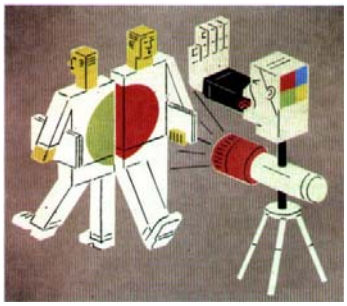


El *firewall* de Windows OneCare Live presenta avisos fáciles de comprender sobre actividad no reconocida en la red.

Como la mayoría de las herramientas antivirus, OneCare Live permite analizar a petición o siguiendo una agenda programada, configurar los archivos que usted quiere examinar y excluir los archivos del proceso de exploración. Actualmente, no examina el correo electrónico de entrada ni de salida, y sólo revisa el tráfico de mensajes instantáneos de MSN Messenger, pero la compañía dice que planea incorporar la posibilidad de examinar el correo electrónico y considerará otros clientes de IM más adelante.

Una capa de protección basada en el comportamiento vigila los archivos para detectar actividades sospechosas, como la modificación de las claves del Registro. Nuestra primera exploración demoró unos 15 minutos, un tiempo bastante aceptable.

El *firewall* de OneCare, que vigila el tráfico de entrada y salida de la red, es una versión renovada de Windows Firewall, que solamente se interesa en el tráfico de entrada. Al probarlo por primera vez, OneCare nos consultó respecto a la actividad de programas que no reconocía, como una actualización de software de iTunes y la actividad de la red de Lotus Notes. Por lo general, no estorbó para nada mientras nos mantuvimos al día con las actualizaciones de seguridad.

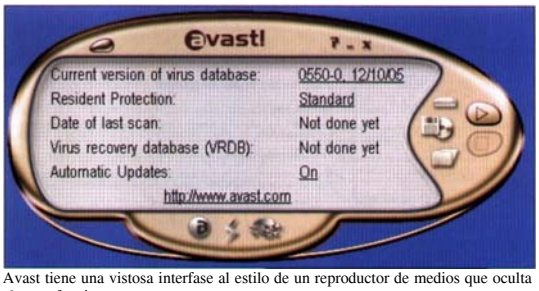


La instalación fue fácil, aunque requirió el uso de Internet Explorer 6 (para comprobar la existencia de actualizaciones de seguridad es necesario usar Internet Explorer 5 o posterior). Un asistente basado en la Web evaluó nuestro sistema para ver si cumplía los requisitos mínimos, además de reconocer los conflictos posibles de software, antes de dejarnos instalar OneCare.

Microsoft dice que OneCare comprobará durante la instalación que usted no tenga un software antivirus conflictivo, pero no reconoció la versión cliente de Symantec Norton AntiVirus Corporate Edition que teníamos instalada en nuestra PC.

Sin embargo, un lector que ofrecía comentarios en nuestro *blog* Today @ PC World (find.pcworld.com/51360) informa que sí detectó y pidió la eliminación de la versión de escritorio de Norton AntiVirus. Microsoft no ha fijado un precio para el paquete, pero la existencia de un botón Purchase Now (Adquiéralo ahora) en el sitio indica que OneCare no será gratuito para siempre.

-Narasu Rebbapragada



Avast tiene una vistosa interfase al estilo de un reproductor de medios que oculta algunas funciones.

PC-cillin volvió a quedar último. Su analizador, armado con definiciones de hace un mes, capturó el 5 por ciento de los gusanos y el 7 por ciento del software clandestino. Trend Micro opina que los problemas causados por la heurística – en particular, su potencial para falsos positivos – son más que los beneficios; como resultado, la compañía decidió no poner tanto énfasis en el desarrollo de la heurística.

LA NECESIDAD DE LA VELOCIDAD

Probamos dos tipos de velocidad en los productos para ver lo rápido que terminan una exploración de virus a petición y, más importante, para ver la agilidad de las compañías para actualizar firmas una vez que brotan nuevos programas maliciosos. El software que más rápido realizó la exploración fue el de Panda, que acabó las pruebas en un tiempo promedio de 1 minuto con 46 segundos; o lo que es igual, siete veces más rápido que el programa más lento, Avast, que quedó rezagado del grupo con un tiempo promedio de 13 minutos y 11 segundos.

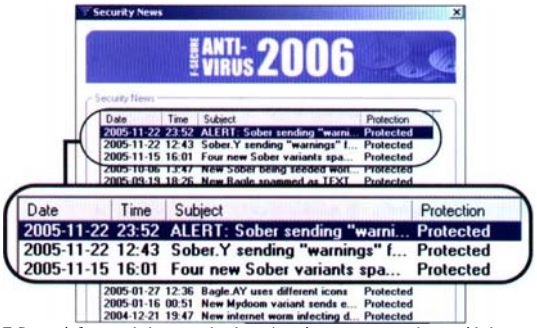
Cuando AV-Test evaluó el tiempo de respuesta de los productos a los brotes, todos respondieron a las incidencias en menos de 12 horas como promedio. Kaspersky fue el más rápido en responder (desde menos de una hora hasta 2 horas); BitDefender y F-Secure le siguieron de cerca con 2 a 4 horas; AntiVir y PC-cillin tuvieron tiempos de respuesta de 4 a 6 horas; Panda demoró de 6 a 8 horas; AVG, Avast y McAfee tardaron entre 8 y 10 horas; y Symantec fue el más perezoso, con 10 a 12 horas.

LAS CARACTERÍSTICAS VARÍAN, LIGERAMENTE

Algunos paquetes ofrecen extras atractivos. Todos bajan archivos de firmas de virus y actualizaciones de la aplicación de una manera automática y regular. La mayoría permite que usted prepare análisis completos o personalizados según su programa definido. Algunos, como el gratuito AVG, son relativamente rígidos, pues permiten sólo los análisis programados de tipos de archivo o de unidades de disco predeterminados. A diferencia de los otros programas evaluados, el de Panda no le permite establecer una agenda de exploración regular; para eso necesitará el Panda Platinum 2006 Internet Security Suite. Muchos programas han adoptado pantallas de consola – similares a las que presenta el Windows XP SP2 Security Center – para proveer una descripción general del estado de su PC. Symantec Norton Protection Center, por ejemplo, le indica cuán segura está su PC al desempeñar actividades comunes como la administración de correo electrónico o la navegación por la Web.

F-Secure y Panda ofrecen noticias de seguridad de última hora desde sus iconos de la bandeja del sistema. BitDefender pone una ventana pequeña llamada File Zone en su escritorio para mostrar una representación gráfica del número de archivos que han sido analizados en los últimos minutos (opción que se puede desactivar).

Todos los productos que probamos vienen con asistencia técnica por correo electrónico, válida mientras dure la suscripción del software antivirus (un año para los programas comerciales e indefinidamente para los gratuitos). BitDefender, F-Secure, Kaspersky, Panda y Micro ofrecen asistencia gratuita por teléfono, por lo menos en los días laborables. La asistencia telefónica de Symantec cuesta US$30 por incidente; McAfee cobre US$3 por minuto por su ayuda. Si usted cree que podría necesitar la asistencia, considere estos precios cuando haga su decisión de compra; una o dos llamadas largas podrían costarle más que el software.



F-Secure informa cabalmente sobre las más recientes amenazas de seguridad.

EL FACTOR CONVENIENCIA

PC-cillin de Trend Micro fue el producto más fácil de usar. Empaqueta mucha información de seguridad en una interfase fácil de comprender.

El Alwil Avast se distingue con una consola única y llamativa – con diferentes *skins* y todo – que luce muy similar a las de algunos reproductores de medios. La consola provee la misma información que las de otros programas, pero oculta algunas características detrás de los botones de iconos circundantes.

Las interfases de los otros programas son bastante básicas. La pantalla de apertura de BitDefender sólo notifica a los usuarios si su protección contra virus y las actualizaciones automáticas están activadas. Las características más útiles se encuentran en pantallas accesibles en el lado izquierdo de la ventana.

Sin embargo, gracias a productos gratuitos como Grisoft AVG Home Edition, usted no tiene que gastar su dinero para ganar la pelea contra los virus conocidos. Aunque no existe el paquete antivirus que pueda proteger completamente a su PC contra las amenazas desconocidas, la elección de uno de los productos mejor clasificados le dará al menos la protección más eficaz que pueda tener hoy en día.

-Tony Bradley y Narasu Rebbapragada

Condensado de PC WORLD México. Abril de 2006. Año XII, número 4.

APÉNDICE C
EL NOMBRE DE LOS VIRUS

Cuando aparece un nuevo código malicioso (ya sea virus, gusano, troyano, etc.) todos los desarrolladores de software antivirus se apresuran a identificarlo, determinar su comportamiento y su modo de transmisión, y a desarrollar una solución para bloquearlo. Forma parte del proceso asignar un nombre al virus, pero pueden surgir problemas si los diferentes desarrolladores de software usan nombres diferentes para designar al mismo virus. Para solucionar este problema, un grupo (al que sólo se puede acceder por invitación) denominado *Computer Anti-Virus Researchers Organization* (CARO) ha desarrollado un convenio de denominación para los virus. Actualmente, la mayoría de las empresas especializadas en software antivirus se adhieren a estas normas generales, aunque algunos añaden prefijos y sufijos propios para sus productos. Aunque los artículos que se publiquen en la prensa suelen hacer referencia a los virus o gusanos por su nombre común (SirCam, Gibe o Goner, por ejemplo), la organización técnica suele estar organizada utilizando nombres estandarizados (W32.Sircam, I-Worm.Gibe, W32/Goner.A@mm).

El nombre de un código malicioso está formado por los siguientes elementos:

- ✓ Un prefijo que identifica el tipo de código. La mayoría de los virus que atacan a los equipos Windows contienen el prefijo W32 (virus de Windows de 32 bits), I-Worm (gusano de Internet), JS (JavaScript) o VBS (virus de Visual Basic Script).
- ✓ Un nombre de familia. Este nombre se suele obtener del propio código del virus.
- ✓ Una versión principal. Si aparece este sufijo, será un número que identifica el tamaño de archivo del virus. La mayoría de los desarrolladores de software antivirus omiten este elemento.
- ✓ Una versión secundaria. Este sufijo suele ser una letra que distingue versiones alternativas de un mismo virus que están en circulación.

Además, algunos nombres utilizan ahora los sufijos @m y @mm para identificar códigos maliciosos que se distribuyen por correo electrónico o envíos masivos, respectivamente.

GLOSARIO

A

Análisis de Riesgos. Es una técnica para recopilar los activos técnicos y operacionales de una empresa, valorarlos en la medida en que el negocio se vería afectado por su pérdida y levantar un mapa de las amenazas a las que están expuestos.

Applet. Es un componente de software hecho en un programa específico que ejecuta en el contexto de otro; por ejemplo, un *applet* hecho en Java, que realiza una función específica, puede ser ejecutado en un navegador Web.

ARPANET (*Advanced Research Projects Agency Network*). Es una red de ordenadores que fue creada por encargo del Departamento de la Defensa de los Estados Unidos como medio de comunicación entre los diferentes organismos estadounidenses. El primer nodo se creó en la Universidad de California y fue la espina dorsal de Internet hasta 1990, tras finalizar la transición al protocolo TCP/IP, iniciada en 1983.

Autómata. Es un ordenador analógico. Se trata de un dispositivo diseñado para manipular la entrada de datos y presentar el resultado de un cálculo o un reconocimiento.

B

BIOS (*Basic Input Output System*). Es el sistema básico de entrada y salida; un código de interfaz que localiza y carga el sistema operativo en memoria, proporciona la comunicación de bajo nivel, el funcionamiento y configuración del hardware básico del sistema (teclado, disco duro, disquetera, memoria, etc.) y la salida básica durante el arranque. El BIOS usualmente está escrito en lenguaje ensamblador.

Búsqueda Heurística. Se trata de métodos en los que la exploración se realiza de manera algorítmica, pero el progreso se logra por la evaluación puramente empírica del resultado. Las técnicas heurísticas no aseguran soluciones óptimas sino solamente soluciones válidas y aproximadas; y frecuentemente no es posible justificar en términos estrictamente lógicos la validez del resultado

C

C/C++. C es un lenguaje de programación creado en 1969, por Ken Thompson y Dennis M. Ritchie en los Laboratorios Bell, como evolución del anterior lenguaje B. Está orientado a la implementación de sistemas operativos; ya que, dispone de estructuras de alto nivel pero, a su vez, de instrucciones de bajo nivel. C++ es diseñado a mediados de los años 80, por Bjarne Stroustrup, como extensión del lenguaje C. Se le considera un lenguaje híbrido; abarca tres paradigmas de la programación: estructurada, genérica y orientada a objetos.

Certificado Digital. Es un documento digital, mediante el cual una autoridad de certificación garantiza la vinculación entre la identidad de un sujeto o entidad y su clave pública. El certificado contiene usualmente el nombre de la entidad certificada, un número serial, fecha de expiración, una copia de la clave pública del titular del certificado (utilizada para la verificación de su firma digital), y la firma digital de la autoridad emisora del certificado.

D

Datos. Son cifras y hechos que no tienen un orden específico y que no han sido analizados.

Delphi. Es un entorno de desarrollo de software, diseñado para la programación de propósito general con énfasis en la programación visual; utiliza como lenguaje una versión moderna de Pascal llamada Object Pascal. Es producido comercialmente por la empresa estadounidense Borland.

DES (*Data Encryption Standard*). Es un algoritmo de cifrado por bloques que utiliza un tamaño de clave de 56 bits. Se basa en una permutación inicial y otra final, además de un proceso de 16

iteraciones intermedias. En cada iteración los bloques son permutados entre sí y después operados por medio de la operación lógica XOR; existe una clave diferente para cada iteración.

Diagrama de Flujo. Representa una forma para especificar los detalles algorítmicos de un proceso.

DOS. Es una familia de sistemas operativos para PC. El nombre está dado por las siglas de *Disk Operating System* (Sistema Operativo de Disco). Existen varias versiones de DOS, de las cuales el más conocido de ellos es el MS-DOS de Microsoft; otros sistemas son el PC-DOS, DR-DOS y, más recientemente, el FreeDOS.

E

ELF (*Executable and Linking Format*). Es un formato para archivos ejecutables, código objeto, librerías compartidas y volcados de memoria. Fue desarrollado por el *UNIX System Laboratory* (USL) para plataformas de 32 bits, a pesar de que hoy en día se usa en una gran variedad de sistemas; principalmente en los de tipo UNIX como Linux, BSD, Solaris o Irix.

F

Firewall. Es un elemento de hardware o software utilizado en una red de computadoras para prevenir algunos tipos de comunicaciones prohibidos. Su accionar será regido por las políticas que la organización responsable de la red haya definido.

H

Hardware. Se denomina hardware (o soporte físico) al conjunto de elementos materiales que componen un ordenador. En dicho conjunto se incluyen los dispositivos electrónicos y electromecánicos, circuitos, cables, tarjetas, armarios o cajas, periféricos de todo tipo y otros elementos físicos.

HTML (*HyperText Markup Language*). Es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto; es el formato estándar de las páginas Web. El

HTML se ha convertido en uno de los formatos más populares para la construcción de documentos y el aprendizaje.

I

Información. Son datos que han sido analizados y organizados de una manera lógica y entendible.

Informática. Es una palabra de origen francés formada por la contracción de los vocablos Información y Automática. La Real Academia Española define a la Informática como “el conjunto de conocimientos científicos y técnicos que hacen posible el tratamiento automático de la información por medio de ordenadores”.

Internet. Es una red de redes, a escala mundial, de millones de computadoras interconectadas con el conjunto de protocolos TCP/IP.

Interrupción. Es una señal recibida por el procesador; indica que se requiere atención inmediata para ejecutar un proceso con determinado nivel de prioridad.

J

Java. Es una plataforma virtual de software, desarrollada por Sun Microsystems, de tal manera que los programas creados en ella puedan ejecutarse sin cambios en diferentes tipos de arquitecturas computacionales.

K

Kernel. Véase núcleo.

L

Lenguaje de Alto Nivel. Es aquel lenguaje que se aparta del código máquina; existe un alto nivel de abstracción entre lo que se pide realizar a la computadora y lo que ésta realmente comprende.

Lenguaje de Bajo Nivel. Es un lenguaje que proporciona poca o ninguna abstracción del microprocesador de un ordenador; consecuentemente, es fácilmente trasladado a lenguaje de máquina.

M

Macintosh. Es el nombre de una serie de computadoras fabricadas y comercializadas por Apple Computer desde 1984.

Memoria *Flash*. La memoria *flash* es una forma evolucionada de la memoria EEPROM que permite que múltiples posiciones de memoria sean escritas o borradas en una misma operación de programación mediante impulsos eléctricos. Las memorias *flash* son de tipo no-volátil; es decir, la información que almacena no se pierde en cuanto se desconecta de la corriente.

Memoria Principal. Es el componente necesario para que se procese la información. En la memoria principal se almacenan los datos y las instrucciones que están próximas a ser ejecutadas por el procesador. Se identifican dos variantes de memoria principal: RAM y ROM. A diferencia de la memoria secundaria, sólo puede guardar información por un período de tiempo corto (mientras el ordenador esté encendido).

Memoria USB. Una memoria USB es un pequeño dispositivo de almacenamiento que utiliza memoria flash para guardar la información sin necesidad de baterías.

Memory Stick. La *memory stick* es un formato de tarjeta de memoria extraíble (memoria flash), comercializado por Sony en octubre de 1998. Es utilizada como medio de almacenamiento de información para un dispositivo portátil (cámaras digitales, dispositivos digitales de música, PDA, teléfonos celulares, PlayStation Portable), de forma que puede ser fácilmente extraído para tener acceso a un ordenador.

Monitorización. Es una amenaza que atentan contra la privacidad, al observar a la victima y su sistema, con el objetivo de establecer sus vulnerabilidades y posibles formas de acceso futuro.

N

Núcleo. También conocido como *kernel*, es la parte fundamental del sistema operativo; es responsable de facilitar a los distintos programas un acceso seguro al hardware del ordenador (gestionar recursos) a través de llamadas al sistema. Hay cuatro grandes tipos de núcleos: los monolíticos, los micronúcleos, los híbridos y los exonúcleos.

O

Ordenador. Un ordenador, computador o computadora es una máquina capaz de aceptar datos de estrada, efectuar con ellos operaciones lógicas y aritméticas, y proporcionar la información resultante a través de un medio de salida; todo ello con la mínima intervención de un operador humano y bajo el control de un conjunto de instrucciones previamente almacenado en el propio ordenador.

P

PE (*Portable Executable*). Es el nuevo formato para archivos ejecutables en sistemas Windows; se inspiró en el formato COFF (*Common Object File Format*) de los sistemas operativos UNIX y mantiene compatibilidad entre Windows y MS-DOS por medio de la cabecera MZ. Su nombre se debe a que es completamente compatible con cualquier versión de Windows 95/98/NT/Me/2000/XP.

PHP (*PHP Hypertext Pre-processor*). Es un lenguaje de programación, usado generalmente para la creación de contenido para sitios Web. Se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico en Web. Últimamente también se utiliza para la creación de otro tipo de programas, incluyendo aplicaciones con interfase gráfica usando la biblioteca GTK+.

Plug and Play. Conocida también por su abreviatura PnP, es la tecnología que permite a un dispositivo informático ser conectado a un ordenador sin tener que configurar *jumpers* ni proporcionar parámetros a sus controladores; para que esto sea posible, el sistema operativo con el que funciona el ordenador debe tener soporte para dicho dispositivo.

Políticas de Seguridad. Conjunto de leyes, reglas y prácticas que regulan la manera de dirigir, proteger y distribuir recursos en una organización para llevar a cabo los objetivos de seguridad.

Programa. Es un conjunto de instrucciones que la CPU de una computadora puede entender y ejecutar.

Programación. Es la creación de programas computacionales.

R

Red de Computadoras. Es una colección interconectada de ordenadores autónomos; se dice que dos ordenadores están interconectados si son capaces de intercambiar información.

RSA. El sistema criptográfico con clave pública RSA recibe su nombre por la inicial del apellido de sus inventores: Ronald Rivest, Adi Shamir y Leonard Adleman. Este algoritmo asimétrico de cifrado funciona con una clave pública de cifrado y una clave privada de descrifrado; se basa en el producto de dos números primos grandes (mayores que 10^{100}) elegidos al azar para conformar la clave de descifrado. La seguridad de este algoritmo radica en que no hay maneras rápidas conocidas de factorizar un número grande en sus factores primos utilizando computadoras tradicionales.

S

Sector de Arranque. Es el sector en el que todos los disquetes y discos duros (incluidos los que sólo contienen datos) tienen un pequeño programa que se ejecuta para cargar el sistema cuando se inicia el ordenador.

Sector de Arranque Maestro. Es un sector de 512 Bytes al principio del disco duro que contiene una secuencia de comandos necesarios para cargar un sistema operativo; es decir, es el primer registro del disco duro (cilindro 0, cabeza 0, sector1), el cual contiene un programa ejecutable y una tabla donde están definidas las particiones del disco duro.

Servicio de Seguridad. Es aquél que mejora la seguridad de un sistema informático y su flujo de información.

Software. Es el conjunto de datos e instrucciones que permite la utilización del ordenador. El software es la parte intangible de la computadora; es decir, programas, aplicaciones, datos, etc.

Spoofing. Es una amenaza que hace referencia al uso de técnicas de suplantación de identidad, generalmente con usos maliciosos o de investigación.

Stream. En C++, un *stream* es una abstracción para referirse a cualquier flujo de datos entre una fuente y un destinatario; convierten cualquier tipo de objeto a texto legible por el usuario y viceversa. Pueden hacer manipulaciones binarias de los objetos y cambiar el formato en que se muestra la salida.

T

Topología. Es la disposición física en la que se conectan los nodos de una red de ordenadores o servidores.

U

Unidad Aritmético-Lógica (ALU). En esta unidad es donde se realizan las operaciones de tipo aritmético y lógico.

Unidad Central de Proceso (CPU). Es el verdadero cerebro del ordenador: el procesador. Se encarga del control y ejecución de las operaciones que se realizan dentro del ordenador, con el fin de realizar el tratamiento automático de la información.

UNIX. Es un sistema operativo multiusuario, con capacidad de simular multiprocesamiento y procesamiento no interactivo. Está escrito en lenguaje C; ofrece facilidades para la creación de programas, sistemas y el ambiente adecuado para las tareas de diseños de software. Existen varias versiones de este sistema: Linux, Solaris, MacOS X, Minix, etc.

USB (*Universal Serial Bus*). El bus serie universal es una interfase que provee un estándar de bus serie para conectar dispositivos a un ordenador personal.

V

Visual Basic. Es un lenguaje de programación desarrollado por Alan Cooper para Microsoft; el lenguaje de programación es un dialecto de BASIC, con importantes añadidos. Su primera versión fue presentada en 1991 con la intención de simplificar la programación, utilizando un ambiente de desarrollo completamente gráfico que facilitara la creación de interfases gráficas.

W

Wildlist. Es un listado de todos los virus que han sido reportados como agentes activos; es decir, los virus que se encuentran en la lista se encuentran propagándose o infectando equipos reales.

FUENTES DE CONSULTA

Fuentes Impresas

Bott, Ed. *Seguridad en Microsoft Windows XP y Windows 2000*. McGraw-Hill. México, 2003.

Ferreyra, Gonzalo. *Virus en las Computadoras*. Alfaomega. México, 1994.

Gottfried, Byron. *Programación en C*. McGraw-Hill. México, 1997.

Levin, Richard. *Virus Informáticos*. McGraw-Hill. México, 1992.

Marcelo, Jesús de. *Virus de Sistemas Informáticos e Internet*. Alfaomega. México, 2000.

Montejano, Claudia. *Estrategia de Seguridad a Partir del Sistema Operativo Linux en los Servidores de la Gerencia de Tecnología Informática del Instituto Mexicano del Petróleo*. Tesis de Licenciatura, UNAM – Facultad de Estudios Superiores Aragón. México, 2005.

Rendón, Uriel. *Diseño y Desarrollo de una Metodología para la Determinación y el Establecimiento de Normas de Seguridad Informática*. Tesis de Licenciatura, UNAM – Facultad de Ingeniería. México, 2004.

Rodríguez, Jorge. *Introducción a la Informática*. Anaya Multimedia. México, 2001.

Rodríguez, Mario. *Creación de un Antivirus Computacional*. Tesis de Licenciatura, UNAM – Facultad de Estudios Superiores Acatlán. México, 2001.

Sanders, Donald. *Informática: Presente y Futuro*. McGraw-Hill. México, 1993.

Savage, Jesús. *Diseño de Microprocesadores*. UNAM. México, 2003.

Zarco, Rogelio. *Estructura de Procedimientos de Soporte Técnico Para la Aplicación y Erradicación de Virus Informáticos*. Tesis de Licenciatura, UNAM – Facultad de Estudios Superiores Aragón. México, 1999.

Fuentes Electrónicas

Enlaces en Internet

ACIMED – Elementos teórico-prácticos útiles para conocer los virus informáticos

http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352003000500004&lng=es&nrm=iso&tlng=es

Agregador de noticias | Fedora-es

http://www.fedora-es.com/

Apple – Mac OS X

http://www.apple.com/la/macosx/

C++ con clase

http://www.conclase.net/c/index.php

Centro de información de virus de CA

http://www3.ca.com/securityadvisor/virusinfo/

Extr@Internet – Información sobre Internet – Virus Informáticos

http://www.x-extrainternet.com/virus.asp

Linux.com: The Enterprise Linux Resource

http://www.linux.com/

Linux para todos – Un buen sitio donde empezar

http://www.linuxparatodos.net/geeklog/

McAfee – Software antivirus y soluciones para la prevención de intrusos

http://www.mcafee.com/mx/

Microsoft Windows: Home Page

http://www.microsoft.com/latam/windows/default.mspx

Panda Software. Antivirus, Antispyware, Antispam y otras amenazas de Internet

http://www.pandasoftware.es/home

PER Antivirus

http://www.perantivirus.com

Seguridad Informática

http://www.obconsultores.com/SegInf/

Sistemas Operativos – FACENA – UNNE

http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/SO0.htm

Symantec Corp.

http://www.symantec.com/index.htm

The short life and hard times of a Linux virus

http://www.librenix.com/?inode=21

Totally Geek :: Virus Source Code Database

http://www.totallygeek.com/vscdb/index.php

Virus Attack! El mejor lugar para defenderte de los virus

http://virusattack.virusattack.com.ar/home/

Virus Bulletin: Independent Malware Advice

http://www.virusbtn.com/index

Virus Informáticos

http://www.geocities.com/ogmg.rm/Index.html

VIRUSPROT.COM – Virus, spam, phishing. Todo sobre seguridad en Internet y Networking

http://www.virusprot.com/

Wikipedia

http://es.wikipedia.org/

Z Virus, antivirus

http://www.zonavirus.com/

Documentos Electrónicos

Ataque a los Sistemas Informáticos

http://www.cybsec.com/ataque.pdf

Computer Virus Prevention: A Primer

http://www.oucs.ox.ac.uk/viruses/documents/artdef.pdf

Computer Virus Propagation Models

http://www.elet.polimi.it/upload/zanero/papers/zanero-serazzi-virus.pdf

Linux as a File, Print & Web Server

http://linux.unimelb.edu.au/server/course/fc4/course-notes.pdf

Los Virus Informáticos

http://omega2.inin.mx/publicaciones/documentospdf/37%20LOS%20VIRUS.pdf

Monografía Teoría e Investigación de Virus y Antivirus

http://gpsis.utp.edu.co/omartrejos/descargas/Monografia%20Virus%20y%20Antivirus.pdf

Norman Book on Computer Viruses

http://download.norman.no/manuals/eng/BOOKON.PDF

Seguridad en Mac OS X

http://images.apple.com/la/macosx/features/security/pdf/Mac_OS_X_Security_TB_esp.pdf

Seguridad Informática – Virus Informáticos

http://www.monografias.com/trabajos5/virusinf/virusinf.shtml

Software Restriction Policies in Windows XP

http://www.virusbtn.com/files/johnlambert_vb2002.pdf

Virus Turorial

http://www.cknow.com/vtutor.zip

What is a Computer Virus?

http://www.go-red.com/pdf/white_paper_comvirus.pdf

What is New in Security for Windows XP Professional & Windows XP Home Edition

Windows XP 64 – Bit Edition Technical Overview

Windows XP Performance

Windows XP Technical Overview

http.//download.microsoft.com/download