LA SEGURIDAD EN LOS SISTEMAS OPERACIONALES

MARÍA DE LOS ANGELES PELÁEZ, GLORIA PIEDAD PONCE,
LUIS FERNANDO COLMENARES, SILVIO BAZANTE
Alumnos del curso de Investigación de VIII Semestre de Ingeniería de Sistemas del ICESI.

INTRODUCCION

Con el pasar de los años y de las generaciones, en la historia de la humanidad, ésta se ha dado cuenta de que la persona que maneja el mundo y lo controla no es la que tiene más poder económico, sino la que maneja la información.

Esto se puede apreciar en esta época, cuando muchas personas con gran poder económico han sucumbido ante el poder de la información. Con los grandes avances tecnológicos de la era, las herramientas desarrolladas anteriormente se han venido perfeccionando y se les ha aumentado su radio de acción. Estas herramientas son los computadores.

El computador se ha vuelto un elemento inseparable en la vida, a tal punto que, a donde vayamos, siempre vamos a encontrar algo controlado por un computador; por ejemplo: estaciones de gasolina controladas por computador; en los bancos, se ven computadores por todos lados; también en las industrias, en los hospitales, etc. Muchas de las actividades que realizan hoy en día los computadores son de vital importancia, porque un error en cualquier gestión puede representar la vida de muchas personas.

A raíz de que los computadores pueden realizar tareas de mucha importancia, es necesario dotarlos de cierta seguridad para evitar que personas ajenas a dicha actividad puedan causar un gran daño, sin proponérselo.

Cuando se menciona la palabra "seguridad" no sólo nos estamos refiriendo a la seguridad física de los equipos, sino también a la seguridad interna del sistema operativo. En esta última es donde más fallan las compañías, ya que creen que al poseer una buena seguridad a nivel físico, nadie va a poder usar el computador y tener acceso a la información. Hoy en día, con el avance de las comunicaciones, no es necesario que una persona tenga que burlar toda esa vigilancia física; simplemente, con un "modem" y un computador, puede conectarse a otro computador, y penetrar el sistema, sin mayor problema.

En el país, la seguridad a nivel del sistema operativo es muy incipiente;

apenas está en una etapa de internacionalización donde se tiene que proteger contra un mundo hostil. Quien maneja la información es quien domina el mundo.

En los países industrializados no se escatiman recursos en el desarrollo de nuevos sistemas, con un mayor grado de seguridad, para que estos sean más confiables y seguros.

1. LA SEGURIDAD EN LOS SISTEMAS OPERACIONALES

En este capítulo se explicará qué es y en qué consiste la seguridad de los sistemas operacionales; qué técnicas se han desarrollado para hacer un sistema más seguro.

También se presentarán varios casos de violación de sistemas operacionales y las consecuencias que trajeron. Además, se analizará cómo se castiga la violación de los sistemas operacionales en Colombia.

1.1. Qué es la seguridad en los sistemas operacionales

Cuando se escucha o se menciona la palabra "seguridad" se piensa inmediatamente en un tipo de seguridad física. Pues bien, cuando se habla de seguridad en el sistema operacional, no se está hablando únicamente de la seguridad física del equipo, sino también, de la seguridad de la información contenida en los equipos.

Cada sistema operacional tiene un nivel de seguridad específico, con el cual se puede manipular el flujo de la información, es decir, el usuario decide quién puede tener acceso a cierta información y quién no.

"El nivel de seguridad que debe proporcionarse a un sistema, depende del valor de los recursos por asegurar".[1]

El concepto de seguridad en los sistemas operacionales es relativamente nuevo. Solamente se vino a hablar de seguridad en los sistemas operacionales a mediados de los años sesenta y setenta. Este campo, al igual que otros relacionados con la tecnología de los computadores, se fue desarrollando a partir de los proyectos militares.

1.2. En qué consiste la seguridad en los computadores

La seguridad en los sistemas operativos no consiste en cualquier dispositivo físico, por fuera del equipo, sino en una serie de programas que se encargan de controlar el acceso de los usuarios a los recursos de un computador.

Básicamente, la seguridad del sistema se concentra en tres actividades que son:

* Auditorías.
* Protección del sistema.
* Protección de datos.

En 1985, el Departamento de Defensa de los Estados Unidos publicó el libro *The Trusted Computer System Evaluation Criteria*. Este libro es conocido como *The Orange Book* (el Libro Naranja). Su importancia se basó en que determinó un estándar en la seguridad de los sistemas operacionales.

1.2.1. *Auditorías*

Las auditorías en los sistemas operativos son el equivalente de las auditorías en los negocios. El objetivo de esta estrategia es determinar si ha ocurrido una violación en el sistema, o si hay alguien que se está apropiando de muchos recursos, cuando no puede hacerlo; o verificar lo que están haciendo los usuarios y poder identificar quiénes son.

Las auditorías se deben adelantar periódicamente para llevar un buen control de la seguridad. Es recomendable realizarla dos a tres veces por mes, pero esto es determinado por el administrador del sistema. También es bueno llevar a cabo auditorías-sorpresa, para que los usuarios no tengan tiempo de cubrir todo tipo de huellas, si han infringido la integridad del sistema.

En algunos sistemas operacionales, realizar las auditorías es un trabajo muy complicado, ya que el administrador del sistema debe vigilar todas las etapas de la auditoría, mientras que en otros es tan fácil que viene siendo una tarea que el mismo sistema realiza, sin que el administrador esté, todo el tiempo, vigilando las etapas del proceso.

1.2.2. *Protección del sistema*

La protección del sistema consiste en prevenir que alguien se adueñe de los recursos del sistema y, por consiguiente, no permita usarlos, o use recursos inhabilitados para los usuarios.

1.2.3. *Protección de datos*

La protección de datos consiste en un control sobre todas las operaciones de los datos, en lo que se refiere a lecturas, escrituras, modificaciones, borrado, etc.; para que los usuarios solamente puedan realizar las operaciones a que tienen derecho.

1.2.4. *El Libro Naranja*

En este libro no se menciona un diseño específico para la construcción de un sistema que sea seguro; sino que se clasifican los sistemas operativos, de acuerdo con la manera como manejan la seguridad.

Para calificar la seguridad de un sistema, el libro utiliza las letras A,B,C,D, en donde las letras B y C tienen subdivisiones. En total, se manejan siete niveles de seguridad. La letra "A" quiere decir que el nivel de seguridad del sistema es muy elevado y que prácticamente es imposible de violar, y así sucesivamente va disminuyendo hasta llegar a la letra "D", la cual quiere decir que el sistema es muy fácil de penetrar.

1.2.4.1 *Categorías definidas por el Libro Naranja*

Las categorías definidas por el Libro Naranja, son las siguientes:

D: Protección mínima, en donde la seguridad solamente se limita a la seguridad física. Un ejemplo de sistemas de categoría D es el famoso DOS.

C1: Protección de seguridad discrecional. Esto significa que la protección y el control de acceso están definidos por el administrador de la máquina; la gran mayoría de los sistemas operacionales UNIX caen dentro de esta categoría.

C2: Protección de acceso controlado. Esto quiere decir que, además de actuar como el nivel C1, lleva registros de auditoría. El ejemplo de sistemas operacionales que cumplen este nivel, es el mismo ejemplo del nivel anterior.

B1: Protección mandataria. El administrador de la máquina no puede definir si existe o no una clave. No pueden existir cuentas sin clave; un ejemplo de sistemas operativos que cumplan con este nivel, son los AS400 de IBM.

B2: Protección estructurada. Existen políticas formales de seguridad y se separan las funciones de administración y operación.

B3: Dominio de la seguridad. Existe la posibilidad de realizar especificaciones, definir permisos por usuarios y por grupos y tener control sobre los periféricos.

A1: Protección verificable. Se define el modelo de protección; existe protección a nivel de la fuente y del código ejecutable, etc. Muy pocos sistemas se pueden acercar a esta categoría; en la actualidad el único sistema que ha sido clasificado como A1, es el sistema operacional SCOMP de Honeywell. Para que un sistema sea clasificado como A1,

1. *Introducción a los Sistemas Operativos*, pág. 447.

se requiere por lo menos un período de estudio de tres años y pasar todas las pruebas a que es sometido el sistema.

1.3. Técnicas de seguridad

Un sistema de computación contiene muchos objetos que necesitan protegerse. Estos objetos pueden ser elementos de "hardware", como unidades centrales de procesamiento, segmentos de la memoria, terminales, unidades de disco, impresoras; o bien pueden ser elementos de "software", como archivos, bases de datos, etc.

Cada objeto tiene un nombre único, por el cual se refiere, y un conjunto de operaciones que se pueden ejecutar con el Read y el Write, las cuales son operaciones adecuadas para un archivo; UP y DOWN tienen sentido con un semáforo. Los objetos son el equivalente del sistema operativo, lo que, en lenguajes de programación, se conoce como tipos de datos abstractos.

Está claro que se necesita contar con alguna manera de prohibir que los procesos den entrada a aquellos objetos para los que no se tiene acceso autorizado. Además, este mecanismo también debe hacer posible limitar los procesos a un subconjunto de las operaciones legales, cuando se necesite. Por ejemplo, el proceso "A" puede tener derecho a leer, pero no a escribir el archivo "F".

Para ofrecer una manera de analizar diferentes mecanismos de protección, conviene presentar el concepto de dominio.

1.3.1. *Dominio de protección*

Un dominio es un conjunto de parejas (objetos, derechos). Cada pareja especifica un objeto y algún subconjunto de las operaciones que se pueden efectuar con él. Un derecho, en este contexto, significa autorización para ejecutar una de las operaciones. Es posible que el mismo objeto esté en múltiples dominios, con diferentes derechos en cada uno.

En cada momento, cada proceso se ejecuta en algún dominio de protección. En otras palabras, existe algún conjunto de objetos que pueden acceder y, por cada objeto, se tiene algún conjunto de derechos. Los procesos también pueden correrse de un dominio a otro, durante la ejecución. Las reglas para el cambio de dominios dependen en gran medida del sistema.

Para llevar un control sobre cuál objeto pertenece a cuál dominio, el sistema utiliza una matriz grande (Figura 1) donde los renglones son los dominios y las columnas los objetos. Cada caja lista los derechos, si hay alguno, que el dominio contiene para el objeto.

OBJETO

Arch	Arch	Arch	Arch	Arch	Arch	Impre	Grafi
R	R,W						
		R	R,W,X	R,W		W	
					R,W,X	W	W

Figura 1. *Matriz de protección*

Dados esta matriz y el número de dominio corriente, el sistema siempre puede indicar, si se permite el intento de acceso, a un objeto específico, de manera particular, a partir de un dominio especificado.

La Figura 2 muestra la matriz de la Figura 1, una vez más, sólo que ahora con los tres dominios como objetos. Los procesos del dominio 1 pueden correrse al dominio 2, pero una vez ahí ya no pueden regresar.

OBJETO

A.1	A.2	A.3	A.4	A.5	A.6	Imp	Gra	D.1	D.2	D.3
R	RW								Int	
		R	RWX	RW		W				
					RWX	W	W			

Figura 2. *Matriz de protección*

1.3.2. *Listas con control de acceso*

En la práctica, el almacenamiento real de la matriz de protección rara vez se hace porque es grande y dispersa. La mayoría de los dominios no tienen acceso en absoluto a la mayoría de los objetos, de manera que el almacenamiento de una matriz grande, vacía, se traduce en una pérdida de espacio, en el disco. Sin embargo, dos métodos prácticos son el almacenamiento de la matriz por renglones o por columnas y el almacenamiento sólo de los elementos no vacíos. Estos métodos son sorprendentemente distintos.

La primera técnica consiste en asociar con cada objeto una lista (ordenada) que contenga todos los dominios a que puede acceder el objeto y que indique cómo hacerlo. A esta lista se le llama "Lista de Control de Acceso" o "ACL".

1.3.3. *Capacidades*

La otra manera de dividir la matriz de protección es por renglones. Cuando se emplea este método en asociación con cada proceso, hay una lista de objetos que pueden acceder, junto con una indicación de qué operaciones se permiten con cada uno, en otras palabras, su dominio. Esta lista se llama "Lista de Capacidades" y los elementos individuales contenidos en ella se denominan "capacidades".

Una lista común de "capacidades" se muestra en la Figura 3. Cada capacidad tiene un "campo de tipo", el cual indica

	Tipo	Derechos	Objeto
0	Archivo	R - -	Apunt al Arch 3
1	Archivo	R W X	Apunt al Arch 4
2	Archivo	R W -	Apunt al Arch 3
3	Archivo	- W -	Apunt a la Impr.

Figura 3. *Lista de capacidades de un objeto.*

de qué tipo de objeto se trata; un "campo de derechos", que es un mapa de bits que indica cuáles de las operaciones legales con este tipo de objeto están permitidas y un "campo de objeto", el cual es un apuntador del objeto mismo.

Como es evidente, las listas de capacidades deben protegerse de la alteración por parte del usuario. Para esto se han propuesto tres métodos:

A) El primer método es un diseño de "hardware" que requiere una arquitectura etiquetada, en el cual cada palabra de la memoria tiene un bit (o etiqueta) extra, el que indica si la palabra tiene capacidad o no.

B) El segundo método consiste en conservar la lista de capacidades, dentro del sistema operativo, y simplemente hacer que los procesos se refieran a las capacidades,por su número de ranura.

C) El tercer método consiste en conservar la lista de capacidades en el espacio del usuario, sólo que poniendo en clave cada capacidad, con una clave secreta, desconocida para el usuario.

Además de los derechos específicos que dependen del objeto, las capacidades suelen tener derecho a genéricos, los cuales son aplicables a todos los objetos. Ejemplos de derechos genéricos son:

- Capacidad de copiado: Crear una nueva capacidad para el mismo objeto.
- Copiar el objeto: Crear un duplicado del objeto con una nueva capacidad.
- Capacidad de eliminación: Suprimir la captación de la lista C, sin afectar el objeto.
- Destruir el objeto: Eliminar en forma permanente el objeto y la capacidad.

1.3.3.1. *Revocación de las capacidades*

La revocación del acceso a un objeto es muy difícil, ya que resulta complicado para el sistema hallar todas las capacidades restantes de cualquier objeto, para devolverlas, ya que éstas pueden estar almacenadas en listas C, por todo el disco. Como solución a este problema, existen dos métodos:

A) El primero consiste en hacer que cada capacidad apunte a un objeto indirecto, en vez de al objeto mismo; al hacer esto, el sistema siempre puede romper esa conexión, con lo cual se invalidan las capacidades.

B) El segundo método consiste en que cada objeto contenga un número largo elegido al azar, el cual también está presente en la capacidad. Cuando una capacidad se presenta para su uso, las dos se comparan. Sólo si concuerdan se permite la operación. El propietario de un objeto puede solicitar que se cambie el número del objeto elegido al azar, con lo cual se invalidan las capacidades existentes.

Ninguno de estos dos métodos permite la revocación selectiva, es decir, retirar, por ejemplo, la autorización de John, pero la de nadie más.

1.3.4. ***Canales de conversión***

Se puede comprobar que aun en un sistema que ha sido rigurosamente probado y se ha concluido que es absolutamente seguro, la fuga de información entre procesos, que en teoría no se pueden comunicar en absoluto, es relativamente directa. Estas ideas se deben a Lampson (1973).

En el modelo de Lampson intervienen tres procesos y se aplica principalmente a sistemas grandes de tiempo compartido. El primer proceso es el cliente, el que desea que el segundo, el servidor, realice algún trabajo. El cliente y el servidor no confían, por completo, el uno en el otro.

El tercer proceso es el colaborador, el cual conspira con el servidor para, en realidad, sustraer datos confidenciales del cliente. El colaborador y el servidor son comúnmente, propiedad de la misma persona. Estos tres procesos se muestran en la Figura 4a.

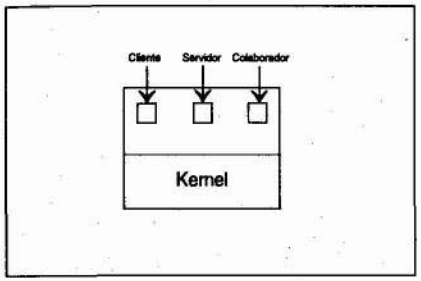


Figura 4(a)

Desde el punto de vista del diseñador del sistema, el objetivo consiste en encapsular al servidor de tal forma que no pueda pasar la información al colaborador. Con una matriz de protección, esto se puede evitar, pero por desgracia, pueden estar disponibles más canales de comunicación sutiles. Por ejemplo, el servidor puede intentar comunicar un flujo de bits binarios, de la manera siguiente. Para enviar un bit uno (1), éste hace lo que sea por determinar un intervalo fijo de tiempo. Para enviar un bit cero (0), éste se bloquea durante la misma longitud de tiempo.

El colaborador puede intentar detectar el flujo de bits, monitoreando con mucho cuidado su tiempo de respuesta. En términos generales, obtendrá una mejor respuesta cuando el servidor envíe un cero (0) que cuando el servidor envíe un uno (1). Este canal se conoce como de conversión y se ilustra en la Figura 4b.

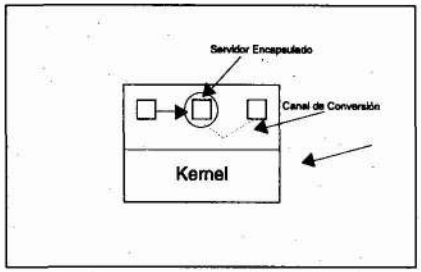


Figura 4 (b)

Desde luego que el canal de conversión es un canal ruidoso, pero la información se puede enviar, con toda confianza, por un canal ruidoso, mediante un código de corrección de errores. El uso de un código de corrección de errores reduce, aún más, el tamaño de la banda del canal de conversión, pero aún sigue siendo lo suficientemente ancho para dejar escapar la información sustancial.

Hallar todos los canales de conversión (sin hablar de su bloqueo) es en extremo difícil. En la práctica, poco se puede hacer al respecto.

1.3.5. ***La criptografía***

En el siguiente tema se hablará de la criptografía, de sus orígenes y de cómo pasó de ser una estrategia militar a una forma de seguridad en los sistemas operacionales.

La criptografía es el arte de comunicarse mediante mensajes secretos; su práctica es tan vieja como lo es la escritura. Su nombre se deriva de las palabras griegas *Kriptos* (escondido) y *Logos* (palabra).

El origen de la criptografía y de su evolución es un misterio, pero se tienen evidencias de que a mediados del siglo 400 A.C. usaban métodos de escritura secreta. En tiempos de Julio César se encontraron evidencias de un método de criptografía que consistía en reemplazar las letras, por la tercera letra que si-

gue en el alfabeto, es decir, la letra A se reemplaza por la letra D y así sucesivamente; a este método se le llama "Julius Caesar Cipher".

En la actualidad, la criptografía está muy desarrollada, y cada vez se recurre a fórmulas matemáticas complejas para codificar los mensajes. Pero esto también es violable y, basta con descubrir cuál es la fórmula utilizada para la codificación para poder descifrar el mensaje. Obviamente, esto no lo puede hacer cualquier persona.

En los sistemas operacionales se utilizan métodos de criptografía para proteger las claves de acceso, o para proteger documentos de vital importancia. Este tema también se ha llevado al cine. Recordemos la película *Héroes al azar,* que consistía en el desarrollo de una fórmula matemática, la cual descifraba cualquier sistema de criptografía utilizado en un sistema operacional, y con él se pretendía entrar en los sistemas de las organizaciones contra el crimen, para borrar todo tipo de evidencias contra los jefes de la mafia de los Estados Unidos.

1.4. Casos de violación a los sistemas operacionales

Se podría llegar a pensar que violar la seguridad de un sistema es difícil y que, por lo tanto, no hay tantas violaciones de los sistemas computacionales.

El problema es de tal magnitud, que sólo en los Estados Unidos las estafas que se originan a raíz de estas violaciones cuestan anualmente entre 500 y 600 millones de dólares, aproximadamente.

La seguridad en los sistemas es la "oveja negra" de la computación. A menudo no recibe respeto, particularmente de la alta gerencia, que está inclinada a invertir los recursos que controla en producción y mercadeo. Estas áreas generan utilidades de una manera más palpable. Pero nosotros, como futuros profesionales, tendremos la misión de disuadir de tal posición a dichas personas, con hechos y cifras. (Por ejemplo, combinar números aterradores con relatos aterradores, y así obtener mejores posibilidades de hacer ver la importancia de la seguridad en los sistemas).

A continuación, se presentarán algunos de los más escandalosos y multimillonarios delitos que han sucedido en los últimos quince años.

Marvin Maki: Pionero en la violación de sistemas.

Este fue el primer delincuente por computador, enjuiciado por la Fiscalía del Distrito de Los Angeles (Estados Unidos, California). Utilizó el computador de la MDSI, que estaba acondicionado para producir una cinta perforada, la cual controlaba la maquinaria de manufactura de la compañía W&R Tool, y procesaba las cuentas por cobrar de esta empresa. También había hallado los códigos de acceso, utilizados por MDSI en sus oficinas francesas e inglesas, y los había utilizado para conseguir acceso a su sistema. La empresa había cambiado sus códigos locales, después que Maki dejó la compañía, pero no había tomado las medidas adicionales para cambiar los códigos en el extranjero.

Su delito consistió en no haber respetado algunos derechos de propiedad que no estaban claramente protegidos, (la toma del tiempo del computador sin autorización; entrar y tomar un poco de tiempo extra), con la única diferencia de que dicho tiempo de trabajo generaba utilidades no para la compañía sino para Marvin Maki, hecho éste que pasaba inadvertido.

La información que se ha podido obtener, acerca de este caso es muy poca, pero se ha incluido en el trabajo para dar una viva muestra de los fracasos que pueden surgir cuando no se cuenta con un apropiado sistema de seguridad, que en este caso debió haber sido más rígido e invulnerable, en cuanto a la asignación de tiempos de trabajo y prioridades, tema éste tan discutido y tratado en el curso de sistemas operativos.

Harold Rossfields Smith: El gran robo del Wells Fargo

Hace nueve años, Harold R. Smith fue condenado a prisión por la Corte Federal de Los Angeles, por haber robado 21.3 millones de dólares, en representación de dos organizaciones que le pertenecían.

El robo consistió en haber transferido fondos de una cuenta a otra, violando el sistema de liquidación de una de las sucursales del banco Wells Fargo. Comenzó cuando un empleado de la sucursal Beverly Drive giró un cheque de caja para cubrir un sobregiro de la cuenta de Smith y envió, tanto el cheque como el formato de liquidación del débito parcial de la sucursal, por medio del sistema de liquidación de ésta, como si estuviera transfiriendo fondos de una sucursal a otra (de la Beverly Drice a la Miracle Mile).

Este sistema de liquidación era básicamente de contabilidad o teneduría de libros, y el banco lo utilizaba para transferir dinero de una sucursal a otra.

El banco empleaba un computador, localizado en San Francisco, para llevar el registro de todas esas transferencias diarias entre sucursales; para hacerlas, se utilizaba un formato estándar (Formato de liquidación de la sucursal) para las entradas ( ) y se le daban instrucciones al computador sobre qué hacer.

El primero paso de la artimaña consistió en que el hombre dentro del banco, (la sucursal Miracle Mile), pagó un dinero al acusado (Smith), por medio de sus compañías.

Como paso siguiente, el débito del formato de liquidación lo entregó al computador, como si fuera una transacción normal. A los diez días, en vez de transmitir la mitad del crédito a Miracle Mile, la reportó al computador, violando el principio de que en una sucursal nunca se deben procesar ambas mitades del tiquete.

Entonces, para engañar al computador, simuló que la mitad del formato de crédito procedía del Miracle Mile, y cambió el código de la mitad del crédito. Puso el código especial de la Miracle Mile en números magnéticos al final de la mitad, del crédito, para evitar violar el principio de que una sucursal nunca usará el número de código de la otra sucursal. Con todo esto, impidió que se produjeran inconsistencias, porque el computador había recibido ambos pagos parciales del formato dentro de los diez días. Y tuvo éxito al engañar al computador para que "pensara" que las dos mitades del formato habían llegado de sucursales diferentes, porque la mitad del débito tenía el código de Beverly Drive y la mitad del crédito, el de Miracle Mile.

El problema surgió en el momento del balance, cuando se observó esta transacción, donde había dos créditos y tan sólo un débito. La razón para que faltara un débito radicaba en que no existía dinero real, entrando al sistema.

Así que, para suplir el débito faltante, el empleado de la Miracle Mile tomó otro tiquete de liquidación de la sucursal y la mitad del débito de este nuevo tiquete y, al tiempo que introducía la mitad del crédito del primer tiquete, incluyó un nuevo débito; y todo quedó balanceado. El computador había recibido las dos mitades de este tiquete dentro de los diez días, con dos números de códigos diferentes, y le dio otros diez días más para conseguir la mitad del crédito del segundo tiquete introducido.

Toda esta maniobra se convirtió en un ciclo vicioso del cual era muy difícil escapar, si no se reponía el dinero (cuya cantidad era imposible de saldar).

La única manera de descubrir el robo era un descuido por parte del empleado de Miracle Mile; hasta que olvidó enviar un tiquete débito y permitió reflejar, en los informes de auditoría, las inconsistencias en las cuentas.

En este caso, se puede apreciar no solamente la importancia de la seguridad en los sistemas, sino también, la necesidad de contar con personas de absoluta confianza para permitirles el acceso a las cuentas de las compañías. Con los computadores y la consecuente dispersión en el acceso a las cuentas de las empresas, el número de estafadores se incrementa dramáticamente.

Hubo deficiencias en los métodos de seguridad interna del sistema; una de las tantas posibles soluciones hubiera podido ser que los parámetros de control se hubieran cambiado ocasionalmente, en vez de investigar siempre los balances.

Jan Hanasz: Mezcla de alta tecnología.

En 1990, en Torun, Polonia, uno de los más sobresalientes científicos del espacio, Jan Hanasz, y tres colegas, fueron llevados a juicio "acusados de haber interrumpido una transmisión estatal por televisión... para instar a los votantes a que boicotearan las elecciones...".

Se afirma que estas personas utilizaron un computador casero, un circuito sincronizador y un transmisor para hacer aparecer mensajes en las pantallas de televisión.

Lo habían hecho, llevando a la práctica el equivalente electrónico de viajar, a dedo, en las ondas de transmisión de la red de televisión polaca. (Crearon un sistema que podía producir señales de televisión, sincronizarlas con las señales de televisión estatal polaca y dirigirlas en la misma ruta. El microcomputador regulaba la emisión de señales para que se sincronizaran con las de la estación del gobierno).

Esto es otro clásico delito por computador, mezclado con la tecnología electrónica y una causa justa. Cada vez más, se realizan grandes esfuerzos en todo el mundo para evitar que se violen los sistemas de comunicaciones por televisión. (De todas maneras, así se utilice cualquier método o procedimiento, si se logra traspasar la barrera de lo permitido, es porque hay fallas y vulnerabilidad en el sistema que protege la ejecución de los procesos).

Casos de violación en sistemas hay muchos y el trabajo se extendería demasiado si se pretendiera seguir exponiendo algunos más. No se busca convertir este trabajo en una recopilación de casos de violación de sistemas; lo que se desea es crear y formar una conciencia de la necesidad de proteger cada cosa que se haga y conocer cómo estas personas logran sus objetivos.

1.5. ¿Cómo se castiga en Colombia?

En Colombia, el delito llevado a cabo a través del computador no está tipificado; esto quiere decir que no hay ninguna ley que pueda actuar contra los delincuentes que usen un computador como medio para llevar a cabo un delito.

Esto no es de extrañarse, ya que Colombia posee un código penal ineficiente y atrasado; y esto es uno de los factores que hace que nuestra justicia sea mala.

Cuando se trató de averiguar por qué no existía una legislación para esto, nos encontramos con la siguiente respuesta: En Colombia, los computadores no están realizando tareas de importancia; además, no es de interés entrar a legislar sobre este campo, ya que no existen las bases para poder hacerlo.

Después de esta respuesta, se formuló el siguiente interrogante: Se supone que alguien redacta un proyecto de ley para tipificar esto como un delito, ¿cuál es el camino que debe seguir para que sea aprobado como ley? La respuesta que se encontró fue más increíble todavía: "ese proyecto de ley tiene que pasar por tanta burocracia, que si no tiene ningún problema se estaría aprobando como ley a los cuatro años". (Esto siguiendo los conductos normales).

Como se ve, antes de elaborar un proyecto de ley, para que se tipifique como delito, la violación de un sistema operacional, es mejor hacer una reforma al Código Penal para actualizarlo y volverlo eficiente.

2. RED

En este capítulo y en el siguiente, se llevará a cabo el caso de estudio. El caso consiste en analizar una red, descubrir las debilidades que tiene y atacar el sistema por esas debilidades. Para su desarrollo se contó con la ayuda del doctor José Hernando Bahamón, del Ingeniero Alvaro Pachón y del administrador de la red, Juan Manuel Madrid.

2.1. Red por atacar

La red por atacar, fue UNIX del ICESI. Se la escogió por las ventajas que se tendrían en cuanto al apoyo logístico; además porque los conocimientos adquiridos servirían para el próximo semestre.

2.2. Configuración de la red

La red en cuestión es una ETHERNET, con topología bus, que permite hacer multipunto. En esta red se encuentran dos servidores, un Sun y un Compaq. Ambos tienen conectadas varias terminales brutas.

El Sun y el Compaq se pueden comunicar entre ellos nada más, a través del protocolo TPC/IP. Las terminales brutas se comunican a través del protocolo CSMA/CD (Carrier Sense Method Acces/Carrier Detect).

El dispositivo físico utilizado para conectar las estaciones y las terminales es un cable coaxial delgado.

2.3. Configuración del sistema

En el tema anterior se mencionó la palabra "Multipunto"; esto quiere decir que el sistema es un sistema multiusuario donde las terminales se pueden comunicar entre sí, con la característica de que cuando habla una terminal todas las demás escuchan.

El criterio conque se maneja el derecho de quién debe usar el cable, se basa en la ley del más fuerte. Es decir, el primero en llegar es quien se adueña del medio de comunicación.

El servidor Sun usa el sistema operativo SUNOS 5.3 y el servidor Compaq usa el UNIX 3.2 de Compaq.

3. VULNERABILIDAD

3.1. Debilidades identificadas

Las siguientes son las debilidades que encontramos en el sistema:

* La primera dificultad que se pensó se iba a tener era: ¿Cómo se iba a entrar al sistema si no se poseía una cuenta? Pero, ¡cuál fue nuestra sorpresa! Se encontró una cuenta que es pública, y lo mejor, sin necesidad de dar clave, y se pudo entrar al sistema.
* El sistema no está protegido contra el procedimiento del *Caballo de*

Troya. Cuando se menciona al *Caballo de Troya,* no nos estamos refiriendo al virus de los microcomputadores, sino a los programas que utilizan el principio empleado en la guerra de Troya.

* No se realizan auditorías.

3.2. **Ataque en las debilidades**

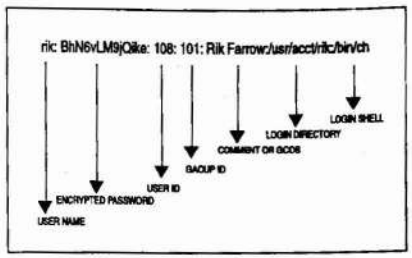
En UNIX, para poder realizar una tarea, se necesita que el grupo de trabajo o la persona tengan una cuenta asignada por el administrador de la red; sin esto no se puede hacer nada, ya que no habría forma de entrar al sistema y utilizar sus recursos.

Al caer en la cuenta de esto, se pensó que no se iba a poder realizar algo, y que se tendría que recurrir a que se nos asignara una cuenta por parte del administrador de la red, y tratar de saltarnos las limitaciones impuestas a nuestra cuenta. Pero, ¡cuál fue nuestra sorpresa! Se pudo entrar al sistema por medio de una cuenta pública de "Reserva", la cual no pedía "password" de entrada.

Si la meta era la de lograr entrar al sistema sin poseer una cuenta, prácticamente se había logrado sin mayor problema y se podría considerar que se había violado la seguridad del sistema.

Al estar en el sistema, se planteó el objetivo de tratar de colocarnos como superusuarios del sistema. Aprovechando una de las cualidades del sistema UNIX, navegamos por todos los directorios hasta que encontramos la lista de las cuentas activas y a quiénes pertenecían. Para una persona que conozca bien UNIX, esta información le es de mucha importancia, ya que puede identificar cuáles son los atributos de una cuenta en especial y tratar de adivinar el password de la cuenta. (Ver Figura 5).

Figura 5.
Estructura del archivo de claves



Para adivinar el "password" de una cuenta, se podría recurrir al método del ensayo y el error, también se podría identificar quién es la persona y, por medio de la sicología, ir tratando de dar con el password, ya que la mayoría de las personas no saben escoger un password adecuado. O por último, el más elaborado, la utilización del programa denominado *Caballo de Troya,* el cual consistía en hacerle creer a la persona que su "password" lo digitó mal la primera vez y que tiene que volver a digitarlo. Este programa lo que hace es que cuando se da la primera vez el "password", él lo toma y lo guarda, y así la persona que hizo el programa nada más tiene que consultar el programa para averiguar cuál es el "password" de esa persona.

Obviamente, realizar este método está fuera del alcance de nuestros conocimientos; sin embargo, el sistema no está protegido contra este tipo de programas y se hubiera podido lograr el robo de un "password" válido.

Otra forma de tratar de descubrir cuáles son los "password" válidos, es hacer un programa que "criptoanalice", ya que los "passwords" están "encriptados". Sobre criptografía se ha hablado mucho, ya que se han desarrollado métodos para descifrar, sin ningún problema, textos "encriptados" en los cuales se consideraba que se había utilizado un buen método. Hay que recordar que el realizar criptografía, a nivel del "software", es complicado; por lo tanto, en sistemas que están diseñados con un nivel medio de seguridad, dicho programa no es muy potente y, por tanto, fácil de descifrar.

Lamentablemente, ninguno de nuestros intentos dio resultado y no se pudo lograr la meta trazada, que era la de colocarnos como super usuarios. Se esperaba que los intentos llevados a cabo fueran detectados por la auditoría, la cual debe ser realizada por el administrador de la red; pero se encontró que no hay auditoría, ya que es muy difícil de llevar a cabo. Se nos informó que se recurría al registro de claves, cuando se tenía la sospecha de que había alguien intentando violar el sistema.

4. **CONCLUSIONES**

* La seguridad en los sistemas operacionales no es cualquier cosa que se pueda pasar por alto.
* Hasta el momento, no se puede afirmar que un sistema es 100% seguro, ya que siempre tendrá un lado débil, por el cual va a fallar, tarde o temprano.
* Sería bueno que desde la misma universidad se empezara a enseñar, a los futuros ingenieros de sistemas, la importancia de la seguridad en los sistemas operacionales.
* Empezar a presionar a la clase política y a los magistrados para que modernicen el Código Penal, para que se pueda hacer una buena legislación en cuanto se refiere al delito por medio del computador.
* Hablar de seguridad en los sistemas operacionales implica también hablar de la seguridad a nivel del "software" y no únicamente de la seguridad física.
* Siempre existirá alguien que trate de violar la seguridad de un sistema, ya sea como un reto, o por hacer una maldad.

BIBLIOGRAFIA

DEITEL M. Harvey, *Introducción a los Sistemas Operativos.*

HUNKA H. Bruce, *Administration and Managen Handbook.*

UNIX *System V relax 4 Administration.*

Grolier Inc. *The New Grolier Multimedia.* Encyclopedia Release 6.

Grandes Estafas por Computador.

Principios Avanzados de Seguridad para UNIX.

GLOSARIO DE TERMINOS

Arch, arc = Archivo.

R = Lectura.

W = Escritura.

X = Ejecución.

Impre = Impresora.

Grafic = Graficadora.