Ingeniería en Desarrollo de Software
3er semestre

Programa de la asignatura:
Programación de sistemas operativos

Unidad 3. Seguridad y protección

Clave:

Licenciatura:	TSU:
15142317	**16142317**

Universidad Abierta y a Distancia de México

Índice

Unidad 3. Seguridad y protección

Presentación de la unidad

Los términos de seguridad y protección de sistemas operativos se utilizan de forma indistinta y sirven para hacer la distinción entre los mecanismos específicos del sistema operativo, los cuales proporcionan seguridad, y el aseguramiento de los archivos para que éstos no sean visualizados ni modificados por usuarios no autorizados.

La **protección** se obtiene por medio de un mecanismo que restringe el acceso a los programas, procesos, usuarios y/o a los recursos definidos por un sistema informático. Este mecanismo debe proporcionar un medio para establecer los controles que se deban imponer.

La **seguridad** ofrece la validación de los usuarios del sistema con el objetivo de proteger la integridad de la información almacenada en el mismo y los recursos físicos del sistema; además, protege el acceso no válido y la destrucción o modificación malintencionada de los datos.

Los temas que se manejarán a lo largo de esta unidad son de vital importancia para un buen desarrollo de un sistema operativo.

Propósito

Al término de esta unidad lograrás:

- Conocer cómo se llevan a cabo los sistemas de seguridad y delimitación en accesos a los archivos, mediante la revisión de los mecanismos de protección que garanticen que sólo los procesos autorizados del sistema puedan operar sobre los segmentos de memoria, CPU y otros dispositivos de entrada y salida.

Competencia específica

- Utilizar las funciones y técnicas para validar las amenazas de un sistema como políticas y mecanismos mediante la diferenciación de seguridad y protección.

3.1. Entorno de seguridad

El uso creciente de los sistemas operativos en ambientes comerciales, gubernamentales, militares e incluso en hogares ha generado que la seguridad esté basada en cómo debe protegerse un sistema contra robos, ataques o cualquier tipo de programas y/o usuarios malintencionados que pueden llegar a afectar el buen rendimiento del sistema operativo.

Al utilizar una gran cantidad de datos, es de vital importancia para el usuario la seguridad, no sólo de un sistema de protección, sino que además se debe considerar el entorno externo del sistema en el que opera. Por lo general, la protección interna de nada sirve, por ejemplo, si la consola del operador está al alcance de personal no autorizado o bien si se pueden extraer los archivos de forma simple. Estos problemas de seguridad no son atribuidos a problemas del sistema operativo, si no que derivan básicamente de problemas administrativos.

Es importante dedicar un esfuerzo a la seguridad considerando el entorno externo en que el sistema opera. La información almacenada en el sistema, así como los recursos propios del sistema deben protegerse contra acceso no autorizado, destrucción o alteraciones, como la modificación accidental de inconsistencias hacia el sistema operativo.

En lo que concierne a la **seguridad de un sistema operativo**, existen varias etapas en las que se debe poner cuidado al tratar de implementar algún medio de seguridad en el sistema, las dos más importantes son: la pérdida de datos y la penetración de intrusos.

Causas más comunes en la pérdida de datos:

1. **Causas naturales**: éstas no tienen nada que ver con el desarrollo del sistema, su implementación ni ambiente de trabajo, pues el sistema operativo se ve afectado por motivos como incendios, inundaciones, terremotos, etc.

2. **Fallas en hardware / software:** se refieren al daño físico en las partes que componen la computadora, como las fallas de CPU, disco duro, memoria, etc., y, fallas en la estructura interna del sistema.

3. **Error humano**: éste puede cometerse de varias formas, el más común es el descuido en la captura de datos; por ejemplo, puede haber errores en la forma de montar algún dispositivo de disco duro o al ejecutar un programa que no debería ejecutarse; la mayoría de estas causas, derivan de la pérdida de información y pueden evitarse manteniendo un constante y adecuado respaldo de la información.

Existen también varios tipos de amenazas a la seguridad que pueden afectar a la integridad del sistema, así como a la información. Para evitar esto se deben cumplir los siguientes **requisitos**:

- **Confidencialidad**: para un sistema es muy importante mantener un nivel de confidencialidad, para que cuando se acceda a la información, ésta sea más fácil de comprender y que sólo los usuarios autorizados tengan los permisos de lectura.

- **Integridad**: es importante que la información que se maneja dentro un sistema sea la más completa posible y que se pueda editar sólo por los usuarios autorizados.

- **Disponibilidad**: hoy en día, el desarrollo de sistemas exige que los elementos estén siempre disponibles y en línea para que los usuarios autorizados tengan acceso a la información.

Al desarrollar un sistema se busca poder garantizar la seguridad propia del sistema operativo; para esto, es necesario implementarla en todos los niveles, ya que por menor que sea la debilidad en el sistema, la información contenida en el mismo podría colapsar. Con la intensión de mantener una seguridad en el sistema, es necesario implementar como mínimo los requisitos anteriores, que permitan establecer el esquema de protección básico para el desarrollo de un sistema operativo.

Además de ofrecer y garantizar durante el desarrollo la seguridad, el sistema es quien debe proveer de mecanismos de protección para la implementación de las características de seguridad. Puesto que, sin la capacidad de autorizar a los usuarios y procesos al controlar su acceso y registro de tareas, sería prácticamente imposible poder implementar estas medidas, pues el sistema estaría restringido para poder llevarlas a cabo. Lo ideal es que el sistema tenga la posibilidad de establecer un enfoque global de protección, soportándose tal vez por mecanismos de protección hardware. La mayor parte de los aspectos de seguridad resultan ser complicados, porque a medida que los usuarios malintencionados conocen las vulnerabilidades de los sistemas de seguridad, éstos pueden ser atacados.

A medida que van creciendo los sistemas y las necesidades de los usuarios, es necesario establecer medidas de seguridad que respondan a los requerimientos del sistema. Existen diferentes tipos de amenazas que pueden llegar afectar la integridad de los sistemas operativos y que pondrían al sistema operativo en un riesgo latente, o bien, dejarlo inservible, las cuales se explican ampliamente a continuación**.**

Las siguientes imágenes “señalan la naturaleza de las amenazas con que se enfrenta cada clase de elemento” (Stallings, 2005, p. 573).

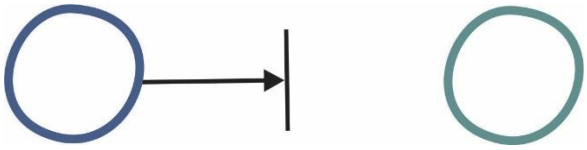
Para comprender el funcionamiento de estas amenazas revisa el flujo normal de información que surge cuando se va generando desde el origen como un archivo o una región de memoria principal hacia su destino; la siguiente imagen representa un ejemplo de este flujo de transmisión de datos sin ningún tipo de bloqueo o interrupción hacia el destino de la información.

Flujo normal de información. Tomada de Stallings, 2005, p. 573.

Interrupción

Como ya se ha mencionado, una de las amenazas más comunes es la **interrupción** que se considera una de las amenazas que afectan a la disponibilidad del sistema operativo puede ser que por un ataque se genere una interrupción de los procesos o de la administración de archivos, esto tiene como consecuencia dejar al sistema inútil.

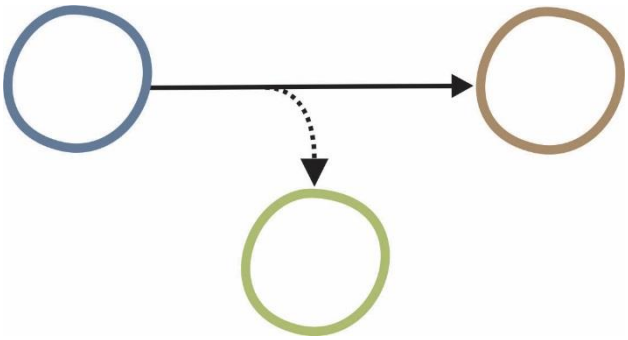
La siguiente imagen muestra un ejemplo de cómo se puede generar una interrupción del flujo de información, la cual es causada por amenaza, bloqueo o falla en el flujo de la información hacia su destino; este ejemplo indica que el sistema operativo quedaría inútil tras esta interrupción.



Flujo de información con interrupción
Tomada de Stallings (2005, p. 573).

Intercepción

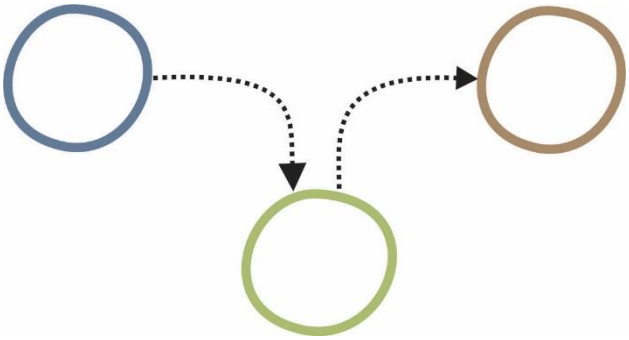
La **intercepción** es otro tipo de amenaza. Un ejemplo típico de este tipo ocurre a partir de un **acceso de forma inesperada** y sin autorización, la cual llega a afectar la integridad de la información. La siguiente imagen representa un ejemplo de flujo de información, donde el envío de la información a su destino es interceptado de forma no autorizada para poner en peligro la integridad de la información.



Flujo de información con interrupción
Tomada de Stallings (2005 p. 573).

Alteración del flujo de información

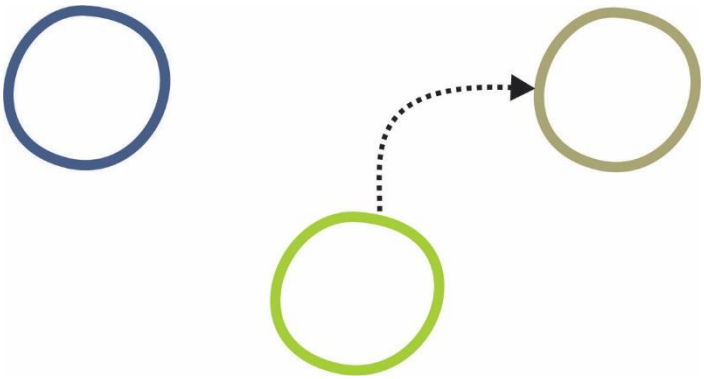
Existe un tipo de amenaza más latente que afecta de forma directa la integridad de la información. En la siguiente imagen se representa un claro ejemplo de una **alteración del flujo de información**, donde una amenaza accede de forma no autorizada y modifica el flujo de información a su destino, alterando de forma directa la información del sistema operativo.



Flujo de información con alteración
Tomada de Stallings (2005, p. 573).

Invención

En la **invención** ocurre que, dentro de un flujo de información, ingresa la amenaza insertando datos falsos al sistema operativo que dejan fuera el origen de la información. En la siguiente imagen se muestra como ocurre esto.



Flujo de información con invención
Tomada de Stallings (2005, p. 573).

El problema más común al que se enfrenta un sistema operativo, y que es bastante difícil de afrontar es la modificación o alteración de la información, lo cual provoca un mal funcionamiento y afecta la integridad de las funciones. En la siguiente tabla se muestran las diferentes amenazas a las que se enfrenta cada clase de elementos.

Elemento	Disponibilidad	Confidencialidad	Integridad
Hardware	Robo o inutilización de equipos, eliminando el servicio.		
Software	Eliminación de programas, denegando el acceso a los usuarios.	Realización de copias no autorizadas del software.	Alteración de un programa en funcionamiento haciendolo fallar durante la ejecución o haciendo que realice alguna tarea no pretendida.
Datos	Eliminación de archivos, denegando el acceso a los usuarios.	Lecturas de datos no autorizadas. Un análisis de datos estadísticos revela datos ocultos.	Modificación de archivos existentes o invención de nuevos archivos.
Lineas de comunicación	Destrucción o eliminación de mensajes. Las líneas de comunicación o redes se hacen No disponibles.	Lectura de mensajes. Observación de la muestra de tráfico de mensajes.	Mensajes modificados, retardos, reordenados o duplicados. Invención de mensajes falsos.

Elementos que intervienen en la seguridad e integridad de la información
Tomada de Stallings (2005, p. 575).

Las amenazas a la integridad de la información son una preocupación constante para los desarrolladores de sistemas operativos, ya que la confidencialidad, disponibilidad e integridad del sistema estarían en peligro, debido a los diferentes tipos de amenazas que están a la orden del día. De acuerdo con su complejidad éstas se clasifican en: pasivas y activas.

Las **amenazas pasivas** son aquellas que sólo interceptan el flujo de la información afectando la confidencialidad de la misma, este tipo de amenazas tienen el objetivo de espiar y divulgar el contenido de la información. Estas amenazas son difíciles de ser detectadas, pues no generan alteración al administrador de archivos y tampoco alteran la funcionalidad del sistema operativo. Una posible medida podría ser la prevención para evitar el filtrado.

Las **amenazas activas** son delicadas para la integridad del sistema. En éstas se encuentra latente la alteración del flujo de datos, o bien, la creación de un flujo falso que pone en riesgo el funcionamiento normal del sistema operativo. En la mayoría de los casos una amenaza activa es difícil de prevenir en forma absoluta, pues requeriría de una protección de todos los servicios y rutas de comunicación.

3.1.1. Clasificaciones de la seguridad

En la actualidad, la clasificación de la seguridad proviene de acuerdo con el requisito propio de cada sistema operativo, los criterios existentes para determinar la clasificación de la seguridad, de acuerdo con “el departamento de defensa de los EEUU, especifica cuatro clasificaciones de seguridad para los sistemas: A, B, C, D, esta especificación se usa ampliamente en dicho país para determinar la seguridad de una instalación y modelar soluciones de seguridad” (Silberschatz, 2006, p. 547).

Tomando como base de referencia dicha clasificación, se ubican cuatro niveles de seguridad de acuerdo con los requerimientos necesarios que debe cumplir el desarrollo de un sistema operativo:

- **Nivel D**. Este nivel de protección es considerado como el nivel mínimo que requiere la protección de la integridad del sistema operativo. Los sistemas operativos diseñados con procesos individuales o mono usuarios, por su naturaleza de administración simple de archivos no es muy común o frecuente el uso de un nivel superior de la información.

- **Nivel C**. Está considerado como un nivel superior al mínimo de seguridad requerido para un sistema operativo. En este nivel se implementan mecanismos de protección de recursos que estarían bajo la responsabilidad de los usuarios, quienes tendrán los privilegios de realizar modificaciones. Se considera que los sistemas comerciales como Linux, Windows, por mencionar sólo unos, caen en esta categoría, este nivel a su vez se clasifica en dos subniveles:
    - C1. Este subnivel integra algunos controles que permiten a los usuarios proteger su información de lectura, escritura o eliminación de archivos.
    - C2. Este subnivel es la parte superior del nivel C1 donde existe un usuario con privilegios superiores que permiten la auditoría de la información, tal como un administrador del sistema.

- **Nivel B**. Este nivel seguridad cuenta con los parámetros de los niveles anteriores, adicionando para el incremento de la seguridad, la amplitud de controles y etiquetas de seguridad, dominio y mecanismos estructurados de protección que son de utilidad para el refuerzo de restricción de acceso no autorizado.

- **Nivel A**. Técnicamente este nivel cumple con todas las características que tiene un nivel B, pero utiliza niveles específicos de diseño y técnicas de validación para el filtrado de la información.

3.1.2. Verificación de autenticidad de usuarios

Frecuentemente se van incrementado los niveles de complejidad para la creación de contraseñas o password utilizados para la validación de acceso a la información. Muchas aplicaciones codifican, por medio de algún algoritmo de encriptación este método provee

un nivel de seguridad más avanzado que la utilización simple de caracteres, comparándolo con la lista de contraseñas disponibles para la validación. Resulta un tanto simple para algunos usuarios o aplicaciones malintencionadas descifrar el contenido y complejidad de las contraseñas para vencer la autenticidad de éstas, teniendo como resultado el acceso a la información de algún sistema; ya sea para modificar el flujo de la información como en la intercepción, o algo más delicado, como en la inversión.

Según Tanenbaum (2003):

"Muchos esquemas de protección se basan en el supuesto de que el sistema conoce la identidad de cada usuario. El problema de identificar a los usuarios cuando inician se denomina verificación de autenticidad de usuarios. La mayor parte de los métodos de verificación de autenticidad se basan en identificar algo que el usuario conoce, tiene o es contraseña".

La **validación** o **verificación física** es muy amplia y tiene muchas aplicaciones, ventajas y desventajas en comparación con otros medios de verificación; el desarrollador es quien planea qué tipo o clasificación de seguridad se implementaría para la validación y acceso de usuarios; algunas de las medidas preventivas que se deben considerar para el incremento de la seguridad, son:

- Registrar los inicios de sesión y log de actividades.
- Bloqueo de inicios de sesión por fecha o por intentos erróneos de validación.
- Encriptación, caducidad y modificación constante de contraseñas.
- Especificar estación de trabajo válida para el acceso al sistema.

Existen varias formas sencillas para la protección de contraseñas: una es el cambio regular de ésta, para evitar que algún intruso pueda descifrarla. Hay otras variaciones utilizadas para la validación y autentificación de usuarios: las identificaciones físicas: éstas se basan en algún dispositivo físico para la autorización de ingreso al sistema, el cual utiliza un algoritmo de validación completamente distinto a una contraseña, estos dispositivos son la llave de acceso al sistema y son útiles para saber si se trata del usuario propietario.

Algo que ocurre de forma común es que las tarjetas magnéticas que son validadas por un lector magnético el cual determina si es válida o no la tarjeta no tienen un alto grado de

efectividad en cuestiones de seguridad, ya que es común el extravío o falsificación de estos dispositivos. Por otro lado, las huellas digitales, lectura de retina o patrón de voz, forman parte de un método físico de verificación de usuarios. Estos métodos consisten en la verificación y autenticidad física de las características únicas de cada usuario, las cuales resultan difíciles de falsificar.

Estas medidas son algunas de las más usadas y son, bajo el criterio del desarrollador, el tipo de seguridad a implementar de acuerdo con el nivel y uso de la información que sea considerada como importante.

3.1.3. Validación y amenazas al sistema

Hasta el día de hoy todavía existen amenazas contra los sistemas operativos cada vez más sofisticadas, que aprovechan los puntos débiles; por eso, es importante que se validen y se consideren al diseñar un sistema operativo.

Principales amenazas que se deben validar al diseñar un sistema operativo

Como se ha dicho existen muchos tipos de amenazas, por lo cual, lo mejor es prevenir todas éstas desde el diseño, teniendo en cuenta buenas prácticas de prevención. Estas amenazas se conocen como **virus** que atacan y actúan de muchas maneras; a continuación se mencionan algunas de ellas:

Troyano

Ingresa cuando el sistema no tiene seguridad, permite accesos a otros tipos de archivos que hacen daño, en ciertos momentos se activan y ejecutan actividades que dañan la consistencia y funcionalidad del sistema operativo.

Exploits (secuencia de aprovechamiento)

Localizan un punto débil en el sistema y ejecutan acciones que no deberían ser ejecutadas causando caos en el sistema.

Rootkits (secuencia de nulidad)

Su origen está en el lenguaje UNIX y son herramientas que entran como administradores tomando el control.

Backdoors (secuencia de salida emergente)

Como su traducción al español lo dice, significa “puerta trasera”, la cual es abierta para que otros sistemas dañinos puedan entrar. Éstos abren una puerta trasera en el sistema para que el creador de malware entre en el sistema y lo domine a su antojo. El objetivo es crear una red computadoras infectadas con el mismo.

Keyloggers (registro de teclas o pulsaciones)

Registra la pulsación de las teclas y clic para enviarlas a un usuario no autorizado puede instalarse como hardware o aplicación.

Para consultar algunas buenas prácticas sobre la seguridad en los sistemas informáticos y recomendaciones para asegurar los sistemas operativos, consulta la obra de Jiménez Rojas (2008), en los *Materiales de desarrollo de la unidad 3.*

3.2. Concepto y objetivos de protección

Por una necesidad de mantener la integridad y confiabilidad de los sistemas, la protección de éstos se vuelve prioridad para todo desarrollo de sistemas operativos; una de las necesidades de la protección es impedir el acceso y violación a la información del sistema de archivos.

La protección es la fuente de control y restricción de acceso a los sistemas, administración de los recursos y procesos, su objetivo principal es proveer de un mecanismo que tenga la facultad de establecer políticas de restricción, y crear bloqueos a usuarios malintencionados.

Cuando un sistema o parte del sistema no está protegido, no tiene la confiabilidad, integridad y mucho menos la disponibilidad de la información. Debido al mal uso de la

información que puede llegar a sufrir modificaciones derivadas del acceso de usuarios malintencionados. Silberschatz (2006) establece que:

> Frecuentemente, podemos utilizar un principio director a lo largo de un proyecto, como puede ser el diseño de un sistema operativo. Ajustarnos a este principio simplifica las decisiones de diseño y hace que el sistema continúe siendo coherente y fácil de comprender. Uno de los principios directores clave y que ha resistido al paso del tiempo a la hora de proporcionar protección es el **principio del mínimo privilegio**. Este principio dicta que a los programas, a los usuarios, incluso a los sistemas se les concedan únicamente los suficientes privilegios para llevar a cabo a sus tareas.

Se considera que cuando un sistema operativo cumple con el principio de mínimo privilegio, durante el desarrollo cuando se integran características y/o mecanismos de protección que cubran las necesidades de poder minimizar los daños causados por usuarios malintencionados. Este principio tiene la bondad de poder ofrecer un entorno más seguro al sistema, por ello es de suma importancia que sea considerado durante la planificación y desarrollo, de lo contrario no lograría su objetivo de protección.

Un ejemplo que cubre con el principio mínimo de privilegio es determinarle al sistema operativo qué modificaciones se pueden realizar dentro del mismo sistema operativo, tales como instalación de programas o modificaciones a los ya instalados.

3.2.1. Mecanismos de protección

Durante el desarrollo del sistema, la parte fundamental que se debe considerar es la seguridad y protección del sistema operativo; se deben de tener en cuenta al menos los problemas potenciales mencionados en temas anteriores, para solucionar esa parte se pueden utilizar técnicas para establecer las políticas y mecanismos de protección.

De acuerdo con Tanenbaum, (2003, p. 447) “en algunos sistemas, la protección se impone mediante un programa llamado monitor de referencias. Cada vez que se intenta un acceso a un recurso que pudiera estar protegido, el sistema pide primero al monitor de referencias verificar” si un acceso está permitido.

Los diferentes tipos de mecanismos de protección se encuentran clasificados en:

- **Dominio de protección**. Este punto considera al sistema de cómputo como grupo global de software y hardware, cada una de las partes que lo conforman tienen su propio nombre, características y objetivo; mediante el cual se podrán realizar operaciones con archivos y manejo de información. Durante la ejecución de un proceso sólo se podrá tener acceso a los recursos que tiene autorizados para realizar sus tareas. En la siguiente imagen, se muestra un ejemplo de tres dominios, cada uno contiene sus propios objetos con la autorización para poder escribir (W), leer (R) y ejecutar (X), se puede apreciar que la impresora está en dos dominios distintos al mismo tiempo, debido a que los archivos de cada dominio hacen referencia a la misma impresora conectada al sistema de cómputo.

Dominios de protección con sus propios objetos y derechos de aplicación
Tomada de Tanenbaum (2003, p. 447).

- **Listas de control de acceso**. El objetivo de esta técnica consiste en asociar los registros de una lista ordenada que contenga la mayor cantidad de dominios y que pueda ingresar al objeto.
- **Capacidades**. Este método distingue las características de cada objeto, las cuales indican las operaciones permitidas que puede realizar, así la lista muestra los objetos y sus capacidades para lograr procesar la información y con esto clasifica los objetos de acuerdo con sus capacidades facilitando el compartimiento de subdominios.
- **Matriz de acceso**. Este modelo de protección puede completar su estructura por medio de abstracción de datos compuestos por una colección de derechos de acceso a la información, este tipo de modelo proporciona el mecanismo factible para

definir e implementar un control específico para la asociación de procesos y dominios de forma dinámica y estática, los procesos deben poder conmutar de un dominio a otro y permitir la modificación controlada del contenido de las entradas de la matriz de acceso, éstas por lo general requieren de operaciones adicionales, tales como: copy, owner and control.

3.2.2. Funciones del sistema de protección

La principal característica que distingue un sistema de protección es que cumpla con los requerimientos de protección de los procesos del sistema contra los procesos de los usuarios; proteja los procesos de los usuarios contra los de otros usuarios; proteja la administración de la memoria, y proteja los dispositivos.

Para cumplir con dicha característica, durante el desarrollo del sistema operativo es conveniente dar flexibilidad a la estructura de los datos para imponer una variedad de políticas y mecanismos de protección, exigiendo que cumpla con el mínimo de requerimientos para el control de pérdida de datos.

Más adelante se mostrarán algunos de los mecanismos que se pueden utilizar para asegurar los archivos, segmentos de memoria y otros dispositivos administrados por el sistema operativo. A partir de ellos, se evidenciará que el principal objetivo de todo sistema operativo es mantener la confiabilidad, integridad y disponibilidad del sistema. El uso de estos mecanismos contribuye evitando el mayor número de amenazas que pueden afectar el rendimiento del sistema.

Las funciones principales para proveer un buen sistema de protección, son:

- **Establecer políticas de uso de recursos**. Las políticas son descripciones de lo que se desea proteger y cómo hacerlo y se pueden establecer por usuario, administrador del sistema o bien sobre el diseño y/o desarrollo.
- **Mecanismos de protección**. Su uso común es controlar el acceso de programas o procesos a los recursos del sistema.

- **Monitoreo de amenazas**. Se pueden establecer rutinas de control que permitan ingresar al sistema, o bien, rechazar la petición de acceso a determinadas aplicaciones o procesos.

Queda claro que es importante y necesaria la protección de un sistema operativo, para prevenir violaciones malintencionadas de acceso al flujo de datos de un proceso dentro del sistema. Por ello, la conveniencia de asegurar cada componente que forme parte del sistema, pues un recurso o componente no asegurado no puede defenderse contra alguna variación al flujo de datos.

3.2.3. Implementación de matrices de acceso

Una matriz de acceso tiene como finalidad la protección del sistema, el cual se puede interpretar como una tabla, en donde se definen filas que representan los sujetos que tienen acceso a los objetos y las columnas que representan los objetos (que pueden ser: discos, cintas, procesadores, otros dispositivos de almacenamiento). Todas las entradas a la matriz van en relación con una serie de derechos de acceso a la información. La siguiente tabla muestra un ejemplo mínimo de una matriz de acceso:

SUJETOS / OBJETOS	**FUNCION 1**	**IMPRESORA**
D1	LECTURA	IMPRIMIR
D2	LECTURA / ESCRITURA	

Al iniciar con la implementación de una matriz de acceso y al ir llenando datos, la matriz tendrá espacios vacíos, esto lleva a una serie de desventajas en el rendimiento del sistema operativo. Los métodos para la implementación de una matriz de acceso son de utilidad para aquellas matrices dispersas y de poca utilidad, a continuación se describe cada uno de estos métodos para su implementación:

- **Tabla global.** Este primer método, tiene la función más simple de acceso para una tabla compuesta por tripletas (dominio, objeto, conjunto-derechos), suponiendo que se tiene un dominio *Da*, un objeto *Oa* y sus derechos de lectura Ra. Si *Da* intenta el acceso en *Oa* mediante alguna operación Ma (matriz de acceso*)*, se analizará toda

la tabla en búsqueda de la tripleta <Da, Oa, Ra> y que Ma pertenezca a Ra, si Da encuentra Oa realiza Ra, en caso contrario sigue realizando el proceso de búsqueda, o bien, hasta que se genere algún error. Este tipo de implementación resulta estar creciendo considerablemente conforme van generándose operaciones de E/S, por lo que no es posible mantenerla en memoria principal. Una desventaja que presenta es que se tienen que generar operaciones adicionales de E/S.

- **Lista de acceso para los objetos.** En este método se van almacenando los datos por columna en la matriz para ir asociando cada objeto dentro de una lista pares ordenados. Siguiendo el ejemplo de la tripleta anterior, se tiene que si Da intenta ingresar a Oa mediante la operación M se permite si se encuentra <Da, Ra> y M tenga pertenencia en Ra. Su ventaja radica en la facilidad de poder agrupar los dominios.

- **Listas de capacidades para los dominios.** La capacidad de este método, consiste en que los derechos de acceso a un objeto se almacenan en la matriz por filas con su dominio; por lo general las listas de capacidades están asociadas con un dominio, pero un proceso que se ejecute en el dominio no podrá ingresar de forma directa sobre ésta. Cada elemento de la matriz es denominado como capacidad y la lista de capacidades debe protegerse por el sistema operativo.

  “Las listas de capacidades se propusieron originalmente como una especie de puntero seguro, para satisfacer la necesidad de protección de los recursos que se preveía que iban a ser necesarios a medida que los sistemas informáticos multiprogramados se generalizaran” (Silberschatz, 2006, p. 494).

- **Mecanismo de bloqueo-clave.** “El esquema de bloqueo-clave, es un compromiso entre las listas de acceso y las listas de capacidades. Cada objeto tiene una lista de patrones de bit distintos, denominados bloqueos. De forma similar, cada dominio tiene una lista de patrones de bit distintos, denominados claves” (Silberschatz, 2006, p. 494). Para que un determinado proceso sea ejecutado dentro de un dominio

podrá hacerlo si sólo cuenta con una clave que corresponda con uno de los bloqueos del objeto.

La mayoría de los sistemas requieren de alguna implementación de matrices de acceso para localizar la información para un proceso determinado y es decisión del desarrollador del sistema operativo determinar qué método sería el más óptimo para la implementación de las matrices de búsqueda.

Cierre de la unidad

Has concluido la tercera unidad del curso. A lo largo de ésta se revisaron conceptos básicos sobre la seguridad y respecto a cómo se clasifica la seguridad, cómo se verifica la autenticidad de los usuarios y las principales amenazas que se deben considerar al diseñar un sistema operativo; en cuanto a la protección, qué mecanismos se utilizan para el sistema de protección y cómo se implementan la matrices de acceso.

Es aconsejable que revises nuevamente la unidad en caso de que los temas que se acaban de mencionar no te sean familiares o no los recuerdes; de no ser el caso, ya estás preparado(a) para seguir con la Unidad 4. Diseño de sistemas operativos, donde continuarás aprendiendo bases para el diseño. Todo esto con el fin de obtener un prototipo final al concluir la última unidad de la asignatura Programación de sistemas operativos.

Para saber más

Si deseas conocer más información acerca de la seguridad de los sistemas operativos, se recomienda visitar los siguientes sitios:

- Seguridad de los Sistemas Operativos Infosec UNAM (2017). *Seguridad en Sistemas Operativos*. *Publicaciones.* Ciudad de México: Laboratorio de Seguridad Informática Centro Tecnológico, FES Aragón, UNAM. http://infosec.aragon.unam.mx/tematicas/view/89

- Facultad de Ingeniería UNAM (n.d.). *Seguridad Informática.* Facultad de Ingeniería. Laboratorio de Redes y Seguridad. http://redyseguridad.fi-p.unam.mx/proyectos/seguridad/ServAutenticacion.php

Fuentes de consulta

- Candela, S. y García, C. (2007).*Fundamentos de sistemas operativos. Teoría y ejercicios resueltos*. España: Paraninfo.
- Jiménez Rojas, J.R. (2008,10 de abril).La seguridad informática y el usuario final. En *Revista Digital Universitaria, 9* (4). México: Coordinación de Publicaciones Digitales. DGSCA-UNAM. Recuperado de http://www.revista.unam.mx/vol.9/num4/art20/art20.pdf
- Ortiz Pabón, H. J. (2005). *Sistemas operativos modernos*. Medellín: Universidad de Medellín.
- Rodríguez Fernández, L. E. (2010). *Diseño y desarrollo de una interfaz de sistema operativo mediante una entidad de inteligencia artificial con soporte para lenguaje natural*. Raleigh. Carolina del Norte-Vigo: Lulú Press-Universidad de Vigo. Departamento de Enxeñería Telemática.
- Silberschatz, A. (2006). *Fundamentos de sistemas operativos*. (7a. ed.). España: McGraw-Hill.
- Stallings, W. (2005). *Sistemas operativos modernos: aspectos internos y principios de diseño*. México: Pearson, Prentice Hall.
- Tanenbaum, A. (2003).*Sistemas operativos modernos*. México: Pearson Educación.