Instituto Politécnico Nacional

TAREA 15

Sistemas Operativos

Integrantes:
Mora Ayala José Antonio
Ramírez Cotonieto Luis Fernando
Torres Carrillo Josehf Miguel Ángel
Tovar Jacuinde Rodrigo

Profesor:
Cortés Galicia Jorge

Seguridad

El problema de la seguridad

Estos mecanismos funcionan adecuadamente mientras que los usuarios respeten los usos previstos y el tipo de acceso que se hubiera pensado para esos recursos.

Las violaciones de seguridad o la mala utilización de un sistema pueden clasificarse en dos categorías: intencionadas (maliciosas) o accidentales.

Los mecanismos de protección forman la base de la defensa frente a posibles accidentes.

- Ruptura de la confidencialidad. Este tipo de violación implica la lectura no autorizada de determinados datos (o el robo de información). Típicamente, el objetivo de los intrusos es una ruptura de la confidencialidad. La captura de datos secretos en un sistema oen unflujo de datos, como por ejemplo información relativa a tarjetas de crédito o información personal para fingir una identidad ficticia, puede reportar al intruso un beneficio monetario directo.

- Ruptura de la integridad. Este tipo de ataque implica la modificación no autorizada de los datos. Estos ataques pueden, por ejemplo, provocar que se atribuya una cierta responsabilidad a alguien que es inocente o que se modifique el código fuente de una aplicación comercial de cierta importancia.

- Ruptura de la disponibilidad. Esta violación de seguridad implica la destrucción no autorizada de datos. Algunos atacantes prefieren causar daño y adquirir un cierto renombre en lugar de obtener beneficios financieros. La sustitución de la página de entrada de un sitio web es un ejemplo común de este tipo de ruptura de la seguridad.

- Robo de servicio. Este tipo de violación de seguridad implica el uso no autorizado de recursos. Por ejemplo, ur intruso (o un programa de intrusión) puede instalar un demonioen un sistema que actúe como servidor de archivos.

- Denegación de servicio. Esta violación de seguridad implica impedir el uso legítimo del sistema. Los ataques de denegación de servicio (DOS, denial-of-service) son en ocasiones accidentales.

Los atacantes utilizan diversos métodos estándar en sus intentos de romper la seguridad delos sistemas. El más común es la mascarada, en la que un participante en una comunicación preten de ser otra persona (u otro host).
Mediante la mascarada, los atacantes rompen la autenticación.

Otro ataque común consiste en reproducir un intercambio de datos previamente capturado.

Otro tipo de ataque es el ataque por interposición (man-in-the-middle), en el cual un atacante se introduce dentro del flujo de datos de una comunicación, haciéndose pasar por el emisor a ojos del receptor y viceversa.

resulta imposible garantizar una protección absoluta del sistema frente a los abusos de carácter malicioso, pero podemos hacer que el coste para aquel que lo intencos teseatanalto como para disuadir a la mayoría de los intrusos.

Para proteger un sistema, debemos adoptar las necesarias medidas de seguridad en cuatro ruveles distintos:

1. Físico. El nodo o nodos que contengan los sistemas informáticos deben dotarse de medidas de seguridad físicas frente a posibles intrusiones armadas o subrepticias por parte de potenciales intrusos. Hay que dotar de seguridad tanto a las habitaciones donde las máquinas residan como a los terminales o estaciones de trabajo que tengan acceso a dichas máquinas.

2. Humano. La autorización de los usuarios debe llevarse a cabo con cuidado, para garantizar que sólo los usuarios apropiados tengan acceso al sistema. Sin embargo, incluso los usuarios autorizados pueden verse "motivados" para permitir que otros usen su acceso.

3. Sistema operativo. El sistema debe autoprotegerse frente a los posibles fallos de seguridad accidentales o premeditados. Un proceso que esté fuera de control podría llegar a constituir un ataque accidental de denegación de servicio. Asimismo, una cierta consulta a un servicio podría conducir a la revelación de contraseñas o un desbordamiento de la pila podría permitir que se iniciara un proceso no autorizado. La lista de posibles fallos de seguridad es casi infinita.

4. Red. Son muchos los datos en los modernos sistemas informáticos que viajan a través de líneas arrendadas privadas, de líneas compartidas como Internet, de conexiones inalámbricas o de líneas de acceso telefónico.

Si queremos poder garantizar la seguridad del sistema operativo, es necesario garantizar la seguridad en los primeros dos niveles.

Cualquier debilidad en uno de los niveles altos de seguridad (físico o humano) permitirá puentear las medidas de seguridad que son estrictamente de bajo nivel (del nivel del sistema operativo)

Para soportar un esquema global de protección hacen falta mecanismos de protección hardware.

Amenazas relacionadas con los programas

Los procesos son, junto con el kernel, el único medio de realizar un trabajo útil en una computadora. Por tanto, un objetivo común de los piratas informáticos consiste en escribir un programa que cree una brecha de seguridad o que haga que un proceso normal cambie su comportamiento y cree esa brecha de seguridad.

Caballo de Troya

Muchos sistemas tienen mecanismos para permitir que programas escritos por unos usuarios sean ejecutados por otros. Si estos programas se ejecutan en un dominio que proporcione los derechos de acceso del usuario ejecutante, los otros usuarios podrían utilizar inapropiadamente estos derechos.

Lo que se hace es buscar en esa ruta un archivo con dicho nombre y ejecutar el archivo. Todos los directorios de esa ruta de búsqueda deben ser seguros, porque de lo contrario podría introducirse un caballo de Troya en la ruta de ejecución del usuario y ejecutarse accidentalmente.

Una variante de caballo de Troya es un programa que emula el tipico programa de inicio de sesión. Un usuario que no esté advertido tratará de iniciar la sesión en un terminal y observará que aparentemente ha escrito mal su contraseña; después, vuelve a intentarlo y esta vez lo hace con éxito. Lo que ha sucedido es que su clave de autenticación y su contraseña han sido robadas por el emulador de inicio de sesión, que fue dejado ejecutándose en el terminal por parte del ladrón.

Otra variante del caballo de Troya es el spyware. Los programas spyivare acompañan en ocasiones a ciertos programas que el usuario haya decidido instalar.

El spyware es un ejemplo a pequeña escala de un problema a gran escala: la violación del principio del mínimo privilegio. En la mayoría de las circunstancias, un usuario de un sistema operativo no necesita instalar demonios de red. Dichos demonios llegan a instalarse debido a dos errores.

En primer lugar, un usuario podría estar operando con más privilegios de los necesario, permitiendo que los programas que ejecute tengan más acceso al sistema del necesario; este es un caso de error humano, que es una de las vulnerabilidades de seguridad más habituales.

En segundo lugar, un sistema operativo puede permitir, de manera predeterminada, más privilegios de los que un usuario normal necesita.

Puerta trasera

El diseñador de un programa o un sistema puede dejar detrás suyo un agujero en el software ques sólo él sea capaz de utilizar.

Podría incluirse una puerta trasera inteligente dentro del propio compilador. El compilador generaría código objeto estándar junto con la puerta trasera, independientemente de qué código puente se estuviera compilando. Esta actividad es particularmente peligrosa, ya que un análisis del código fuente del programa no permitiría revelar ningún problema. Sólo el código fuente del compilador contendría la información necesaria para detectar la puerta trasera

Las puertas traseras plantean un difícil problema porque, para detectarlas, tenemos que azar todo el código fuente de todos los componentes de un sistema.

Desbordamiento de pila y de búfer

El ataque por desbordamiento de pila o de búfer es la forma más común para que un atacante externo al sistema, a través de una conexión de red o de acceso telefónico, obtenga acceso no autorizado al sistema objetivo. Los usuarios autorizados del sistema también pueden utilizar este tipode ataque para escalar sus privilegios.

Utilizando un método de prueba y error, o examinando el código fuente del programa atacado, si es que está disponible, el atacante determina la vulnerabilidad y escribe un programa para hacer lo siguiente:

1. Desbordar un campo de entrada, un argumento de línea de comandos o un búfer de entrada (por ejemplo, en un demonio de red) hasta escribir en la zona correspondiente a la pila.

2. Sobreescribir la dirección actual de retorno de la pila, sustituyéndola por la dirección de los códigos de ataque cargados en el paso 3.

3. Escribir un fragmento simple de código en el siguiente espacio de la pila, que incluye loscomandos que el atacante quiera ejecutar, como por ejemplo arrancar un programa shell.

Cuando se invoca una función en una arquitectura informática típica, las variables definidas
localmente a la función (conocidas en ocasiones con el nombre de variables automáticas), los parámetros pasados a la función y la dirección a la que volverá el control cuando la función termine

El ataque por desbordamiento de búfer resulta especialmente pernicioso porque pueden lanzarse ataques de un sistema a otro y ese código de ataque puede ser transmitido.

Virus

Los virus son auto-replicantes y están diseñados para "infectar" otros programas. Pueden causar estragos en un sistema modificando o destruyendo archivos y provocando funcionamientos inadecuados de los programas y fallos catastróficos del sistema.

Un virus es el fragmento de código integrado dentro de un programa legítimo.

Al igual que la mayoría de los ataques de penetración, los virus son muy específicos de las arquitecturas, de los sistemas operativos y de las aplicaciones.

Los virus suelen propagarse a través de correo electrónico, siendo el correo basura el vector más común. También pueden propagarse cuando los usuarios descargan programas infectados desde servicios de compartición de archivos de Internet o cuando intercambian discos infectados.

¿Cómo funcionan los virus? Una vez que un virus alcance una máquina objetivo, un programa conocido como lanzador de virus inserta el virus en el sistema. El lanzador de virus es usualmente un caballo de Troya, que se ejecuta por otras razones pero cuya principal actividad consiste en instalar el virus. Una vez instalado, el virus puede hacer una de varias cosas. Existen literalmente miles de virus distintos, pero se los puede clasificar en varias categorías generales.

Archivo. Un virus de archivo estándar infecta un sistema insertándose a un archivo y modi-
ficando el inicio del programa para que la ejecución salte al código del virus.

Arranque. Un virus de arranque infecta el sector de arranque del sistema, ejecutándose
cada vez que el sistema se arranca y antes de que se cargue el sistema operativo.

Macro. La mayoría de los virus están escritos en un lenguaje de bajo nivel, como por ejem-
plo ensamblador o C. Los virus de macro están escritos en un lenguaje de alto nivel, como
Visual Basic. Estos virus se activan cuando se inicia un programa capaz de ejecutar
Código fuente. Un virus de código fuente busca código fuente y lo modifica para incluir :
el virus y ayudar a su distribución.

Polimórfico. Este tipo de virus cambia cada vez que se instala, para evitar su detección por parte del software antivirus. Los cambios no afectan a la funcionalidad del virus, sino que sólo modifican la signatura del virus.

Cifrado. Un virus cifrado incluye código de descripción junto con el virus cifrado, dejen nuevo para evitar la detección. El virus se descifra primero y luego se ejecuta.

Encubierto. Este insidioso virus trata de evitar la detección modificando partes del sistema que podrían ser usadas para detectarlo.

Túnel. Este tipo de virus trata de evitar la detección por los programas antivirus instalándose asimismo en la cadena de rutinas de tratamiento de interrupciones.

Multiparte. Los virus de este tipo son capaces de infectar múltiples partes de un sistema, incluyendo los sectores de arranque, la memoria y los archivos.

Acorazado. Los virus acorazados están codificados de tal manera que resultan difíciles de desentrañar y de comprender por parte de los investigadores que desarrollan programas antivirus.

Esta amplia variedad de virus es probable que continúe creciendo.

Generalmente, los virus son el tipo de ataque de seguridad más dañino; y como son bastante efectivos, continuarán desarrollándose y distribuyéndose.

Amenazas del sistema y de la red

Las amenazas basadas en programas utilizan típicamente un fallo en los mecanismos de protección de un sistema para atacar a los programas.

Las amenazas del sistema y de la red crean una situación en la que se utilizan inapropiadamente los recursos del sistema operativo y los archivos del usuario.

Es importante destacar que las mascaradas y los ataques por reproducción también resultan comunes en las redes que interconectan los sistemas.

La generalización de este concepto es que el compartir secretos (para demostrar la identidad y en forma de claves de cifrado) es una necesidad para la autenticación del cifrado, y que esa compartición resulta más sencilla en aquellos entornos (por ejemplo con un único sistema operativo) en los que existan métodos seguros de compartición.

Gusanos

Un gusano es un proceso que utiliza un mecanismo de reproducción para afectar al rendimiento del sistema. El gusano crea copias de sí mismo, utilizando recursos del sistema y en ocasionesimpidiendo operar a todos los demás procesos.
En las redes informáticas, los gusanos son particularmente potentes, ya que pueden reproducirse de un sistema a otro y colapsar una red completa.

Escaneo de puertos ea

El escaneo de puertos no es un ataque, sino más bien un método para que los piratas informáticos detecten las vulnerabilidades del sistema que puedan ser atacadas.

El escaneo de puertos se realiza normalmente de forma automatizada, lo que implica utilizar una herramienta que trate de crear una conexión TCP/IP a un puerto o rango de puertos específicos.

Puesto que los escaneos de puertos son detectables, se suelen realizar desde sistemas zombi. Dichos sistemas son máquinas independientes y previamente comprometidas que están prestando un servicio normal a sus propietarios al mismo tiempo que son utilizadas inadvertidamente para propósitos inconfesables, incluyendo la realización de ataques por denegación de servicio y la retransmisión de correo basura.

Los zombis hacen que resulte particularmente difícil perseguir a los piratas informáticos, ya que resulta muy difícil determinar el origen del ataque y la persona que lo ha iniciado.

Lanzar un ataque que impida el uso legítimo de un sistema resulta, frecuentemente, más sencillo que irrumpir en una máquina o instalación.

Los ataques de denegación de servicio se realizan generalmente a través de la red. Se los puede clasificar en dos categorías. El primer caso es el de los ataques que consumen tantos recursos de la máquina atacada que prácticamente no puede realizarse con ella ningún trabajo útil.

Generalmente, es imposible prevenir los ataques de denegación de servicio. Los ataques utilizan los mismos mecanismos que la operación normal. Todavía más difíciles de prevever y de solucionar son los ataques distribuidos de denegación de servicio (DDOS, distributed denial-of-service attacks). Estos ataques se inician desde múltiples sitios a la vez, dirigidos hacia un objetivo común, normalmente por parte de programas zombis.

La criptografía como herramienta de seguridad

Existen muchas defensas frente a los ataques informáticos, que abarcan toda la gama que va desde la metodología a la tecnología. La herramienta de carácter más general que está a disposición de los usuarios y de los diseñadores de sistemas es la criptografía.

Comúnmente, se utilizan las direcciones de red para inferir los emisores y receptores potencia les de los mensajes que circulan por la red. Los paquetes de red llegan con una dirección de 0-25, como por ejemplo una dirección IP. Y cuando una computadora envía un mensaje, indicas quién es el receptor pretendido del mismo especificando una dirección de destino.

La criptografía moderna se basa en una serie de secretos, denominados claves, que se distribuyen selectivamente a las computadoras de ES una red y se utilizan para procesar mensajes.

La criptografía permite al receptor de un mensaje verificar que el mensaje ha sido creado por alguna computadora que posee una cierta clave: esa clave es el origen del mensaje. De forma similar, un emisor puede codificar su mensaje de modo que sólo una computadora que disponga de una cierta clave pueda decodificar el mensaje, de manera que esa clave se convierte en el destino.

Cifrado ES

Un algoritmo de cifrado consta de los siguientes componentes:

+ Un conjunto K de claves.
+ Un conjunto M de mensajes.
+ Un conjunto C de mensajes de texto cifrado.
Una función E: K > (M —> C). Es decir, para cada k e K, E(k) es una función para generar
mensajes de texto cifrado a partir de los mensajes de texto en claro. Tanto E como E(k) para cualquier k deben ser funciones computables de manera eficiente.

Una función D: K>(C —« M). Es decir, para cada k e K, D(k) es una función para generar
mensajes de texto en claro a partir de los mensajes de texto cifrado. Tanto D como D(k) para cualquier k deben ser funciones eficientemente computables.

Existen dos tipos principales de algoritmos de cifrado: simétricos y asimétricos.

Cifrado simétrico

En un algoritmo de cifrado simétrico, se utiliza la misma clave para cifrar y para descifrar, es decir, E(k) puede deducirse a partir de D(k) y viceversa.

Un algoritmo de cifrado de flujo está diseñado para cifrar y descifrar un flujo de bytes o bits, en lugar de un bloque. Esto resulta útil cuando la longitud de una comunicación pueda hacer que un algoritmo de cifrado de bloques sea demasiado lento. La clave se introduce en un generador de bits pseudoaleatorio, que es un algoritmo que trata de producir bits aleatorios. La salida del generador, cuando se le alimenta con una clave, es lo que se denomina flujo de clave. Un flujo de clave es un conjunto infinito de claves que pueden usarse para el flujo de texto

Cifrado asimétrico

El algoritmo RSA de cifrado es un algoritmo de cifrado de bloque de clave pública y es el algoritmo asimétrico más ampliamente

El uso de un mecanismo de cifrado asimétrico comienza con la publicación de la clave pública del destino. Para la comunicación bidireccional, el origen debe también publicar su clave pública.

Autenticación EE

Hemos visto que el cifrado ofrece una manera de restringir el conjunto de posibles receptores de un mensaje. El proceso de restringir el conjunto de potenciales emisores de un mensaje se denomina autenticación.

La autenticación es, por tanto, complementaria al cifrado. De hecho, algunas ES veces sus funciones se solapan.

Un algoritmo de autenticación consta de los siguientes componentes: o es:
+Un conjunto K de claves.
+Un conjunto M de mensajes.
+Un conjunto A de autenticadores. +
+ Una función S : K > (M > A).
+ Una función V:K> (MxA > {true, false}).

La propiedad crítica que un algoritmo de autenticación debe poseer es esta: para un mensaje, una computadora puede generar un autenticador a e A tal que V(k)(m, a) = true sólo si posee S(k). Así, una computadora que posea S(k) podrá generar autenticadores para los mensajes de modo que cualquier otra computadora que posea V(k) pueda verificarlo.
Al igual que hay dos algoritmos de cifrado, hay también dos variedades principales de algoritmos de autenticación. El primer paso para comprender estos algoritmos consiste en analizar las funciones hash.

Si el cifrado permite demostrar la identidad del emisor de un mensaje, entonces ¿por qué necesitamos algoritmos de autenticación separados? Existen tres razones principales:

+ Generalmente, los algoritmos de autenticación requieren menos cálculos (con la excepción de las firmas digitales RSA).

+ Un autenticador de un mensaje casi siempre es más corto que el mensaje y su texto cifrado correspondiente. Esto mejora el uso del espacio y reduce el tiempo de transmisión.

+ En ocasiones, deseamos disponer de la posibilidad de autenticación, pero no de la confidencialidad.

La autenticación es un componente de muchos aspectos de la seguridad.

Distribución de claves

Sin ninguna duda, una gran parte de la batalla entre criptógrafos (aquellos que inventan los mecanismos de cifrado) y criptoanalistas (aquellos que intentan romperlos) se encuentra en las claves.

En ocasiones, esa distribución se hace fuera de banda.

El problema se encuentra en la autenticación; lo que necesitamos es demostrar quién (o qué posee una determinada clave pública. Una forma de resolver este problema consiste en utilizar certificados digitales. Un certificado digital es una clave pública firmada digitalmente por un organismo de confianza. El organismo de confianza recibe la prueba de identificación de alguna entidad y certifica que la clave pública pertenece a dicha entidad.

Implementación de los mecanismos criptográficos

Usualmente, los protocolos de red se organizan en niveles, actuando cada nivel como un cliente del nivel inferior.

La criptografía puede incluirse en casi cualquier nivel del modelo ISO.

Generalmente, la seguridad del nivel de red se ha estandarizado en IPSec, que define formatos de los paquetes IP que permiten la inserción de autenticadores y el cifrado del contenido de los paquetes.

¿En qué lugar de la pila de protocolos es mejor incluir la protección criptográfica? En general, no hay una respuesta definitiva.

Por otro lado, la protección en los niveles inferiores de la pila de protocolos puede resultar insuficiente para los protocolos de los niveles superiores.

SSL

SSL 3.0 es un protocolo criptográfico que permite que dos computadoras se comuniquen de forma segura; es decir, cada una de ellas puede establecer limitaciones que garanticen que el transmisor o receptor de los mensajes sea la otra computadora.

Es quizá el protocolo criptográfico más comúnmente empleado actualmente en Internet, dado que es el protocolo estándar mediante el que se comunican de forma segura los exploradores web con los servidores web.

Este certificado es una estructura que contiene lo siguiente:
+ Varios atributos attrs del servidor, tal como su nombre distintivo univoco y su nombre E común (DNS).
+ La identidad de un algoritmo de cifrado público E() para el servidor La clave pública k, de ese servidor.
+Un intervalo de validez interval durante el que el certificado debe considerarse válido.
+ Una firma digital a para la información anterior, proporcionada por la certificación de autoridad CA, es decir, a = S(kTllattrs, E(k,), interval))

En esta situación, el cliente y el servidor calculan las claves siguientes a partir de ms:
+ Una clave de cifrado simétrica k?" para cifrar mensajes del cliente al servidor.
+ Una clave de cifrado simétrica k*" para cifrar mensajes del servidor al cliente.
+ Una clave de generación MAC k;" para generar autenticadores para los mensajes del cliente al servidor.
* Una clave de generación MAC K;. para generar autenticadores para los mensajes del servidor al cliente.

Autenticación de usuario

La exposición anterior sobre autenticación hacia referencia a mensajes y sesiones

El sistema de protección depende de la capacidad de identificar los programas y procesos que están actualmente en ejecución, lo que a su vez depende de la capacidad de identificar a cada usuario del sistema.

Generalmente, la autenticación de usuario se basa en una o más de tres cuestiones: la posesión de algo (una clave o tarjeta) por parte del usuario, el conocimiento de algo (un identificador de usuario y una contraseña) por parte del usuario y/o un atributo del usuario (huella digital, patrón retinal o firma).

Contraseñas

El método más habitual para autenticar la identidad de un usuario consiste en usar contraseñas.

Cuando el usuario se identifica a sí mismo mediante un ID de usuario o un nombre de cuenta, se le pide una contraseña. Si la contraseña suministrada por el usuario coincide con la contraseña almacenada en el sistema, el sistema supone que el propietario de la cuenta está accediendo a la misma.

A menudo, en ausencia de esquemas de protección más completos, se usan contraseñas para proteger objetos del sistema informático. Las contraseñas pueden considerarse un caso especial de las claves o de las capacidades.

Vulnerabilidades de las contraseñas

Las contraseñas son extremadamente comunes porque son fáciles de comprender y utilizar

Existen dos formas habituales de adivinar una contraseña. Una forma consiste en que el intruso (persona o programa) conoce al usuario o tiene información acerca de él. Con demasiada frecuencia, las personas emplean como contraseñas informaciones obvias, como los nombres de sus mascotas o de sus parejas.

Las contraseñas pueden averiguarse mediante mecanismos de monitorización visual o electrónica.

El último tipo de amenaza relativa a las contraseñas, la transferencia ilegal, es resultado de la propia naturaleza humana. La mayor parte de las instalaciones de computadoras tienen una regla que prohíbe a los usuarios compartir cuentas. En ocasiones, esta regla se implementa por razones contables, pero con frecuencia se impone para mejorar la seguridad.

Las contraseñas pueden ser generadas por el sistema o seleccionadas por el usuario. Las contraseñas generadas por el sistema pueden ser difíciles de recordar, por lo que en consecuencia los usuarios las anotarán.

Contraseñas cifradas

Los sistemas UNIX utilizan el cifrado para evitar la necesidad de mantener en secreto su lista de contraseñas. Cada usuario tiene una contraseña. El sistema contiene una función que es extremadamente difícil (los diseñadores esperaran que imposible) de invertir, pero fácil de calcular. Es decir, dado un valor x, es fácil calcular el valor de la función f(x). Sin embargo, dado un valor de la función A(x), es imposible calcular x.

Esta función se emplea para codificar todas las contraseñas y sólo se almacenan las contraseñas codificadas

El fallo de este método es que el sistema ya no tiene el control sobre las contraseñas.

Otra debilidad de los métodos de contraseñas de UNIX es que muchos sistemas UNIX sólo tratan los ocho primeros caracteres como significativos.

Contraseñas de un solo uso

Para evitar los problemas de la intercepción de contraseñas y las miradas de los fisgones por encima del hombro, un sistema podría usar un conjunto de contraseñas emparejadas. Cuando se inicia una sesión, el sistema selecciona aleatoriamente una pareja de contraseñas y presenta una parte de la misma; el usuario debe suministrar la otra parte.

En este sistema de contraseña de un solo uso, la contraseña es diferente en cada caso.

Biométrica

Otra variante del uso de contraseñas en los mecanismos de autenticación implica el uso de medidas biométricas. Los lectores palmares o de manos se usan habitualmente para dotar de seguridad a los accesos físicos, como por ejemplo, el acceso a un centro de datos.

Los lectores de huellas digitales son muy precisos y su relación coste-efectividad es buena, por lo que en el futuro serán de uso común.

Protección

Los procesos en un sistema operativo deben protegerse de las actividades realizadas por otros procesos.

El concepto de protección hace referencia a un mecanismo para controlar el acceso de los programas, de los procesos o de los usuarios a los recursos definidos por el sistema informático.

Se distingue entre los conceptos de protección y seguridad, en que la seguridad es una medida de la confianza que se puedan preservar la integridad de un sistema y de sus datos, la garantía de seguridad es un tema mucho más amplio.

Objetivos de la protección

La protección se concebía originalmente como algo asociado a los sistemas operativos multiprogramados, de modo que los usuarios que ni fueran de confianza pudieran compartir de manera segura un espacio físico de nombres común.

Los conceptos modernos de protección han evolucionado para incrementar la fiabilidad de cualquier sistema complejo que haga uso de recursos compartidos.

Razones:
Impedir una violacion maliciosa e intencionada de una restricción de acceso por parte de un usuario.

La necesidad de garantizar que cada componente del programa activo en un sistema utilice los recursos del sistema solo en ciertas formas que sean coherentes a las políticas establecidas.

Los mecanismos de protección pueden mejorar la fiabilidad deter asneo ps errores latentes en las interfaces definidas entre los distintos su sistemas componentes.

La detección temprana del errores de interfaz puede a menudo impedir que un su sistema correcto se vea contaminado por otro.

El papel de la protección en un sistema informático es proporcionar un mecanismo para la imposición de políticas que gobiernen el uso de recursos.

Las políticas de uso de recursos pueden variar según la aplicación y también pueden variar a lo largo del tiempo.

Los mecanismos determinan como se llevara algo acabo
Las políticas deciden que es lo qué hay que hacer

Principios de la protección

Podemos utilizar un principio director a lo largo de un proyecto, como puede ser el diseño de un sistema operativo.
Manteniendo que sea coherente, lógico y sencillo.

Un sistema operativo se ajusta al principio del mínimo privilegio implementara sus características, programas, llamadas al sistema y estructuras de datos de modo que el fallo o compromiso de un componente provoquen un dato mínimo y no permitan realizar mas que un daño mínimo.

Dicho sistema operativo también proporcionara llamadas al sistema y servicios que permitan escribir aplicaciones con controles de acceso a granularidad fina, mecanismos para activar lols privilegios, la creación de pistas de auditorias, etc.

La pista de auditoria permite al programador, ala suministrador del sistema o a los miembros de las fuerzas del orden revisar todas las actividades realizadas en el sistema que estén relacionadas con los mecanismos de protección y seguridad.

La gestión de los usuarios con el principio del mínimo privilegio implica crear una cuenta separada para cada usuario.

El principio de mínimo privilegio puede ayudar a obtener un entorno informático mas seguro.

Dominio de protección

Un sistema informático es una colección de procesos y objetos.

Objetos hardware y software

Los objetos son esencialmente tipos abstractos de datos

Las operaciones posibles pueden depender de cada objeto.

Los archivos de datos pueden crearse, abrirse, leerse, escribirse,
cerrarse y borrarse; los archivos de programa pueden leerse, escribirse, ejecutarse y borrarse.

Á un proceso sólo se le debe permitir acceder a aquellos recursos para los que tenga autorización. Además, en cualquier instante determinado, un proceso sólo debería poder acceder a aquellos recursos que necesite actualmente para completar su tarea. Este segundo requisito, al que comúnmente se denomina principio de la necesidad de conocer, resulta útil a la hora de limitar la cantidad de daño que un proceso erróneo pueda provocar en el sistema

Estructura de dominios

Cada dominio define un conjunto de objetos y los tipos de operaciones que pueden invocarse sobre cada objeto. La capacidad de ejecutar una operación sobre un objeto es un derecho de acceso. Un dominio es una colección de derechos de acceso, cada uno de los cuales es una pareja ordenada <nombre-objeto, conjunto-derechos>.

Los dominios no tienen por qué ser disjuntos, sino que pueden compartir derechos de acceso.

La asociación entre un proceso y un dominio puede ser estática, si el conjunto de recursos disponibles para el proceso está fijo durante la vida del proceso, o dinámica.

Si la asociación es dinámica, habrá disponible un mecanismo para permitir la conmutación de dominio, permitiendo al proceso conmutar de un dominio a otro. También podemos permitir que se modifique el contenido de un dominio. Si no podemos cambiar el contenido de un dominio, podemos proporcionar el mismo efecto creando un nuevo dominio con el contenido modificado y conmutando a este nuevo dominio cuando queramos cambiar el contenido del dominio.

Un dominio puede llevarse a la práctica de diversas formas:
+ Cada usuario puede ser un dominio. En este caso, el conjunto de objetos a los que se podrá acceder dependerá de la identidad del usuario. La conmutación de dominios tiene lugar cuando cambia el usuario, es decir, generalmente cuando un usuario cierre la sesión y otro usuario la incite.

+Cada proceso puede ser un dominio. En este caso, el conjunto de objetos a los que se podrá acceder dependerá de la identidad del proceso. La conmutación de dominio tendrá lugar cuando un proceso envíe un mensaje a otro proceso y espere una respuesta.

UNIX

En el sistema operativo UNIX, un dominio está asociado con el usuario. La conmutación de dominio se corresponde con un cambio temporal en la identificación del usuario. Este cambio se lleva a cabo a través del sistema de archivos de la forma siguiente: con cada archivo hay asociada una identificación de propietario y un bit de dominio (conocido como bit setuid); cuando el bit setuid
está activado y un usuario ejecuta dicho archivo, el ID de usuario se configura con el valor correspondiente al propietario del archivo; sin embargo, cuando el bit está desactivado, el ID de usuario no se modifica

MULTICS

.

En el sistema MULTICS, los dominios de protección están organizados jerárquicamente en una estructura de anillos concéntricos. Cada anillo se corresponde con un único dominio.

Los anillos están numerados de 0 a 7. Sean D, y D; dos anillos de dominio cualquiera entonces D; es un subconjunto de D,, es decir, un proceso que se ejecute en el dominio D; tiene más privilegios que otro que se ejecute en el dominio D,. Un proceso que se ejecute en el dominio será el que tenga más privilegios. Si sólo existen dos anillos, este esquema es equivalente al modo a
monitor-usuario de ejecución, donde el modo monitor corresponderá a D, y el modo usuario corresponderá a D.
dinámicamente nuevos objetos y nuevos dominios e incluirlos en el modelo de la matriz de acceso. Sin embargo, sólo hemos mostrado que el mecanismo básico existe; los diseñadores del sistema y los usuarios deben tomar las decisiones de política relativas a qué dominios deben poder acceder a qué objetos y en qué
manera.

Matriz de acceso

Nuestro modelo de protección puede contemplarse de forma abstracta como una matriz, denominada matriz de acceso. Las filas de la matriz de acceso representan dominios y las columnas representan objetos. Cada entrada de la matriz está compuesta de un conjunto de derechos de acceso.
Puesto que la columna define los objetos explícitamente, podemos omitir el nombre del objeto del derecho de acceso.

La matriz de acceso proporciona el mecanismo apropiado para definir e implementar un control estricto de la asociación tanto estática como dinámica entre procesos y dominios.

Los derechos de copia y el derecho de propietario permiten a un proceso modificar las entradas de una columna. También hace falta un mecanismo para modificar las entradas de una fila. El derecho control sólo es aplicable a los objetos dominio. Si accessincluye el derecho control: entonces un proceso que se ejecute en el dominio D, puede eliminar cualquier derecho de acceso de la fila.

Estas operaciones sobre los dominios y la matriz de acceso no son en sí mismas importantes,
pero ilustran la capacidad del modelo de la matriz de acceso para permitir la implementación y control de requisitos de protección dinámicos. Pueden crearse dinámicamente nuevos objetos y nuevos dominios e incluirlos en el modelo de la matriz de acceso. Sin embargo, sólo hemos mostrado que el mecanismo básico existe; los diseñadores del sistema y los usuarios deben tomar las decisiones de política relativas a qué dominios deben poder acceder a qué objetos y en qué
manera.

Implementación de la matriz de acceso

La implementación más simple de la matriz de acceso es una tabla global compuesta de un con junto de tripletas ordenadas <dominio, objeto, conjunto-derechos>. Cada vez que se ejecuta una operación M sobre un objeto O; dentro del dominio D,, se analiza la tabla global en busca de una tripleta <D,, Oj, R>, con M e R,. Si se encuentra esta tripleta, se permite que la operación continúe; en caso contrario; se genera una condición de excepción (o error).

Esta implementación tiene varias desventajas. La tabla es usualmente muy grande y no puede,por tanto, ser conservada en memoria principal, por lo que hacen falta operaciones adicionales de E/S. Á menudo se utilizan técnicas de memoria virtual para gestionar esta tabla.

Listas de acceso para los objetos

Cada columna de la matriz de acceso puede implementarse como una lista de acceso de un objeto.

Listas de capacidades para los dominios

En lugar de asociar las columnas de la matriz de acceso con los objetos en forma de listas de acceso, podemos asociar cada fila con su dominio. Una lista de capacidades para un dominio es una lista de objetos junto con las operaciones permitidas sobre esos objetos. Cada objeto.se suele representar mediante su dirección o nombre físico, denominada capacidad. Para ejecutar la operación sobre el objeto, el proceso ejecuta la operación M especificando la capacidad (o puntero) el objeto O, como parámetro.

La lista de capacidades está asociada con un dominio, pero un proceso que se ejecute en dominio no puede nunca acceder directamente a ella. Por el contrario, la lista de capacidades en sí misma un objeto protegido, mantenido por el sistema operativo.

Las capacidades se propusieron originalmente como una especie de puntero seguro, para satisfacer la necesidad de protección de los recursos que se preveía que iba a ser necesaria a medir que los sistemas informáticos multiprogramados se generalizaran.

Para proporcionar una protección inherente, debemos distinguir las capacidades de otros objetos y debemos interpretarlas mediante una máquina abstracta sobre la que se ejecuten los programas de mayor nivel.

Un mecanismo de bloqueo-clave A

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 distintivos, denominados bloqueos. De forma similar, cada dominio tiene una lista de patrones de bit distintivos, denominados claves. Un proceso que se ejecute dentro de un dominio podrá acceder a un objeto sólo si dicho dominio tiene una clave que se corresponda con uno de los bloqueos del objeto.

Comparación

Vamos ahora a comparar las diversas técnicas de implementación de las matrices de acceso. La utilización de una tabla global resulta muy simple.

El mecanismo de bloqueo-clave, como hemos mencionado, representa un compromiso entre las listas de acceso y las listas de capacidades. El mecanismo puede ser a la vez efectivo y flexible, dependiendo de la longitud de las claves. Esas claves pueden pasar seguramente de un dominio y a otro.

Control de acceso

Á cada archivo y directorio se le asignan un propietario, un grupo o posiblemente una lista de usuarios y para cada una de estas entidades se asigna una información de control de acceso. Podemos añadir una función similar a otros aspectos de un sistema informático.

En un sistema de protección dinámico, puede que necesitemos en ocasiones revocar derechos de acceso a objetos compartidos por diferentes usuarios. En este punto pueden surgir diversas cuestiones acerca de la revocación:
+ Inmediata o diferida
+ Selectiva o general.
+ Parcial o total.
+ Temporal o permanente.

I proceso puede entonces tratar de volver a adquirir la capacidad. Si se ha revocado el acceso, el proceso no será capaz de efectuar esa readquisición.

+ Retropunteros. Con cada objeto se mantiene una lista de punteros, que hace referencia atodas las capacidades asociadas con ese objeto

+ Indirección. Las capacidades apuntan indirectamente, en lugar de directamente, a los objetos. Cada capacidad apunta a una entrada univoca dentro de una tabla global, que a su vez apunta al objeto.

+ Indirección. Las capacidades apuntan indirectamente, en lugar de directamente, a los objetos. Cada capacidad apunta a una entrada univoca dentro de una tabla global, que a su vez apunta al objeto.

+ Claves. Una clave es un patrón distintivo de bits que puede asociarse con una capacidad.
Esta clave se define en el momento de crear la capacidad y no puede ser nunca modificada ni inspeccionada por el proceso que posee la capacidad.

Este esquema no permite la revocación selectiva, ya que con cada objeto sólo hay aso-
ciada una clave maestra. Si asociamos una lista de claves con cada objeto, entonces podrá
implementarse la revocación selectiva. Finalmente, podemos agrupar todas las claves en

+ Una tabla global de claves. Una capacidad será válida sólo si su clave se corresponde con
alguna de las claves de la tabla global. Implementamos la revocación eliminando de la tabla esa clave que se corresponda.

En los esquemas basados en claves, las operaciones de definición de claves, de inserción de claves en listas y de borrado de claves de las listas no deben estar disponibles para
todos los usuarios.

Sistemas basados en capacidades

En esta sección, vamos a repasar dos sistemas de protección basados en capacidades. Estos sistemas varían tanto en lo que se refiere a su complejidad como en el tipo de políticas que pueden implementarse sobre ellos.

Hydra

Hydra es un sistema de protección basado en capacidades que proporciona una considerable flexibilidad. El sistema conoce e interpreta un conjunto fijo de posibles derechos de acceso. Estos derechos incluyen formas básicas de acceso tales como el derecho de leer, escribir o ejecutar segmento de memoria. Además, un usuario (del sistema de protección) puede declarar otros derechos. La interpretación de los derechos definidos por el usuario se lleva a cabo exclusivamente por el programa de usuario, pero el sistema proporciona protección de acceso para el uso de esto derechos, además de para el uso de los derechos definidos por el sistema. Estas característica constituyen un avance significativo en la tecnología de protección.

Hydra también proporciona un mecanismo de amplificación de derechos.

Sistema CAP de Cambridge

En el sistema CAP de Cambridge se ha adoptado un enfoque distinto para la implementación del mecanismo de protección basado en capacidades. El sistema de capacidades de CAP es más simple y superficialmente menos potente que el de Hydra.

La interpretación de una capacidad software se deja completamente al arbitrio del subsistema, a través de los procedimientos protegidos que contenga. Este esquema permite implementar diversas políticas de protección.