

# POLÍTICAS Y MODELOS DE SEGURIDAD



por

**Estela E. Vogelmann Martínez**

estelavogelmann@hotmail.com

Trabajo de Adscripción a Sistemas Operativos  
2008

Dpto. de Informática  
FACENA - UNNE  
Argentina



# Índice general

<b>1. Políticas y Modelos de Seguridad</b>	<b>1</b>
1.1. Introducción . . . . .	1
1.2. La necesidad de las políticas . . . . .	2
1.3. Las políticas de seguridad . . . . .	3
1.4. Aplicación - Políticas Específicas . . . . .	5
1.4.1. Políticas de confidencialidad . . . . .	5
1.4.2. Políticas de integridad . . . . .	5
1.4.3. Grupo de políticas . . . . .	6
1.4.4. Políticas de conflicto de intereses . . . . .	6
1.5. Sistema de políticas . . . . .	6
1.6. Ejemplos de políticas . . . . .	7
1.7. Uso de funciones en materia de políticas . . . . .	9
1.8. Políticas Estándares . . . . .	9
1.9. Normas para las políticas . . . . .	10
1.10. Políticas de lenguajes . . . . .	10
1.11. Políticas en conflictos . . . . .	10
1.12. Problemas con las políticas no apropiadas . . . . .	11
1.13. Propiedades e interacciones de las políticas . . . . .	11
1.14. Políticas y diseño de sistemas de seguridad . . . . .	12
1.15. Modelos de seguridad . . . . .	13
1.15.1. La matriz de acceso . . . . .	13
1.15.2. Control de Acceso basado en funciones - RBAC (Role - Based Access Control) . . . . .	16
1.15.3. Autorización implícita . . . . .	17
1.15.4. Los modelos multinivel . . . . .	18
1.16. El modelo de Clark-Wilson . . . . .	21
1.17. Modelos y diseño de sistemas de seguro . . . . .	21
1.18. El modelo Monitor de Referencia . . . . .	22

<b>Bibliografía</b>	<b>25</b>
<b>Índice alfabético</b>	<b>29</b>

# Índice de figuras

1.1. The authorization pattern. . . . .	14
1.2. The RBAC pattern. . . . .	17
1.3. Multilevel security pattern. . . . .	20
1.4. Class diagram for the reference mon . . . . .	24
1.5. Sequence diagram for enforcing security of requests. . . . .	24



# Capítulo 1

## Políticas y Modelos de Seguridad

### 1.1. Introducción

Los requerimientos de seguridad que involucran las tecnologías de la información, en pocos años han cobrado un gran auge, y más aún con las de carácter globalizador como los son la de Internet y en particular la relacionada con la Web, la visión de nuevos horizontes explorando más allá de las fronteras naturales, situación que ha llevado a la aparición de nuevas amenazas en los sistemas computarizados [1].

Ante este esquema las instituciones se ven inmersas en ambientes agresivos donde el delinquir, sabotear, robar se convierte en retos para delincuentes informáticos universales conocidos como Hackers, Crakers, etc., es decir en transgresores.

Conforme las tecnologías se han esparcido, la severidad y frecuencia las han transformado en un continuo riesgo, que obliga a las entidades a crear medidas de emergencia y políticas definitivas para contrarrestar estos ataques y transgresiones.

De esta manera, las políticas de seguridad en informática emergen como el instrumento para concientizar a los miembros de las instituciones acerca de la importancia y sensibilidad de la información y servicios críticos, de la

superación de las fallas y de las debilidades, de tal forma que permiten a la institución cumplir con su misión.

El proponer estas políticas de seguridad requiere un alto compromiso con la institución, agudeza técnica para establecer fallas y deficiencias, constancia para renovar y actualizar dichas políticas en función del ambiente dinámico que nos rodea.

## 1.2. La necesidad de las políticas

Las políticas de alto nivel son pautas sobre la seguridad de la información [2]. Cada institución tiene un conjunto de políticas de negocio, explícitas o implícitas, de las cuales algunas son las políticas de seguridad. Sin políticas es imposible la creación de sistemas seguros, no sabremos lo que debemos proteger y cuánto esfuerzo se debe poner en materia de seguridad.

Una política de seguridad se divide en los estados de un sistema autorizado y no autorizado. Eso significa, que ni siquiera podemos hablar de seguridad sin políticas porque no sabremos lo que dice que debemos evitar. La necesidad y el valor de las políticas han sido reconocidos recientemente, y ahora hay incluso una conferencia anual dedicada a las políticas [3]. También existen paquetes de políticas [4], software para la generación automática de la política [5], e incluso un libro se ha dedicado a este tema [6].

La institución de políticas de seguridad incluye las leyes, normas y prácticas que regulan cómo una institución gestiona y protege los recursos. Hay políticas globales que afectan a todos los aspectos del negocio y políticas más especializadas que se refieren a la división de las funciones básicas. Algunas de estas políticas pueden ser impuestas o sugeridas de fuentes externas, por ejemplo, la legislación, el gobierno o las normas de la industria. Hay algunas cuestiones interesantes acerca de quién debe definir la política exterior o las normas de lo que una institución debe seguir. Por ejemplo, todos los bancos como lo exige la ley del gobierno de EE.UU., deben enviar a sus clientes una declaración de su política de privacidad, indicando la información personal que recogen, cuales comparten, y detalles similares.

Los sistemas informáticos de la institución deben hacer cumplir estas políticas que se ven reflejas en sus mecanismos. El modelo de capas se puede utilizar para describir cómo se estructuran las políticas, pasando de las políticas de

las instituciones del más alto nivel para permitir a las políticas específicas de los usuarios que acceden a los datos, las políticas de encriptación, etc.

### 1.3. Las políticas de seguridad

A través de los años se han desarrollado varias políticas como las más convenientes para crear o configurar sistemas de seguridad. Se enumeran algunas de ellas aquí (más detalles se pueden encontrar en [2]):

- *Sistemas Abiertos / Cerrados*: En un sistema cerrado, nada es accesible a menos que se autorice expresamente; en un sistema abierto o institución todo es accesible a menos que esté explícitamente denegado. Es evidente que un sistema seguro debe ser cerrado. En las instituciones en que la seguridad de la información es muy importante, por ejemplo, los bancos, utilizan una política cerrada. En cambio, instituciones cuyo objetivo es la difusión de información, tales como bibliotecas, usan políticas abiertas.
- *Menos privilegio (lo que necesita conocer)*: Las personas o cualquier entidad activa que necesita acceder a recursos computacionales deben ser autorizadas sólo para tener acceso a los recursos que necesitan para desempeñar sus funciones. Esta política suele combinarse con la política de sistemas cerrados.
- *Maximizar el intercambio*: Hay instituciones que quieren hacer a la información lo más accesible posible [7]. Aquí puede ser la información privada o de otra índole, pero la idea es maximizar el uso de la misma.
- *Autorización*: Las normas explícitas deben definir quién puede utilizar qué recursos y cómo. Las autorizaciones podrán permitir o denegar el acceso y podrán imponer las condiciones de acceso.
- *Obligación*: Estas políticas definen qué debe o no debe realizarse en un conjunto de datos [8].
- *Separación de los derechos*: Las funciones críticas deben ser asignadas a más de una persona o sistema. Por ejemplo, la persona que decide la compra de un producto no es la misma que la que en realidad hace los pedidos del producto.

- *Auditoria*: Una auditoria debe llevar un registro de lo que se hizo y en qué momento. Esto ayudará a prevenir futuros ataques y es importante para fines de rendición de cuentas.
- *Control Centralizado / Descentralizado*: En un sistema descentralizado sus unidades o divisiones tiene autoridad para definir sus propias políticas o mecanismos de aplicación en la medida en que no violen las políticas globales.
- *Propiedad y administración*: En muchos sistemas el usuario cree que algunos datos se convierte en su propiedad y tiene todos los derechos sobre el mismo. Una política administrativa separa la administración de los datos de su uso. La propiedad puede violar la separación de los derechos cuando el usuario de la información también es su administrador, la institución de datos entra en un conflicto de intereses, pero es aceptable para los archivos personales. La propiedad no es una buena política de seguridad de los sistemas institucionales a pesar de que se utiliza comúnmente en la mayoría de sistemas operativos.
- *Rendición de cuentas individuales*: Las personas o los procesos deben ser identificados y sus actuaciones grabadas y revisadas.
- *Roles*: Los roles implican un grupo de derechos que se le da a los usuarios de acuerdo a sus funciones. Los derechos de los roles podrían seguir las políticas de menor privilegio.
- *Nombre o número dependiendo de su control de acceso*: El acceso de control está designado por su número o por las clases incluidas en sus instancias.
- *Contenido -dependiendo del control de acceso-*: El acceso a los datos depende de los requerimientos de los archivos específicos.
- *Contexto -dependiendo del control de acceso-*: El acceso a los datos depende de que otra información también la requiere; por ejemplo, uno no puede buscar los salarios y los nombres a la vez. Otra interpretación de esta política se basa en la decisión de acceso del sistema o en el estado en que se encuentra el trabajo.
- *Historia -dependiendo del control de acceso-*: Se considera todos o subgrupos de requerimientos para la decisión de acceso.

## 1.4. Aplicación - Políticas Específicas

Las políticas se aplican a cualquier sistema seguro. Algunas aplicaciones requieren políticas más específicas, por ejemplo, los militares ponen más énfasis en el secreto, mientras que la oficina jurídica está más interesada en la integridad.

### 1.4.1. Políticas de confidencialidad

- *Clasificación de documentos*: Los documentos son clasificados en función de la sensibilidad de su información. A las personas se les da autorizaciones. La política define una relación entre la clasificación y las autorizaciones. Por ejemplo, el juego de las clasificaciones pueden ser los niveles jerárquicos: secreto top, secreto, confidencial, público, y un usuario habilitado para un nivel dado puede leer todos los documentos en su nivel o por debajo.
- *Categorías*: Definen particiones verticales de los niveles, por ejemplo, el Ejército, la Armada. Ahora, no sólo la clasificación debe ser adecuada para leer un documento, sino también el usuario debe coincidir con la categoría o estar incluida en la categoría del documento.
- *Originator controlled (ORCON)*: Un documento sólo se libera a las personas o unidades que estén en una lista específica hecha por el inventor.
- *Acceso a lo total*: Los usuarios están autorizados a leer sólo los valores de los datos agregados, por ejemplo, el promedio de los salarios, el promedio de calificaciones del estudiante. Estas políticas son particularmente importantes cuando se trata de la privacidad de las personas.

### 1.4.2. Políticas de integridad

- *La integridad de los documentos*: Un documento no puede ser modificado o sólo se puede registrar las modificaciones.
- *Cambio limitado*: Los datos sólo se pueden modificar en la forma prescrita.

### 1.4.3. Grupo de políticas

- *Acciones autorizadas*: Las personas sólo pueden realizar acciones para las que fueron autorizadas.
- *Rotación de los derechos*: Una tarea no debe ser realizada siempre por la misma persona.
- *Operación de la secuenciación*: Los pasos de algunas tareas deben llevarse a cabo en un orden específico.

### 1.4.4. Políticas de conflicto de intereses

- *Política de Muralla*: La información se agrupa en clases de “conflicto de intereses” y a una persona se le permite el acceso a la mayoría de un conjunto de información de esa clase.
- *Conflicto de roles*: Un usuario no puede tener dos funciones que pueden implicar un conflicto de intereses.

## 1.5. Sistema de políticas

La mayoría de estas políticas pueden aplicarse también a bajo nivel, a algunos aspectos del sistema. Por ejemplo, un proceso debe ejecutarse con la menor cantidad de privilegios que necesita para desempeñar sus funciones. Otras políticas de sistema definen el uso específico de algún sistema, por ejemplo, una cuenta de usuario / contraseña [5], en la política se determinan aspectos tales como la longitud de las contraseñas, que caracteres pueden tener o no, y la frecuencia con que debe ser cambiada.

Se pueden definir políticas para el diseño y el uso de cualquier aspecto de un sistema informático. Un ejemplo de separación de servicio en estos niveles es la separación de la aplicación de una regla a partir de la autorización de almacenamiento y mantenimiento de las normas. Algunas de las políticas del sistema proceden directamente de políticas similares en un nivel más alto, otros se utilizan para controlar aspectos específicos de la arquitectura correspondiente al nivel.

Moffett y Sloman [9] clasifican las políticas de sistemas de seguridad en tres niveles:

- *Políticas generales:* Estas se aplican a cualquier institución. Ejemplos de ello son el control de acceso a la administración para la seguridad de los administradores.
- *Políticas específica:* Estas se refieren a organizaciones específicas, por ejemplo, haciendo hincapié en la integridad y en la confidencialidad.
- *Reglas de acceso:* Define especificaciones para el acceso a recursos determinados.

Un error común es definir las políticas de bajo nivel sin utilizar políticas de alto nivel como referencia. Por ejemplo, Visa requiere que los comerciantes en línea que utilizan sus tarjetas: instalen un cortafuegos, mantengan los parches de seguridad actualizados, cifren datos transmitidos y almacenados, etc. Estas políticas son demasiado detalladas para ser eficaces y son restrictivas al que las utiliza, ya que no se basan en políticas de más alto nivel.

Algunas políticas se pueden aplicar a varios sistemas, por ejemplo:

- *Aislamiento o contención:* Un sistema debe estar aislado de los sistemas externos, un proceso debe ser aislado de otros procesos.
- *Compartir el control:* Recursos o información deben ser compartidos por los procesos o sistemas de forma controlada, sin perjuicio de las autorizaciones.
- *Sistemas sin memoria:* Un programa no debe tener ningún vestigio de sus ejecuciones pasadas. Por ejemplo, un programa para calcular los impuestos no deben tener ninguno de los valores que ha utilizado en el pasado.

En general, el aislamiento y la participación en el control se excluyen mutuamente cuando se aplica a un proceso específico, pero se pueden combinar cuando se habla de un conjunto de procesos, que puede ser aislado en su conjunto, pero podrá compartir recursos entre ellos.

## 1.6. Ejemplos de políticas

Muchas políticas comunes se refieren a aspectos de autorización. Las autorizaciones definidas deben ajustarse a las necesidades de la aplicación. Por

ejemplo, Anderson menciona cómo en el Reino Unido una oficina del gobierno trató de imponer la política de múltiples niveles en los sistemas médicos y no funcionó porque no encajaba con los requisitos; pacientes que quieren controlar el uso de sus registros, a veces esto no se permite en el modelo multinivel.

Lo siguiente es un posible conjunto de las políticas de un sistema universitario, asumiendo también la política de un sistema cerrado:

- Un instructor puede ver toda la información sobre el curso que está enseñando.
- Un instructor puede cambiar las calificaciones de los estudiantes en el curso en que enseñanza.
- Un estudiante puede ver sus calificaciones del curso que está realizando.
- Un director de departamento puede añadir o suprimir cursos en su departamento.
- Los miembros del profesorado puede acceder a información sobre sí mismos.
- Un estudiante puede inscribirse en un curso.
- Un director de departamento puede ver información sobre su departamento y pueden cambia la información sobre profesores y cursos.
- Un decano puede ver la información de todos los departamentos en su universidad o facultad.

Algunas políticas pueden ser muy complejas y dependen de los valores de las variables involucradas. Un ejemplo de política compleja [10]: “Un usuario puede ver los registros de cada empleado que supervisa si el usuario tiene un sueldo mayor que los demás empleados”. La autorización se aplica utilizando las políticas y normas de autorización gestionadas por medio de algún sistema de administración de seguridad. Debido a esto, algunos proveedores, por ejemplo, Microsoft y Sun, sus normas se refieren a sí mismas como políticas.

Las políticas pueden referirse a múltiples políticas, por ejemplo [10]: a sí mismas como políticas:

- Un plan de vuelo se puede clasificar si la lista de pasajeros incluye a funcionarios con nombre específica.

- Un plan de vuelos clasificados sin clasificar puede ser una vez que el vuelo se ha completado.

## 1.7. Uso de funciones en materia de políticas

Es importante definir las funciones con respecto a la información producida o utilizada en una institución o el sistema. Algunas posibles funciones con respecto a los documentos son:

- *Fuente*: La persona que emite un documento.
- *Autorizador*: La persona que controla el acceso sobre el documento.
- *Depositario*: La persona que guarda el documento de control y su uso.
- *Usuario*: La persona que lee o modifica el documento.
- *Auditor*: La persona que chequea las acciones, resultados, y los controla.

También se puede definir las funciones apropiadas para las personas de acuerdo a sus funciones de trabajo y asignar los derechos de acuerdo con estas funciones, por ejemplo, gerente, secretaria, estudiante, y el instructor. En el ejemplo anterior de la universidad, los papeles utilizados son el instructor, estudiante, profesor, secretario, director de departamento y decano. Cada función puede tener algunos subroles, por ejemplo, un profesor puede ser un instructor de una tesis, un miembro del comité de departamento, y un investigador.

## 1.8. Políticas Estándares

En los EE.UU. la primera institución gubernamental a cargo de las políticas de seguridad fue el Departamento de Defensa. Se publicó un documento de referencia definiendo los diferentes niveles de seguridad. Este documento (conocido como el Libro Naranja) enumera una serie de requisitos para sistemas de seguridad que pueden ser considerados como políticas de evaluación de la seguridad. Más tarde, el Instituto Nacional de Estándares y Tecnología (NIST) ha desarrollado un conjunto de documentos conocidos como los Criterios Comunes [11]. Otras políticas se han definido por ECMA y la ISO. Las políticas para aplicaciones especializadas, se han definido:

- La información médica: BMA en el Reino Unido y la HIPAA en los EE.UU.
- La información financiera: La Ley Sarbanes - Oxley de los EE.UU.

## 1.9. Normas para las políticas

El Modelo de Política del Núcleo de Información (PCIM) es un modelo de política para ampliar el Modelo Común de información (CIM), desarrollado por el Grupo de Tareas de gestión distribuida (DTMF) y la Política de grupo de trabajo IETF .

La CIM define objetos genéricos que incluyen sistemas, elementos administradores del sistema, elementos físicos y lógicos, y los servicios. Se define una política de Estado y sus componentes, condiciones y acciones. Una política de Estado es la forma <condition set> hacer <action list>.

Las normas de política pueden ser simples o grupales (un patrón compuesto). Las condiciones y acciones pueden ser parte de normas específicas o ser almacenados en los repositorios de uso común por varias normas. El uso de repositorios es un aspecto de la aplicación, que no debería haber sido mezclada con la estructura de la regla lógica. El PCIM también ofrece modelos detallados de las condiciones y acciones.

## 1.10. Políticas de lenguajes

IBM ha desarrollado una Política de Lenguaje Fiduciario. Este usa XML para definir los criterios de asignación de clientes a las funciones y la autorización de recursos. Las normas no pueden ser heredadas y tienen otras restricciones [12].

## 1.11. Políticas en conflictos

Es posible que los objetos a que se refiere una política se superpongan con los de otra política. Por ejemplo, una política puede indicar que un objeto sea accesible a un usuario, mientras que otra política puede negarlo.

En estos casos el conflicto puede resolverse mediante políticas tales como “permisos tienen prioridad”, “negaciones tienen prioridad”, o mediante la adición explícita a las prioridades de cada Estado.

## 1.12. Problemas con las políticas no apropiadas

Un ejemplo de un caso real ilustra lo que sucede cuando las políticas no se definen o no se aplican [13]. Un ex-empleado de Global Crossing Holdings Ltd. Descontento con esta, colocó numerosos nombres, SSN, y fechas de nacimiento de empleados de la empresa en su sitio web. La empresa había permitido que todos los desarrolladores de software tengan pleno acceso de lectura a la información sobre los empleados y el cliente, el sistema de facturación era accesible para leer y escribir sobre un gran número de empleados.

El primer problema fue la no aplicación de la necesidad de conocer la política, no era necesario que los desarrolladores de software tengan acceso a los datos operativos. El segundo problema fue similar, el acceso a la información de facturación debería haberse limitado sólo a aquellos que tenían una necesidad de funciones de su trabajo, es decir, la falta de conocimiento de roles para que el acceso se basaba en estos. Aparentemente no tenían un sistema cerrado.

## 1.13. Propiedades e interacciones de las políticas

Algunas políticas pueden ser representadas formalmente con el uso de modelos como se muestra a continuación, mientras que otros en su mayoría se describen con palabras. La lógica difusa se ha utilizado para hacer las políticas de palabras más precisas [14]. Un buen conjunto de políticas pueden ser reutilizable, como lo demuestra el hecho de que existen políticas de pre-empaquetados [15].

Todas las políticas aplicadas en un sistema interactúan unas con otras. Idealmente, deberían colaborar o convergen [14]. A veces pueden superponerse, lo que puede dar lugar a la redundancia innecesaria. La peor situación es cuando las políticas entran en conflicto entre sí, porque esto puede dar lugar a vulnerabilidades de seguridad [8]. Por ejemplo, la privacidad puede entrar en conflicto con la rendición de cuentas. El uso transnacional de los datos a menudo resulta en conflictos de política. Los sistemas distribuidos requieren

la coexistencia de muchas políticas y las metapolíticas son necesarias para coordinar las mismas [16].

### 1.14. Políticas y diseño de sistemas de seguridad

Una vez que tenemos una lista de las amenazas a nuestro sistema podemos decidir cuáles de estas amenazas son importantes y cómo podemos evitarlas, de acuerdo con las políticas de la institución, es decir, las políticas que guiarán la selección de los mecanismos específicos que necesitamos para poner fin a las amenazas. Por ejemplo, si el secreto es importante, tenemos que protegerlo contra virus o caballos de Troya que puede comprometerlo.

Las políticas son también importantes para la evaluación de un sistema seguro, si se aplica a un sistema sus políticas, será seguro para nuestros propósitos. Un sistema complejo deberá tener múltiples políticas de apoyo, incluyendo una variedad de políticas para el control de acceso [17].

Las políticas de seguridad deberían reflejarse en los mecanismos de seguridad utilizados en los distintos niveles de la arquitectura. Los mecanismos de más bajo nivel deben aplicar las políticas definidas por los de alto nivel. La mayoría de los sistemas comerciales no aplican las políticas descritas anteriormente, por ejemplo, en Unix un archivo creador se convierte en su administrador y usuario, esto viola la política de separación de servicio. También es importante definir las políticas de seguridad en un contexto, por ejemplo, un sistema específico o nivel.

Cuando tenemos las jerarquías de las políticas que pueden tener conflictos, es importante resolverlos antes de continuar con el diseño más detallado.

En este momento podemos considerar los casos de uso del sistema para definir los derechos que los usuarios deben tener para poder llevar a cabo sus funciones [18]. Los casos de uso pueden ser ampliados con la declaración de nuevas políticas.

Algunas políticas de seguridad pueden ser más precisas mediante la utilización de modelos semi-formales o formales. Un modelo nos permite analizar las propiedades de seguridad y es la base para el diseño del sistema.

## 1.15. Modelos de seguridad

Los modelos de seguridad son más precisos y detallados que la expresión de las políticas y se utilizan como directrices para crear y evaluar sistemas. Por lo general, pueden describirse de manera formal o semi-formal. Los modelos pueden ser obligatorios o discrecionales. En un modelo discrecional, los titulares de derechos pueden ser autorizados a transferir en su discreción. En un modelo obligatorio sólo papeles designados están autorizados a conceder derechos y los usuarios no pueden transferirlos.

Una clasificación divide a los modelos ortogonales en los que se basan en la matriz de acceso, acceso basado en funciones de Control, y los modelos multinivel. Los dos primeros son modelos de control de acceso, mientras que los últimos son intentos de un control de flujo de información. Los modelos obligatorios y discrecionales pueden ser combinados con el de matriz de acceso y el de modelos multinivel.

### 1.15.1. La matriz de acceso

Aunque se presentó como un modelo para el funcionamiento de los sistemas de seguridad [19], la matriz de acceso (AM) es un modelo de seguridad que se pueden aplicar a cualquier sistema. En su forma inicial, define un modelo discrecional, pero puede ser limitado para que sea un modelo obligatorio. Se puede controlar tanto la confidencialidad como la integridad.

El modelo define un conjunto de sujetos  $S$  (requieren entidades), un conjunto de objetos protegidos  $O$  (las entidades lo solicitan), y un conjunto de tipos de acceso  $T$  (la forma en que el objeto se puede acceder). En un sistema operativo, los temas son los procesos, los objetos son los recursos del sistema, y los tipos de acceso suelen ser leer, escribir y ejecutar. En un DBMS, los temas son los usuarios, los objetos son elementos de base de datos, y los tipos de acceso son recuperar, actualizar, insertar y borrar. En los sistemas orientados a objetos, los objetos protegidos son las clases o los objetos y los tipos de acceso son la clase de operaciones (métodos). Una combinación (sujeto, objeto protegido, tipo de acceso) o  $(s, o, t)$  es una norma de autorización. Un modelo para describir las normas de autorización se da en la figura 1.1 de la página 14.

Un amplio modelo de acceso puede incluir también: un predicado, una

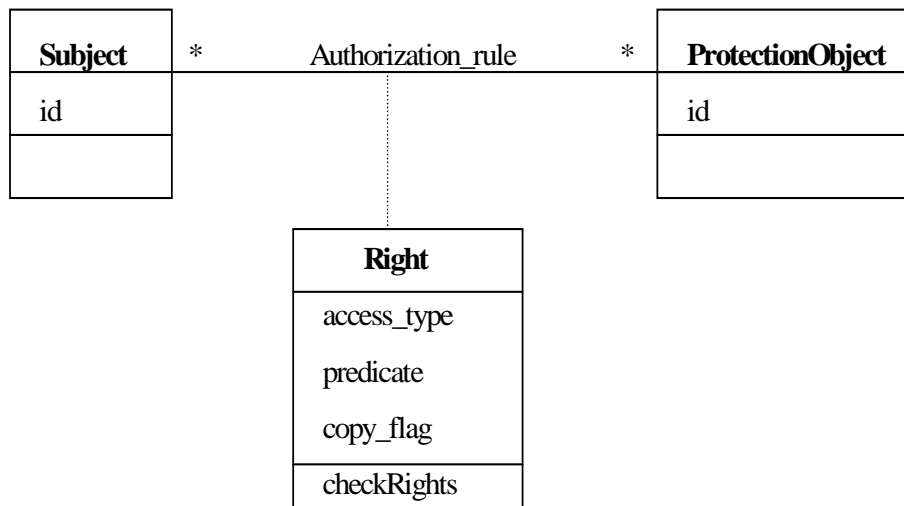


Figura 1.1: The authorization pattern.

condición, una copia de la bandera, un autorizador. El predicado define el contenido que dependen las limitaciones en el acceso, la condición establece una condición que debe ser cierto en el caso de la norma a aplicar (un guardia), si la copia bandera 'en la' autoriza al objeto de la norma de conceder el derecho a otros usuarios, y el autorizador indicó que ha realizado esta autorización. En este caso, la autorización de la regla tiene la forma  $(a, s, o, t, p, c, f)$ . Los sistemas de bases de datos generalmente utilizan un subconjunto  $(s, o, t, p)$ .

En la matriz original de Lampson [19], tenía el concepto de propietario, que, como hemos discutido anteriormente, es una violación del principio de separación de servicio. Tenía también el concepto de “controlador”, la definición de derechos especiales de un proceso sobre el otro.

La matriz de Lampson y su extensión [20], incluyen las operaciones de modificar la matriz y permiten la propagación de los derechos. Llamamos a estas operaciones “administrativa” de operaciones, ya que normalmente serían utilizadas por un administrador de seguridad. Estos incluyen:

- $\text{transfer}\{t/t^*\}$  to  $M(s,o)$ —la transferencia puede ser destructiva o no, dependiendo de la política.
- $\text{grant}\{t/t^*\}$  to  $M(s,o)$ — una subvención en tanto otorgante y el conce-

sionario tiene derecho después de la concesión.

- delete t from M(s,o)– se elimina el derecho de un sujeto.
- read M(s,o)- inspección del derecho de un sujeto por un objeto.
- create object o
- delete object o
- create subject s
- delete subject s

Harrison, Ruzzo y Uhlman [21], ampliaron y formalizaron este modelo (modelo de la Dependencia de Derechos Humanos), para demostrar seguridad. La principal diferencia entre este modelo y la matriz de acceso descrito anteriormente es la forma en que la matriz es cambiada. Utilizan muchas de las mismas operaciones, pero añaden un conjunto de comandos de la aplicación a estas operaciones. Un comando tiene la estructura:

```
Command c(x1, x2, .xk) //the x's stand for s or o
if t1 in M(s1,o1) and
if t2 in M(s2,o2) and
..
if tm in M(sm,om)
then op1,op2,.., opn
end
```

En particular, se demostró que el problema de seguridad para el acceso a la matriz no está resuelto. Esto significa que a partir de un estado inicial, donde un sujeto s no tiene derecho sobre un objeto o, no es posible decidir si s puede obtener el derecho en un determinado número de pasos. Algunas variaciones de la matriz de acceso se han propuesto para tratar de obtener un modelo en el que la seguridad es resuelta.

Sabemos que en un simple modelo, el recorrido de concesión [22], la seguridad es lograda, el problema es encontrar algo entre tener acceso a la concesión

y a la matriz de que la seguridad sigue siendo alcanza. Un enfoque más práctico es utilizar una versión obligatoria de la matriz de acceso, tales como RBAC, donde en general los usuarios no pueden transferir sus derechos.

Una aplicación de la matriz de acceso debe tener una manera de almacenar adecuadamente la autorización de las normas, interceptar las peticiones del usuario o el programa, para luego comparar la solicitud de acceso a la matriz para decidir si otorgarlo o no. Este es el interceptor de monitor de referencia. Un patrón del monitor de referencia da al final del capítulo y se ilustra el modelo utilizado para describir los patrones en la literatura estándar. Normalmente, una petición (s', o ', t ") es comparada por el monitor de referencia con las normas de acceso. Si hay una (s, o, t), el acceso es validado y se ha completado la solicitud, de lo contrario la solicitud es denegada.

Las políticas especiales son necesarias cuando un sujeto o un objeto puede implicar otros, lo que se discute en la sección de autorización implícita. Otra consideración es cómo decidir cuando hay predicados involucrados.

### **1.15.2. Control de Acceso basado en funciones - RBAC (Role - Based Access Control)**

RBAC puede considerarse como una variación de la matriz de acceso, donde los sujetos sólo pueden ser funciones. Un rol corresponde a un trabajo o funciones dentro de un puesto de trabajo, por ejemplo, un profesor puede tener las funciones de profesor, investigador, consejero, presidente de tesis, y otros. Los derechos se asignan a las funciones, no a los individuos. Si los usuarios son asignados a las funciones y los derechos sólo dado por un administrador de este tipo se convierte en un modelo de carácter obligatorio.

RBAC convenientemente puede aplicar las políticas de mínimos privilegios, y la separación de funciones. Menos privilegios pueden ser implementados mediante la asignación a cada rol, sólo los derechos que necesita para desempeñar sus funciones. La separación de funciones puede ser implementada a través de roles mutuamente excluyentes. Un patrón para RBAC se muestra en la figura 1.2 de la página 17 [23]. Este modelo utiliza también el concepto de período de sesiones para limitar aún más los derechos utilizados en un momento dado y para hacer cumplir la separación de servicio.

Una manera de hacer cumplir la política de mínimos privilegios es asignar derechos a las funciones de casos de uso [18]. Los casos de uso se utilizan



y. NET componente autorización. Asimismo, se ha aplicado a bases de datos orientadas a objetos, donde los derechos de acceso pueden ser heredados a lo largo de la generalización o agregación de jerarquías.

#### 1.15.4. Los modelos multinivel

Este tipo de modelo corresponde a las múltiples políticas en que los datos se clasifican en niveles de sensibilidad y los usuarios tienen acceso de acuerdo con sus autorizaciones. Debido a la forma del control de seguridad también han sido llamados modelos de flujo de datos, que permite controlar el flujo de datos entre los niveles. Estos modelos se han formalizado en tres formas diferentes:

- *El modelo de La Bell-Padula:* Destinados a controlar las fugas de información entre los niveles.
- *El modelo de Biba:* Que controla la integridad de los datos.
- *El modelo de celosía:* Generaliza los niveles parcialmente ordenados de los modelos anteriores utilizando el concepto de matemática de celosías.

#### Modelo de confidencialidad La Bell - Padula

Este es un modelo de confidencialidad. Que clasifica los temas y datos en niveles de sensibilidad. Son ortogonales estos niveles, los compartimientos o categorías están definidos, y corresponden a las divisiones o agrupaciones dentro de cada nivel. La clasificación,  $C$ , de los objetos de datos define su sensibilidad. Del mismo modo, los usuarios o temas en general tienen sus niveles. En cada nivel superior de acceso de la matriz se va refinando el control de acceso.

Un nivel de seguridad se define como un par (nivel de clasificación, conjunto de categorías). Un nivel de seguridad domina otro si y sólo si su nivel es mayor o igual que las otras categorías y su nivel incluye las otras categorías. Dos propiedades, conocidas como “no leer” y “no escribir”, definen un flujo seguro de información:

- **Propiedad de seguridad simple (ss):** Un sujeto  $S$  puede leer un

objeto O sólo si su clasificación domina la clasificación del objeto, es decir,  $C(s) \Rightarrow C(o)$ . Esta es la no-lectura de la propiedad.

- **\*- Propiedad:** Un sujeto S que puede leer un objeto o se le permite escribir sobre un objeto p sólo si la clasificación de p domina la clasificación de la o, por ejemplo, el  $C(p) \Rightarrow C(o)$ . Esta es la no escritura de la propiedad.

Este modelo también incluye sujetos de confianza que se les permite violar el modelo de seguridad. Estos son necesarios para el desempeño de las funciones administrativas (por ejemplo, desclasificar documentos, el aumento de un usuario de liquidación), pero hace que la prueba de las propiedades de seguridad sean menos creíbles. Este modelo se complementa con el modelo de integridad Biba a continuación. La figura 1.3 de la página 20 muestra un patrón para describir este modelo.

### El modelo de integridad Biba

El modelo de Biba clasifica los datos en los niveles de integridad y define dos propiedades dobles de seguridad simple y \* propiedades.

Este modelo incluye las propiedades:

- **Propiedad de seguridad simple:** Un sujeto S puede modificar un objeto o sólo si  $(s) > I(o)$ .
- **Integridad \*- propiedad:** Si un sujeto s tiene acceso para leer un objetos o con el nivel de integridad  $I(o)$ , s puede escribir en el objeto sólo si  $p(o) > I(p)$ .

### El modelo de celosía

Una celosía es una estructura matemática que consta de elementos parcialmente ordenados, donde cada par de elementos tiene un límite superior mínimo y un máximo límite inferior. Como las celosías no son estrictamente de orden jerárquico pueden modelar una mayor variedad de sistemas. Sin embargo, son más difíciles de aplicar que las jerarquías simples. También son difíciles de utilizar, por ejemplo, es muy difícil hacer una celosía de una serie

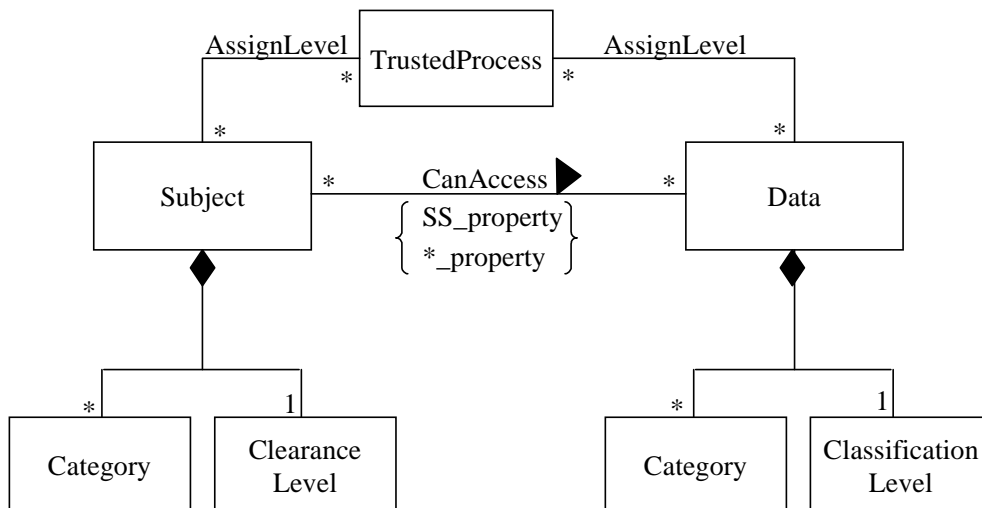


Figura 1.3: Multilevel security pattern.

de elementos de datos con diferentes limitaciones de acceso. Por todo ello, se utilizan con poca frecuencia en la práctica. Una descripción detallada puede encontrarse en D. Denning [28].

### Aplicación de los modelos multiniveles

Los modelos multiniveles son teóricamente los más seguros de los tres modelos básicos de seguridad. Es posible construir DBMS y sistemas operativos que siguen múltiples enfoques. Sin embargo, son difíciles de aplicar porque requieren el etiquetado de los objetos protegidos y de los procesamiento independiente en cada nivel, de lo contrario tenemos la posibilidad de convertirlos en canales (los canales son encubiertas de la banda ancha de canales bajos los cuales pueden ser utilizados para la fuga de información entre los niveles). También son complejos de utilizar, se necesita al menos dos modelos, uno para la confidencialidad y otro para la integridad. Además, en instituciones distintas de la militar es difícil clasificar a las personas y los datos en niveles.

Tienen valor para la aplicación de sistemas que necesitan capas o compartir sus funciones o su administración de la seguridad, por ejemplo, sistemas operativos y servidores de seguridad.

## 1.16. El modelo de Clark-Wilson

Cuando los militares estudiaron el control de la seguridad en los EE.UU., este modelo les llama la atención sobre el hecho de que para las empresas, la integridad de la aplicación era un aspecto mucho más importante que la confidencialidad.

Se hace hincapié en las transacciones bien realizadas y en la separación de servicio.

## 1.17. Modelos y diseño de sistemas de seguro

Estos modelos definen una visión de estados de un sistema seguro, si se parte de un estado inicial seguro y todas las transiciones de estado son seguros vamos a estar siempre en un estado seguro. Esta definición no tiene en cuenta si el estado inicial o las transiciones son significativas o si contradicen las políticas de las instituciones.

Hay ejemplos de tres de las posibles combinaciones de los modelos. Los modelos basados en la matriz de acceso discrecional se han utilizado en la mayoría de sistemas operativos y DBMS hasta hace poco. RBAC es el modelo más común de los sistemas modernos, incluyendo sistemas operativos, DBMS, y servidores de aplicaciones web. Los modelos multiniveles se han utilizado sólo en sistemas militares, aunque, son útiles para controlar los ataques en diferentes partes de un sistema. En particular, Joshi [29] discute la idoneidad de estos modelos para aplicaciones basadas en web. El RBAC se lo considera como el modelo más adecuado, pero en el futuro debe extenderse a consideraciones dinámicas y basadas en tareas.

Una vez que hemos decidido acerca de las políticas que queremos para un sistema dado, el siguiente paso es convertirlas en modelos. Debemos tratar de encontrar el modelo (o combinación de modelos) que coincide con la política de requisitos. Por ejemplo, si tenemos un sistema en el que los usuarios deben tener determinados tipos de accesos a los documentos, es necesario algún tipo de matriz de acceso. Podemos definir los sujetos de esta matriz de acuerdo con las funciones de usuario y si los usuarios no deben conceder o recibir derechos de otros usuarios, está claro que necesitamos RBAC.

El patrón de la figura 1.2 de la página 17 se utiliza como referencia para

definir el sistema: ¿Es necesario el concepto de período de sesiones? ¿Necesitamos estructuras de roles? Normalmente, este modelo cubrirá sólo una parte de los requisitos de la política, y políticas complementarias deben utilizarse. El siguiente paso consiste en reflejar el modelo seleccionado en los niveles inferiores.

## 1.18. El modelo Monitor de Referencia

Presentamos ahora la plantilla que vamos a utilizar para describir los patrones.

- *Intento*

Hacer cumplir las autorizaciones cuando un sujeto solicita un objeto protegido.

- *Contexto*

Un entorno multiprocesamiento en el que los sujetos solicitan objetos protegidos para llevar a cabo sus funciones.

- *Problema*

Si no se define las autorizaciones correctamente es lo mismo que no tenerlas, los sujetos pueden realizar todo tipo de acciones ilegales. Cualquier usuario puede leer cualquier archivo, por ejemplo. ¿Cómo podemos controlar las acciones de los sujetos?.

- *Fuerzas*

Las siguientes fuerzas afectan a la solución:

- - Definir las normas de autorización no es suficiente, debe aplicarse siempre que un sujeto formule una solicitud a un objeto protegido.

- Existen muchas posibles aplicaciones, necesitamos un modelo abstracto de ejecución.

- *Solución*

Definir un proceso abstracto que intercepta todas las peticiones de recursos y controles para el cumplimiento de las autorizaciones.

La figura 1.4 de la página 24 muestra un diagrama de clases con un monitor de referencia. En esta figura `Set_of_Authorization_Rules` denota un conjunto de normas de autorización organizado de una manera conveniente. La figura 1.5 de la página 24 muestra un diagrama de secuencia que muestra cómo se realiza la comprobación.

- *Usos conocidos*

La mayoría de los sistemas operativos modernos aplican este concepto, por ejemplo, Solaris 9, Windows 2000, AIX, y otros. El administrador de seguridad de Java es otro ejemplo.

- *Consecuencias*

Las ventajas incluyen:

1. Si se interceptan todas las peticiones, podemos asegurarnos de que cumplan las normas.
2. La aplicación no se ha limitado al uso de procesos abstractos.

Las desventajas son:

1. Las implementaciones específicas (concretas Monitores de referencia) son necesarios para cada tipo de recurso.
2. Comprobar cada solicitud puede resultar una pérdida de rendimiento intolerable. Es posible que tengamos que realizar algunas comprobaciones en tiempo de compilación. Otra posibilidad es el factor de los controles, por ejemplo al abrir un archivo, con algunos procesos de confianza que no se comprueban.

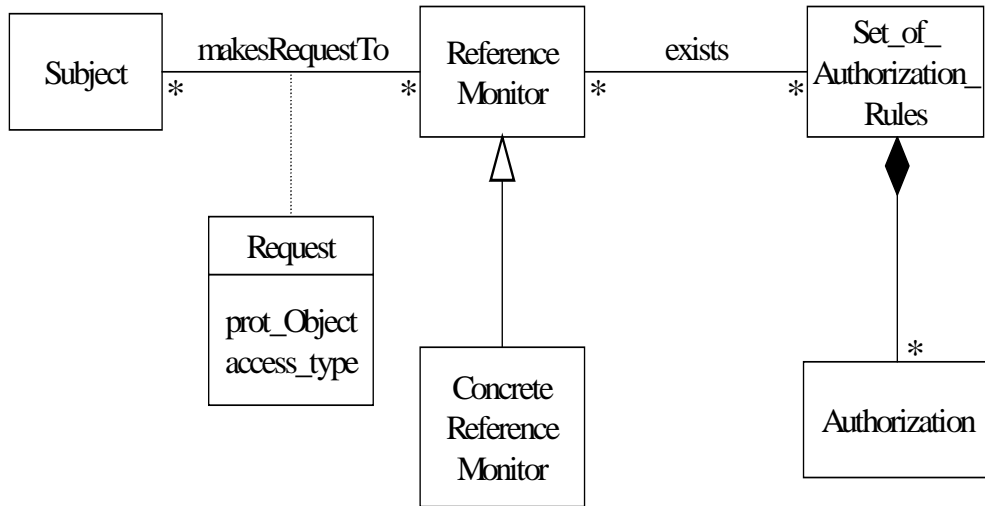


Figura 1.4: Class diagram for the reference mon

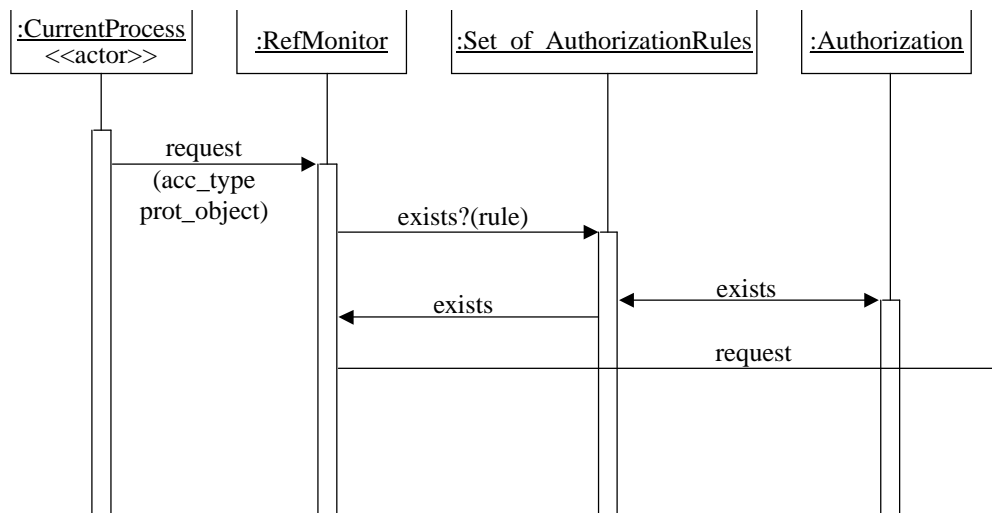


Figura 1.5: Sequence diagram for enforcing security of requests.

# Bibliografía

- [1] Cano; Heimy J. *Pautas y Recomendaciones para Elaborar Políticas de Seguridad Informática (PSI)*. Universidad de los Andes, Colombia, 1998.
- [2] C.Wood; E.B.Fernandez; and R.C. Summers. *Data base security: requirements, policies, and models*. IBM Systems Journal, vol. 19, No 2,229-252, 1980.
- [3] *Policy 200X: Workshop on Policies for Distributed Systems and Networks*. <http://www-dse.doc.ic.ac.uk/events>.
- [4] D. Blacharski. *Emerging Technology: Create order with a strong security policy*. <http://www.networkmagazine.com/article/NMG20000710S0015>, Network Magazine, July 2000.
- [5] M. Andress. *An overview of security policies*. <http://searchsecurity.techtargt.com>, December 2002.
- [6] S. Barman. *Writing information security policies*. New Riders Publ., 2002.
- [7] E.B.Fernandez; R.C.Summers and C. Wood. *Database security and integrity*. Addison-Wesley, 1981.
- [8] E. Lupu and M.Sloman. *Conflict analysis for management policies*. May 1997.
- [9] J.D.Moffett and M. Sloman. *The source of authority for commercial access control*. Computer, IEEE, 59-69, February 1988.
- [10] M. Schaefer. *Reflections on current issues in trusted DBMS*. ARCA Systems, August 1990.
- [11] *Common Criteria home page*. <http://csrc.nist.gov/cc>.

- [12] M. Sloman and E. Lupu. *Security and management policy specification*. IEEE Network, 10-19, March/April 2002.
- [13] J. Vijayan. *Employee data exposed on web*. <http://www.computerworld.com>, Computerworld, Feb. 11, 2002.
- [14] H. Hosmer. *Multiple security policies for business*. Notes for a tutorial at the IFIP/SEC'95 Conference, 1995.
- [15] C.C. Wood. *Information security policies made easy, Version 7*. <http://www.pentasafer.com/products/vsapolicybook.htm>, 2000.
- [16] W.E. Kuenhauser and M. von Kopp Ostrowski. *A framework to support multiple security policies*. Procs. of the 7th Annual Canadian Comp. Security Symp, 1995.
- [17] W.E. Kuenhauser. *A paradigm for user-defined security policies*. Procs. of the 14th IEEE Symp. On Reliable Distributed Systems, 1995.
- [18] E.B. Fernandez and J.C. Hawkins. *Determining role rights from use cases*. Procs. 2nd ACM Workshop on Role-Based Access Control, 121-125. <http://www.cse.fau.edu/ed/RBAC.pdf>, 1997.
- [19] B.W. Lampson. *Protection*. Procs. 5th Annual Conf. on Info. Sciences and Sys., 437-443. Reprinted in ACM Operating Sys. Review, 8, 1 (January 1974), 18-24, 1971, 1974.
- [20] G.S. Graham and P. Denning. *Protection: Principles and practice*. AFIPS Conf. Procs., 40, SJCC, 417-429, 1972.
- [21] M.A. Harrison; W.L. Ruzzo; and J.D. Ullman. *Protection in operating systems*. Comm. of the ACM, 19, 461-471, 8 (August 1976).
- [22] L. Snyder. *Formal models of capability-based protection systems*. IEEE Trans. on Computers, Vol. C-30, No 3, 172-181, March 1981.
- [23] E.B. Fernandez and R. Pan. *A pattern language for security models*. Procs. of PLoP 2001, <http://jerry.cs.uiuc.edu/plop/plop2001>, 2001.
- [24] E. B. Fernandez; J. Wu and M. H. Fernandez. *User group structures in object-oriented databases*. Proc. 8th Annual IFIP W.G.11.3 Working Conference on Database Security, Bad Salzdetfurth, Germany, August 1994.

- [25] R. Sandhu et al. *Role-Based Access Control models*. Computer , vol. 29 , No2,38-47, February 1996.
- [26] GMU Laboratory for Information Security Technology. <http://www.list.gmu.edu>.
- [27] E.B. Fernandez; R.C.Summers and T.Lang. *Definition and evaluation of access rules in data management systems*. Procs. First Int. Conf. On Very Large Databases, 268-285, Boston, MA, 1975.
- [28] D.E. Denning. *Cryptography and data security*. Addison-Wesley, 1982.
- [29] J.B.D.Joshi; W.G.Aref; A. Ghafoor and E. H. Spafford. *Security models for web-based applications*. Comm. of the ACM, vol. 44, No. 2,38-44, February 2001.



# Índice alfabético

CIM, 10

ECMA, 9

funciones

en materia de políticas, 9

introducción, 1

ISO, 9

modelo

de Clark-Wilson, 21

monitor de referencia, 22

modelos

de seguridad, 13

modelos y diseño

de sistemas de seguros, 21

NIST, 9

PCIM, 10

políticas

de seguridad, 3

diseños de sistemas de seguridad,

12

específicas, 5

estándares, 9

necesidad, 2

normas, 10

RBAC, 16

sistema

de políticas, 6

