Fundamentos de la Seguridad Informática

Políticas y Modelos de Seguridad (II)

El Modelo Chinese Wall

Prevención de conflicto de interés

- Modela reglas de acceso en una consultora en la que sus analistas deben asegurar que no se generará conflicto de interés entre sus diferentes clientes
- Política de seguridad: ningún flujo de información puede causar un conflicto de interés

Conflicto de Interés

- Es un concepto bien conocido
- En el mundo financiero un ejemplo es el del analista de mercado que trabaja para una institución financiera proveyendo servicios para negocios corporativos
- Este analista debe preservar la confidencialidad de la información provista a él por el cliente, por lo tanto no puede trabajar para empresas si tiene conocimiento interno de los planes y estado de situación de una empresa competidora
- Sin embargo el analista puede asesorar corporaciones que no están en competencia entre ellas así como poder manejar información general del mercado

Política Chinese Wall

- Introducida por Brewer y Nash en 1989
- La motivación para este trabajo fue evitar que información sensible de una compañia sea revelada a compañias competidoras a través del trabajo de consultores financieros
- Dinámicamente establece los permisos de acceso de un usuario basado en los accesos que ya ha efectuado el mismo

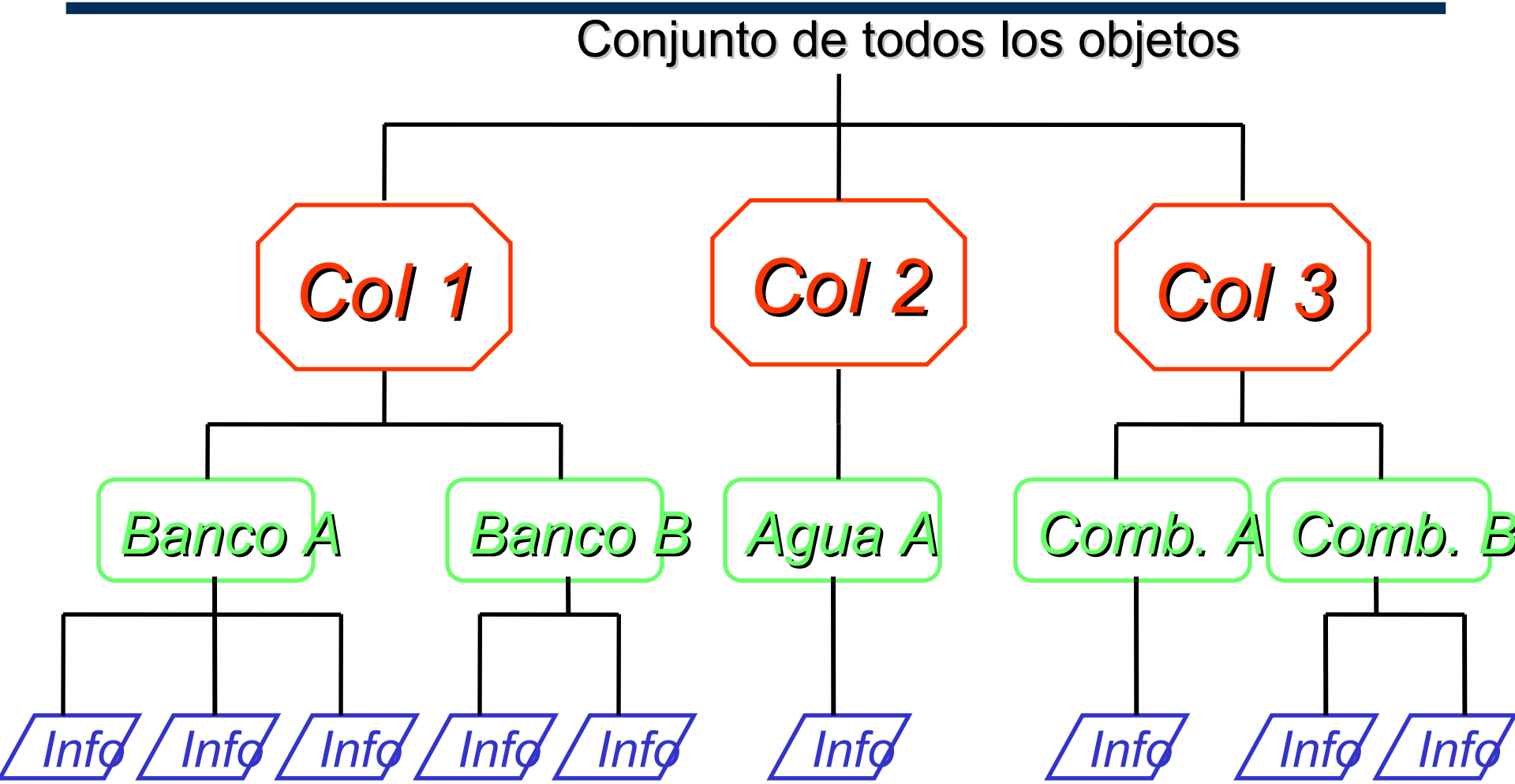
Información Sanitizada

- Brewer y Nash han reconocido la necesidad que tienen los analistas de poder comparar información que ellos tienen con la de otras compañias
- Por lo tanto establecen que las restricciones de acceso pueden no ser aplicadas sobre información sanitizada
- Generalmente, sanitización se realiza disfrazando la información de una corporación de forma de prevenir que se conozca la identidad de la misma

Estado

- Un conjunto de compañias C
- Los sujetos (S) son los analistas, los objetos (O) son unidades de información
- Objetos son organizados en 3 niveles
  - Información
  - Dataset (conjunto de datos de una compañia c)
    - DS: O -> C (función que retorna el dataset de la compañia del objeto argumento)
  - Clases de conflicto de interés (CoI)
    - (CoI: O -> $\wp(C)$)

Clasificación de Datos



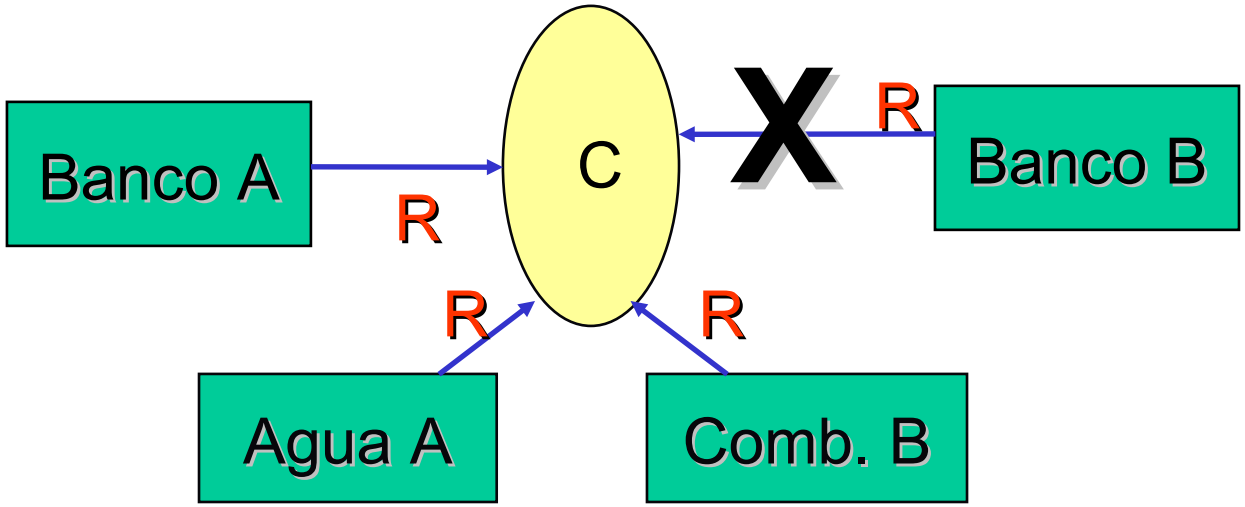
Estado

- El nivel de seguridad de un objeto o es el par (CoI(o), DS(o))
- Información sanitizada es la que no contiene elementos sensibles. El nivel de seguridad es ({}, DS(o))
- Historia de los accesos: $N_{so} = N[s,o] : Bool$
- Políticas de seguridad
  - Prevención de flujo de información directo (Regla Read)
  - Regulación de acceso para escritura (Regla Write)

Prevención de flujo

- Prevenir que un sujeto sea expuesto a un conflicto de intereses: si *s* quiere acceder al objeto *o* entonces
  - El objeto *o* debe pertenecer a un dataset de compañia que el sujeto ya maneja,
  - O si no la compañia de *o* no está en conflicto con las de los objetos que *s* ya ha accedido
- *La propiedad ss*: un sujeto *s* puede acceder a un objeto *o* sólo si para todo *o'* con Nso' = true
  - DS(o) = DS(o'), o si se cumple que
  - DS(o) $\notin$ CoI(o')
- Flujo de información indirecto es todavía posible

Propiedad ss (regla Read)



Comparación con Bell-LaPadula

- La Política Chinese Wall es una combinación de elección libre y control mandatorio
- Inicialmente un sujeto es libre de acceder a cualquier objeto que él desee
- Una vez que la elección inicial se ha hecho , una *Chinese Wall* es creada para ese usuario en torno al conjunto de datos a los que pertenece el objeto
- Notar que Chinese Wall puede ser combinada también con políticas DAC

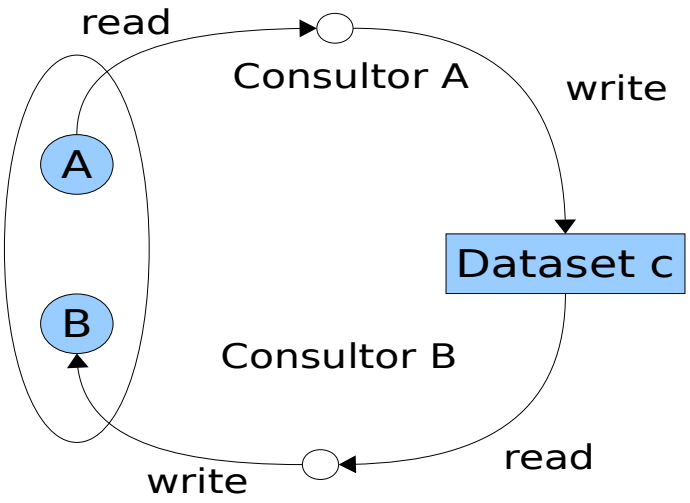
Flujo indirecto

- La regla Read no previene flujo indirecto de información
- Considerar el siguiente caso:
  - Consultor A tiene acceso a
    - Comb. A y Banco A
  - Consultor B tiene acceso a
    - Comb. A y Banco B

  Si a Consultor A se le permite leer Banco A y escribir en Comb A, él podría transferir información acerca de Banco A que puede ser leída por Consultor B

Regulación de escritura

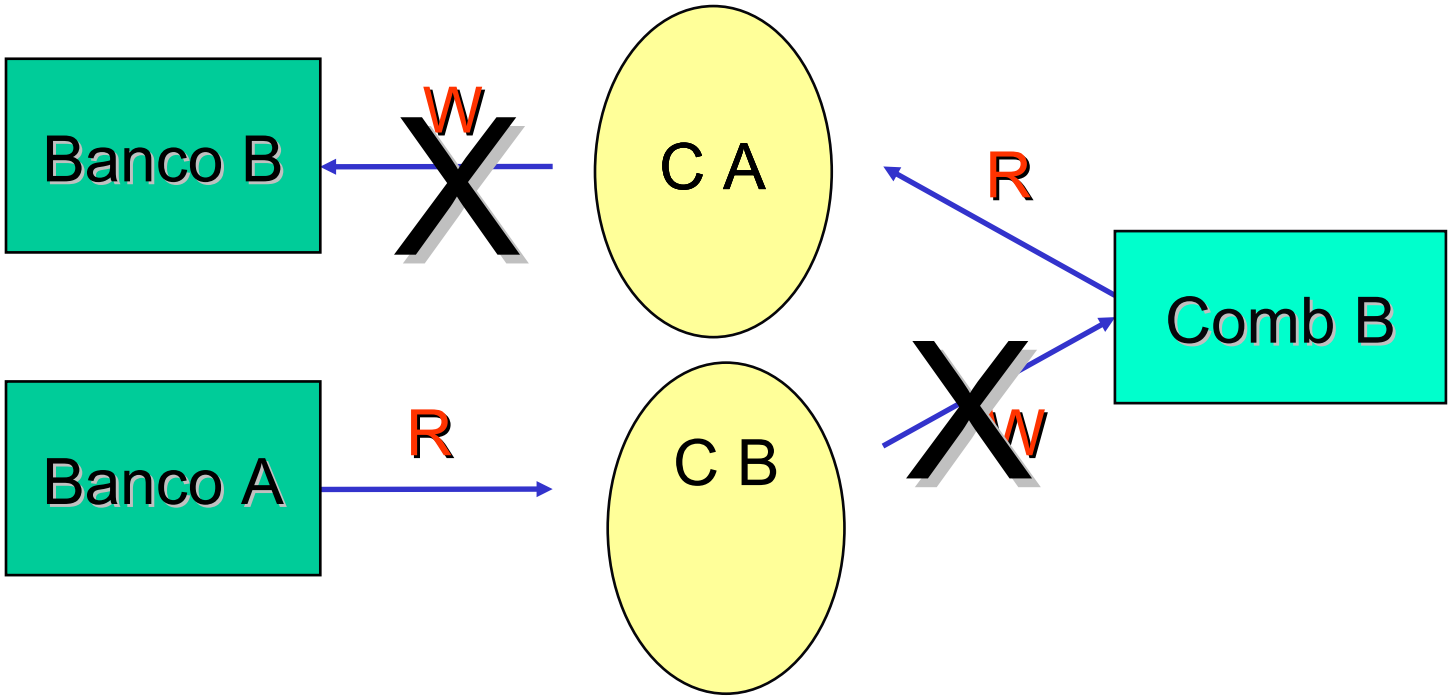
- Flujo de información indirecto en una clase de conflicto de interés



Regulación de escritura

- Acceso de escritura a un objeto *o* es permitido para *s* sólo si, para todo *o'* que puede ser leido por *s, o'* pertenece al mismo dataset que *o* o sino está sanitizado
- *La propiedad* *: Un sujeto *s* puede escribir en un objeto *o*, sólo si, todo o' para el que *s* tiene acceso de lectura
  - DS(o) = DS(o'), o se cumple que
  - CoI(o') = {}

Propiedad * (regla Write)



Confinamiento de información

En definitiva, lo que la regla write establece es que:

El flujo de información es confinado al conjunto de datos de la compañia

Observaciones

- La propiedad * no permite que información que no haya sido sanitizada pueda fluir hacia fuera del dataset
- En contraste con BLP, donde la asignación de permisos de acceso es usualmente estática, en este modelo los permisos de acceso son reasignados en cada transición de estado
- Clearance de un sujeto es una HW mark que puede escalar en el lattice de lectura (***Sandhu 93***)

El Modelo RBAC

RBAC - Motivaciones

- Un problema importante al gestionar sistemas de envergadura es la complejidad que presenta la administración de la seguridad
- Cuando el número de objetos y sujetos es alto, el número de autorizaciones también tiende a ser muy alto
- Más aún, si la comunidad de usuarios varía en forma muy dinámica, el número de operaciones de otorgamiento como de revocación de permisos se torna difícil de gestionar

RBAC - Motivaciones

- A menudo los usuarios no son los propietarios de la información a la que acceden. La corporación o el organismo es el verdadero propietario de los datos.
- El control a menudo está basado en las funciones de los usuarios más que en la propiedad de la información
- RBAC ha sido propuesto como un enfoque alternativo a DAC y MAC con el objetivo de simplificar la gestión de control de acceso y proveer soporte directo a la implementación de control de acceso basado en funciones de trabajo

RBAC - Conceptos básicos

- Roles representan funciones dentro de una organización y las autorizaciones son otorgadas a los roles en vez de a los usuarios (sujetos)
- Los usuarios son entonces autorizados a “jugar” roles, y por lo tanto a adquirir las autorizaciones que le son otorgadas a esos roles

RBAC - Beneficios

- Como los roles representan funciones organizacionales, un modelo RBAC puede directamente soportar políticas de seguridad de la organización
- El otorgamiento y la revocación de autorizaciones a usuarios es fuertemente simplificada
- Los modelos RBAC han demostrado ser neutrales respectos a las políticas que permiten enforzar

Grupo de Seguridad Informática

DAC, MAC y RBAC

- MAC controla acceso en función de etiquetas de seguridad
- DAC controla acceso a un objeto en función de los permisos definidos para ese objeto por su propietario
- RBAC es un componente independiente de control de acceso, que puede convivir sin problemas con DAC y/o MAC

RBAC - estándar

- Proveedores de DBMS han reconocido la importancia y las ventajas de RBAC, actualmente la mayoría de los DBMSs comerciales soportan algún tipo de característica de RBAC
- Asimismo, existe consenso en relación a un estándar de modelo RBAC
- El modelo del NIST [Sandhu,Ferraiolo,Kuhn 00] representa el primer paso hacia la definición de ese estándar
- Existe una definición provista por ANSI: American National Standard for Information technology – role based access control. ANSI INCITS 359-2004, February 2004

Modelo del NIST

- Se definen tres niveles incrementales de capacidades funcionales
  - *Core* RBAC – también llamado *Flat* RBAC
  - *Hierarchical* RBAC
  - *Constrained* RBAC

RBAC- Conceptos básicos

- *Usuario* – es definido como un ser humano, una máquina, un proceso, un agente inteligente autónomo, etc.
- *Rol* – es una función dentro del contexto de una organización con una semántica asociada en relación a su autoridad y responsabilidad

RBAC- Conceptos básicos

- *Permiso* – es un modo de acceso que puede ser ejercitado sobre los objetos del sistema. Tanto los objetos como los permisos son dependientes del dominio. Por ejemplo, en el caso de bases de datos, el conjunto de objetos incluye tablas, columnas, y filas, y el conjunto de modos de acceso incluye operaciones insert, delete y update.

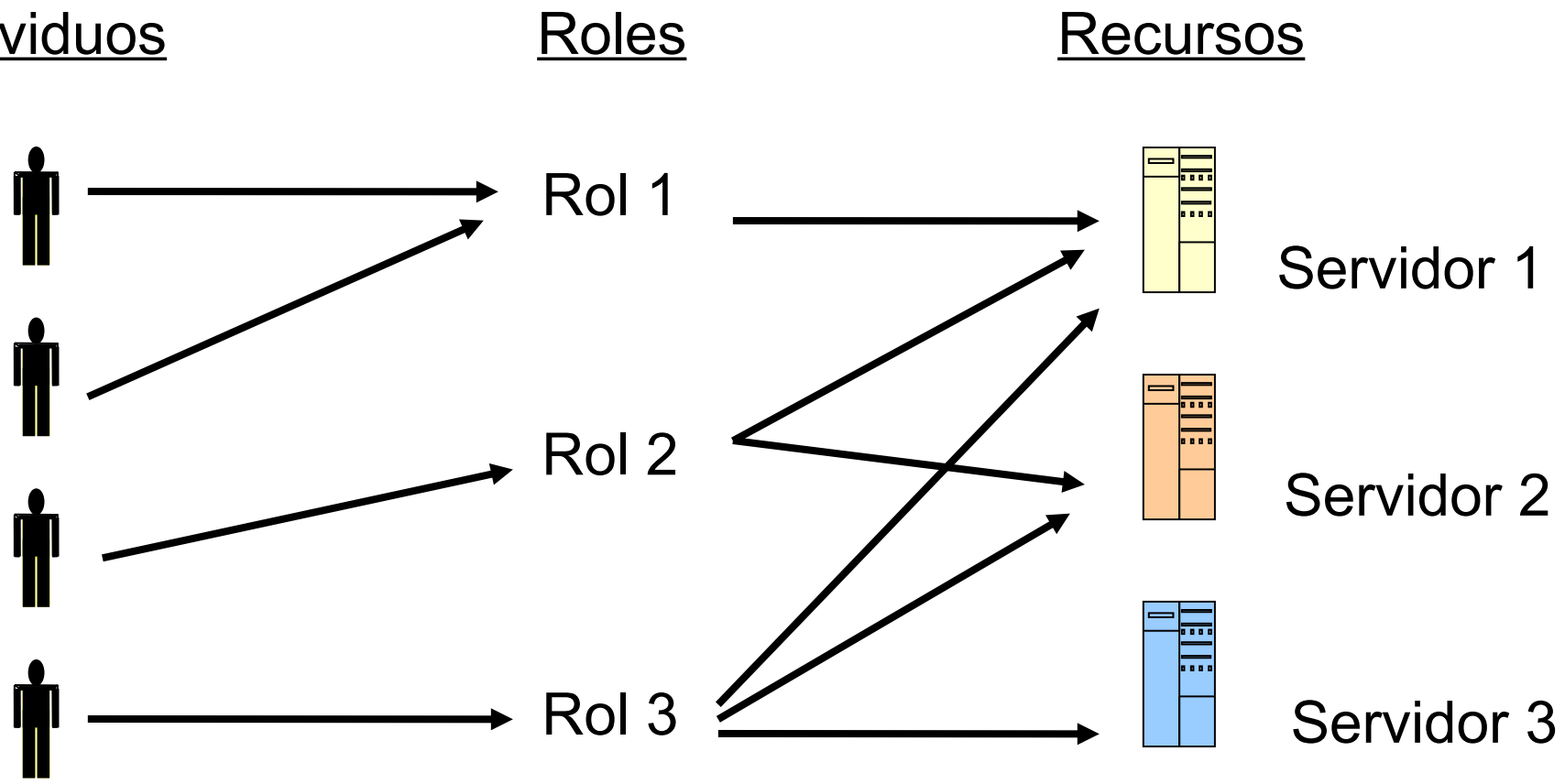
Grupo de Seguridad Informática

RBAC- Conceptos básicos

- *Sesión* – es una instancia particular de una conexión de un usuario al sistema tal que define el sub-coinjunto de roles activados por el usuario. En un momento dado, el usuario puede tener activadas varias sesiones concurrentemente.

  Cuando un usuario se loguea en el sistema, él/ella establece una sesión y, durante esa sesión, puede requerir activar un subconjunto de los roles que está autorizado a jugar. El usuario obtiene todos los permisos asociados con los roles que ha activado en la sesión

Role-Based AC

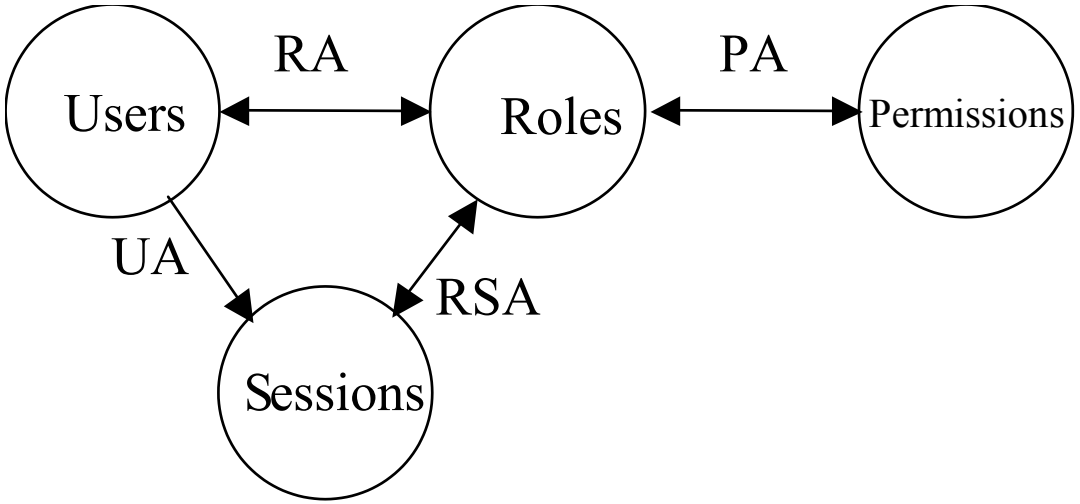


Usuarios cambian con frecuencia, no así los Roles

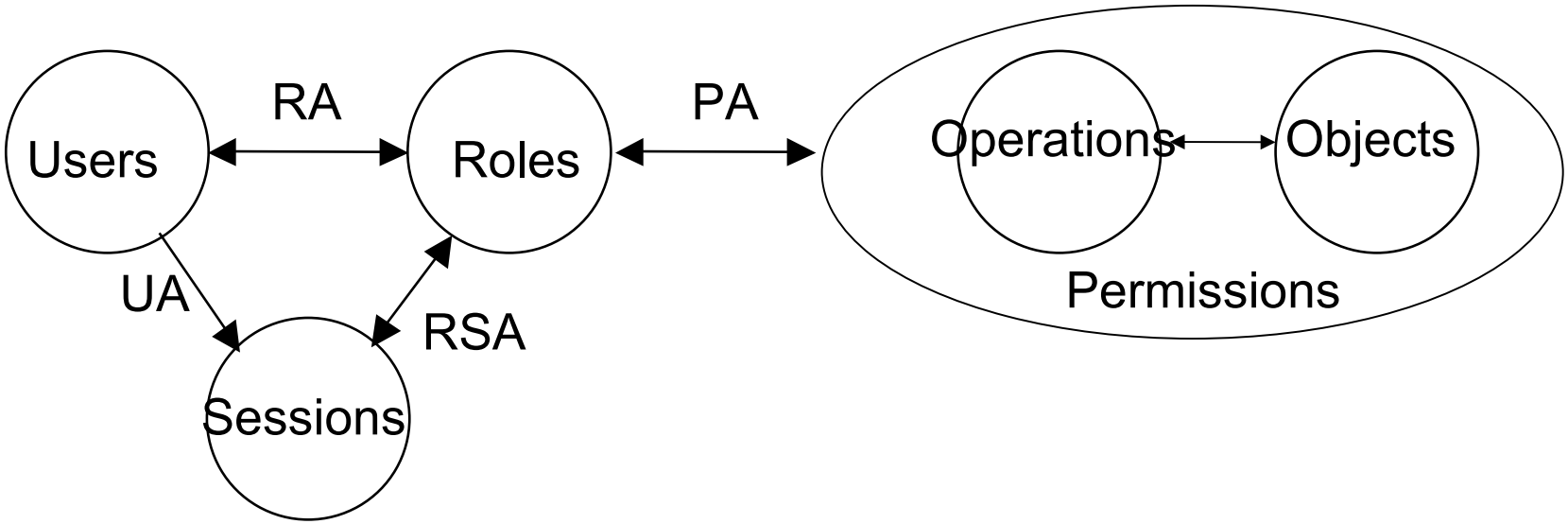
La Familia de Modelos RBAC (Sandhu)

Modelos	Jerarquías	Constraints
$RBAC_0$ (Core)	NO	NO
$RBAC_1$ (Hierarchical)	SI	NO
$RBAC_2$ (Constrained)	NO	SI
$RBAC_3$ (Full)	SI	SI

Core RBAC



Core RBAC - Permisos



Core RBAC - Conjuntos, Funciones, y Relaciones

- *USERS*, *ROLES*, *OPS y OBS*
- $RA \subseteq USERS \cdot ROLES$, un mapping many-to-many de asignación de usuarios a roles
- $assigned\_users: (r: ROLES) \rightarrow 2^{USERS}$, usuarios asociados a un rol dado. Formalmente:
  - $assigned\_users(r) = \{u \in USERS \mid (u, r) \in RA\}$
- $PRMS = 2^{(OPS \cdot OBS)}$, el conjunto de permisos
- $PA \subseteq PRMS \cdot ROLES$, un mapping many-to-many de asignación de permisos a roles
- $assigned\_permissions: (r: ROLES) \rightarrow 2^{PRMS}$, permisos asociados a un rol. Formalmente:
  - $assigned\_permissions(r) = \{p \in PRMS \mid (p, r) \in PA\}$

Core RBAC - Conjuntos, Funciones, y Relaciones

- $Op \in PRMS \rightarrow OPS$, mapping que provee el conjunto de operaciones asociadas con el permiso *p*
- $Ob \in PRMS \rightarrow OBS$, mapping que provee el conjunto de objetos asociados con el permiso *p*
- *SESSIONS* = el conjunto de sesiones
- $session\_users \in SESSIONS \rightarrow USERS$, mapping que retorna el usuario asociado a una sesión
- $session\_roles \in SESSIONS) \rightarrow 2^{ROLES}$, mapping que provee el conjunto de roles activados en la sesión *s*. Formalmente
  - $session\_roles\ (s) = \{r \in ROLES \mid (session\_users\ (s), r) \in RA\}$
- $avail\_session\_perms\ (s : SESSIONS) \rightarrow 2^{PRMS}$, los permisos disponibles para un usuario en una sesión. Formalmente:
  - $avail\_session\_perms\ (s) = \cup_{r \in session\_roles(r)} assigned\_permissions\ (r)$

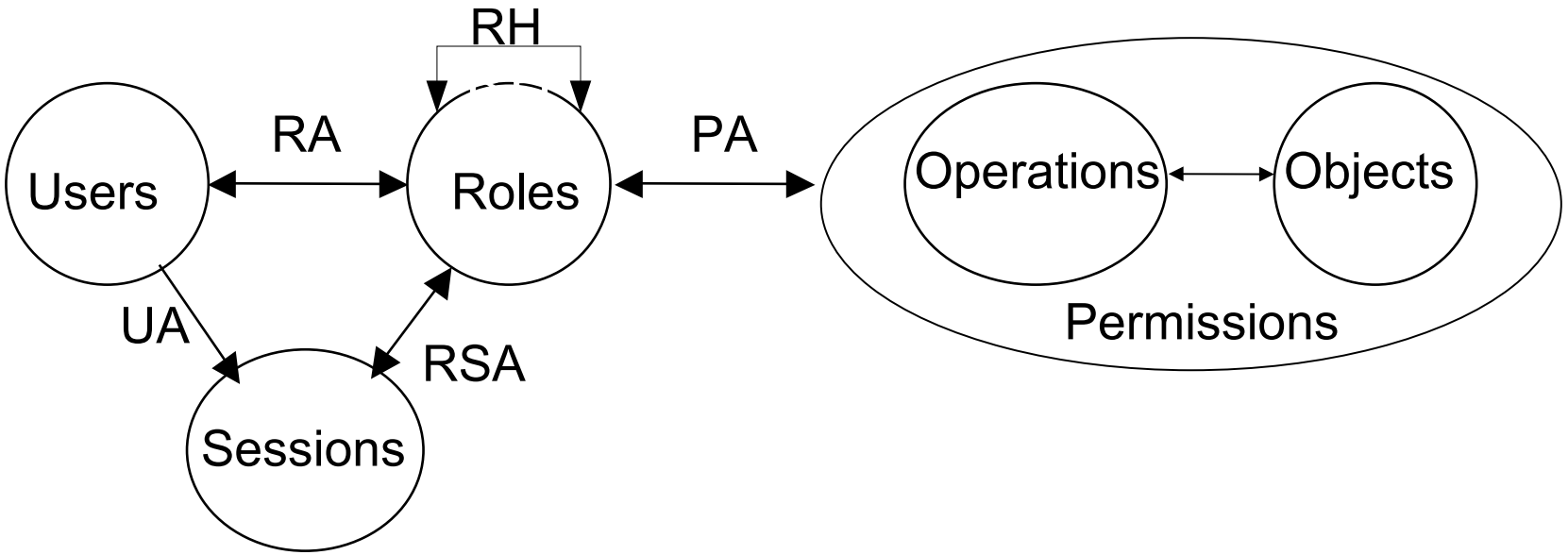
Core RBAC - Sesiones

- La noción de sesión es bastante abstracta – se define como "*a mapping between a user and an activated subset of roles that are assigned to the user*"
- Distinción básica:
  - *Single-role activation (SRA)* Sólo un rol puede ser activado
  - *Multi-role activation (MRA)* Múltiples roles pueden ser activados en una sesión, restricciones dinámicas de separación de responsabilidades (separation of duty, SoD) pueden ser usadas para restringir activación concurrente de algunos roles
  - Compromisos en el uso de estos dos tipos de sesiones

Hierarchical RBAC - Motivaciones

Las jerarquías de roles son un intrumento natural que se puede usar para estructurar los roles de forma que reflejen la línea de autoridad y responsabilidad de una organización

Hierarchical RBAC - Jerarquías de Roles



RH: Relación rol/rol que define herencia de privilegios
Refleja estructuras organizacionales
Dos tipos

- Jerarquía general
- Jerarquía limitada

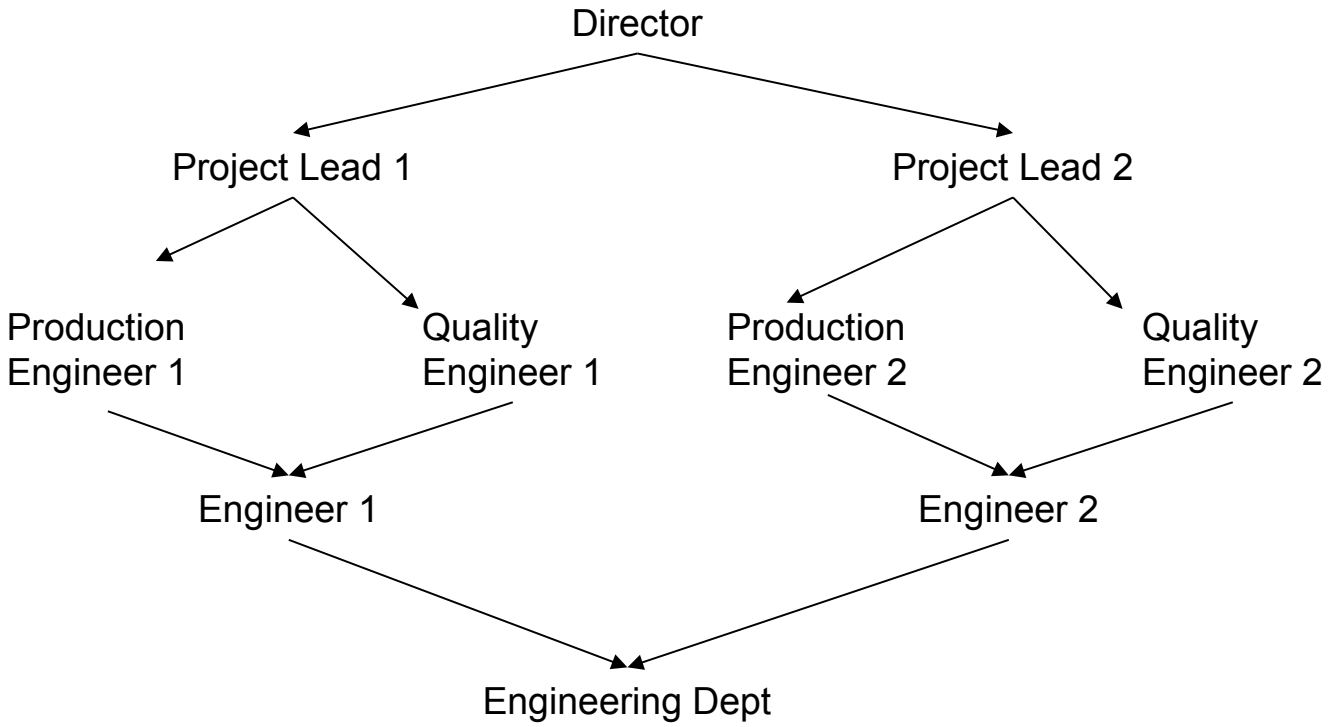
Hierarchical RBAC - Modelo

- $RH \subseteq ROLES \cdot ROLES$ es una relación definida sobre el conjunto de roles del sistema

Tiene que ser irreflexiva y acíclica. Generalmente se refiere a la misma como la *relación de dominance; si* $(r_i, r_j) \in RH$ se dice que r_i domina a r_j

- También se define un orden parcial $\geq$ que es la clausura reflexo-transitiva de *RH*.
- Un sistema RBAC puede elegir entre almacenar $\geq$ o computarla cuando es necesario

Ejemplo de RH



Hierarchical RBAC - Semántica

- *User Inheritance* (*UI*):

  Todos los usuarios autorizados para un rol *r* lo son también para todo rol *r' tal que* $r \geq r'$

- *Permission Inheritance* (*PI*):

  Un rol *r* tiene autorización para todos los permisos para los que todo rol *r'*, tal que $r \geq r'$, está autorizado

- *Activation Inheritance* (*AI*):

  La activación de un rol *r* automáticamente activa todos los roles *r'*, tal que $r \geq r'$. Esta semántica tiene sentido sólo si son usadas sesiones MRA

Constrained RBAC

- Constrained RBAC es un modelo RBAC con la capacidad de soportar políticas que controlan *Separation of Duties*
- Dos categorías principales:
  - SoD Estática
  - SoD Dinámica

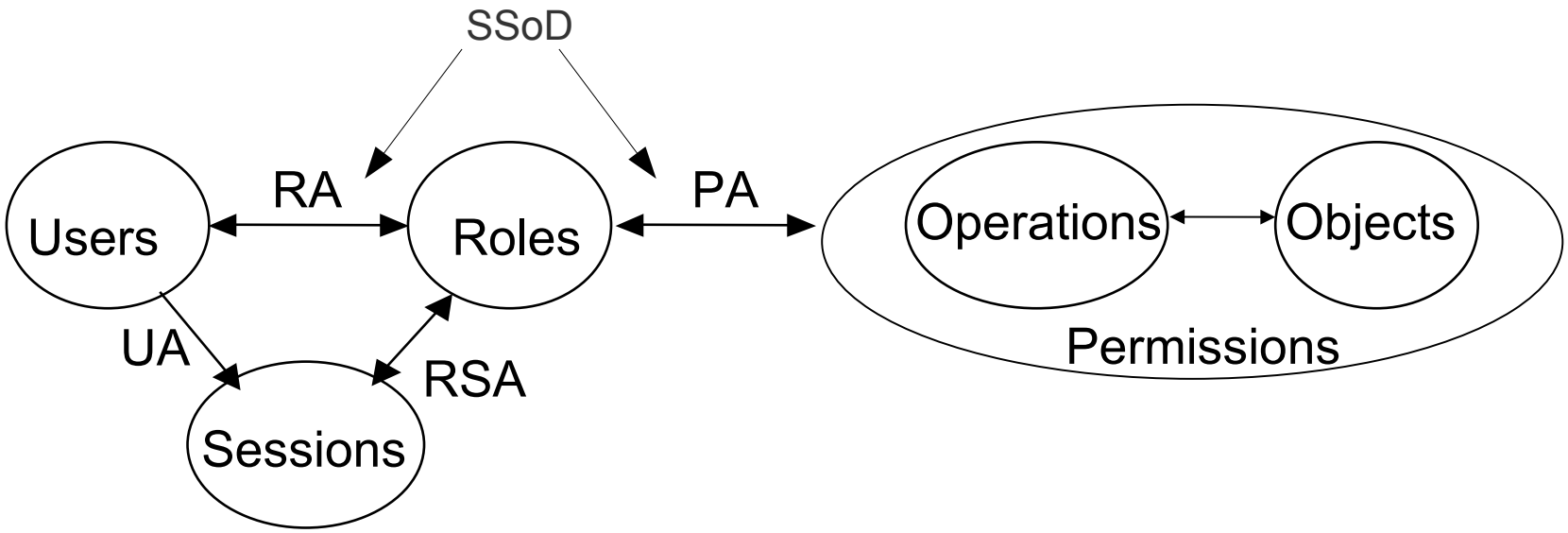
Políticas de SoD

- Permiten aplicar políticas de control de conflicto de interés para prevenir que haya usuarios que puedan exceder un nivel razonable de autorización para su posición
- Permiten asegurar que las fallas de omisión o colaboración dentro de una organización pueden ser causadas solamente como resultado de complicidad entre individuos

Definiciones SoD

- ANSI: “Dividir responsabilidad por la información sensible de forma que un individuo actuando solo no pueda comprometer la seguridad del sistema de procesamiento de datos”
- U.S. Office of Management and Budget’s Circular A-123: “Key duties and responsibilities in authorizing, processing, recording, and reviewing official agency transactions should be separated among individuals”.

Constrained RBAC - Restricciones estáticas (SSoD)

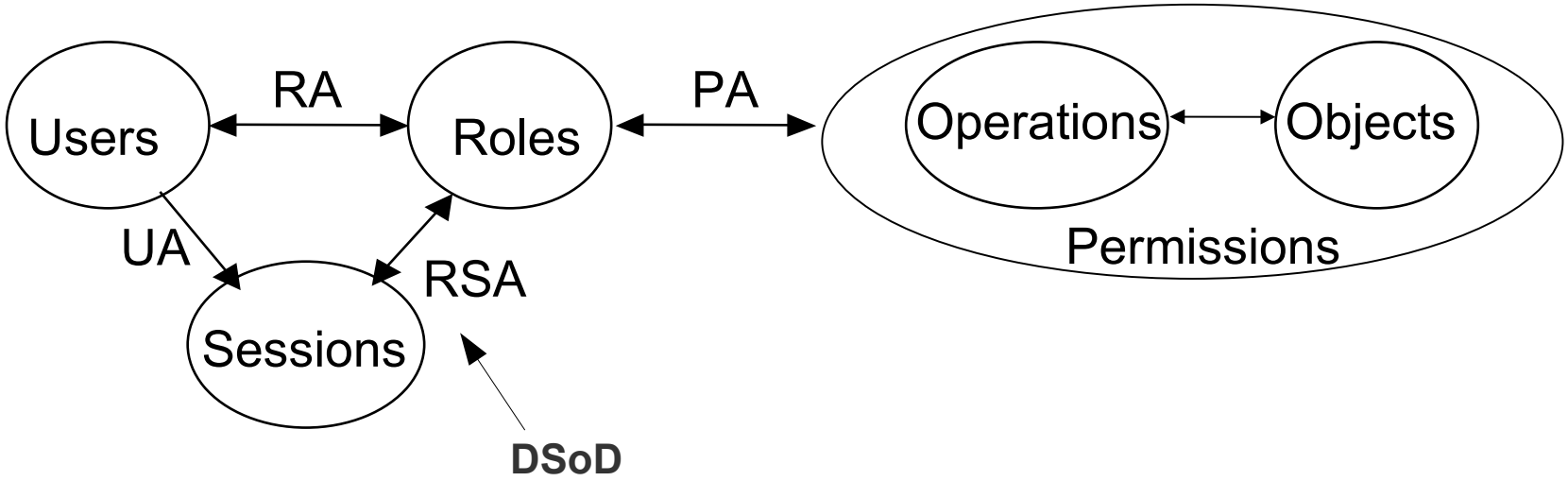


Políticas de SSoD previenen fraudes restringiendo acciones administrativas y por lo tanto restringiendo la combinación de privilegios disponibles para los usuarios

Constrained RBAC – Restricciones estáticas (SSoD)

- SSoD define restricciones sobre el conjunto de roles y en particular sobre la habilidad de formar relaciones *RA (asignación de roles)*.
- Ningún usuario es asignado a *t* o más roles en un conjunto de *m* roles
- Previenen que una persona sea autorizada a usar más roles de los que necesita
- Estas restricciones pueden ser enforzadas (estáticamente) cuando los usuarios son asignados a roles

Constrained RBAC - Restricciones dinámicas (DSoD)



Políticas previenen fraude restringiendo los roles que pueden ser activados en una sesión, restringiendo así los privilegios disponibles para los usuarios

Ejemplo: ningún usuario puede activar a la vez el rol cajero y supervisor de cajero en una sesión, aún cuando este usuario podría ser asignado a cualquiera de los dos roles

Importante para la aplicación de *least privilege*

Grupo de Seguridad Informática

Constrained RBAC – Restricciones dinámicas (DSoD)

- Estas restricciones permiten limitar el número de roles que un usuario puede activar en una sesión
- Ejemplos:
  - Ningún usuario puede activar *t* o más roles del conjunto de roles en cada sesión de un usuario
  - Si un usuario ha usado el rol r_1 en una sesión, él/ella no puede usar el rol r_2 en la misma sesión
- Para poder aplicar estas restricciones es necesario registrar los roles que un usuario ha activado en el transcurso de una sesión

RBAC – Administración del Sistema

- Operaciones Administrativas
  - Crear, Borrar, Mantener elementos y relaciones
- Reviews Administrativos
  - Operaciones de Query
- Funciones de Nivel de Sistema
  - Creación de sesiones de usuario
  - Activación/desactivación de roles
  - Aplicación de restricciones
  - Cálculo de decisiones de acceso

Grupo de Seguridad Informática

Referencias y bibliografía

- **D. Gollman,** *Computer Security***,** Wiley, 2006.
- **E. Bertino**, *Notes of Information Security course*, Purdue University, 2005.
- **D.F.C Brewer, M.J. Nash**, *The Chinese Wall Security Policy*, Proc. IEEE Symp. Research in Security and Privacy, 1989.
- **Rick Wayman**, *What is the "Chinese Wall" and why is it in the News,* ResearchStorck.com, 2001.
- **Ravi S. Sandhu**, *A Lattice Interpretation of the Chinese Wall Policy*. Proc. Of 15th NIST-NCSC National Computer Security Conference, October 1992, Baltimore USA.
- **V. Atluri, S. Chun, P. Mazzoleni**, *A Chinese Wall Security Model for Decentralized Workflow Systems.* Proc. of 8th ACM Conference on Computer and Communications Security (CCS-8), Novembre 200119 Philadelphia, USA

Grupo de Seguridad Informática

Referencias y bibliografía

- **R.S. Sandhu et al**, *Role-Based Access Control Models*. IEEE Computer, 1996.
- **R. Sandhu, D. Ferraiolo, D. Kuhn**, *The NIST Model for Role Based Access Control: Toward a Unified Standard*. In Proc. 5th ACM Workshop on Role Based Access Control, pp.47-63, 2000.
- **E. Bertino, P. A. Bonatti, E. Ferrari**, *Trbac: A temporal role-based access control model*. ACM Transactions on Information and System Security, Vol. 4, No. 3, pp 191–223, August 2001.
- **K.J. Biba,** *Integrity Considerations for Secure Computer Systems*. MITRE report TR-3153, 1977.