Control de Acceso

Curso 2013/14

Grado Gestión Informática Empresarial

Asignatura Auditoría y Seguridad Informática

Profesores Alfredo Cuesta Infante
alfredo.cuesta@ajz.ucm.es
Alberto Herrán González
aherran@ajz.ucm.es

Contenidos

Principios básicos

Prevenir el uso no autorizado, o de un modo no autorizado, de un **recurso**

- Control de acceso $\neq$ Autenticación **!!**

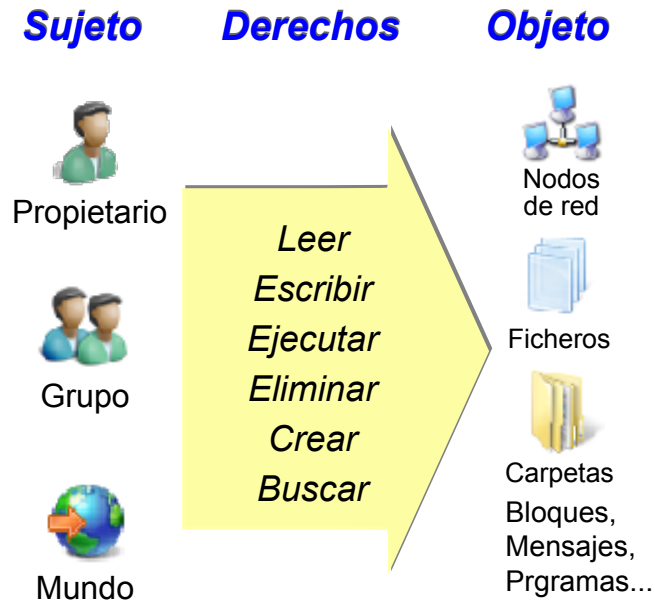
Usuario $\rightarrow$ Autenticación $\rightarrow$ *Sistema* $\rightarrow$ Control de acceso $\rightarrow$ *Recursos*

Definiciones

Sujeto La entidad capacitada para acceder a los *objetos* (Usuarios o procesos)

Objeto El recurso cuyo acceso es controlado

Derecho La acción que realiza el *sujeto* sobre el *objeto*



Principio de privilegio mínimo

"A cada sujeto de un sistema se le otorga el conjunto de privilegios más restrictivo (o la autorización más baja) necesario para el desempeño de sus tareas autorizadas."
La aplicación de este principio limita el daño que puede generar un accidente, error o uso no autorizado

Requerimientos de un sistema de control de acceso

Granularidad

- **Grano grueso** (*coarse grain*) significa que se asignan privilegios masivamente. Proporciona una rebaja en la carga de trabajo.
- **Grano fino** (*fine grain*) significa que se asignan privilegios de manera individual, a usuarios o archivos concretos.
- El sistema de control de acceso debería proporcionar **ambos**.

Grano grueso

Grano fino

Políticas abiertas y cerradas

- En las **políticas abiertas** las autorizaciones especifican qué accesos están prohibidos. Todo lo demás está abierto.
- En las **políticas cerradas** las autorizaciones especifican qué accesos están permitidos. Todo lo demás está prohibido.

Combinación de politicas y resolución de conflictos

Si hay varias políticas de control de acceso se debe contar con un protocolo para resolver los conflictos que puedan surgir.

Control de Acceso Discrecional

En inglés *Discretionary Access Control* o DAC

En qué consiste

- Método tradicional: acceso según solicitante + reglas de acceso (autorizaciones)
- Discrecional = No sujeto a reglas.
  Los usuarios pueden cambiar los permisos de aquellos ficheros que les pertenecen.
- 3 estructuras de datos:
  - Matriz de accesos
  - Listas de control de acceso (ACL)
  - Tickets de capacitación

Matriz de accesos

- Sujetos frente a objetos
- Cada celda contiene los derechos
- Generalmente es una matriz *dispersa*
  $\rightarrow$ Se emplean las otras dos

ACL

- Descomposición de la matriz en cols.
- Cada sujeto tiene un elemento asociado en el ACL del objeto
- Fácil de averiguar todos los sujetos que tienen un derecho sobre un objeto

Tickets de capacitación

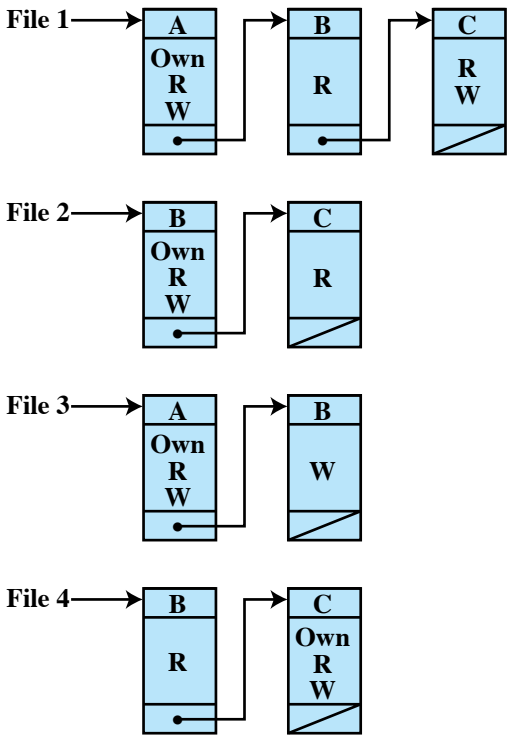
- Descomposición de la matriz en filas
- Cada elemento de la lista es un *ticket*
- Cada objeto tiene un ticket
- Cada sujeto tiene una lista de tickets
- Fácil de averiguar los objetos sobre los que el sujeto tiene un derecho

Ejemplo de estructuras DAC

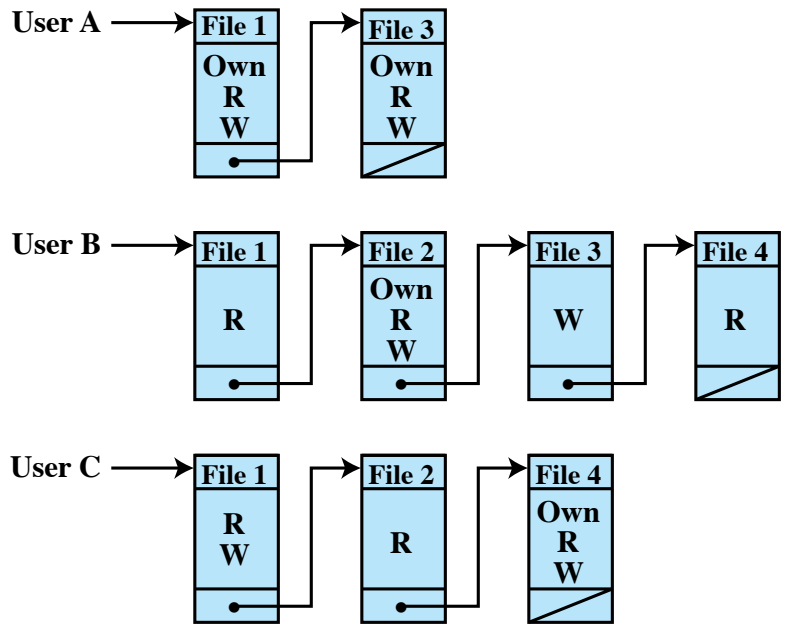
¿Tiene el sujeto **S** el derecho **D** sobre el objeto **O**?

		OBJECTS			
		File 1	File 2	File 3	File 4
SUBJECTS	User A	Own Read Write		Own Read Write	
	User B	Read	Own Read Write	Write	Read
	User C	Read Write	Read		Own Read Write

(a) Access matrix



(b) Access control lists for files of part (a)

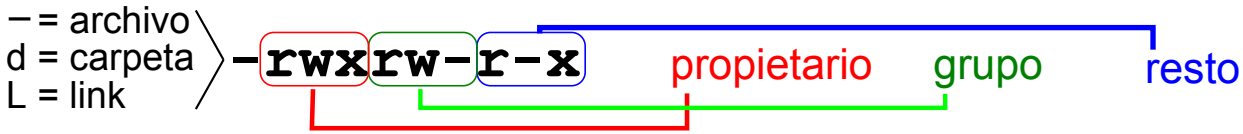


(c) Capability lists for files of part (a)

DAC en la práctica

Estandar POSIX
(Portable Operating System Interface)

- 9 bits que asignan:
  - 3 permisos sobre un objeto:
    lectura (r), escritura (w) y ejecución (x)
  - 3 clases: propietario, al grupo o a otros.



- El estardar no implementa ACLs
- ↓ Los grupos pueden no reflejar la estructura organizativa
- ↓ Diferentes administradores y aplicaciones aplican sus propios trucos para extender estas capacidades.
- ∗ En 1998 se retiró el patrocinio a los grupos de trabajo para estendarizar ACLs.

SO basados en Unix

- Cada usuario tiene un UID (*User ID*) que es único.
  * Almacenados en el archivo `/etc/passwd`
- Cada usuario pertenece a un grupo principal y, opcionalmente, a otros grupos.
- Cada grupo tiene un GID (*Group ID*) que es único.
  * Almacenados en el archivo `/etc/group`
- Los ficheros son *propiedad* de quien los crea.
- Implementaciones:
  - **Linux:** Parche con las propuestas en el Kernel de Nov.2002
  - **FreeBSD:** *Minimal ACL* + *Extended ACL*

Minimal ACL

Protección clásica basada en permisos de 9 bits = UGO (**U**ser **G**roup **O**ther)

Extended ACL

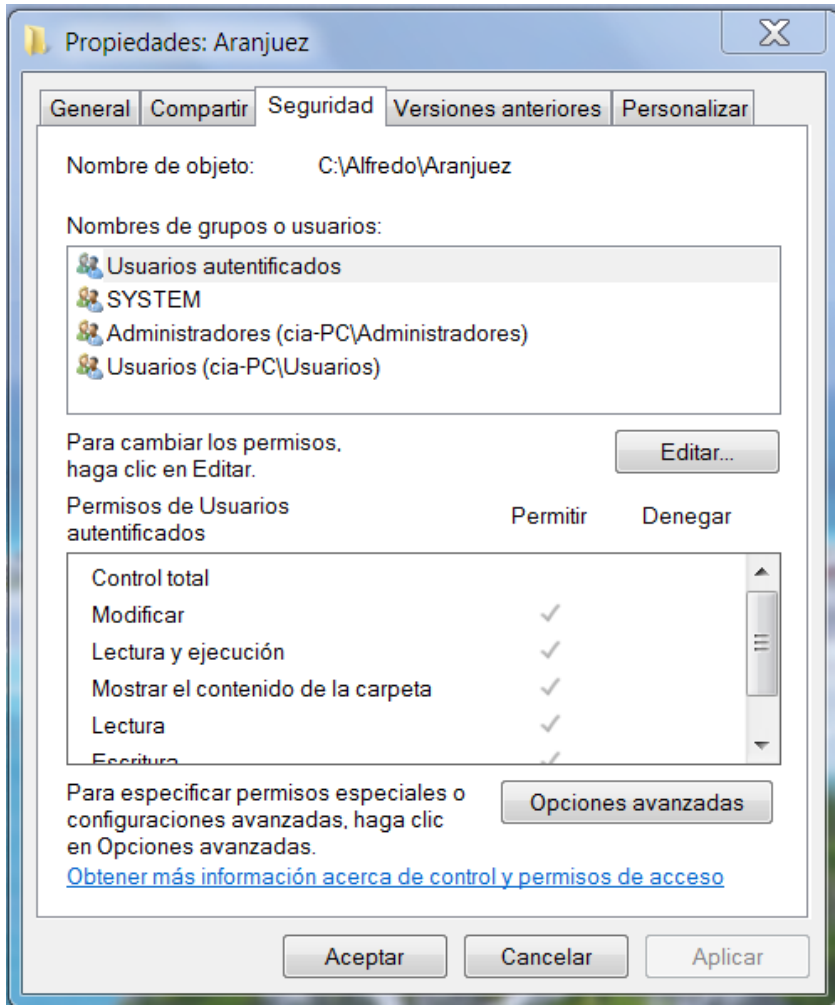
- Se añaden grupos y usuarios a la lista ACL,
- cada uno de ellos con permisos de 3 bits extra +
- una máscara que impide permisos si no coinciden con ella.

Ej. Si la máscara es `mask::rw-`,
y hay un permiso para el usuario `user:alfredo:r-x`,
entonces el usuario `alfredo` sólo tiene pérmiso de lectura ya que es el único que coincide en permiso y en máscara.

ACLs en Windows y MacOS

Windows y MacOS

- El sistema de archivos FAT no tiene control de accesos. Todos los usuarios pueden acceder a todos los recursos.
- Con el sistema de archivos NTFS, desde WinXP en adelante sí.
- Sólo los usuarios que tienen permisos sobre un objeto pueden ver la pestaña *Seguridad* haciendo clic-derecho y seleccionando *propiedades*.
- Alternativamente se puede utilizar el comando `cacls`
- MacOS X implementa los mismos estandares POSIX que Linux. También implementa el estandar NFSv4 cuando se utiliza el sistema de archivos HFS+

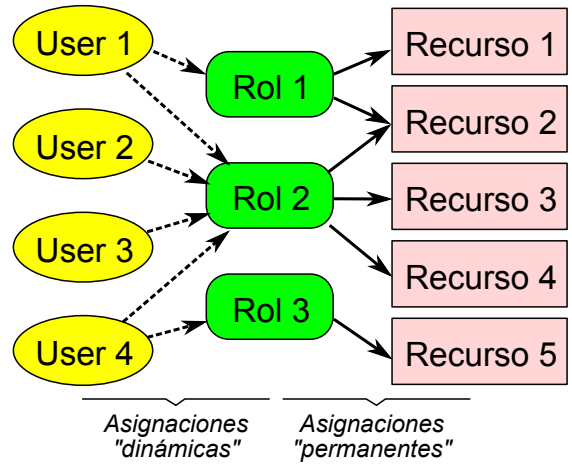


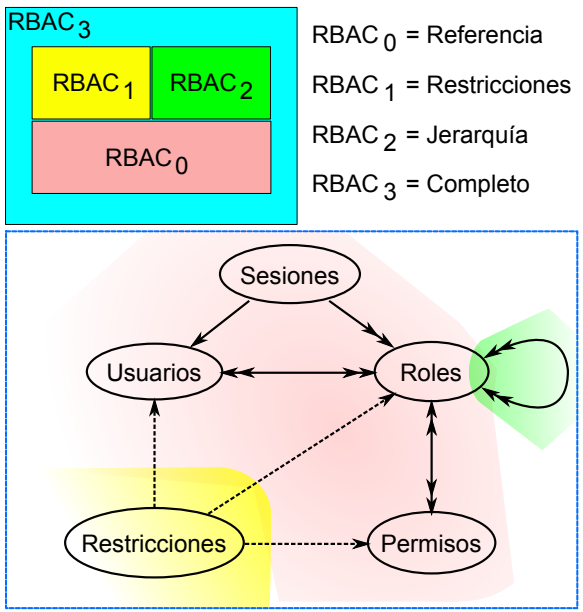
Control de Acceso Basado en Roles (1/3)

En inglés *Role-Based Access Control* o RBAC

¿Por qué hace falta otro?

- Las políticas de seguridad son *dinámicas* porque
  - Los usuarios cambian con frecuencia de rol en la organización, pero un rol suele necesitar siempre los mismos recursos para su tarea.
  - Es muy frecuente dar de alta o baja a los usuarios pero muy raro crear o eliminar roles.
- Los roles reflejan la estructura organizativa.





Modelo Shandu *et al.* 1996

Consiste en un modelo de referencia $RBAC_0$ al que se le añaden 2 componentes: $RBAC_1$ y $RBAC_2$.

- **$RBAC_0$:** Relaciona usuarios y roles con sesiones, y roles con permisos.
- **$RBAC_1$:** Añade restricciones
- **$RBAC_2$:** Añade jerarquías

En el gráfico de la izquierda
las flechas con 1 punta indican relacion 1 a 1;
y con 2 puntas relación 1 a varios.

Control de Acceso Basado en Roles (2/3)

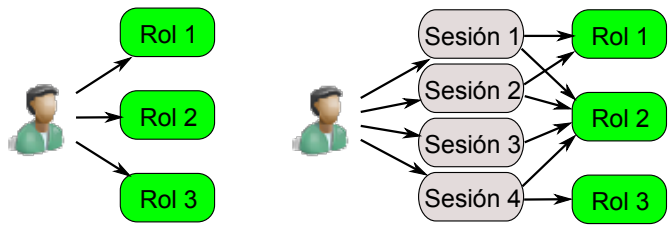
Terminología

Permiso Aprobación de un modo concreto de acceder a uno o más recursos.

Sesión Relación temporal entre un usuario y los roles.

- El usuario establece la sesión sólo con los roles necesarios para la tarea que debe realizar.
- Proporciona flexibilidad y granulidad.
  Grano *fino* es la posibilidad de especificar tareas muy concretas
- Se asegura el principio de privilegio mínimo.
- Una sesión no es una asignación de roles al usuario **!!**

Asignación de roles vs. Establecimiento de sesiones



Jerarquía Relaciones entre los roles

- Más responsabilidad tendrá más autorización de acceso.
- Roles subordinados a otros tendrán un subconjunto de los derechos de los roles por encima.
- Un rol puede heredar los derechos de los roles subordinados.

Restricciones Limitaciones sobre usuarios, roles o permisos.

- **Roles mutuamente exclusivos.**
  La sesión limita el acceso a 1 único rol.
  Un permiso sólo puede ser otorgado a un conjunto de roles.
- **Cardinalidad.** Es la limitación en cuanto al número de roles.
  Imponer un máximo de 5 roles por sesión.
  Imponer un máximo de 3 roles para el permiso de borrado.
- **Prerequisitos.** Un permiso/rol se otorga sólo si ya tiene otro necesario.

Control de Acceso Basado en Roles (3/3)

Modelo NIST 2001

En 2001 el *National Institute of Standards and Technology* incorpora la
Especificación Funcional Administrativa y de Sistemas
(*System and Administrative Functional Specification*).

Dicha especificación clasifica las funciones en 3 categorías

- **Funciones administrativas**
  Capacitan para crear, borrar o gestionar elementos o relaciones entre ellos.
- **Funciones de ayuda/soporte al sistema**
  Funciones para la gestión de sesiones y el control de acceso.
- **Funciones de revisión**
  Capacitan para realizar consultas a los elementos o relaciones.

En la siguiente transparencia hay ejemplos de las funciones de cada tipo para cada uno de los componentes del NIST2001:

1. Núcleo
2. Jerarquías
3. Separación Estática de tareas (SSD)
4. Separación Dinámica de tareas (DSD)

Componentes	Funciones		
	Administrativas	**Ayuda / Soporte**	**Revisión**
Nucleo	-- Añadir / Borrar -- Añadir / Borrar roles, -- Crear / Eliminar asignaciones usuario - rol -- Crear / Eliminar asignaciones permiso - rol	-- Crear una sesión con un conjunto de roles activo por defecto. -- Añadir / Borrar un rol a una sesión. -- Comprobar que el sujeto de una sesión tiene permiso para acceder al objeto que quiere.	Funciones para revisar todos los elementos, relaciones, usuarios, asignaciones de roles, sesiones…
Jerarquías	-- Crear / Borrar una relacion de herencia inmediata entre 2 roles (*) -- Crear un nuevo rol, Ascendente / Descendente de otro que ya existe (*)		Ver pemisos y usuarios asociados, ya sea directamente o mediante herencia, a cada rol.
Relaciones estáticas de separación de tareas (Static Separation of Duty , SSD)	-- Crear conjuntos excluyentes de roles (un usuario se asigna a un conjunto). -- Asignar *cardinalidad* = número máximo de usuarios permitidos en una SSD		Ver las propiedades de cada SSD
Relaciones dinámicas de separación de tareas (Dynamic Separation of Duty , DSD)	Igual que SSD (limitar los permisos de un usuario) pero ahora las limitaciones se hacen efectivas a partir de un núm. de sesiones abiertas a la vez.		

(*) El rol R1 es descendiente de R2 si R1 incluye todos los permisos de R2 (hereda de R2) y todos los usuarios asignados a R1 son también asignados a R2