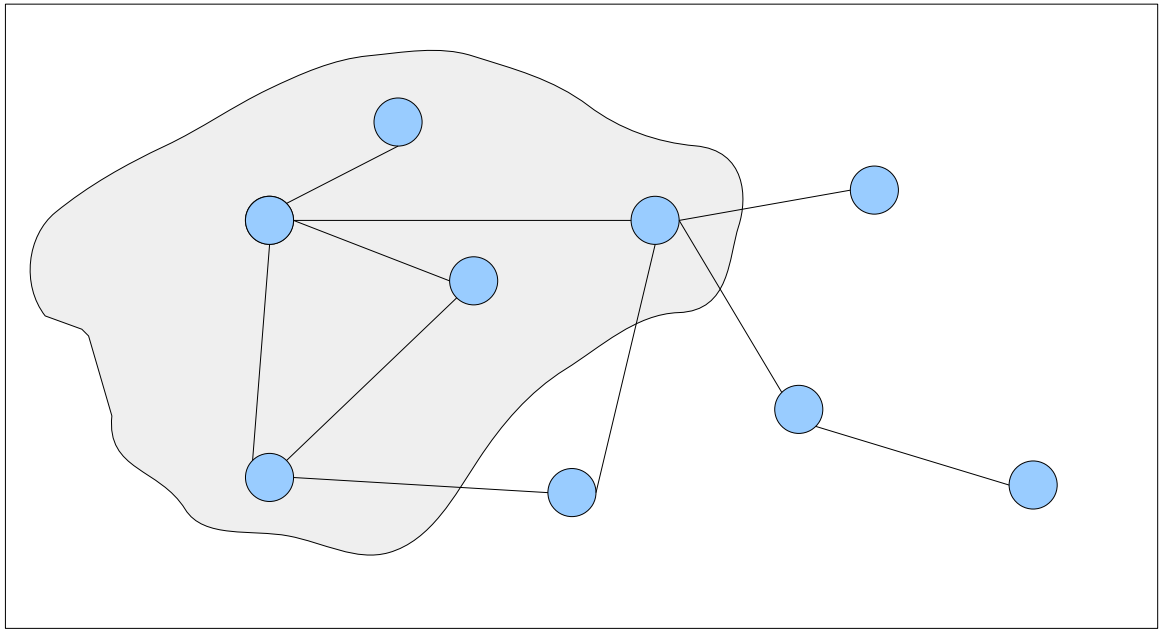
La idea de la segunda parte de la materia es que cuando salimos del mundo teorico y entramos al mundo real, no alcanza solo con criptografia.
La gran mayoria de los criptografos son matematicos, pero en realidad en la practica tenemos mas requerimientos.

Criptografía y Seguridad

Políticas de seguridad

Política de seguridad

- Es un enunciado que parte los estados de un sistema en autorizados (o seguros), y no autorizados.

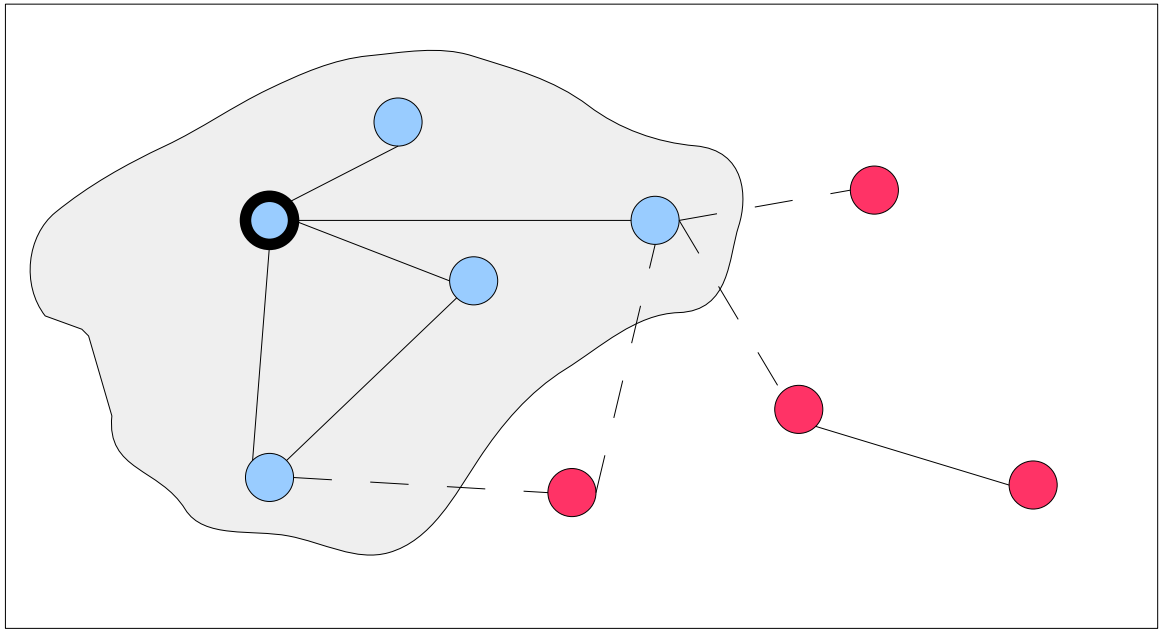


- Si el sistema entra en un estado no autorizado ocurrio una violación de segurdad

Sistema seguro

- Es un sistema que comienza en un estado autorizado y no puede entrar en un estado no autorizado

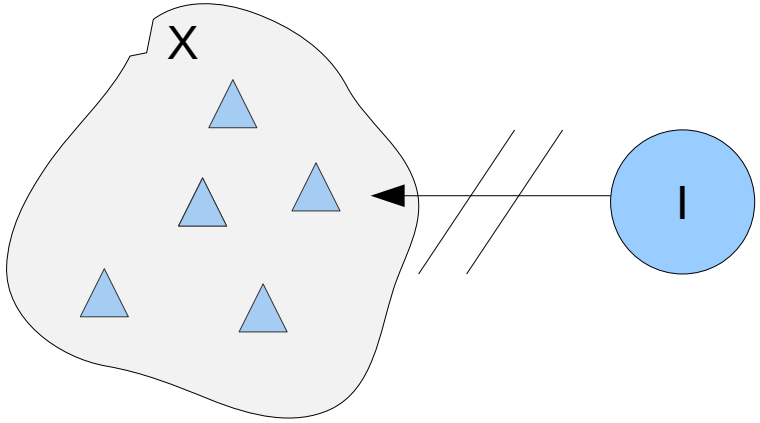
La definicion de sistema seguro es teorica. En la practica, hay tantos estados que es imposible clasificarlos todos.



Confidencialidad

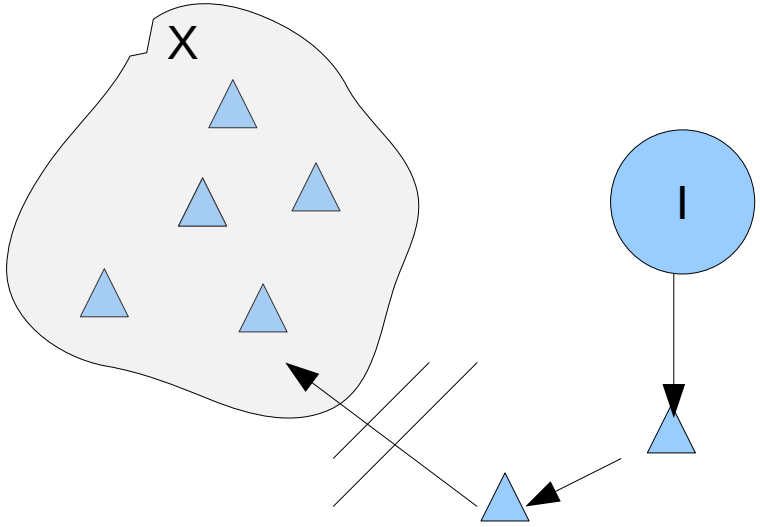
- Sean
  - X = conjunto de entidades
  - I = Información
- Propiedad de confidencialidad
  - I es confidencial para X si ningún miembro de X puede obtener información de I

La confidencialidad entonces es una relacion entre un conjunto de entidades e informacion.
Esa informacion es confidencial si ninguna entidad del conjunto puede extraer NADA de informacion (ni siquiera parcial) de I.



Confidencialidad (2)

- Sean
  - X = conjunto de entidades
  - I = Información
- Propiedad de confidencialidad
  - I es confidencial para X si ningún miembro de X puede obtener información de I



X No puede acceder a I ni directa ni indirectamente.
Ejemplo: como evitamos que un usuario que tiene acceso ilegitimo a la informacion se la haga llegar a otro.

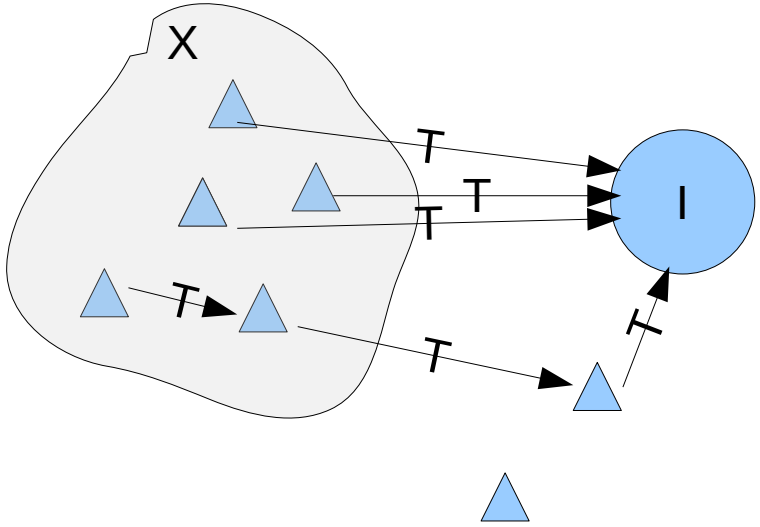
¡Ni siquiera por vías Indirectas!

Integridad

- Sean
  - X = conjunto de entidades
  - I = Información o recurso
- Propiedad de integridad
  - I es íntegro para X si todo miembro de X confía en I

La integridad tambien es una relacion entre un conjunto de entidades X y un pedazo de informacion I.
El concepto de integridad se basa en la confianza.

La integridad es importante, porque NO se puede construir un sistema seguro sin confiar en nada.
La idea es minimizar al maximo la base de confianza del sistema. Este es el eslabon mas debil, y la confianza en el sistema entero parte de ahi y de relaciones de integridad que van extendiendo el sistema.



Tipos de integridad

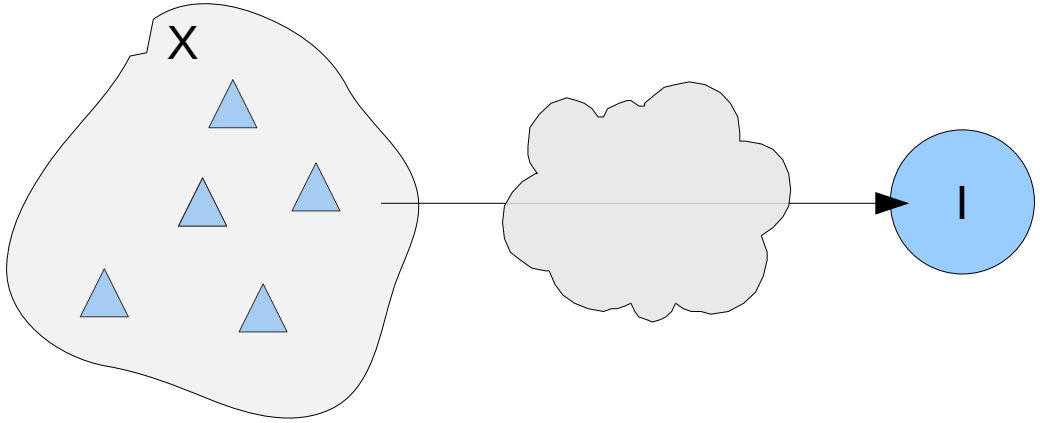
- Integridad de datos: Confianza en transporte y almacenamiento
- Integridad de origen: Confianza en el origen del dato o la identidad que representa
- Garantía: confianza en que el recurso o programa funciona como debería

Para programas ejecutables por ejemplo

Disponibilidad

- Sean
  - X = conjunto de entidades
  - I = Recurso
- Propiedad de disponibilidad
  - I está disponible para X si todo miembro de X puede acceder a I cuando lo requiera

Esto no se puede resolver solo con criptografia.
Involucra un concepto temporal, ya que todo miembro de X debe poder hacer uso de I en el momento exacto que lo necesite.
En el campo de seguridad, no nos interesa protegernos ante cortes de luz y esas cosas, sino que buscamos prevenirnos de atacantes que puedan tirar abajo el sistema por sobrecarga o similares.



Paradigmas de Control de Acceso

- Las políticas se centran en controlar el acceso a objetos:

En sistemas de uso comun, como sistemas operativos o servicios genericos, lo mas usado son los paradigmas de acceso discrecional.

Aunque los 3 problemas anteriores tengan naturalezas distintas, todos se basan en el acceso de entidades hacia objetos.
Por esto, se concluyo que todas se deben poder resolver con la misma solucion, que es el control de accesos.

- Acceso Discrecional (DAC)
  - Reglas arbitrarias (adhoc)

Arbitrarias en el sentido de que no son universales. Si tengo N sistemas, cada sistema tiene sus propias reglas de acceso.

  - Mecanismos puntuales
  - Opcional:Acceso controlado por creadores
    - Quien crea la información controla el acceso a la misma
- Acceso Mandatorio (MAC)
  - Reglas prefijadas
  - Mecanismos del sistema
  - No pueden ser alterados

Los paradigmas de acceso mandatorio son lo contrario a los de acceso discrecional.
Estos se usan en sistemas hechos a medida (para una empresa por ejemplo).
La idea es que existen una serie de reglas prefijadas que no pueden ser alteradas.
El concepto basico es que se define una serie de roles, y cada rol tiene ya definido que puede hacer y que no. NO se puede cambiar las aptitudes de cada rol.
Cualquier modificacion a un MAC termina siendo una modificacion a todo el sistema. Por ende, los paradigmas MAC son mas seguros pero mas dificiles de cambiar. Se usan para sistemas a medidas para empresas por ejemplo.
En cambio, los DAC permiten su uso para herramientas mucho mas versatiles (como seria Google Docs o sistemas operativos)

Modelos

- Describen familias de políticas
- Proveen un marco teórico común
  - Permiten reutilizar demostraciones
  - Simplifican el desarrollo de políticas

Esto nos permite implementar un modelo sin tener que estar redemostrando las garantias ante cada cambio

Modelo Bell-LaPadula

- Es un modelo de política militar
  - Se centra en garantizar confidencialidad
- Conceptualización del sistema:
  - Dividir el sistema en Sujetos y Objetos
  - Transición: (Sujeto, Objeto, Acción)
  - Las acciones se clasifican en Lectura y Escritura
- El modelo (versión simplificada):
  - Una lista ordenada de Niveles
  - Cada objeto y sujeto tienen un nivel asignado
  - ¿Cómo se restringe el acceso?

El modelo Bell-LaPadula prioriza totalmente la confidencialidad por sobre la integridad.
Se le dice politica militar a todo lo que priorice la confidencialidad sobre el resto de las garantias.

Los sujetos pueden ser usuarios y los objetos pueden ser archivos por ejemplo

Las transiciones se ven como: usuario Pablo leyendo archivo 5.

Necesitamos una simplificacion, porque si consideramos todas las transiciones como el conjunto de todas las ternas de ese tipo, nos queda algo larguisimo.

Vamos a tener el nivel mas confidencial, uno menos confidencial, etc. Luego asignamos cada sujeto y objeto a un nivel de seguridad.
A partir de esto nuestro sistema dicta las reglas.

Modelo Bell-LaPadula - Lectura

(se la llama condicion simple de seguridad)

- La información fluye hacia arriba, no hacia abajo
  - Se permite leer información de menor nivel
  - Se prohibe leer información de mayor nivel
- Condición de seguridad simple (reducida)
  - S puede leer O si y solo si L(O) ≤ L(S) y S tiene permiso para leer O

(el nivel de confidencialidad del objeto es <= a la del sujeto

- Se combina acceso mandatorio y discrecional

El modelo Bell-LaPadula es hibrido, porque combina MAC y DAC

Recordar que las reglas mandatorias son universales, y las reglas discrecionales son particulares. Por este motivo, las reglas mandatorias tienen mas peso que las discrecionales.

Importante: Los accesos discrecionales solo pueden restringir a los mandatorios, no contradecirlos

Entonces la discrecional solo sirve para restringir aun mas a la mandatoria

Modelo Bell-LaPadula - Escritura

- La información fluye hacia arriba, no hacia abajo
  - Se permite escribir información de mayor nivel
  - Se prohibe escribir información de menor nivel
- Condición de cierre (* property - reducida)

condicion de cierre o propiedad estrella

  - S puede escribir O si y solo si L(S) ≤ L(O) y S tiene permiso para escribir O

Esta regla de escritura es contraintuitiva. Yo no tengo acceso a informacion que no puedo leer pero si puedo escribirla?
Sin embargo, no es tan raro. Por ejemplo, nosotros completamos un formulario en la pagina web de una empresa. Otro ejemplo es que cuando alguien esta en el campo de batalla envia informacion a algun cuartel, y cuando la envia ya no la ve mas.
En conclusion, no es tan raro el caso en el cual alguien escribe cosas en un lugar al que no puede volver a acceder.
El concepto basico es que cuando una persona con alto nivel de confidencialidad escribe informacion a un nivel menor de confidencialidad, puede filtrar informacion.

Modelo Bell-LaPadula - DAC

- El acceso discrecional se define con una matriz de acceso:

La matriz tiene a todos los sujetos y todos los objetos. La idea es que aca se restringen los accesos dados por el modelo mandatorio.
Entonces, no podemos permitir cosas que el modelo mandatorio habia prohibido, pero si podemos asignar restricciones adicionales al mismo.
Esto se conoce como matriz de acceso.

	O1	...	On
S1	R, W		R
...			
Sn	W		

Importante: La matriz de acceso RESTRINGE los accesos dados por el modelo mandatorio

Modelo Bell-LaPadula

- El modelo original define 4 niveles
  - Top Secret (TS)
  - Secret (S)
  - Confidential (C)
  - Public (P)
- Pero funciona para cualquier cantidad de niveles

Modelo Bell-LaPadula - Ejemplo

- Ejemplo

  Generalmente, en este tipo de modelos no se trabaja sobre objetos y sujetos puntuales, sino que nos referimos al estereotipo de los mismos (tipos de sujeto y tipos de objetos).

  - Sujetos: Diseñador, Gerente, Director
  - Objetos: Producto X, Balances

- Etiquetado:

  Entonces, en el modelo Bell LaPadula solo necesitamos definir un nivel de seguridad para cada tipo de sujeto y para cada tipo de objeto.

  - L(Diseñador) = Confidential
  - L(Gerente) = Secret
  - L(Director) = Top Secret
  - L(Producto X) = Confidential
  - L(Balances) = Secret

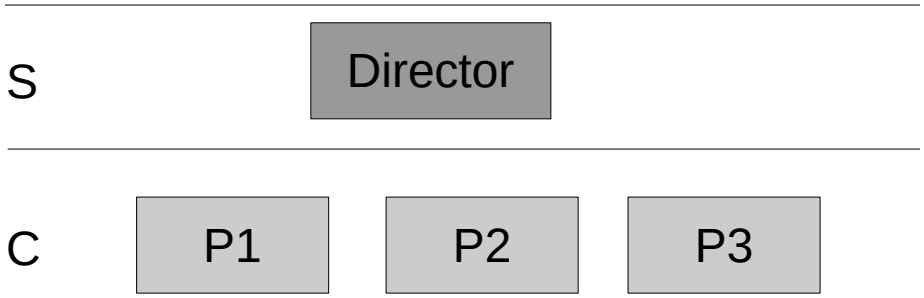
¿Qué acciones están permitidas?

Por ejemplo, el director puede leer producto X y balances, pero no puede escribir nada.
La idea es que si el director pudiera escribir a producto X, entonces tal vez estaria modificando indirectamente a los balances por ejemplo.

Modelo Bell-LaPadula

- ¿Que ocurre si no existe un orden completo?
- Ejemplo:
  - Varios proyectos aislados
  - No se ven entre sí
  - Un director ve todos los proyectos

Obviamente en cualquier escenario no trivial, el modelo simplificado se queda corto.
Por ejemplo, cuando tengo varios proyectos distintos que deben estar en el mismo nivel de seguridad pero que no se deberian poder ver entre si.
A priori, esto se podria resolver utilizando reglas discrecionales. Pero esto es muy poco escalable, porque tendria que ir restringiendo el acceso a todos los usuarios. Y si se van creando nuevos proyectos es muy dificil de mantener.



No se puede GARANTIZAR la aislacion entre Px y Py!

Modelo Bell-LaPadula

- Modelo completo
  - Además de niveles se definen categorias
  - Describen el tipo de información
  - Las categorías no están ordenadas
  - A cada objeto y sujeto se le asigna un compartimento
  - Compartimento = (Nivel, {categorias})

Los niveles estaban ordenados, pero las categorias no. Esto significa que no hay categorias mas importantes que otras.

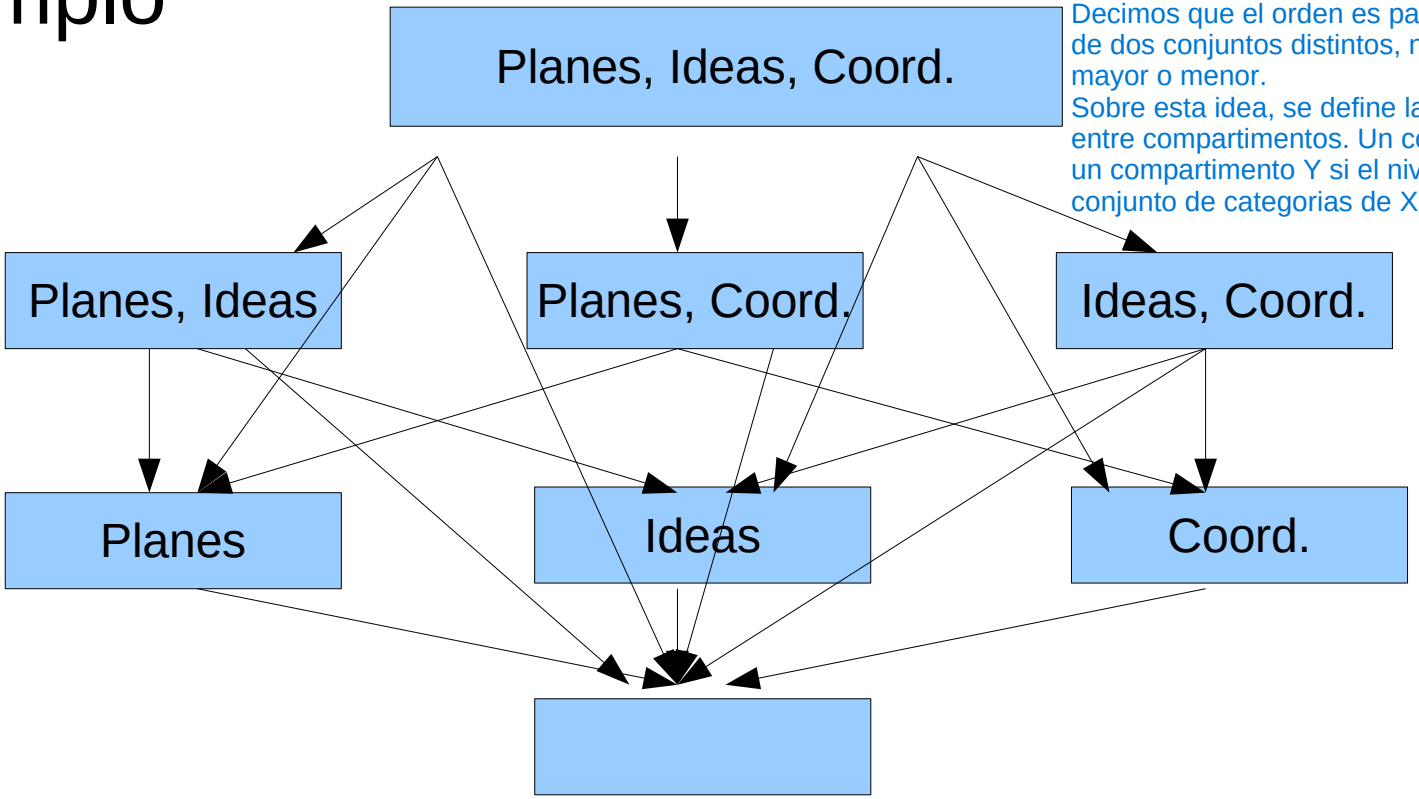
Modelo Bell-LaPadula

- Dominancia:

$$Sea\ , l \in L\ , C \subseteq CAT$$
$$(l\ , C)\ dom\ (l'\ , C') \Leftrightarrow l' \leqslant l \wedge C' \subseteq C$$

Las categorias no tienen orden, pero si le podemos dar un orden parcial.
Esto se logra usando la propiedad de conjuntos de la idea de que si un conjunto contiene a otro, entonces es mayor.
Decimos que el orden es parcial, porque si partimos de dos conjuntos distintos, no podemos decir cual es mayor o menor.
Sobre esta idea, se define la relacion de dominancia entre compartimentos. Un compartimento X domina a un compartimento Y si el nivel de X >= nivel de Y y el conjunto de categorias de X contiene al de Y.

- Ejemplo



Modelo Bell-LaPadula

- Compartimentos
  - Conjunto parcialmente ordenado
  - Dado A y B puede ocurrir que:
    - A dom B
    - B dom A
    - Ni A dom B ni B dom A
  - Ejemplos:
    - (TS, {X}) dom (TS, {})
    - (TS, {X}) dom (S, {X})
    - (S, {X}) no dom (P, {Y})
    - (P, {Y}) no dom (S, {X})

Modelo Bell-LaPadula - Lectura

- La información fluye hacia arriba, no hacia abajo
  - Se permite leer información de menor nivel
  - Se prohibe leer información de mayor nivel
- Condición de seguridad simple
  - S puede leer O si y solo si L(S) **dom** L(O) y S tiene permiso para leer O
- Se combina acceso mandatorio y discrecional

Modelo Bell-LaPadula - Escritura

- La información fluye hacia arriba, no hacia abajo
  - Se permite escribir información de mayor nivel
  - Se prohíbe escribir información de menor nivel
- Condición de cierre
  - S puede escribir O si y solo si L(O) **dom** L(S) y S tiene permiso para escribir O

Modelo Bell-LaPadula

- Teorema básico de la seguridad
  - Si un sistema comienza en un estado seguro y sus transiciones satisfacen la condición simple de seguridad y la condición de cierre, todos los estados del sistema son seguros
- Garantiza que no existe flujo de información en el sentido de la condición simple
- ¿Para que existe la condición de cierre?
  - Para garantizar que no existan caminos indirectos que violen la condición simple

El problema de la comunicación

- A le envía un mensaje a B (B dom A)
  - Como B dom A, la regla de cierre permite enviar el mensaje (escritura)
- B le contesta
  - Como B dom A, la regla de cierre prohíbe enviar el mensaje
  - Problemas!!!

La idea es que B solicita una reduccion de nivel de acceso para poder contestarle a A. Generalmente ocurre una señal visual, y B no tiene acceso a la informacion de su nivel de seguridad anterior mientras este en este modo de reduccion de nivel.
En otros casos, se utiliza un tercero que intercepta el mensaje de B hacia A y puede borrar algunas partes para proteger confidencialidad.

- El modelo Bell-LaPadula establece la posibilidad de disminuir el nivel de acceso temporalmente
  - MaxLevel y CurLevel / MaxLeve dom CurLevel
  - La disminución de acceso debe ser solicitada explicitamente

Principio de tranquilidad

- Usuarios y objetos no cambian sus niveles luego de ser creados

Hasta ahora el modelo viene ignorando que pasa cuando una persona tiene que cambiar de nivel. Por ejemplo, un empleado asciende a CEO y ahora tiene nivel maximo.

- ¿Que pasaría se los niveles cambiasen?
- Subir el nivel de un objeto

Por ejemplo, me confundi y clasifique un dato como de menor nivel al que era. El problema es que cuando me di cuenta y lo subo de nivel, tal vez alguien de menor nivel ya lo habia leido.

  - La información fue leída por usuarios de menor nivel
  - Viola principio de seguridad simple
- Bajar el nivel de un objeto

Tambien existe el caso de bajar el nivel de un objeto. Por ejemplo, cuando tengo informacion que ya hice publica, le puedo bajar el nivel de seguridad al minimo.

  - Problema de desclasificación
  - Viola principio de cierre

Hay que tener en cuenta que Bell LaPadula se creo teniendo en mente que los objetos siempre tendran el mismo nivel de seguridad

Políticas de integridad

- Se concentran en preservar la integridad
- Mayor uso en ambientes comerciales
- Requerimientos muy diferentes a las políticas de confidencialidad
  - Prevenir modificación de datos por entidades no autorizadas
  - Prevenir modificaciones no autorizadas de datos por entidades autorizadas
  - Asegurar que los datos representan la información que se supone deben representar

Un lugar en donde las politicas de integridad tienen mas importancia que las politicas de seguridad es una agencia de noticias. Su sistema interno no se enfoca en esconder la informacion, sino que buscan que la misma sea confiable. Esta relacion de confianza puede ser en la informacion o en la ejecucion del programa (por ejemplo que el programa no tenga troyanos). La definicion de troyano es que un programa haga algo que no se aviso o no se pretendia que hiciera.

Modelos de integridad de Biba

- Base para los tres modelos:
  - Conjunto de sujetos S, objetos O, Niveles de Integridad I

    A todo sujeto y a todo objeto se le asigna un nivel de integridad. Existen 3 operaciones (lectura, escritura, ejecucion)
  - Relación $< : I \times I$, dominancia del 1ro sobre el 2do
  - *i*: $S \cup O \rightarrow I$, *nivel de integridad de una entidad*
  - *r*: $S \times O$, *pares* $s \in S, o \in O$ *donde se puede leer o*
  - *w*: $S \times O$, *ídem para escritura*
  - *x*: $S \times O$, *ídem para ejecución*

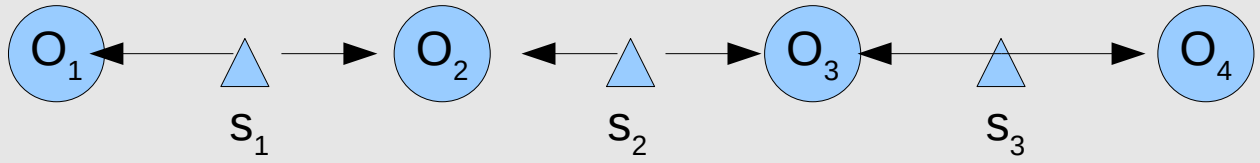
Es parecido a Bell LaPadula. Divide al sistema en sujetos y objetos, y en niveles de integridad (LaPadula era en niveles de confidencialidad).
Se le asigna a todo sujeto y objeto un nivel de integridad.
Se definen 3 relaciones entre sujetos y objetos.

Niveles de integridad

- A mayor nivel, mayor confianza de que
  - Un programa se ejecutará correctamente o detectará errores en sus entradas
  - Un dato es preciso y/o fiable
- Clasificación normal
  - No confiable — Untrusted
  - Ligeramente confiable — Slightly trusted
  - Confiable — Trusted
  - Altamente confiable — Highly trusted
  - Intachable — Unimpeachable

Camino de transferencia de información

- Camino que puede seguir la información para llegar desde un objeto a otro
- Secuencia de objetos o_1, ..., o_{n+1} y sujetos s_1, ..., s_n tal que s_i r o_i y s_i w o_{i+1} para todo i, $1 \leq i \leq n$.



¿Como fluye información de O_1 a O_4?

S_1 lee O_1 y escribe O_2

S_2 lee O_2 y escribe O_3

S_3 lee O_3 y escribe O_4

Low-Water-Mark Policy (1er modelo)

- Idea: Si un sujeto usa información poco confiable, se vuelve poco confiable

Un sujeto es tan confiable como la informacion que usa

- Reglas:

1. Un sujeto solo puede escribir un objeto de integridad menor o igual a la de el.
2. Cuando un sujeto lee un objeto, el nivel de integridad del sujeto baja al minimo entre el sujeto y el objeto
3. Un sujeto solo puede ejecutar un objeto de integridad mayor o igual a la de el

1. $s \in S$ puede escribir $o \in O$ si y solo si $i(o) \leq i(s)$.
2. Si $s \in S$ lee $o \in O$, entonces $i'(s) = min(i(s), i(o))$ es el nuevo nivel de integridad de s
3. $s_1 \in S$ puede ejecutar$s_2 \in S$ si y solo si $i(s_2) \leq i(s_1)$

La diferencia con Bell LaPadula es que NO restringimos la lectura, sino que le asignamos un "castigo". Un sujeto puede leer todos los objetos que quiera, pero si lee un objeto poco confiable entonces su nivel de confianza maximo de escritura baja al del objeto.

- Previene contra:
  - Modificaciones directas que bajarian el nivel de integridad
  - Modifcaciones indirectas con informacion de menor nivel de integridad

Restricción en el flujo de información

- Si hay un camino de transferencia entre o_1 y o_n la aplicación de la política requiere $i(o_j) \leq i(o_1)$ para todo $1 < j \leq n$
- Demostración (idea):
  - Asumir que existe un camino de transferencia y las operaciones se realizan ordenadamente (s_1 lee o_1, s_1 escribe o_2, s_2 lee o_1 ...)
  - Por indicucción $i(s_1) = \min(i(s_1), i(o_1))$. Luego de k lecturas $i(s_k) = \min(i(o_1), i(o_2), ..., i(o_k))$
  - La última escritura requiere $i(o_n) \leq i(s_n) \leq i(o_1)$

Low-Water-Mark - Problemas

- Los niveles de integridad de los sujetos decaen con el uso del sistema
  - Eventualmente nadie puede acceder o generar objetos de niveles altos de integridad
- Alternativas: modificar los niveles de integridad de los objetos en lugar de los sujetos
  - Problema similar: los objetos se degradan hasta llegar a los niveles mas bajos de integridad

En un sistema dinamico, en donde por ejemplo una consultora hace un analisis de una empresa (y cuando termina el analisis se va a trabajar para otro), este sistema funciona.
Sin embargo, si el sistema tiene que funcionar en periodos largos de tiempo, entonces va bajando el nivel de todos hasta que no se puede producir nada con un nivel alto de integridad.

Ring Policy (2do modelo)

- Considera solamente el problema de la modificación directa de objetos
- Reglas:
  1. $s \in S$ puede escribir $o \in O$ si y solo si $i(o) \leq i(s)$.
  2. **Cualquier sujeto puede leer cualquier objeto**
  3. $s_1 \in S$ puede ejecutar $s_2 \in S$ si y solo si $i(s_2) \leq i(s_1)$
- Los niveles de integridad son estaticos
- Previene contra la modificación directa
- Permite utilizar información de menor nivel de confianza para generar información de mayor nivel

Ring policy es parecido a Low Watermark pero saca el castigo por la lectura de un objeto de menor nivel. La idea es que si un sujeto confiable lee un objeto poco confiable, confiamos en la capacidad del sujeto para curar esa informacion, y mantener ese nivel confiable que tenia. La unica regla que cambia con Low Watermark es la 2.

Esto funciona cuando el grupo de sujetos es chico y se puede controlar y verificar su integridad. El sistema de low watermark nos garantiza que no va a haber modificaciones indirectas de la integridad.

Strict Integrity (3er modelo)

- Similar al modelo Bell-LaPadula
- Reglas:
  1. $s \in S$ puede leer $o \in O$ sii $i(s) \leq i(o)$
  2. $s \in S$ puede escribir $o \in O$ sii $i(o) \leq i(s)$
  3. $s_1 \in S$ puede ejecutar $s_2 \in S$ sii $i(s_2) \leq i(s_1)$
- Se pueden agregar categorias y controles discrecionales para obtener el dual de Bell-LaPadula
- Mantiene la misma restricción en el flujo de información

Un sujeto puede escribir un objeto si y solo si domina a ese objeto. Estas reglas tienen un parecido tremendo con Bell LaPadula.
Este sistema funciona bien si no se cambian los niveles.

Modelo de la pared China

- Modelo híbrido
  - Toma en cuenta confidencialidad e integridad
- Se concentra en el problema de conflictos de interés
  - De amplio uso en ámbitos bursátiles y judiciales
  - Algunos países exigen medidas que prevengan problemas de conflicto de interés
- Ejemplos:
  - Impedir que un trader represente a dos clientes que compiten en el mercado
  - Impedir que un abogado represente a dos empresas competidoras

Hay muchas implemenaciones de pared china que no son solo informaticas. Los empleados que representan a empresas competidoras tienen que ir a oficinas distintas en pisos distintos, etc.

Esto permite que una firma de abogados represente a dos empresas competidoras en dos areas distintas entre si. De esta manera, se le obliga a la firma a utilizar el modelo de la Pared China para evitar que el bufete le entregue informacion de una empresa a la otra empresa (ya sea a proposito o sin querer). Aca sirve esta modelo porque luego de que se termine el caso, la informacion se hace publica.

Concepto

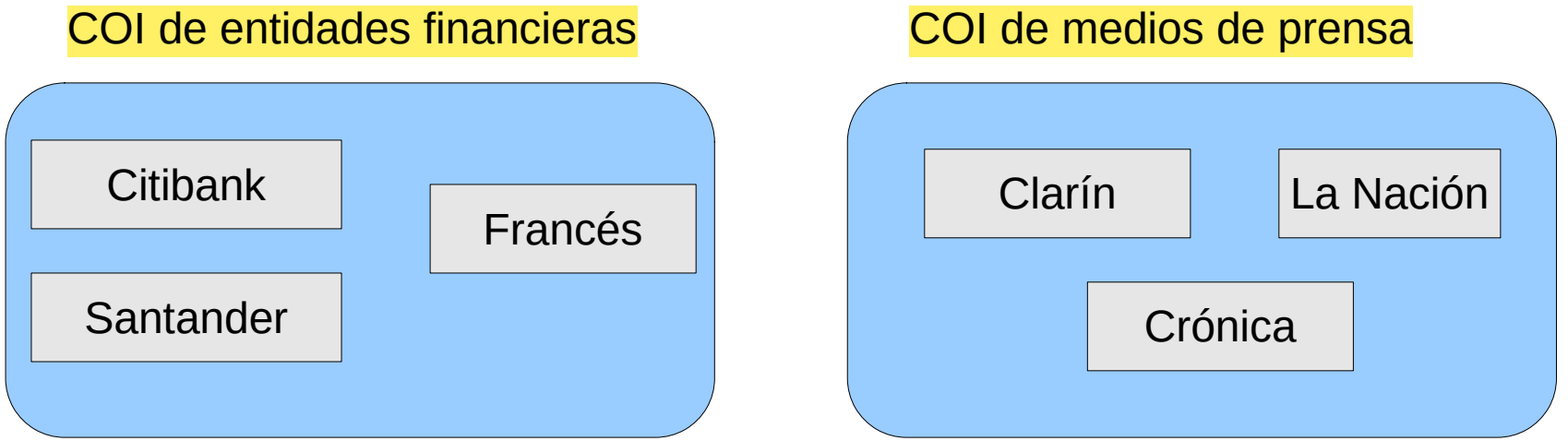
- Agrupar las entidades en clases de conflicto de interés
- Controlar el acceso de sujetos a a cada clase
- Controlar la escritura a todas las clases para impedir que se mueva información en contra de la política
- Permitir que datos desclasificados sean vistos por todos

La pared china involucra un concepto temporal. Generalmente, en la mayoria de los juicios cuando terminan la informacion se vuelve publica, por lo que se la puede desclasificar.

Definiciones

- Objetos. Items de información
- Company dataset (CD): conjunto de objetos relacionados con la misma empresa o grupo
- Conflict of interest Class (COI): contiene CDs de empresas con conflicto de interes
  - Se asume que cada objeto pertenece a exactamente un CD y a un COI

Ejemplo


COI de entidades financieras
Citibank
Francés
Santander
COI de medios de prensa
Clarín
La Nación
Crónica

Idea

- Un usuario nuevo puede leer cualquier objeto
  - Una vez que leyo un objeto no puede acceder a objetos de otras empresas que entren en conflicto. Es decir, leer un objeto tiene consecuencias (me puede limitar de leer otros objetos)

- Por ejemplo, un asesor puede leer objetos pertenecientes al CD La Nación
  - Pero una vez hecho esto no puede leer objetos del CD Clarin
  - Aunque nada impide que lea objetos del CD Santander

Elemento temporal

- Si S lee cualquier CD de un COI, no puede volver a leer otra CD del mismo COI, *nunca.*
  - El acceso depende de la historia de S
  - Se impide que use información que obtuvo antes para tomar decisiones que afecten a intereses en competencia
- El elemento temporal es un nuevo requerimiento que no es capturado en el modelo Bell-LaPadula

Esto puede ser un problema, entonces agregamos mas adelante una condicion de declasificacion

CW–Condición Simple de Seguridad

- s puede leer o si alguna de las condiciones se cumple:

(si yo ya accedi al CD, puedo seguir accediendo)

  - Existe un o' leído previamente tal que CD(o) = CD(o')
    - Se accedió previamente a un dato de la empresa
  - Para todo o' leído previamente, COI(o) ≠ *COI(o')*
    - No se accedió a algún dato en una categoría de conflicto de interés (hasta que deja de ser confidencial)
  - o es un objeto público (desclasificado)
    - Información que dejó de ser confidencial (por ejemplo: balance anual del período anterior)
    - Información sanitizada (donde se eliminan partes confidenciales)

Escritura

Control mas complicado para prevenir flujos indirectos

Considerar:

- s_1 accede a objetos de Clarín y Santander
- s_2 accede a objetos de La Nación y Santander
- Ninguno esta en una situación de conflicto de interés.

Pero si s_1 *escribiera* un objeto en DS(Santander), podría estar volcando información de Clarín, y de esa forma hacer que s_2 la tenga disponible.

CW- Propiedad de cierre

- S puede escribir o si se cumplen las siguientes condiciones: se tienen que cumplir simultaneamente
  - S puede leer o según la condición simple de seguridad
  - Para todo objeto no público o', si s puede leer o' entonces CD(o') = CD(o)
- Significa que para escribir un objeto es necesario que todo objeto accesible para la entidad pertenezca al mismo grupo
  - Por ejemplo: el dato es escrito por un miembro de la empresa

La segunda regla es muy restrictiva, y viene a solucionar el problema de la slide anterior.
Implica que para escribir informacion, se debe estar trabajando unicamente sobre el dataset de esa empresa.
Es por esto que firmas de consultoria como McKinsey, que dan consultoria a miles de empresas en simultaneo, tengan empleados que trabajan unicamente en una empresa.
Luego hay jefes que supervisan a todos esos empleados, pero que NO pueden escribir informacion, solo leerla.

Composición de políticas

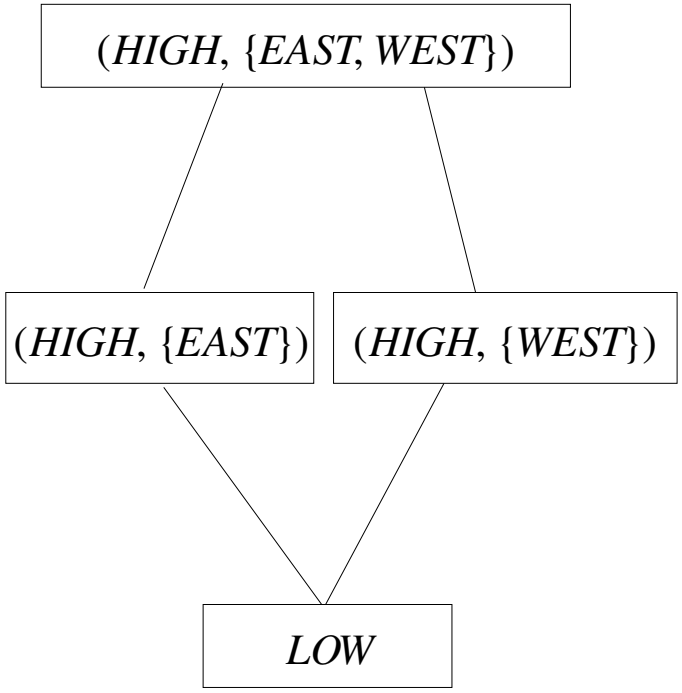
- El problema:
  - Conectar dos sistemas seguros
- Las preguntas
  - ¿La composición de los dos sistemas será segura?
  - ¿Se puede crear una política única consistente con ambas?

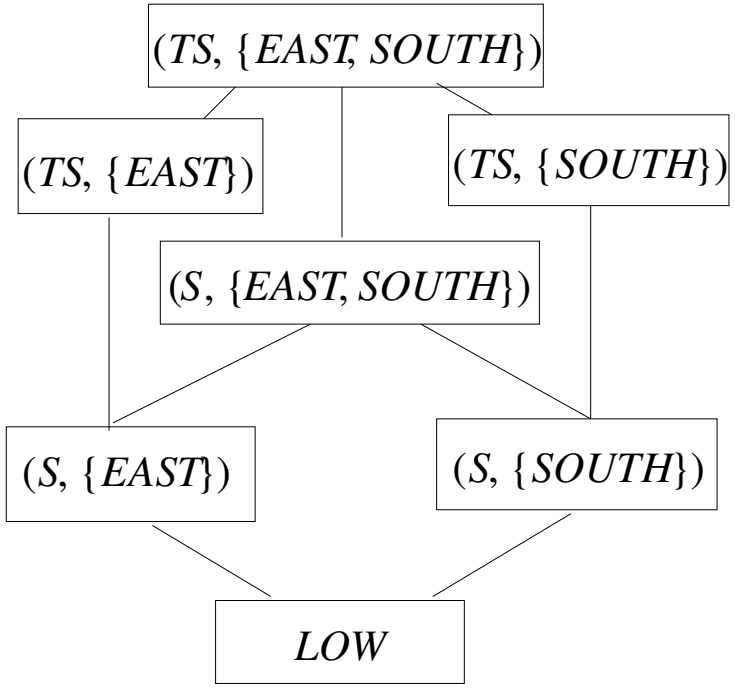
En general, ninguna empresa regula todos sus sistemas con el mismo modelo de seguridad. Surge el problema de que pasa cuando tengo dos sistemas seguros y quiero combinarlos. Sin embargo, este es un problema abierto (no tiene solucion)

Composición de políticas

- Supongamos dos sistemas que siguen un modelo del tipo Bell-LaPadula
- ¿Cual es el modelo compuesto del sistema conjunto?
- Problemas
  - Los grupos no tienen orden total
  - Es necesario establecer una correspondencia entre niveles de seguridad

Ejemplo





Si nosotros quisieramos combinar estos dos Bell LaPadula, surgen estos problemas para combinarlos:
- LOW vale lo mismo en ambos sistemas?
- HIGH es equivalente a TOP SECRET?
- EAST es lo mismo en ambos sistemas?

Encima en este problema queremos combinar dos Bell LaPadula. Si quisieramos combinar un Bell LaPadula con una pared china el problema es aun mayor.

Analisis del ejemplo

- Determinar orden de los niveles
  - Por ejemplo, S < HIGH < TS
- Determinar equivalencia de categorias
  - Por ejemplo: east representa lo mismo
- El modelo complementario tendría:
  - 4 niveles (LOW < S < HIGH < TS)
  - 3 categorías (SOUTH, EAST, WEST)
  - Notar que es una nueva política

Composición

- Si los modelos de políticas son iguales
  - Si se puede cambiar la política de los componentes (reemplazarla por el modelo compuesto) la composición es trivial
  - Si no se puede, hay que demostrar que la composición cubre los requerimientos de las políticas de los componentes. Muy dificil

Composición

- Si los modelos de políticas son diferentes
  - ¿Que significa seguro en este contexto?
  - ¿Que política domina la composición?
- No hay una única solución
- Posibles principios guía:

Son dos principios distintos

  - Cualquier acceso permitido por la política de seguridad de un componente debe estar permitido por la política emergente (autonomía)

Privilegia funcionalidad por sobre seguridad. Si en uno de los sistemas podemos hacer algo y en el otro no, en el resultante se puede hacer.

  - Cualquier acceso prohibido por la política de seguridad de un componente debe ser prohibido por la política emergente (seguridad)

Privilegia seguridad por sobre funcionalidad.

Consecuencias

- La política compuesta satisface la seguridad de las políticas de los componentes
- ¿Si algún caso no esta explícitamente permitido o prohibido por alguna política?
  - Permitirlo (modelo original de Gong & Quiam)
  - Prohibirlo (principio de denegación por defecto)

Podemos pensar el sistema resultante como un diagrama de Venn. Dadas dos politicas que queremos combinar, entonces cuando caemos en la interseccion podemos usar el principio de autonomia o el de seguridad.
Sin embargo, cuando caemos en donde no hay interseccion, tenemos que definir si permitirlo (dando prioridad a funcionalidad) o prohibirlo (dando prioridad a seguridad).
En conclusion, combinar dos sistemas es tan dificil que generalmente pueden surgir problemas de seguridad.

Ejemplo

- Systema X: Bob no puede leer archivos de Alice
- Sistema Y: Eve y Lilith pueden leer los archivos del otro
- Composición:
  - Por el sistema Y: bob podría leer archivos de Alice.
  - Pero el sistema X prohíbe explícitamente esto
  - Metodología: la metodologia es por fuerza bruta. Habria que quitar las relaciones no permitidas y listo. Esto es un problema NP porque crece de manera exponencial.`
    - Crear el conjunto de accesos posibles (expanción de relaciones transitivas)
    - Quitar las relaciones no permitidas
    - Determinar el número de relaciones que deben ser quitadas es un problema NP

Lectura recomendada

Capítulo 4
Capítulo 5: 5.1-5.4
Capítulo 6: 6.1-6.2
Capítulo 7: 7.1
Capítulo 8: 8.1

Computer Security Art and Science
Matt Bishop