Resumen Teórica 3 : Políticas de seguridad

Tomás F. Melli

August 2025

Índice

1 Introducción **2**
1.1 Definiciones formales de conceptos ya vistos . . . 2
1.1.1 Confidencialidad . . . 2
1.1.2 Integridad . . . 2
1.1.3 Disponibilidad . . . 3
1.1.4 Mecanismo . . . 3

2 Tipos de políticas **3**
2.1 Política de confidencialidad . . . 3
2.1.1 Modelo Bell-Lapadula . . . 3
2.2 Covert Channel . . . 5
2.3 Política de integridad . . . 5
2.3.1 Principios de operación . . . 5
2.3.2 Modelo BIBA . . . 5
2.3.3 Clark-Wilson . . . 6
2.3.4 El rol de la confianza . . . 6
2.4 Política de híbrida . . . 6
2.4.1 Modelo Pared China . . . 6
2.5 ORCON (Originator Controlled Access Control) . . . 8
2.6 DRM (Digital Rights Management) . . . 8
2.7 Políticas de seguridad en sistemas de información clínica . . . 8
2.8 SELinux (Security Enhanced Linux) . . . 9
2.9 Windows Mandatory Integrity Control (MIC) . . . 9

1 Introducción

Si consideramos a un sistema como un autómata finito (modelo matemático que se utiliza para representar y controlar sistemas que pueden estar en un número limitado de estados y que responden a entradas de manera predecible) con un conjunto de funciones que permiten cambiar de estado, entonces una política de seguridad es **una declaración que particiona un sistema en dos conjuntos de estados**:

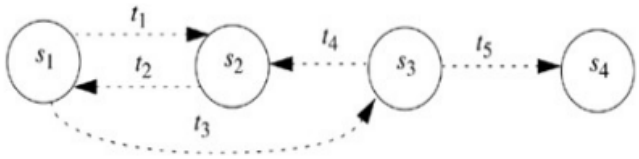
1. **Autorizados** (seguros) : son los estados en los que el sistema puede entrar.
2. **No Autorizados** (no seguros): si el sistema entra en uno de estos estados habrá una violación de seguridad.

Decimos que un sistema es **seguro** si comienza en un estado autorizado, y nunca entra en un estado no autorizado. Un sistema seguro bajo una política puede no serlo bajo otra.
Decimos que tenemos un **problema de seguridad** cuando un sistema entra en un estado no autorizado entonces tenemos un problema de seguridad.

Ejemplo

Dada la política de seguridad,

$$A = \{s_1, s_2\}$$
$$NA = \{s_3, s_4\}$$



Este sistema no es seguro ya que desde un estado autorizado puede pasar a un estado no autorizado. Si el arco de s_1 a s_3 no estuviera presenta, sería seguro.

1.1 Definiciones formales de conceptos ya vistos

1.1.1 Confidencialidad

Repasemos la definición formal,

> Sea X un conjunto de entidades e I una cierta información. Entonces I tiene la propiedad de confidencialidad con respecto de X si ningún$x \in X$ puede obtener información sobre I.

Ejemplo

Tomamos X como un conjunto de alumnos, I como las respuestas de un examen. I es confidencial con respecto a X si los alumnos no pueden obtener las respuestas del examen.

1.1.2 Integridad

Repasamos la definición formal,

> Sea X un conjunto de entidades e I una cierta información. Entonces I tiene la propiedad de integridad con respecto de X si todos los $x \in X$ confían en I.

Ya hablamos en la teórica 1 sobre los **tipos**,

- **Integridad de los datos** : confiamos en que el medio de transporte o de almacenamiento no cambian la información.
- **Integridad de origen** : cuando la información es sobre el origen de algo o sobre identidad.

1.1.3 Disponibilidad

Repasamos la definición formal,

> Sea X un conjunto de entidades e I una cierta información. Entonces I tiene la propiedad de disponibilidad con respecto de X si todos los $x \in X$ pueden acceder a I.

Tenemos dos tipos :

1. **Tradicional** : La disponibilidad es binaria, o sea, la información está disponible y el usuario puede acceder o la información no está disponible y el usuario no puede acceder.
2. **Calidad de servicio** : se mide en términos de niveles de servicio. Se define mediante un SLA (Service Level Agreement), un acuerdo que establece un nivel mínimo de disponibilidad que debe cumplirse. Un servicio cloud tiene un SLA de 99.9% de disponibilidad entonces puede haber interrupciones pequeñas, pero debe garantizar que el servicio funcione la mayor parte del tiempo.

1.1.4 Mecanismo

Un mecanismo de seguridad es una entidad o un procedimiento que hace cumplir una parte de la política de la seguridad. Por ejemplo, Control de acceso (asignar permisos a los recursos) o no permitir que las personas inserten CDs o dispositivos USB de almacenamiento en una computadora para controlar lo que ingresa a los sistemas.

2 Tipos de políticas

2.1 Política de confidencialidad

Desarrollada principalmente para proteger la confidencialidad.

2.1.1 Modelo Bell-Lapadula

Fue desarrollado en 1973 por David Elliott Bell y Leonard J. LaPadula en el MITRE Corporation. Surgió para sistemas militares y gubernamentales que necesitaban manejar información clasificada y altamente sensible. El objetivo principal era garantizar la confidencialidad de la información, minimizando riesgos de filtración entre niveles de seguridad. Permitió formalizar políticas de seguridad multinivel en sistemas informáticos.
Este modelo combina acceso Mandatorio y acceso Discrecional (más adelante lo desarrollo). Tiene 4 niveles de clasificación de seguridad ordenados :

1. Alto Secreto : Nivel más alto
2. Secreto
3. Confidencial
4. No Clasificado : nivel más bajo

En este modelo, los **sujetos tienen habilitaciones** (clearance) de seguridad $L(S)$ y los **objetos tienen clasificaciones** de seguridad $L(o)$. Cuando nos referimos a ambos, hablamos de clasificaciones.

Ejemplo

Clasificación de Seguridad	***sujeto***	***objeto***
Alto Secreto	Tamara	Legajos de Personal
Secreto	Samuel	Archivos correo Electrónico
Confidencial	Claire	Logs de Actividad
No clasificado	Ulaley	Guía Telefónica

Es decir que, Tamara puede leer todos los archivos, Claire no puede leer los legajos de personal o los archivos de correo electrónico y Ulaley sólo puede leer la guía telefónica.

Principios

Ante un requerimiento, el sistema lo admite o rechaza considerando la habilitación del sujeto que lo solicita y la clasificación de seguridad del objeto solicitado. Para ello, se aplican **dos principios**

Condición Simple de Seguridad

Condición Simple de Seguridad es el nombre formal de la regla en el modelo BLP (Bell-Lapadula) que define quién puede leer un objeto. Se trata de una regla combinada. De forma coloquial se la llama **Read Down**
El sujeto **s** puede leer el objeto **o** si y sólo si $L(o) \leq L(s)$ y **s** tiene permiso para leer el objeto **o**.
El modelo combina control de acceso mandatorio (relaciones entre niveles de seguridad) y control discrecional (el permiso requerido).

Star Property (propiedad estrella *)

Es la regla de escritura en el modelo BLP. O sea, define cuándo un sujeto puede escribir un objeto. También es llamada **Write Up**.
El sujeto **s** puede escribir el objeto **o** si y sólo si $L(s) \leq L(o)$ y **s** tiene permiso de escritura para **o**. El modelo combina control de acceso mandatorio (relaciones entre niveles de seguridad) y control discrecional (el permiso requerido).

Extensión

Se expande el concepto agregando **categorías o compartimentos**. Las categorías representan distintas áreas de información dentro de un mismo nivel, y no responden a un esquema jerárquico. El **nivel o etiqueta de seguridad** se representa como el par **(A,C)**

```
(habilitación, conjunto de categorías)
```

Ejemplos :

- (Alto Secreto, { NATO, MERCOSUR, NOFORN })
- (Confidencial, { MERCOSUR, NOFORN })
- (Secreto, { NATO, NOFORN })

Niveles y Dominancia

La **dominancia** significa que un nivel de seguridad es "mayor o igual" que otro, considerando nivel jerárquico y categorías. Por tanto, se define formalmente como

$$(A, C)\ dom\ (A', C') \iff A' \leq A\ \wedge C' \subseteq C$$

Ejemplos :

- (Alto Secreto, {NATO, NOFORN}) dom (Secreto, {NATO})
- (Secreto, {NATO, Mercosur}) dom (Confidencial, {NATO, Mercosur})
- (Alto Secreto, {NATO}) ¬dom (Confidencial, {Mercosur})

Condición Simple de Seguridad - Versión Extendida

El sujeto **s** puede leer el objeto **o** si y sólo si $L(s)$ *dom* $L(o)$ y **s** tiene permiso para leer **o**. El modelo combina control de acceso mandatorio (relaciones entre niveles de seguridad) y control discrecional (el permiso requerido).

Star Property - Versión Extendida

El sujeto **s** puede escribir el objeto **o** si y sólo si $L(o)$ *dom* $L(s)$ y **s** tiene permiso para escribir **o**. El modelo combina control de acceso mandatorio (relaciones entre niveles de seguridad) y control discrecional (el permiso requerido).

2.2 Covert Channel

Un covert channel es un medio no autorizado por el sistema para filtrar información de un nivel de seguridad más alto a uno más bajo. Es decir, aunque la política de seguridad prohíba el acceso directo, un sujeto puede transmitir información por métodos indirectos. Las políticas de confidencialidad (como las de Bell-LaPadula) prohíben la filtración de información hacia niveles no autorizados: No read up, no leer información más sensible. No write down, no escribir información hacia niveles inferiores. Un covert channel viola la política de confidencialidad, porque permite que un sujeto con nivel alto "filtre" información a un nivel más bajo, sin pasar por los mecanismos de control del sistema. Representa una brecha de seguridad que no se controla con MAC ni DAC estándar.

2.3 Política de integridad

El objetivo de una política de integridad es **preservar los datos y su integridad** (asegurar que los datos y sistemas sean correctos, consistentes y confiables, evitando modificaciones no autorizadas). Para establecer una política de integridad hay que identificar las maneras autorizadas en las cuáles la información puede ser alterada y cuáles son las entidades autorizadas para alterarla. Para lograrlo, se aplican **principios de operación** que guían cómo se diseñan y manejan los sistemas:

2.3.1 Principios de operación

Segregation of Duties (separación de tareas)

Ninguna persona o proceso debería tener control total sobre todas las fases de un proceso crítico. El objetivo es evitar errores o fraudes, ya que se requiere cooperación entre diferentes roles. Ejemplo: pasar un sistema del entorno de desarrollo al entorno de producción.

Separation of Functions (separación de funciones)

Cada función del sistema debe realizarse en entornos distintos y controlados para reducir riesgos. El objetivo es minimizar el impacto de errores o accesos indebidos. Ejemplo: los sistemas se programan y prueban en el entorno de desarrollo no en el de producción.

Accountability / Auditing (Auditabilidad)

Todos los cambios y operaciones críticas deben ser registrados y verificables. El objetivo es permitir revisión independiente y detectar posibles violaciones de integridad. Ejemplo: el proceso de pasar un sistema a producción debe ser auditado, los auditores deben tener acceso al estado del sistema y a los logs.

2.3.2 Modelo BIBA

Fue propuesto por Kenneth J. Biba en 1977 como un modelo de seguridad para mantener la integridad de la información. Surgió como complemento al modelo Bell-LaPadula, que se enfocaba en confidencialidad. Mientras BLP protege que la información no se filtre hacia niveles inferiores, Biba protege que la información no sea modificada por niveles no autorizados. Su objetivo principal es evitar la corrupción de datos, especialmente en entornos donde la exactitud y confiabilidad son críticas como sistemas financieros, bases de datos corporativas, control industrial. El modelo Biba es un modelo de control de acceso mandatorio (MAC) que enfoca la seguridad en la integridad, no en la confidencialidad. Define **niveles de integridad** para sujetos (usuarios, procesos) y objetos (archivos, datos, sistemas). Lo que significa que cuánto más alto el nivel de integridad, más confianza en que

- Un programa ejecutará correctamente
- La información es correcta y/o confiable

Principios básicos

- **Read Up** : un sujeto **s** puede leer un objeto **o** si y sólo si $i(s) \leq i(o)$. Esto es, un sujeto con alto nivel de integridad no puede leer datos de bajo nivel de integridad, para no contaminarse.
- **Write Down** : un sujeto **s** puede escribir un objeto **o** si y sólo si $i(o) \leq i(s)$. Esto es, un sujeto con bajo nivel de integridad no puede modificar datos de alto nivel de integridad.

2.3.3 Clark-Wilson

Propuesto por David D. Clark y David R. Wilson en 1987. Surgió como un modelo práctico de integridad para sistemas comerciales, especialmente para bases de datos financieras y sistemas de transacciones. A diferencia de Biba, que es más académico y teórico, Clark-Wilson está pensado para aplicaciones del mundo real, enfocándose en mantener integridad de datos transaccionales. Algunas políticas de integridad utilizan esta noción de **transacción**. Esto es, **el sistema comienza en un estado inicial consistente y se realizan una serie de operaciones (transacciones) atómicas (es decir, no se interrumpen, y en caso de fallar hacen un rollback al estado inicial) que llevan el sistema de un estado consistente a otro consistente**.

2.3.4 El rol de la confianza

La seguridad informática no se trata solo de reglas y algoritmos; también se trata de en qué elementos del sistema decidimos confiar. Esa confianza es inevitable, pero debe ser consciente y bien fundamentada. Las políticas de integridad no solo definen cómo proteger los datos, sino también en qué y en quién confiamos dentro de un sistema. Ya hablamos sobre el modelo BIBA, este se centra en integridad multinivel. En el caso del modelo Clark-Wilson, este hace foco en las transacciones y la separación de tareas.
Otros escenarios de la vida cotidiana son los **parches**. Implícitamente confiamos en:

- Que el parche viene del vendedor del sistema operativo y que no fue modificado.
- Que el vendedor probó correctamente el parche antes de la liberación.
- Que el ambiente de prueba del vendedor se corresponde con nuestro ambiente.
- Que el parche se instaló correctamente.

Con esto en mente, llegamos a la conclusión de que el rol de la confianza en el cualquier política o procedimiento de seguridad es importantísimo, ya que asumir que estos están basados en asumir hechos, que de ser incorrectos, destruyen todo lo construido. Es indispensable tener esto en mente, ya de lo contrario, si no entendemos en que se basa la política, el mecanismo o el procedimiento de seguridad, se pueden asumir cosas inválidas y llegar a conclusiones erróneas.

2.4 Política de híbrida

2.4.1 Modelo Pared China

En los años 80, surgió en el ámbito de las finanzas y consultoría empresarial una preocupación concreta:

- Las empresas de consultoría, auditoría o asesoría financiera trabajan con información muy sensible de clientes competidores.
- Un mismo consultor podía tener acceso a datos confidenciales de dos bancos, petroleras o aseguradoras que compiten directamente.
- El riesgo era que esa información se usara de forma inadecuada, comprometiendo la confidencialidad y la imparcialidad.

Para responder a este problema, Brewer y Nash propusieron en 1989 el Chinese Wall Security Policy (política del "Muro Chino" o **Pared China**). El nombre proviene de la metáfora de un muro informativo dentro de la propia organización, que separa los datos de distintos clientes competidores, evitando que una persona que accedió a información de uno pueda acceder a la del otro.

Problema

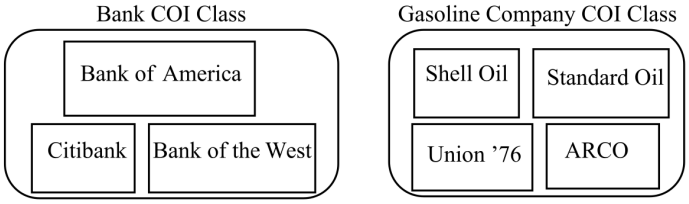
- Armando es un analista de mercado que asesora al Banco Mayo en temas de planes corporativos de negocios.
- Se le solicita que también aconseje al Banco Junio en los mismos temas.
- Se produce un **Conflicto de interés**, porque al tene información interna de uno de ellos, como ser planes, estado financiero, etc, puede obtener ventajas en la forma en la que asesora al otro.

La forma de pensar la solución a este problema de la política Pared China es :

- Organizar entidades en **clases de "conflicto de interés"**.
- Controlar el acceso de los sujetos a cada clase.
- COntrolar la escritura a todas las clases para asegurarse que la información no es pasada de una a otra violando las reglas.
- Permite que todos vean la información "esterilizada" (ejemplo : los balances que son públicos).

Definición formal de los elementos del modelo Pared China

- **Objetos**: Son los elementos de información concretos, por ejemplo, un documento financiero, un reporte de auditoría, una base de datos de clientes de una empresa. Cada objeto está asociado a una compañía específica.
- **Company Dataset (CD)** : Es el conjunto de objetos que pertenecen a una misma compañía. Se denota como CD(O), es decir: "el dataset de compañía al que pertenece el objeto O".
- **Clase de Conflicto de Interés (COI)** : Agrupa a varios Company Datasets de compañías que compiten entre sí. Se denota como COI(O), es decir: "la clase de conflicto de interés a la que pertenece el objeto O". Podemos ver que cada objeto sólo puede pertenecer a una COI (El objetivo del modelo es evitar que un sujeto acceda a información de compañías competidoras. Para que esta restricción sea clara y aplicable, cada objeto debe poder clasificarse sin ambigüedad en un único dataset de compañía (CD) y, por lo tanto, en una única clase de conflicto de interés (COI). Si un objeto perteneciera a dos COI, no habría una regla clara sobre qué accesos están prohibidos.).



Contexto dinámico de los accesos previos

Las decisiones de acceso no dependen solo de niveles de seguridad o roles, sino también de qué información ya accedió el usuario en el pasado.Si Armando lee cualquier CD en un COI, no podrá **NUNCA** leer otro CD en ese COI. Ya que es posible que la información que obtuvo anteriormente le sirva para decisiones posteriores. Se define PR(S) como el conjunto de objetos que S ya leyó (viene de *prior read*, siempre se inicializa vacío y el primer acceso de lectura siempre se garantiza).

Condición de seguridad simple (Lectura)

Un sujeto **s** puede leer el objeto **o** si y sólo si alguna de estas condiciones se cumple :

1. Acceso dentro del mismo dataset (CD). Existe un objeto **o'** tal que **s** ha leído **o'** y CD(o') = CD(o). O sea, **s leyó previamente algún dato en el dataset de la compañía** (ejemplo: si un analista ya vio un documento del Banco A, puede seguir leyendo otros documentos de Banco A.).

   $$\exists\ o'\ \texttt{tal que}\ o' \in PR(s)\ \wedge\ CD(o') = CD(o)$$

2. No hay conflictos de interés (COI).

   $$\forall o' \in O,\ o' \in PR(s)\implies COI(o') \neq COI(o)$$

   Es decir, **s** no leyó ningún objeto de algún CD(o) en la misma COI (ejemplo : si nunca accedió a información de bancos, entonces puede leer por primera vez información del Banco A.).

Son tres las condiciones de Seguridad Simple :

1. Ignora los datos esterilizados.
2. Inicialmente, PR(s) = $\emptyset$, por eso la petición de lectura inicial es concedida.
3. **o** es un objeto esterilizado

Datos esterilizados (sanitized data)

Los datos esterilizados son versiones de la información sin contenido sensible o identificable. El modelo aclara que estos datos no generan conflictos de interés, por lo que se pueden leer sin restricciones. Veamos lo siguiente. La información Publica puede pertenecer a un CD :

- Como está disponible públicamente, no surge ningún conflicto de intereses.
- Por eso, no debería ser restringido el acceso de ningún analista.
- Típicamente, toda la información sensible de esa información es removida antes de hacerla pública (esterilización o sanitizing).

Star Property (Escritura)

Supongamos la siguiente situación, Armando y Nancy trabajan en la misma agencia financiera. Armando puede leer el CD del Banco 1, y el CD de la compañía de Gas. Nancy puede leer el CD del Banco 2, y el CD de la compañía de Gas. Si Armando pudiera escribir al CD de GAS, Nancy podría Leerlo. Indirectamente, Nancy podría leer información sobre el Banco 1, un claro conflicto de intereses. Por ello, para la escritura sucede que **s puede escribir en o si y sólo si se cumplen las dos siguientes premisas** :

1. La CW-condición (chinese wall) simple de seguridad permite a **s** leer **o**. (No se puede escribir en un objeto al que uno no tenga acceso de lectura válido.)
2. Consistencia de datasets: Para todo objeto no esterilizado **o'**, si **s** puede leer **o'**, entonces CD(o') = CD(o). (todos los objetos no esterilizados a los que **s** tiene acceso deben pertenecer al mismo Company Dataset que **o**.)

En otras palabras, **s** puede escribir un objeto si todos los objetos (no esterilizados) que puede leer están en el mismo dataset. La escritura se permite únicamente cuando no hay riesgo de mezclar información sensible de compañías competidoras.

2.5 ORCON (Originator Controlled Access Control)

En seguridad de la información, surge a menudo la necesidad de controlar no solo quién accede a un documento, sino también cómo se redistribuye una vez generado. ORCON surge en el ámbito militar estadounidense, específicamente en el manejo de información clasificada. Problema: los documentos clasificados generados por una autoridad debían diseminarse sólo a personas autorizadas, y no podía permitirse que los receptores redistribuyeran libremente esa información. Ejemplo: un memorándum del Secretario de Defensa debía llegar a sus subordinados, pero no podía circular más allá sin autorización explícita.
Problema a resolver: Una organización quiere asegurarse de que los documentos que genera no se diseminen libremente, sino que su distribución quede bajo el control del originador (la persona u organismo que los creó).
ORCON se diseñó como un modelo híbrido, combinando:

- La autoridad del originador para decidir quién recibe información (aspecto DAC).
- La prohibición obligatoria de redistribución por receptores (aspecto MAC).

2.6 DRM (Digital Rights Management)

DRM es un sistema de control de acceso enfocado en contenidos digitales, usado para garantizar que los medios se utilicen solo por quienes tienen derechos, limitando copias, compartición y reproducción no autorizada. Es un modelo híbrido ya que :

- Como DAC: El contenido digital tiene un propietario (editorial, titular de derechos) que decide inicialmente quién puede usarlo y bajo qué condiciones.
- Como MAC: Una vez establecido el acceso, las reglas son obligatorias: los usuarios no pueden transferir, copiar o modificar los derechos asignados sin autorización. El sistema de DRM aplica estas restricciones automáticamente a todos los usuarios y dispositivos autorizados, sin depender de la voluntad del usuario.

2.7 Políticas de seguridad en sistemas de información clínica

Los sistemas de información clínica manejan datos extremadamente sensibles, por lo que las políticas de seguridad se diseñan con tres objetivos críticos:

- Confidencialidad del paciente: Solo personas autorizadas (médicos, enfermeros, personal administrativo relevante) pueden acceder a la información médica. Evita divulgación indebida de datos personales y médicos.
- Autenticidad de los registros: Asegura que la información registrada realmente proviene de quien dice haberla generado (por ejemplo, un médico autorizado).
- Integridad de los registros: Garantiza que los datos no sean modificados de manera inapropiada. Ejemplo: un médico solo puede actualizar el historial de su paciente dentro de los límites permitidos; nadie más puede alterar esos registros.

Las entidades entoces son, los **pacientes**, la **información personal de salud** y el **médico**.

Principios de Acceso

1. **Principio 1**: Cada registro médico tiene una lista de control de acceso que incluye los individuos o grupos que pueden leer y agregar información al registro. El sistema debe controlar que sólo aquellos en la lista pueden acceder al registro.

2. **Principio 2**: Uno de los médicos en la ACL, denominado médico responsable, debe tener permisos para agregar a otros médicos a la ACL.

3. **Principio 3**: El médico responsable debe notificar al paciente los nombres en la ACL cada vez que abre su registro médico. Salvo casos explícitos en estatutos, o en casos de emergencia, el médico responsable debe obtener el consentimiento del paciente.

4. **Principio 4**: El nombre del médico, la fecha y hora de acceso a un registro médico debe ser registrado. También el borrado de información.

Otros : Definen la forma en que se crean registros, cuando se puede borrar los mismos, como se agrega información de un registro médico a otro, como se fuerza el cumplimiento de estos principios, etc.

2.8 SELinux (Security Enhanced Linux)

Se originó a principios de los 2000s. SELinux fue desarrollado por la Agencia de Seguridad Nacional de Estados Unidos (NSA) junto con colaboradores de la comunidad Linux. El objetivo principal era añadir controles de seguridad obligatorios (Mandatory Access Control, MAC) a Linux, que originalmente solo tenía DAC (Discretionary Access Control). La idea era proteger sistemas críticos frente a exploits, malware o errores de configuración, agregando restricciones de acceso obligatorias sobre archivos, procesos y recursos del sistema. Se trata de una política de control de acceso híbrida ya que combina :

- Aspecto DAC (Discrecional): Linux originalmente implementa DAC clásico: el propietario de un archivo o recurso puede decidir quién tiene permisos de lectura, escritura o ejecución. SELinux no elimina estos permisos; los mantiene y respeta como parte del control.

- Aspecto MAC (Mandatorio): SELinux superpone políticas obligatorias sobre DAC. Cada objeto y proceso recibe una etiqueta de seguridad (security context), y las políticas MAC determinan qué puede hacer cada sujeto con cada objeto. Estas restricciones son imposibles de ignorar por el usuario; incluso si el propietario DAC lo permitiera, el sistema puede denegar el acceso según la política MAC.

2.9 Windows Mandatory Integrity Control (MIC)

MIC fue implementado a partir de Windows Vista (lanzado en 2007). Se diseñó para proteger la integridad del sistema operativo y los datos ante malware y procesos no confiables. La idea central era evitar que procesos de menor integridad afecten a objetos de mayor integridad, siguiendo principios del modelo Biba de integridad. Todos los objetos del sistema (archivos, carpetas, procesos, usuarios) reciben un nivel de integridad. Windows define **niveles** como:

- **Low** : procesos no confiables.
- **Medium** : nivel por defecto para usuarios estándar y objetos sin etiqueta.
- **High** : usuarios administrativos o procesos críticos.
- **System** : procesos y objetos esenciales del sistema operativo.

Cada usuario o proceso también tiene un nivel de integridad asignado. Y tiene las siguientes restricciones principales:

- Un usuario o proceso no puede elevar el nivel de integridad de un objeto por encima del suyo propio. Esto asegura que procesos de baja integridad no puedan modificar objetos de alta integridad, previniendo escalamiento de privilegios.
- El nivel medio es el nivel por defecto para usuarios estándar y objetos sin etiquetas.