Modèles de politique de contrôle d'accès

Romain Laborde
laborde@irit.fr

Introduction: **Propriétés** de sécurité

- La propriété de **confidentialité** consiste à dissimuler une information ou une ressource.
  - Elle s'applique aussi à l'existence même de l'information ou de la ressource.
- Par exemple, dans le cadre des réseaux, les administrateurs des sites peuvent désirer cacher la nature de leur réseau (topologie, équipements installés, configurations des équipements, serveurs installés, etc.).
- Peut être obtenue par le biais du service de
  - confidentialité,
  - contrôle d'accès
  - => service d'identification/authentification

Introduction: **Propriétés** de sécurité

- L'**intégrité** fait référence à la confiance que l'on peut avoir dans une information ou une ressource.
  - Les *mécanismes de prévention* : but de maintenir l'intégrité d'une information en bloquant les tentatives de modification non autorisées ou de manière non autorisée.
    - un utilisateur n'ayant pas le droit de modifier une donnée qui tente de la modifier.
    - un utilisateur possédant ce droit mais qui effectue cette tâche dans un but frauduleux.
  - Les *mécanismes de détection* rapportent simplement le fait que l'intégrité d'une information ou d'une ressource n'est plus garantie.

Introduction: **Propriétés** de sécurité

- La propriété de **disponibilité** fait référence à la capacité d'utiliser une information ou une ressource autorisée.
- Cette propriété est très importante car pour un utilisateur, un service qui n'est pas accessible équivaut à la non existence de ce service.
- peut être causé par malveillance comme les attaques de déni de service ou de déni de service distribué.
  - Ce type d'attaque est très difficile à détecter car il faut être capable de distinguer les flux de données correspondant à une attaque des flux de données légitimes.
- peut aussi résulter d'une mauvaise étude du système ou d'une mauvaise configuration de celui-ci.

Introduction: **Services** de sécurité

- Cinq services définis par la norme ISO 7498-2
  - **le contrôle d'accès** fournit des mécanismes permettant de prévenir l'utilisation non autorisée de ressources.
  - **l'identification/authentification** définit l'identité de l'entité et la valide.
  - **la confidentialité** protège les données d'une divulgation non autorisée.
  - **l'intégrité** détecte/prévient de toute modification, insertion ou suppression de données non autorisée. De la même manière que pour la confidentialité, généralement le service d'intégrité est souvent restreint aux problèmes de transfert de données sur un réseau.
  - **la non répudiation** évite qu'une des parties impliquées dans une communication renie avoir participé totalement ou en partie à cette communication.
  - **la traçabilité** permet de retrouver les opérations effectuées par une entité. Il s'agit alors de pouvoir journaliser les évènements du système.

Introduction: Politique de sécurité

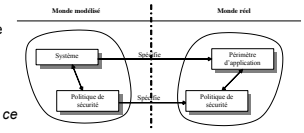
- Définition:
  - Une *politique de sécurité* consiste en un ensemble de documents décrivant *les principes ou règles* auxquelles se conforment les personnes qui reçoivent un droit d'accès au capital technologique et informatif de l'entreprise. Ces documents à caractère non technique donnent aux responsables de l'entreprise *les axes à suivre*.

Introduction: Services de sécurité

- La politique de sécurité étant une vision à long terme de la sécurité, elle doit être pérenne.
- Elle décrit les besoins et non les moyens.
  - Les moyens sont appelés à être modifiés fréquemment pour tenir compte des avancées technologiques, des modifications de l'architecture, etc.
  - Les besoins sont eux fonctions de l'organisation et des activités de l'entreprise.

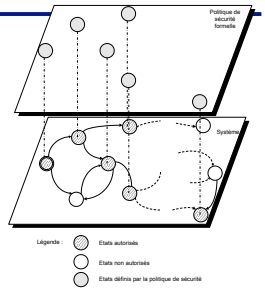
Les politiques de contrôle d'accès formelles

- Objectif:
  - pouvoir décrire de manière non ambigüe un problème afin de pouvoir par exemple évaluer mathématiquement les solutions répondant à ce problème.
- Définition:
  - Une *politique de sécurité formelle* est un énoncé qui définit *ce qui est autorisé et ce qui ne l'est pas* dans le contexte d'un modèle formel.



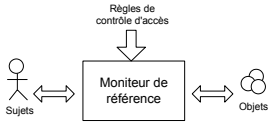
Les politiques de contrôle d'accès formelles

- Le comportement du système est un ensemble S d'états dans lequel le système peut entrer.
- La politique de sécurité formelle définit un sous-ensemble P d'états dits autorisés tel que $P \subseteq S$
- Les mécanismes de sécurité doivent garantir que le système ne rentre pas dans un état $s \in S/P$



Le moniteur de référence

- Moniteur de référence:
  - la représentation conceptuelle de l'ensemble des mécanismes qui mettent en œuvre les règles définies par la politique de contrôle d'accès.



Les modèles de contrôle d'accès

- la caractéristique commune à tous ces modèles de contrôles d'accès est qu'ils considèrent trois ensembles :
  - **l'ensemble des sujets S** qui représentent les entités qui exécutent des requêtes sur les objets (par exemple les utilisateurs, les processus). Ces entités sont dites actives.
  - **l'ensemble des objets O** qui représentent les entités qui reçoivent ou contiennent de l'information (par exemple les fichiers). Ces entités sont dites passives.
  - **l'ensemble des droits d'accès R** des sujets sur les objets.
- Une politique de contrôle d'accès consiste donc en un ensemble de relations sur $S \times O \times R$.
- Les différences qui existent entre ces modèles portent sur les droits d'accès, la façon de grouper les sujets/objets et donc les relations sujets-droits-objets.

Les modèles de contrôle d'accès

- Les droits:
  - **le droit de lecture** - un sujet possédant ce droit peut récupérer l'information contenue dans un objet (par exemple lire un fichier).
  - **le droit d'écriture** - un sujet possédant ce droit peut modifier ou ajouter de l'information dans un objet (par exemple écrire dans un fichier).
  - **le droit d'exécution** - le droit d'exécution consiste en un droit d'accès qui n'est ni lire, ni écrire (par exemple exécution d'un processus).
- Un quatrième droit d'accès qui est le droit d'administration ou droit de possession n'est utilisé que dans certains modèles et de manière différente.

Modèles de politiques discrétionnaires

- Les *modèles de politiques discrétionnaires* (Discretionary Access Control - DAC) considèrent que chaque sujet peut *détenir un droit de possession* sur un objet.
- Ce droit particulier permet à son propriétaire (qui est souvent le créateur de l'objet) d'ajouter ou soustraire des droits d'accès pour lui ou pour les autres.

Le modèle de matrice de contrôle d'accès

- Introduit par Lampson [Lampson 1971] puis améliorée par Graham et Denning [Denning 1971, Graham *et al.* 1972]
- Une matrice de contrôle d'accès est une fonction $A : S \times O \rightarrow 2^R$ qui donne pour chaque couple (sujet, objet), l'ensemble des droits associés.
- Tout sujet peut avoir le droit de possession
- L'analyse s'effectue sur des états appelés états de protection qui sont représentés par des triplets (S, O, A) où S est l'ensemble des sujets, O l'ensemble des objet et A la matrice de contrôle d'accès.
  - L'état du système change lorsque un sujet, un objet ou un droit d'accès est ajouté ou supprimé.

Problème modèle matrice d'accès

- Problème de la gestion de la propagation des droits
  - Le sujet s_1 crée le fichier *toto.txt*.
  - Il en devient *propriétaire*.
  - Il décide de donner le droit de *lecture* au sujet s_2 mais pas au sujet s_3. Cette règle constitue sa politique de contrôle d'accès.
  - Lorsque s_2 lit le fichier *toto.txt*, il décide de recopier le contenu dans un nouveau fichier *copie_toto.txt*.
  - Maintenant, s_2 détient les droits de *possession* sur le fichier *copie_toto.txt* qui contient la même information de *toto.txt*.
  - Il décide d'en donner le droit de *lecture* à s_3.

  => s_3 peut *lire* l'information contenue dans *copie_toto.txt* et donc *toto.txt*

Le modèle HRU [Harrison *et al.* 1976]

- Amélioration du modèle de Lampson pour la vérification des droits
- HRU utilise une matrice de contrôle d'accès classique, la différence réside en ce que HRU précise les *commandes* qui peuvent lui être appliquées

Entrer a **dans** A[s,o]	**Supprimer** a **de** A[s,o]
Créer le sujet s	**Détruire le sujet** s
Créer l'objet o	**Détruire l'objet** o

- A partir de ces opérations élémentaires, il est possible de créer des commandes.
- Exemple
  - Soit *copie* le droit de copie et *lire* le droit de lecture :

commande *délivrer_lecture (s1, s2, f)*
  si *lire* $\in A[s1, f]$ **et** *copie* $\in A[s1, f]$
  alors
    entrer *lire* **dans** *A[s2, f]*
fin

Modèle HRU: résultats fondamentaux

- Comment déterminer si un tel système est sécurisé ?
- Ce problème est posé par rapport à la matrice de contrôle d'accès par le *problème de protection* (safety problem) :
  - Existe-t-il une séquence d'opérations qui amène un droit d'accès α à se retrouver dans une case de la matrice d'accès où il ne devrait pas être ?
- Trois théorèmes fondamentaux concernant la complexité du problème de protection ont été démontrés:
  - le problème de protection est *indécidable dans le cas général* [Harrison *et al.* 1976],
  - le problème de protection est *décidable pour les systèmes à mono-opération*, i.e., dont les commandes ne contiennent qu'une seule opération élémentaire [Harrison *et al.* 1977].
  - le problème est *décidable pour les systèmes mono-conditionnels*, i.e. dont les commandes ne contiennent qu'une condition, *privés des opérations détruire* [Harrison *et al.* 1977].
- Par conséquent, la question de protection est indécidable dans le cas général. Elle le devient dans le cas de systèmes restreints.

Autres études sur ce problème de protection

- Le modèle Take-Grant [Jones *et al*. 1976]
  - démontre que le problème de protection est décidable pour un système donné
- Le modèle de protection schématique (SPM)
  - différence qui existe entre un modèle pour lequel la question de protection est décidable d'un modèle où ce n'est pas le cas [Sandhu 1988, Sandhu 1989, Sandhu 1992a, Ammann *et al*. 1992]
- Le modèle de matrice de contrôle d'accès typée (TAM [Sandhu 1992b])
  - Les opérations sont les mêmes que le modèle HRU mise à part pour l'opération *créer* qui prend en compte les types
  - Sandhu a prouvé que pour les modèles MTAM (Monotonic Typed Access Matrix) - un modèle TAM privé des opérations supprimer, détruire sujet et détruire objet :
    - Le problème de protection est décidable pour des systèmes avec des schémas MTAM acycliques et qu'il est NP-complet,
    - Le problème de protection est décidable pour des systèmes avec des schémas MTAM acycliques ternaires (i.e., trois opérations par commandes) et qu'il est polynomial par rapport à la taille initiale de la matrice de contrôle d'accès.

Vers les modèles de politiques obligatoires

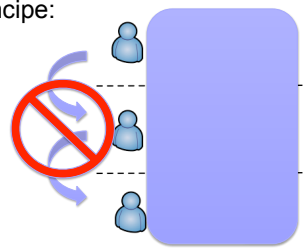
- Les modèles de politiques discrétionnaires ont une approche complètement distribuée des droits d'administration (droit de *possession*, *take*, *grant*, etc.).
- Cette approche pose le *problème de protection* qui peut se résumer à : est ce qu'un droit spécifique ne va pas être détenu par la mauvaise personne ?
- Une autre famille de modèles formels, dits d'autorisation obligatoire, règle ce problème en centralisant l'autorité d'administration
  - => les droits de type *possession* ou *grant* n'existent pas

Le modèle Bell LaPadula [Bell *et al.* 1973]

- Modèle militaire permettant de définir des politiques garantissant la **propriété de confidentialité**, i.e. qui prévient la divulgation non autorisée d'information.
- Les permissions d'accès sont définies à travers une matrice de contrôle d'accès et un ensemble de niveaux de sécurité.
- Etant donné qu'il n'est pas possible d'ajouter/supprimer des droits comme dans les politiques DAC, les politiques de type Bell-LaPadula considèrent les flux d'informations qui se produisent quand un sujet observe ou modifie un objet et non l'état de protection du système.

Le modèle de BLP

- Principe:



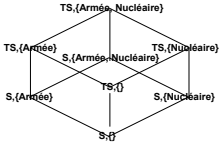
22

Le modèle Bell LaPadula [Bell *et al.* 1973]

- Il s'appuie sur l'association de différents niveaux de confidentialité aux sujets (appelés dans ce cas niveaux d'habilitation) et aux objets (appelés dans ce cas niveaux de classification)
  - Pour un objet *o*, sa *classification fo*(*o*) est un moyen de représenter le danger que peut constituer la divulgation de l'information contenue dans cet objet.
  - Pour un sujet *s*, c'est une *habilitation fs*(*s*) qui désigne la confiance qui lui est accordée.
- Chaque niveau $n = (cl, C)$ est caractérisé par ses deux attributs:
  - *cl* : une classification prise dans un ensemble totalement ordonné, par exemple : {non-classifié<confidentiel<secret}.
  - *C* : un compartiment défini par un ensemble de catégories i.e. il décrit le type d'information, par exemple {nucléaire, défense}.

Le modèle Bell LaPadula [Bell *et al.* 1973]

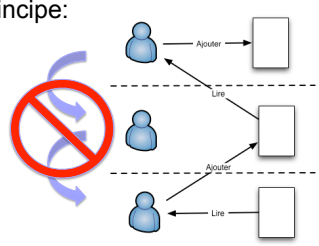
- Les niveaux constituent un *treillis* par une relation de dominance notée "$\leq$" et définie par : Soit $n = (cl, C)$, $n' = (cl', C')$,
  $n \leq n' \Leftrightarrow cl \leq cl' \wedge C \subseteq C'$. On dit alors que n' domine n.



- Un treillis complet est un ensemble partiellement ordonné L tel que tout sous-ensemble X de L possède :
  - 1) Un plus grand minorant appartenant à L,
  - 2) un plus petit majorant appartenant à L.

Le modèle de BLP

- Principe:



25

Sémantique des droits

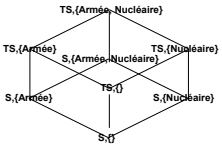
- Le modèle considère les quatre droits d'accès suivants :
  - *exécuter* : accès sans modification ni observation ;
  - *lire* : observer sans modifier ;
  - *ajouter* (*append*) : modification sans observation, par exemple dans le cas des écritures dans les fichiers d'audits (*audit logs*) ;
  - *écrire* : observation et modification.

Etat du système

- Chaque état du système v ∈ V est représenté par un quadruplet (b, a, f, h) où :
  - b ∈ $\mathcal{P}$(S×O×P) où $\mathcal{P}$(S×O×P) est l'ensemble de tous les sous-ensembles possible de S*O*P, indique quel sujet possède quel droit sur quel objet avec S l'ensemble des sujets, O l'ensemble des objets et P = {lire, écrire, ajouter, exécuter} ;
  - a ∈ A la matrice de contrôle d'accès pour cet état avec A l'ensemble des matrices de contrôle d'accès possibles ;
  - f ∈ M est un triplet qui définit les niveaux de classification des objets et d'habilitation des sujets.
    - f_s le niveau maximal d'un sujet,
    - f_c le niveau courant d'un sujet
    - f_o le niveau d'un objet;
  - et h ∈ H représente la hiérarchie qui existe entre les objets.

La condition de simple sécurité

- Un sujet ne peut avoir un droit d'observation sur un objet que si celui-ci domine cet objet
- Définition formelle
  - (s,o,p) ∈ S×O×P satisfait la simple sécurité si et seulement si l'une de ces condition est vérifiée :
    - 1) p = exécuter ou p = ajouter
    - 2) (p=lire ou p=écrire) et $f_o(o) \leq f_s(s)$
- Exemple
  - Est-ce que cette condition est satisfaite lorsque Romain avec le niveau (TS, {Nucléaire}) veut lire:
    - Fichier1 dont le niveau est (S, {Armée})
    - Fichier2 dont le niveau est (TS, {nucléaire, armée})
    - Fichier3 dont le niveau est (S, {nucléaire})



La propriété étoile

- La propriété étoile (*-property) implique qu'un sujet ne peut avoir un droit de modification d'un objet que si cet objet le domine.
- Définition formelle
  - Un état (b, m, f, h) ∈ V satisfait la propriété étoile si et seulement si chaque triplet (s , o, p) ∈ b vérifie l'une des conditions suivantes :
    1) (p = ajouter) ⇒ $f_c(s) \leq f_o(o)$
    2) (p = écrire) ⇒ $f_c(s) = f_o(o)$
    3) (p = lire) ⇒ $f_o(o) \leq f_c(s)$

La propriété de sécurité discrétionnaire

- La propriété de sécurité discrétionnaire fait le lien entre les états du système et la matrice de contrôle d'accès
- Définition formelle
  - Un état (b, a, f, h) ∈ V satisfait la propriété de sécurité discrétionnaire si et seulement si pour chaque triplet (s , o, p) ∈ b, p ∈ a[s, o].

Problème modèle BLP

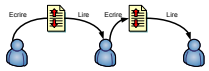
- La dégradation du service provoquée par la *surclassification* des informations.
  - Le niveau d'une information ne peut que croître : si une information non classifiée est utilisée par un sujet habilité au secret, tout objet modifié par ce sujet avec cette information sera classifié secret.
  - Petit à petit, les niveaux de classification des informations croissent de façon automatique, et il faut les "déclassifier", manuellement par un officier de sécurité ou par un processus dit "de confiance" n'obéissant pas aux règles du modèle.
- Système *Z* [McLean 1985, McLean 1987]
  - vérifie bien les trois propriétés, et qui n'est pourtant pas sûr.
  - Le système *Z* est un système où un utilisateur de niveau minimal met les niveaux de tous les sujets et de tous les objets au niveau minimal, et autorise l'accès de tous les utilisateurs à tous les objets. Ceci est possible car le niveau d'un objet peut, lui-même, être mémorisé dans un objet ; la valeur de ce dernier peut donc être modifiée par un utilisateur de niveau minimal (puisque l'écriture dans un niveau dominant est autorisée).
  - Certains arguments de McLean seront remis en question par LaPadula [LaPadula 1988] comme par exemple la nature des règles introduites ou encore l'interprétation du modèle faite par McLean.

Le modèle de Biba [Biba 1977]

- Traite de l'intégrité des systèmes
- Tout comme le modèle BLP:
  - sujet S, d'objets O et de niveaux d'intégrité I
  - relation d'ordre $\preceq \subseteq I \times I$ définit la dominance entre les niveaux d'intégrité
  - fonction i : S∪O → I retourne le niveau d'intégrité d'un sujet ou d'un objet
  - Droits = lire, écrire, exécuter (sur un autre sujet)

Le modèle de Biba

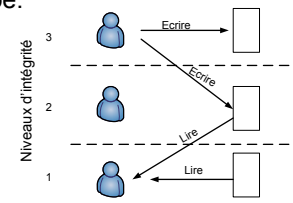
- Notion de chemin de transfert:
  - Un chemin de transfert d'information est une séquence d'objets $o_1, \ldots, o_{n+1}$ et une séquence de sujets $s_1, \ldots, s_n$ tel que pour tout i, s_i lit o_i et s_i écrit o_{i+1}



- Les règles régulant les transferts d'information sont les suivantes :
  - $s \in S$ peut *lire* $o \in O$ si et seulement si $i(s) \preceq i(o)$
  - $s \in S$ peut *écrire* $o \in O$ si et seulement si $i(o) \preceq i(s)$
  - $s_1 \in S$ peut *exécuter* $s_2 \in s$ si et seulement si $i(s_2) \preceq i(s_1)$

Le modèle de Biba

- Principe:



34

Le modèle Clark-Wilson [Clark *et al.* 1987]

- Approche différente des modèles précédents
- Ils posent le problème suivant :
  - Aucun utilisateur, même s'il en possède les droits, ne doit pouvoir corrompre ou supprimer les actifs ou les enregistrements des comptes de la compagnie
- Ce modèle considère donc comme opération de base non pas les droits mais les transactions effectuées
- La politique de Clark et Wilson repose sur deux anciens principes bien connus :
  - les transactions bien formées:
    - les utilisateurs n'ont pas le droit de manipuler les données d'une manière arbitraire, mais seulement au travers des procédures de transformation spécifiques, préservant l'intégrité des données
  - et la séparation des pouvoirs:
    - fondé sur la répartition des pouvoirs entre plusieurs parties, et l'attribution des droits différents, mais complémentaires, à différentes catégories de personnes

Le modèle Clark-Wilson

- Le modèle de Clark et Wilson sépare les données manipulées en deux groupes :
  - les données contraintes (notées CDI pour *Constrained Data Items*)
    - données soumises à des règles de manipulation strictes visant à conserver leur intégrité
  - et les données non contraintes (notées UDI pour *Unconstrained Data Items*)
    - les données dont l'intégrité n'est pas garantie et qui peuvent être manipulées arbitrairement
- Ce modèle s'appuie sur deux notions:
  - un ensemble d'opérations de vérification de l'intégrité des données (IVP) qui valident les CDI ;
  - l'ensemble des opérations de transformation des données (TP) qui représentent l'unique moyen de manipulation des CDI ;

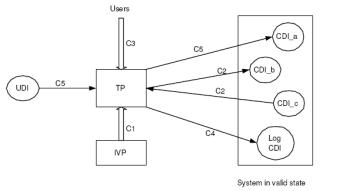
36

Le modèle Clark-Wilson [Clark *et al.* 1987]

- Ces règles nécessitent des techniques pour leur mise en œuvre au niveau du système informatique
  - Les procédures (transactions bien formées) manipulant les données doivent être examinées et bien établies
  - la capacité à les installer et les modifier doit être contrôlée
  - la validité des données doit toujours faire l'objet d'une vérification
  - l'affectation des droits aux utilisateurs ainsi que la séparation des pouvoirs doivent êtres menées à bien,
  - Etc

Le modèle Clark-Wilson

- Règles de certification



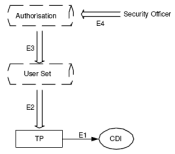
Certification Rules

C1: IVP validates CDI state
C2: TPs preserve valid state
C3: Suitable separation of duty
C4: TPs write to log file
C5: TPs validate UDI

38

Le modèle Clark-Wilson

- Règles de mise en oeuvre



Enforcement Rules

E1: CDIs changed only by authorised TP
E2: Users authorised for TP
E3: Users are authorised
E4: Authorisation lists changed only by security officer

39

Le modèle Clark-Wilson: Analyse

- Analyse 1
  - Le modèle de Clark et Wilson attache des niveaux d'intégrité aux sujets (certifiés par les TPs et non certifiés via les autres procédures) et aux objets (contraints CDI et non contraints UDI)
  - Comme Biba

Le modèle Clark-Wilson: Analyse

- Analyse 2
  - Il ajoute un nouveau concept qui n'existe pas dans le modèle de Biba qui est la *certification*.
  - De plus, ce modèle met en évidence deux préoccupations importantes dans bon nombre de structures :
    - la *traçabilité*, c'est-à-dire la possibilité de reconstituer les actions importantes du point de vue des objectifs de sécurité,
    - et la *séparation des pouvoirs*.

Le modèle Clark-Wilson: Analyse

- Analyse 3
  - Le modèle prend en compte la possibilité que le système dévie de son fonctionnement normal.
    - l'existence de procédures de validation de l'intégrité,
    - les données non-contraintes sont acceptées par certaines procédures de transformation.
  - L'existence d'un historique de l'exécution du système permet alors d'identifier les comportements erronés qui ont pu être à l'origine d'une violation des objectifs de sécurité (par exemple, une faute dans l'implémentation d'une opération, non détectée par la certification).
  - Volonté de définir des politiques de sécurité applicables dans un environnement moins rigide
    - mécanismes capables de fonctionner en l'absence de propriétés de sécurité attendues pour le système,

La muraille de Chine [Brewer *et al.* 1989]

- Traite les propriétés de confidentialité et d'intégrité
- Approche différente
  - définir les conflits d'intérêts qui existent
- Définition 1
  - Un ensemble de données d'une compagnie (CD - Company Dataset) contient l'ensemble des données relatives à une compagnie.
- Définition 2
  - Une classe de conflit d'intérêt (COI - Conflict Of Interest) contient les données des compagnies qui sont en concurrence.

La muraille de Chine [Brewer *et al.* 1989]: Exemple

- Deux banques Banque1 et Banque2 sont en compétition.
- Il existe donc une classe COI contenant les CD de Banque1 et de Banque2.
- Un employé de la Banque1 qui a eu accès aux données confidentielles de cette banque ne peut par la suite accéder aux données de la Banque2 car il pourrait leur divulguer les données confidentielles.
- Néanmoins, cette restriction ne s'applique pas pour toutes les données.
- Le modèle de la muraille de chine considère donc deux types de données :
  - stérilisées (qui ne tombent pas sous le coup de cette restriction)
  - et non stérilisées (qui tombent sous le coup de cette restriction).

La condition de simple sécurité de la muraille de chine

- Soit PR : S → 2^O qui définit pour un sujet l'ensemble des objets qu'il a lus dans le passé
- Un sujet « s » peut lire un objet « o » si et seulement si une de ces conditions est vrai :
  - 1. il existe un objet o'∈ PR(s) tel que CD(o) = CD(o'),
  - 2. pour tous les objets o',
    $$o' \in PR(s) \Rightarrow COI(o) \neq COI(o'),$$
  - 3. l'objet « o » est stérilisé.
- Les accès passés sont pris en compte par le modèle grâce à la fonction d'historique PR !

La propriété-* de la muraille de chine

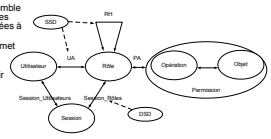
- Un sujet « s » peut écrire sur un objet « o » si et seulement si les deux conditions suivantes sont satisfaites :
  - 1. la condition de simple sécurité permet à « s » de lire « o »,
  - 2. pour tous les objets o' non stérilisés, s peut lire o' ⇒ CD(o) = CD(o').

Le modèle RBAC [ANSI359] Role Based Access Control

- Constatation simple :
  - les permissions accordées aux sujets sont données par rapport aux fonctions qu'ils occupent dans l'organisation
- Définition d'un rôle
  - Une fonction dans le contexte d'une organisation dont la sémantique dépend de l'autorité et des responsabilités des utilisateurs qui sont assignés à ce rôle
- Différence entre notions de rôle et de groupe
  - Un rôle est une fonction dans une organisation. Un groupe est un ensemble d'entités. Par conséquent, l'ensemble des sujets assignés à un même rôle forme un groupe.
  - La réciproque n'est pas forcément vraie car un groupe peut être formé de plusieurs entités associées à des rôles différents

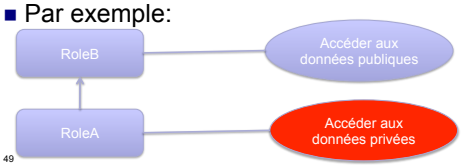
Le modèle RBAC [ANSI359]

- Un *utilisateur* est une entité active, i.e., une personne ou un agent intelligent,
- Un *rôle* a été défini précédemment,
- Une *opération* est une entité qui après invocation exécute des actions à la place de l'utilisateur,
- Un *objet* est une entité qui contient ou reçoit de l'information,
- Une *permission* est une approbation pour réaliser des opérations sur des objets,
- Une *session* permet à un utilisateur de disposer de plusieurs rôles en même temps.
- Le modèle décrit les relations entre les ensembles :
  - UA ⊆ Utilisateur × Rôle,
  - PA ⊆ Permission × Rôle,
  - ≥ ⊆ Rôle × Rôle définit un ordre partiel sur l'ensemble Rôle,telle que rôle1 ≥ rôle2 implique que toutes les permissions assignées à rôle2 sont aussi assignées à rôle1,
  - Session_Utilisateurs : Session → Utilisateur, permet d'établir l'utilisateur d'une session,
  - Session_Rôles(s : Session) ⊆ {r ∈Rôle | (Session_Utilisateurs (s), r) ∈UA} permet d'établir l'ensembles des rôles associés à une session.
  - Assigned-users(r: Role) pas de RH
  - Authorized_users(r:Role) prise en compte de la relation RH



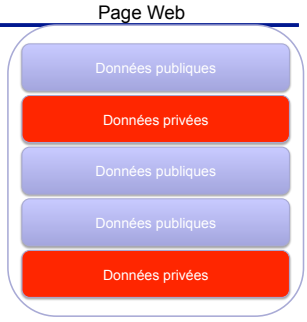
Le modèle RBAC

- Hiérachie de rôles
  - Si RoleA ≥ RoleB alors toute permission accordée à RoleB est aussi accordée à RoleA.
- Par exemple:



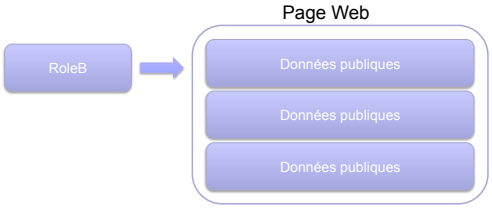
49

Solution



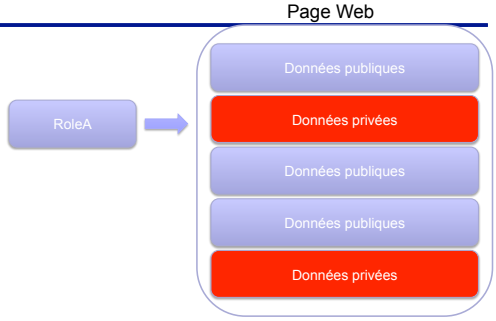
50

Solution (accès RoleB)



51

Solution



52

Le modèle RBAC [ANSI359]

- Possibilité de définir des contraintes permettant d'exprimer la séparation de certains pouvoirs à cause de conflits d'intérêt par exemple
  - Cf modèle Clark-Wilson et Muraille de Chine
- RBAC propose deux types de séparation de fonctions :
  - les séparations de fonction statiques (Static Separation of Duties - SSD)
    - SSD sur UA
      - Empêcher l'assignation d'un utilisateur à deux rôles en conflit
    - SSD sur RH
      - Empêcher qu'une hiérarchie de rôles amène un utilisateur à posséder les permissions de deux rôles en conflit
  - les séparations de fonction dynamique (Dynamic Separation of Duties - DSD).
    - La DSD porte sur Session_Rôles et donc
    - Eviter qu'un utilisateur possède deux rôles en conflit dans une même session.
    - Contrairement à la SSD, la DSD n'implique pas que l'utilisateur ne soit pas assigné à deux rôles. Elle empêche simplement que l'utilisateur bénéficie des permissions des deux rôles en même temps.

Le modèle RBAC [ANSI359]

- Définition de SSD sur RH

Definition 3b. Static Separation of Duty in the Presence of a Hierarchy.

—In the presence of a role hierarchy *static separation of duty* is redefined based on authorized users rather than assigned users as follows.

$$\forall (rs, n) \in SSD, \forall t \subseteq rs: |t| \geq n \Rightarrow \bigcap_{r \in t} authorized\_users(r) = \emptyset.$$

- Définition de DSD

Definition 4. Dynamic Separation of Duty.

—$DSD \subseteq (2^{ROLES} \times \mathbb{N})$ is collection of pairs (rs, n) in *Dynamic Separation of Duty*, where each rs is a role set and n is a natural number ≥ 2, with the property that no subject may activate n or more roles from the set rs in each $dsd \in DSD$. Formally:

$$\forall rs \in 2^{ROLES}, n \in \mathbb{N}, (rs, n) \in DSD \Rightarrow n \geq 2 \wedge |rs| \geq n, \text{ and}$$
$$\forall s \in SESSIONS, \forall rs \in 2^{ROLES}, \forall role\_subset \in 2^{ROLES}, \forall n \in \mathbb{N}, (rs, n) \in DSD,$$
$$role\_subset \subseteq rs, role\_subset \subseteq session\_roles(s) \Rightarrow |role\_subset| < n.$$

Références

- [Ammann *et al.* 1992] Ammann P., Sandhu R., "The Extended Schematic Model", Journal of Computer Security, pp. 335-385, 1992
- [ANSI359] , "Role-Based Access Control", ANSI/INCITS 359-2004, February 2004
- [Bell *et al.* 1973] Bell D., LaPadula L., "Secure Computer Systems : Mathematical Foundations", Rapport Technique MTR-2547, Vol 1, MITRE Corporation, 1973.
- [Biba 1977] Biba K., "Integrity Considerations for Secure Computer Systems", Rapport Technique MTR-3153, MITRE Corporation, 1977.
- [Brewer *et al.* 1989] Brewer D., Nash M., "The Chinese Wall Security Policy", IEEE Symposium on Security and Privacy, pp. 206-214, 1989
- [Clark *et al.* 1987] Clark D., Wilson D., "A Comparison of Commercial and Military Security Policies", IEEE Symposium on Security and Privacy, pp. 184-1994, 1987
- [Denning 1971] Denning D., "Third Generation Computer System", Computing Surveys, pp. 175-216, 1971
- [Graham *et al.* 1972] Graham G., Denning D., "Protection - Principles and Practice", Spring Joint Computer Conference, pp. 417-429, 1972
- [Harrison *et al.* 1976] Harrison M., Ruzzo W., Ullman J., "Protection in Operating Systems", Communication of the ACM, pp. 461-471, 1976
- [Harrison *et al.* 1977] Harrison M., Ruzzo W., "Monotonic Protection Systems", Foundations of Secure Computing, pp. 337-363, 1977

Références

- [Jones *et al.* 1976] Jones A., Lipton R., Snyder L., "A Linear-Time Algorithm for Deciding Security", 17th Symposium on the Foundations of Computer Science, pp. 33-41, 1976
- [Lampson 1971] Lampson B, "Protection", 5th Princeton Symposium of Information Science and Systems, pp. 437-443, 1971
- [LaPadula 1988] Lapadula L., "The 'Basic Security Theorem' of Bell and Lapadula Revisited", Computer Security Foundations Workshop, 1988
- [Lipton *et al.* 1977] Lipton R., Snyder L., "A Linear-Time Algorithm for Deciding Subject Security", Journal of the ACM, pp. 455-464, 1977
- [McLean 1985] Mc Lean J., " A Comment on the 'Basic Security Theorem' of Bell and Lapadula", Information Processing Letter, pp. 67-70, 1985
- [McLean 1987] Mc Lean J., "Reasoning About Security Models", IEEE Symposium on Security and Privacy, pp. 123-131, 1987
- [Sandhu 1988] Sandhu R., "The Schematic Protection Model : Its Définition and Analysis for Acyclic Attenuating Schemes", Journal of the ACM, pp. 404-432, 1988
- [Sandhu 1989] Sandhu R., "The demand Operation in the Schematic Protection Model", Information Processing Letters, pp. 213-219, avril 1989
- [Sandhu 1992a] Sandhu R., "Expressive Power of the Schematic Protection Model", Journal of Computer Security, pp. 59-98, 1992
- [Sandhu 1992b] Sandhu R., "The Typed Access Matrix Model", IEEE Symposium on Security and Privacy, pp. 122-136, 1992