CHAPITRE IV : PROTECTION & SECURITE DANS LES SYSTEMES D'EXPLOITATION

4.1 PROTECTION :

4.1.1 Introduction :

Référence : Les systèmes d'exploitation, Silberschatz.

Un système d'exploitation doit comporter des moyens et des procédures de protection des objets qu'il permet de manipuler. Les objets à protéger appartiennent à deux grandes catégories :

- les objets persistants tels que les fichiers, les périphériques, ..etc.
- les objets temporaires, comme les processus, les espaces de mémoire crée créés, ..etc.

La protection consiste à empêcher qu'un utilisateur puisse altérer un fichier qui ne lui appartient pas et dont le propriétaire ne lui en a pas donné l'autorisation, ou encore à empêcher qu'un processus en cours d'exécution ne modifie une zone mémoire attribuée à un autre processus sans l'autorisation de celui-ci, par exemple.

Sommairement, on peut dire que la protection d'un objet informatique se pose dans les termes suivants, inspirés des concepts mis en oeuvre par certains systèmes d'exploitation, comme Multics et Linux :

- Un objet (processus, fichier, segment mémoire) a un propriétaire identifié, généralement l'utilisateur qui l'a créé.
- Le propriétaire d'un objet peut avoir conféré à lui-même et à d'autres utilisateurs des droits d'accès à cet objet. Les types de droits possibles sont notamment :
    - droit d'accès en consultation (lecture) ;
    - droit d'accès en modification (écriture, destruction, création) ;
    - droit d'accès en exécution ;
- À chaque objet est donc associée une liste de contrôle d'accès (access control list) qui énumère les utilisateurs autorisés et leurs droits.
- Avant toute tentative d'accès à un objet par un utilisateur, l'identité de cet utilisateur doit être authentifiée.
- Pour qu'un utilisateur ait le droit d'exécuter une action sur un objet, et dans un système informatique cette action est perpétrée par l'intermédiaire d'un processus, il faut en outre que le processus en question possède le *pouvoir* voulu. Le pouvoir est un attribut d'un processus, il peut prendre des valeurs qui confèrent à ce processus des *privilèges* plus ou moins étendus (mode user, mode supersuer).
- La valeur du pouvoir d'un processus peut changer au cours de son exécution. Ainsi un processus qui se déroule dans un mode utilisateur peut faire une demande d'entrée-sortie, ce qui nécessite le mode superviseur. Ceci sera résolu, sous Unix par exemple, par le mécanisme de l'appel système, qui transfère le contrôle, pour le compte du processus utilisateur, à une procédure du noyau qui va travailler en mode superviseur.
- O définit a notion de *domaine de protection* dans lequel s'exécute un processus comme l'ensemble des objets auxquels ce processus a accès et des opérations qu'il a le droit

d'effectuer sur ces objets. Lorsqu'un processus change de valeur de pouvoir, il change par là même de domaine de protection.

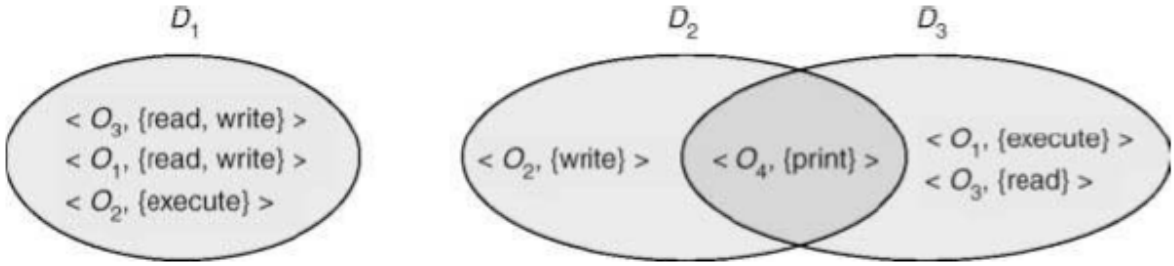
Les dispositifs et procédures de protection du système d'exploitation vont consister à faire respecter les règles qui découlent des droits et pouvoirs énumérés ci-dessus et à empêcher leur violation.

4.1.2 Domaines de protection :

Référence : Les systèmes d'exploitation, Silberschatz.

On définit un **droit d'accès** comme une paire : **(objet / ensemble de droits)**. Exemple (0_1, Read)
On définit un **domaine de protection** comme un ensemble de droits d'accès {DA_1, DA_2, … DA_n}

Les domaines de protection ne sont pas forcément disjoints :



La Liaison « processus / domaine » peut être statique ou dynamique :

- **statique** : l'ensemble de ressources disponibles fixe. Le principe de nécessité d'accès requiert un mécanisme de modification des contenus de domaines.
- **Dynamique** : requiert un mécanisme de « commutation de domaine » (pas nécessairement de modification).

Un domaine peut être soit : un utilisateur, un processus ou une procédure.

- Domaine = utilisateur
  les objets auxquels on peut accéder dépendent de l'utilisateur qui y accède. La commutation de domaine est liée au changement d'utilisateur.
- Domaine = processus
  les objets auxquels on peut accéder dépendent du processus qui y accède. La commutation de domaine est liée à la commutation de contexte.
- Domaine = utilisateur
  les objets auxquels on peut accéder correspondent aux variables utilisées par la procédure. La commutation de domaine est liée à chaque appel de procédure.

4.1.3 Matrices de droits d'accès :

Les matrices des droits d'accès permettent de formaliser le domaine de protection de chaque processus.

Les lignes de la matrice représentent les domaines, et les colonnes représentent les objets.

Exemple :

	O1	O2	O3	O5
D1	Lecture		lecture	
D2				Impression
D3		lecture	Exécution	
D4	Lecture écriture		Lecture écriture	

L'implémentation des matrices de droits d'accès, peut se faire selon plusieurs méthodes :

La table globale : consiste à représenter la matrice par une table constituée d'un ensemble de triplets « domaine, objet, ensemble de droits » . Ainsi , à chaque fois que l'on exécute une opération A sur un objet, il faut rechercher dans la table des triplets si A y figure bien . Dans l'affirmative, l'opération est autorisée, sinon elle est refusée.

Liste d'accès aux objets : On implémente chaque colonne de la matrice par une liste chainée.

Liste des domaines : On implémente chaque ligne de la matrice par une liste chainée.

4.2 SECURITE:

4.2.1 Introduction :

Les problèmes techniques actuels de sécurité informatique découlent directement ou indirectement de l'essor des réseaux, qui multiplie la quantité et la gravité des menaces potentielles. Ces menaces peuvent être : atteinte à la disponibilité des systèmes et des données, destruction de données, corruption ou falsification de données, vol ou espionnage de données, usage illicite d'un système ou d'un réseau, usage d'un système compromis pour attaquer d'autres cibles.

Les personnes qui accèdent à une ressource non publique doivent être *identifiées* ; leur identité doit être *authentifiée* ; leurs droits d'accès doivent être *vérifiés*.

La sécurité des accès par le réseau à une ressource protégée n'est pas suffisamment garantie par la seule identification de leurs auteurs. Sur un réseau local de type Ethernet ou sur Internet, il est possible à un tiers de capter la transmission de données. Les données doivent donc être protégées, grâce aux techniques de chiffrement ou *cryptage*.

4.2.2 Authentification :

L'authentification est la procédure qui consiste, pour un système informatique, à vérifier l'identité d'une entité (personne, ordinateur...), afin d'autoriser l'accès de cette entité à des ressources (systèmes, réseaux, applications...).

Le protocole d'authentification peut appartenir à l'une des familles suivantes :

- *L'authentification simple* : l'authentification ne repose que sur un seul élément ou « facteur » (exemple : l'utilisateur indique son mot de passe).
- *L'authentification forte* : l'authentification repose sur deux facteurs ou plus.

- *L'Authentification unique* : (ou identification unique ; en anglais Single Sign-On ou SSO) est une méthode permettant à un utilisateur de ne procéder qu'à une seule authentification pour accéder à plusieurs applications informatiques (ou sites internet sécurisés).

L'authentification peut se baser sur :

- La vérification d'un « mot de passe » préalablement établi.
- Une « vérification biométrique » (reconnaissance de la voix, empreintes digitales, l'iris, …etc).

4.2.3 Chiffrement :

Le chiffrement, ou cryptage est le procédé grâce auquel on souhaite rendre la compréhension d'un document impossible à toute personne qui n'a pas la clé de déchiffrement.

La sécurité d'un système de chiffrement repose plus sur le secret de la clé de chiffrement que sur l'algorithme lui-même.

Un système de chiffrement est dit :

- *symétrique* quand il utilise la même clé pour chiffrer et déchiffrer.
- *asymétrique* quand il utilise des clés différentes : une paire composée d'une clé publique, servant au chiffrement, et d'une clé privée, servant à déchiffrer. Le point fondamental soutenant cette décomposition publique/privée est l'impossibilité calculatoire de déduire la clé privée de la clé publique.

Les méthodes les plus connues sont le DES, le Triple DES et l'AES pour la cryptographie symétrique, et le RSA pour la cryptographie asymétrique, aussi appelée cryptographie à clé publique.

L'utilisation d'un système symétrique ou asymétrique dépend des tâches à accomplir. La cryptographie asymétrique présente deux intérêts majeurs : elle supprime le problème de transmission sécurisée de la clé, et elle permet la signature électronique. Elle ne remplace cependant pas les systèmes symétriques car ses temps de calcul sont nettement plus longs.