

# Sécurité des systèmes d'exploitation

Contrôle d'accès et contrôle de flux d'information

# Plan

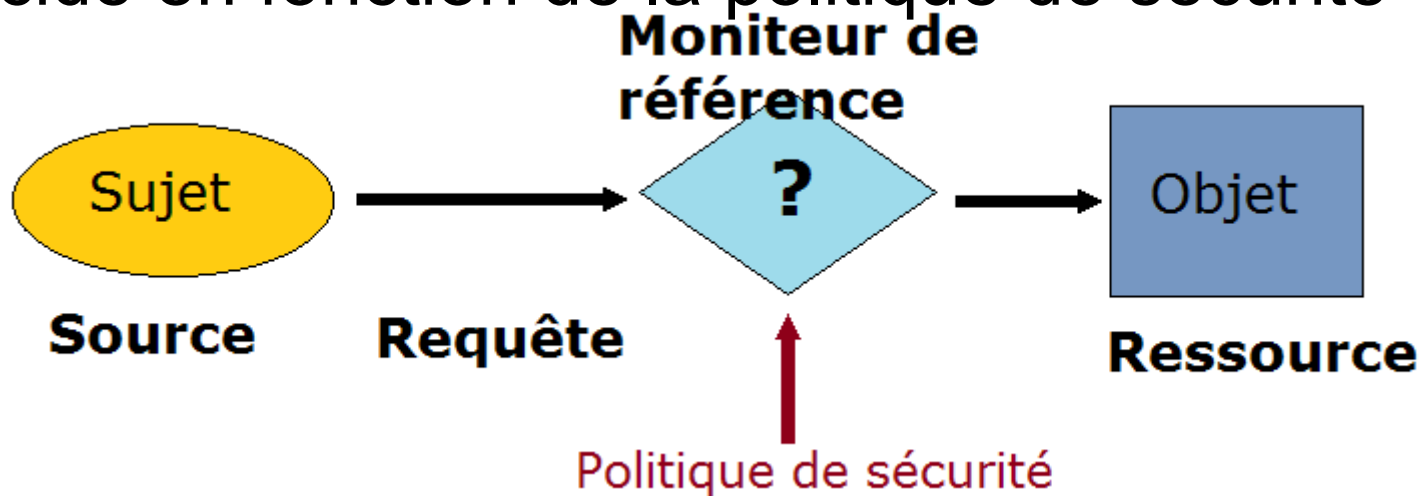
- Contrôle d'accès
- Contrôle de flux d'information
- Contrôle d'accès des systèmes Linux
- Contrôle d'accès des systèmes Windows
- SELinux
- Windows Mandatory Integrity Controls

# Contrôle d'accès

- Se retrouve à plusieurs niveaux :
  - Matériel (gestion de la mémoire) ;
  - Système d'exploitation (système de fichiers) ;
  - Applicatif (bases de données) ;
  - Réseau (pare-feu) ;
  - Physique (accès à un local).
- On parle de mécanisme de contrôle d'accès (ce qui le réalise), mais aussi de modèle de contrôle d'accès

# Contrôle d'accès

- Le modèle générique comprend plusieurs éléments :
  - Des objets ou des ressources ;
  - Des requêtes ;
  - Des sources de requêtes appelées principal ou sujet ;
  - Un moniteur de référence qui traite les requêtes et décide en fonction de la politique de sécurité



# Contrôle d'accès

- Exemples de sujets
  - utilisateur ;
  - programme ;
  - ordinateur ;
  - une combinaison des précédents.
- Moniteur de référence doit satisfaire 3 propriétés
  - Médiation complète : tous les accès aux objets sont surveillés et contraints
  - Protection : ses fonctions ne doivent pas pouvoir être modifiées par un attaquant
  - Comportement sain et prouvé : il doit forcer l'application loyale de la politique de sécurité

# Contrôle d'accès

- La définition des opérations implique des choix importants. En particulier, certaines opérations doivent être regroupées dans une « méta-opération »
- Par exemple :
  - Lire le dossier d'un patient,
  - Écrire un enregistrement dans le journal (à des fins d'audit).
- Un sujet peut être autorisé à effectuer une « méta-opération » mais pas chacune des opérations la composant

# Contrôle d'accès

- Les objets peuvent inclure
  - des blocs de disque ;
  - des fichiers ;
  - des tables, des lignes ou des colonnes de bases de données ;
  - des enregistrements de niveau applicatif comme des entrées de calendrier.
- Définir les objets est une part importante de la conception d'un système de contrôle d'accès

# Contrôle d'accès

- Matrice de contrôle d'accès [Lampson, 1971]

Objets Sujets	Système d'exploitation	Programme 1	Fichier 1	Fichier 2
Utilisateur1	rwX	rw	r	r
Utilisateur2	rx	rx	-	w
Utilisateur3	rx	-	r	r



# Contrôle d'accès

Objets	Système d'exploitation	Programme 1	Fichier 1	Fichier 2
Sujets				
Utilisateur1	rx	rw	r	r
Utilisateur2	rx	rx	-	w
Utilisateur3	rx	-	r	r

- Liste de contrôle d'accès (ou ACL pour Access Control List) : une colonne d'une matrice de contrôle d'accès, attachée à un objet
- Une ACL indique quels sujets peuvent accéder à un objet donné
- La revue d'une ACL peut être simple si elle est compacte
- Révoquer les droits d'accès d'un sujet peut être pénible (il faut balayer tous les objets)

# Contrôle d'accès

Objets	Système d'exploitation	Programme 1	Fichier 1	Fichier 2
Sujets				
Utilisateur1	rx	rw	r	r
Utilisateur2	rx	rx	-	w
Utilisateur3	rx	-	r	r

- Capacité : une paire (objet, opération) pour un sujet donné, cela signifie que le sujet peut réaliser cette opération sur l'objet.
- Une liste de capacités est une ligne de la matrice
- Une capacité ne doit pas pouvoir être forgée par un sujet
- Les capacités sont souvent transmissibles (facilite la délégation)
  - Rend la révocation plus difficile quand une capacité a été transmise à de multiples processus

# Contrôle d'accès

- Les sujets peuvent être organisés dans des groupes
  - Le contrôle d'accès peut être fait sur des groupes plutôt que individuellement par sujet
- Utilisés comme des niveaux d'indirections dans le contrôle d'accès.
  - par ex., le membre d'un groupe  $G$  peut accéder à un fichier  $f$  ;

# Contrôle d'accès

- Contrôle d'accès discrétionnaire (ou DAC pour *Discretionary Access Control*)
- Chaque sujet peut détenir un droit de possession sur un objet
  - Par ex., le propriétaire d'un fichier (souvent le créateur de l'objet) peut d'ajouter ou soustraire des droits d'accès pour lui ou pour les autres
- Ce modèle ne peut pas empêcher des sujets de faire des erreurs.
- Il est compliqué de forcer l'application d'une politique de sécurité au niveau d'un système entier.

# Contrôle d'accès

- Contrôle d'accès basé sur les rôles (ou RBAC pour *Role-Based Access Control*)
  - Définition des différents rôles possibles en fonction du contexte de l'organisation
  - Attribution des droits d'accès et permissions par rôle
  - Chaque utilisateur a accès à une liste de rôle, selon son activité, via un mécanisme de sessions
- Différence entre rôle (fonction dans une organisation) et groupe (ensemble d'entités)
  - l'ensemble des sujets assignés à un même rôle forme un groupe ;
  - mais un groupe peut être formé d'entités avec des rôles différents

# Contrôle d'accès

- Contrôle d'accès obligatoire (ou MAC pour *Mandatory Access Control*)
- Une entité tierce assigne des attributs de sécurité pour les sujets et les objets.
  - Par ex., les objets peuvent être « travail » et « loisir », et les sujets peuvent être « de confiance » ou « invité »
  - Implique que la politique de sécurité ne puisse pas être modifiée par les utilisateurs du système
- Ces attributs contraignent les accès
  - Par exemple, les sujets « invité » ne peuvent pas modifier des objets « travail »

# Contrôle des flux d'information

- Notion de sécurité multi-niveau (ou MLS pour *Multi-level security*)
- Les niveaux de sécurité peuvent se rapporter aux propriétés de :
  - Confidentialité
    - obtenue par des services de chiffrement ou de contrôle d'accès
      - La détection de la perte de confidentialité n'est pas évidente et dépend fortement du contexte
  - Intégrité
    - obtenue en maîtrisant les tentatives de modification
      - La détection de la perte d'intégrité peut être constatée via un motif d'intégrité (somme de contrôle, condensat cryptographique)

# Contrôle des flux d'information

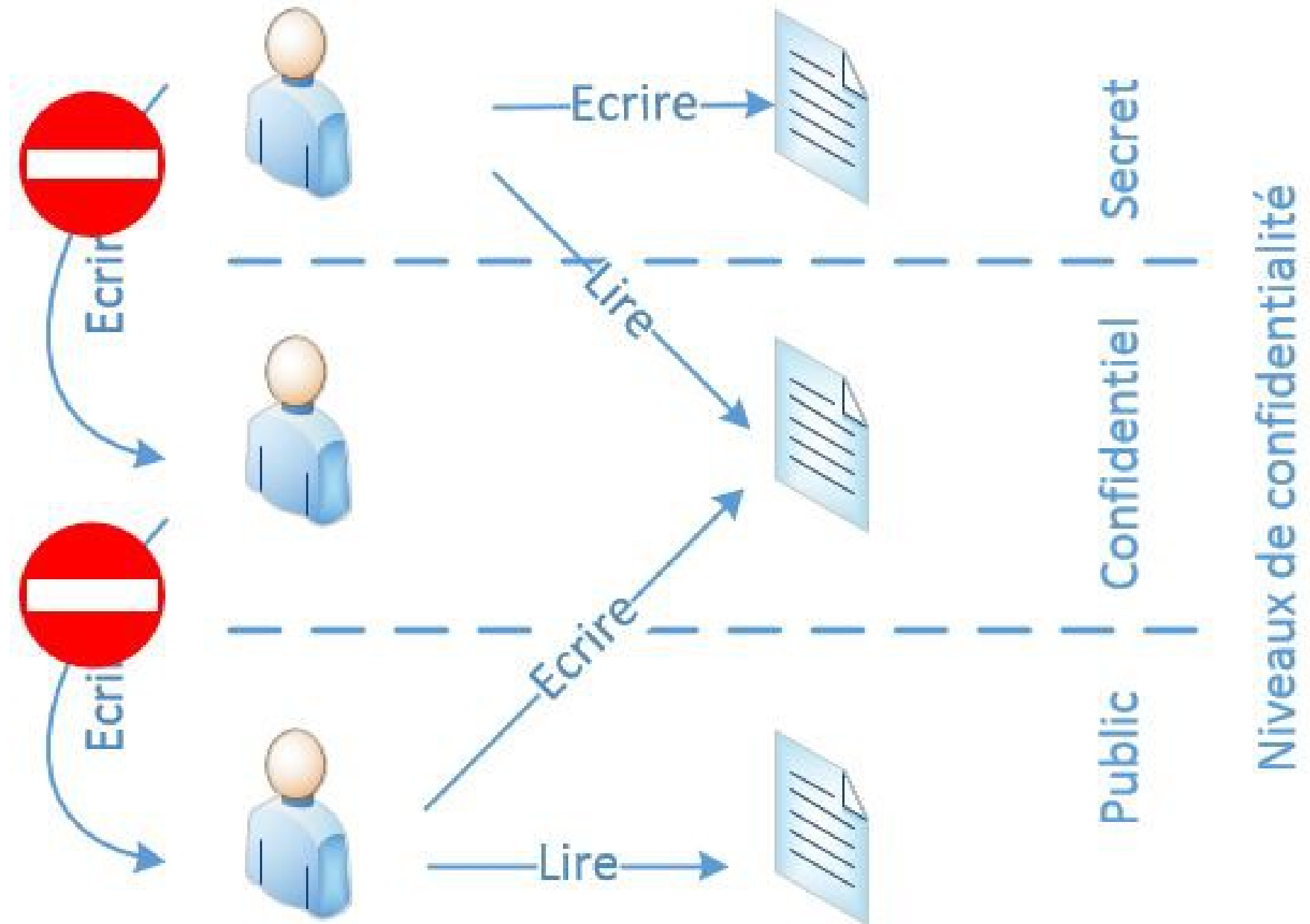
- Les niveaux de sécurité impliquent de définir une classification avec une relation d'ordre
  - Par ex. : public < confidentiel < secret
- Ils s'appliquent aux sujets et aux objets
  - Reçoivent des étiquettes (*label*)
- Le contrôle d'accès obligatoire est souvent associé à des niveaux de sécurité
  - Attribut de sécurité MAC  $\Leftrightarrow$  *label*



# Contrôle des flux d'information

- Modèle **Bell LaPadula** (1973)
  - Traite la confidentialité des informations
- 2 propriétés :
  - propriété de sécurité simple (« No read-up ») : un sujet avec un niveau de sécurité donné ne peut pas lire un objet avec un niveau de sécurité supérieur.
  - propriété \* (« No write-down ») : un sujet avec un niveau de sécurité donné ne peut pas écrire dans un objet avec un niveau de sécurité inférieur.
- Niveaux de sécurité (simplifiés)
  - Niveau de classification pour les objets
  - Niveau d'habilitation pour les sujets

# Contrôle des flux d'information



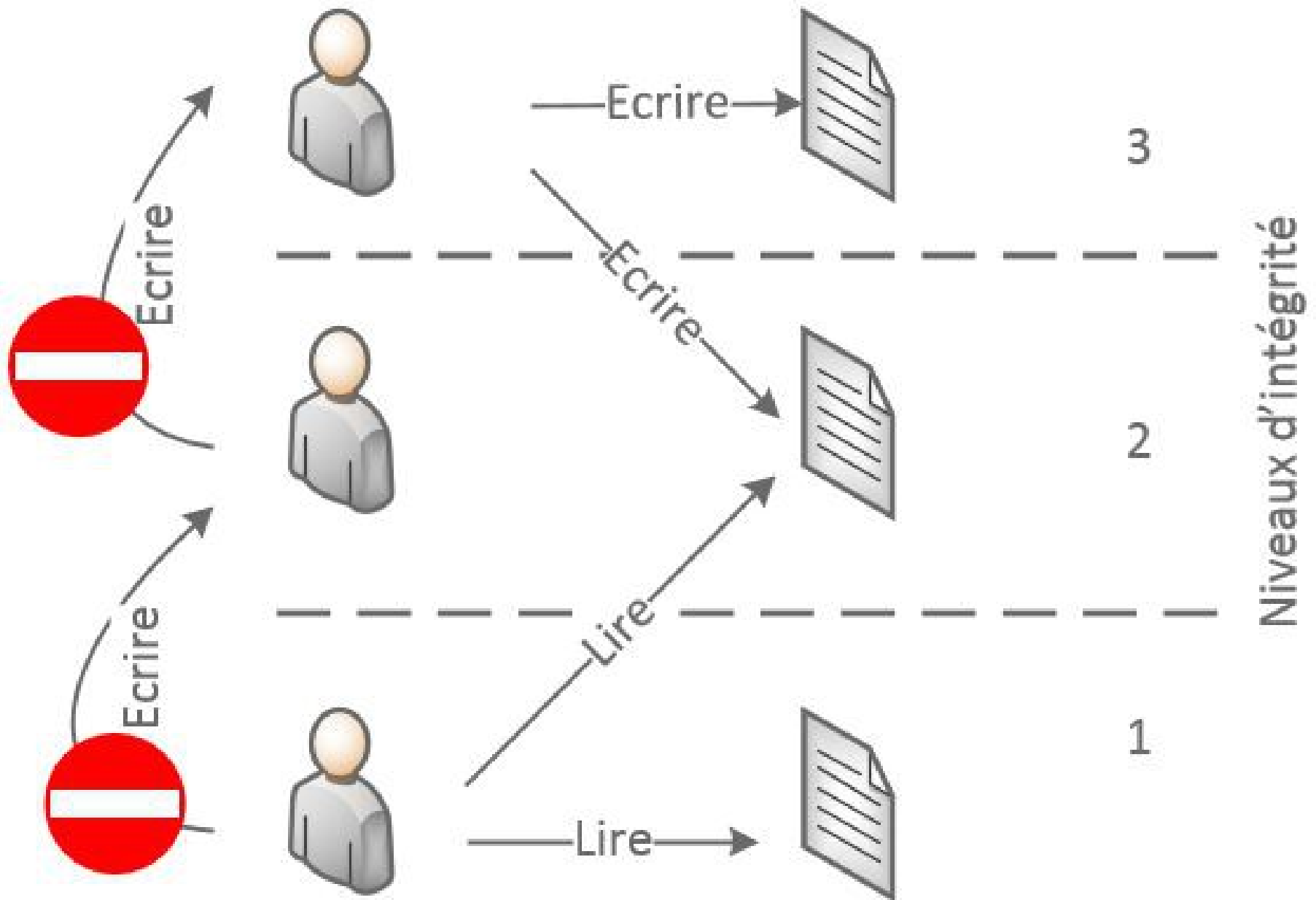
# Contrôle des flux d'information

- Critiques du modèle Bell LaPadula
  - Le niveau d'une information ne peut que croître
    - si une information publique est utilisée par un sujet habilité au secret, tout objet modifié par ce sujet avec cette information sera classifié secret.
  - La croissance du niveau induite par le modèle nécessite d'introduire une procédure de déclassification (par un officier de sécurité ou par un processus de confiance) n'obéissant pas aux règles du modèle
  - Difficile à implémenter tel quel
    - S'applique à un modèle abstrait de système d'exploitation

# Contrôle des flux d'information

- Modèle Biba (1975)
  - Traite l'intégrité des informations et des systèmes
- Présentation simplifiée (limitée à la politique d'intégrité stricte)
  - 2 propriétés miroir de Bell LaPadula :
    - propriété de sécurité simple (« No write-up ») : un sujet avec un niveau de sécurité donné ne peut pas modifier (écrire) un objet avec un niveau de sécurité supérieur.
    - propriété \* (« No read-down ») : un sujet avec un niveau de sécurité donné ne peut pas observer (lire) un objet avec un niveau de sécurité inférieur.
- Relation d'invocation : définit la capacité d'un sujet à invoquer un autre sujet

# Contrôle des flux d'information



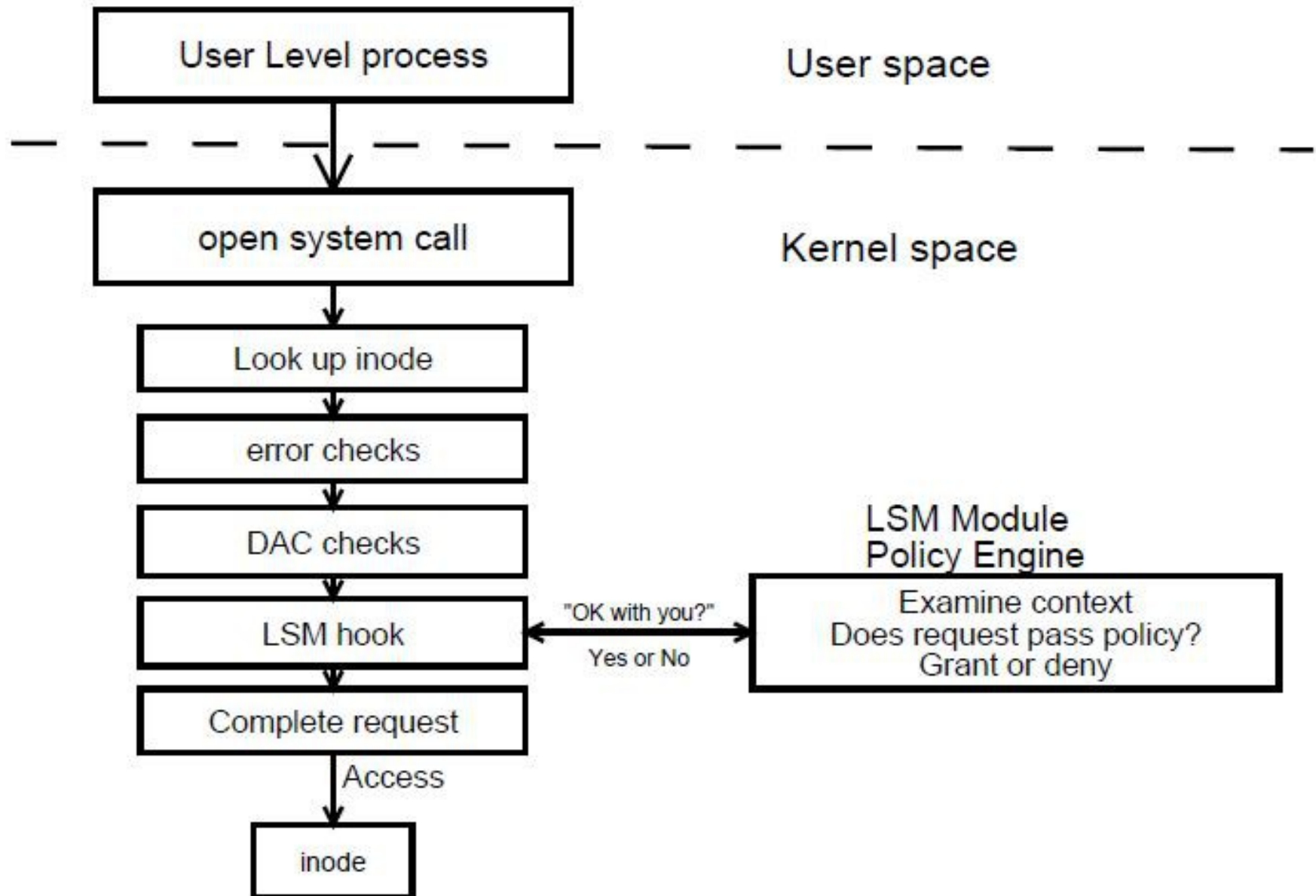
# Contrôle des flux d'information

- Critiques du modèle Biba
  - Les sujets tendent à se trouver vers le niveau d'intégrité le plus bas
    - Un sujet doit migrer vers un niveau d'intégrité inférieur pour accéder à un objet de niveau inférieur
  - Difficile à implanter tel quel dans un système d'exploitation
    - Comment le noyau (niveau d'intégrité haut) peut chercher une mise à jour au travers du réseau (niveau d'intégrité bas) ?

# Contrôle d'accès des systèmes Linux

- Contrôle d'accès se fait sur la base de l'UID du sujet
- Les processus possèdent des attributs UID, notamment :
  - UID
    - Identifiant du propriétaire du processus
  - EUID (effective user ID)
    - Identifiant utilisé pour lors des contrôles de permission et qui peut être différent de l'identifiant du propriétaire
  - Le même concept s'applique aux groupes

# Contrôle d'accès des systèmes Linux



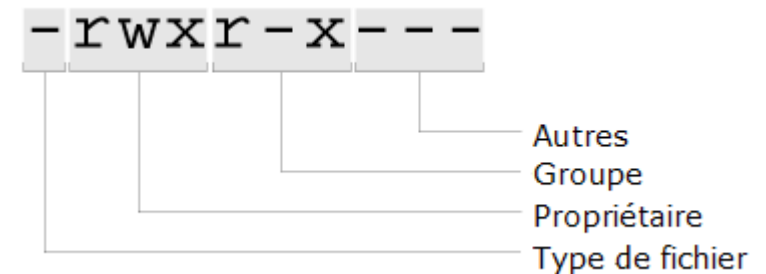


# Contrôle d'accès des systèmes Linux

- Permissions UNIX

- Fichiers (type « - »)

- r : accès en lecture
    - w : accès en écriture
    - x : exécuter



- Répertoires (type « d »)

- r : lister les fichiers
    - w : insérer ou retirer des fichiers
    - x : déterminer le statut du contenu d'un répertoire (par ex., les permissions, le propriétaire, la taille, ...), en faire le répertoire courant (commande `cd`) et ouvrir des fichiers

- Représentation octale

- $r = 4, w = 2, x = 1 \Rightarrow$  « `rwxr-x---` » correspond à 750

# Contrôle d'accès des systèmes Linux

- Permissions spéciales

`-rwSr-Sr-T`

Sticky  
SGID  
SUID

- Fichiers exécutables

- S (SUID) : le processus qui exécute le programme a un EUID identique au propriétaire du programme
    - S (SGID) : le processus qui exécute le programme a un EGID identique au groupe du programme
    - T (Sticky) : ignoré par le noyau Linux

- Répertoire :

- S (SGID) : hérite le groupe du répertoire
    - T (Sticky) : les fichiers du répertoire peuvent être supprimés ou renommés uniquement par le propriétaire (ou root)

# Contrôle d'accès des systèmes Linux

- Problématique avec les binaires SUID (ou SGID) appartenant à root
  - L'EUID vaut 0 donc le programme a accès à tous les privilèges du super-utilisateur.
- Les options de montage de systèmes de fichiers peuvent avoir un impact sur les permissions :
  - noexec : empêche une exécution directe des binaires du système de fichiers
  - nosuid : empêche la prise en compte des bits SUID et SGID (\*)
  - ro : monte le système de fichier en lecture seule

# Contrôle d'accès des systèmes Linux

- ACL POSIX

- Option du noyau, doit avoir été pris en compte lors de sa compilation

- Par ex. sur un système CentOS 7

```
[root@centos7 ~]# grep ACL /boot/config-3.10.0-229.20.1.el7.x86_64
CONFIG_EXT4_FS_POSIX_ACL=y
CONFIG_XFS_POSIX_ACL=y
CONFIG_BTRFS_FS_POSIX_ACL=y
CONFIG_FS_POSIX_ACL=y
CONFIG_GENERIC_ACL=y
CONFIG_TMPFS_POSIX_ACL=y
CONFIG_NFS_V3_ACL=y
CONFIG_NFSD_V2_ACL=y
CONFIG_NFSD_V3_ACL=y
CONFIG_NFS_ACL_SUPPORT=m
CONFIG_CIFS_ACL=y
```

- Se manipule avec des utilitaires spécifiques (getfacl & setfacl) :

- L'ajout des utilisateurs ou des groupes se fait avec des permissions « rwx »
  - Permet de définir des permissions par défaut (mécanisme d'héritage)
  - Se détecte avec ls (le +)

```
[superviseur@centos7 test-acl]$ ls -l
total 4
-rw-rw-r--. 1 superviseur superviseur 0 3 janv. 17:32 file
-rw-rw-r--+ 1 superviseur superviseur 0 3 janv. 17:32 file1
-rw-rw-r--. 1 superviseur superviseur 0 3 janv. 17:32 file2
```

# Contrôle d'accès des systèmes Linux

- Certains programmes nécessitent un des privilèges du super-utilisateur (uid 0) pour fonctionner mais rarement l'intégralité
  - Tout ou rien => absence de granularité
- Les capacités (*capabilities*) Linux consistent en un découpage de l'ensemble de ces privilèges
- Apparues dans le noyau 2.2
- Le noyau 2.6.24 a apporté un support des capacités au niveau système de fichiers (attribut étendu *security.capability*) via VFS
  - utilitaires setcap et getcap pour manipuler les capacités au niveau du système de fichiers

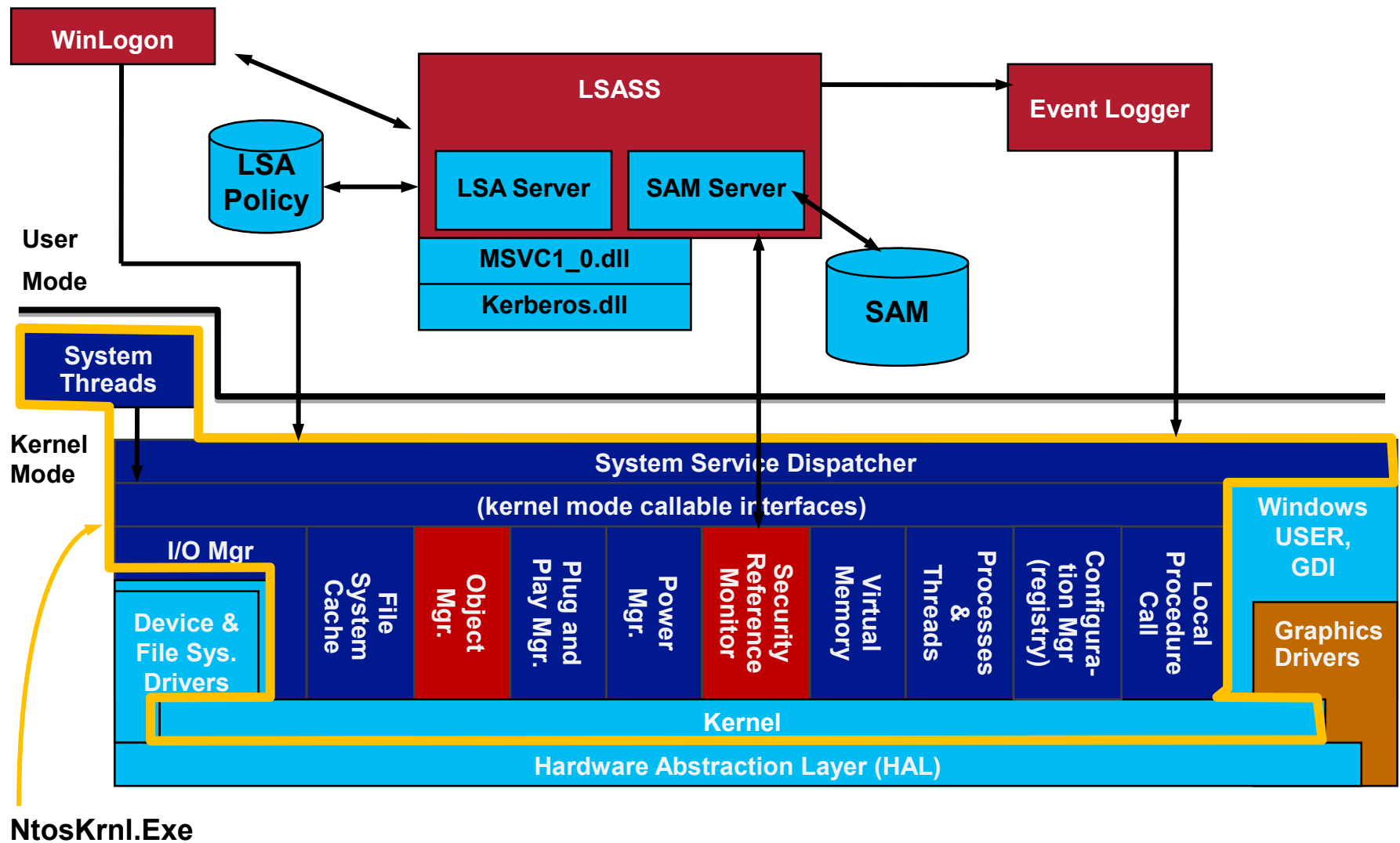
# Contrôle d'accès des systèmes Linux

SSE - Contrôle d'accès

Capacité	Version de noyau
CAP_CHOWN, CAP_DAC_OVERRIDE, CAP_DAC_READ_SEARCH, CAP_FOWNER, CAP_FSETID, CAP_KILL, CAP_SETGID, CAP_SETUID, CAP_SETPCAP, CAP_LINUX_IMMUTABLE, CAP_NET_BIND_SERVICE, CAP_NET_BROADCAST, CAP_NET_ADMIN, CAP_NET_RAW, CAP_IPC_LOCK, CAP_IPC_OWNER, CAP_SYS_MODULE, CAP_SYS_RAWIO, CAP_SYS_CHROOT, CAP_SYS_PTRACE, CAP_SYS_PACCT, CAP_SYS_ADMIN, CAP_SYS_BOOT, CAP_SYS_NICE, CAP_SYS_RESOURCE, CAP_SYS_TIME, CAP_SYS_TTY_CONFIG	2.2
CAP_MKNOD, CAP_LEASE	2.4
CAP_AUDIT_WRITE, CAP_AUDIT_CONTROL	2.6.11
CAP_SETFCAP	2.6.24
CAP_MAC_OVERRIDE, CAP_MAC_ADMIN	2.6.25
CAP_SYSLOG (modifie CAP_SYS_ADMIN)	2.6.37
CAP_WAKE_ALARM	3.0
CAP_BLOCK_SUSPEND	3.5
CAP_AUDIT_READ	3.16

# Contrôle d'accès des systèmes Windows

- Composants de sécurité Windows



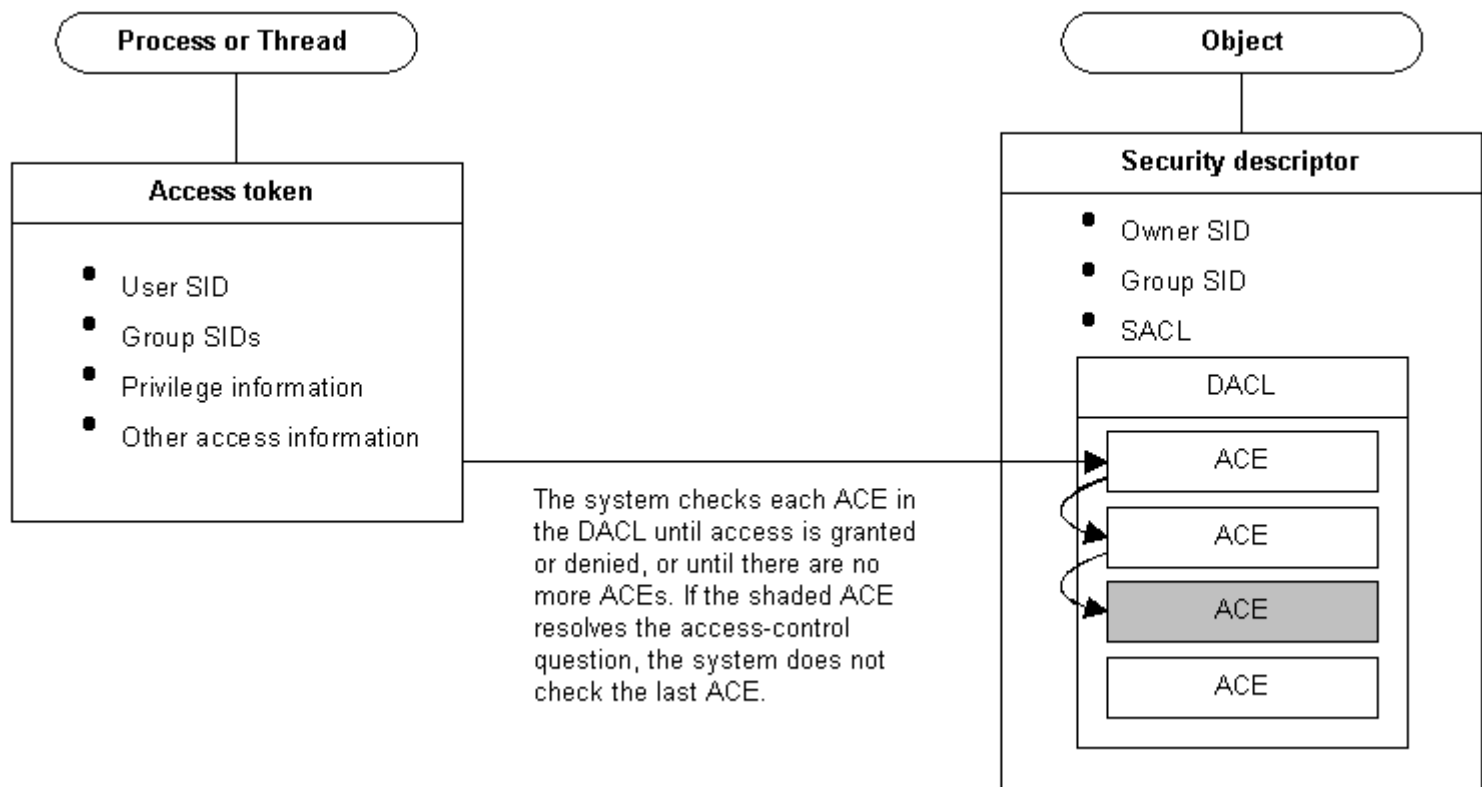
# Contrôle d'accès des systèmes Windows

- Le « Security reference monitor » (SRM) est un composant du noyau, en charge de :
  - définir les jetons d'accès ;
  - réaliser les contrôles d'accès sur les objets ;
  - manipuler les privilèges utilisateurs ;
  - générer des messages dans les journaux de sécurité.
- Pour le contrôle d'accès, la routine utilisée par le SRM est **SeAccessCheck**
  - Tient compte notamment du descripteur de sécurité de l'objet cible, du contexte de sécurité du sujet, du niveau d'accès demandé et des privilèges détenus



# Contrôle d'accès des systèmes Windows

- Le système compare les informations du jeton d'accès avec les informations de sécurité du descripteur de sécurité.



Source : MSDN

# Contrôle d'accès des systèmes Windows

- Un jeton d'accès est un objet qui décrit un contexte de sécurité, avec notamment :
  - le SID du compte utilisateur ;
  - les SID des groupes dont l'utilisateur est membre ;
  - une liste de privilèges portés par l'utilisateur ou les groupes de l'utilisateur ;
  - le SID du groupe primaire ;
  - la source du jeton...

# Contrôle d'accès des systèmes Windows

- Un descripteur de sécurité contient certaines informations de sécurité, notamment :
  - des identifiants de sécurité (utilisateur et groupe primaire) ;
  - une ACL discrétionnaire ;
  - une ACL système (informations relatives à la journalisation).
- Les « Securable Objects » sont des objets qui ont un descripteur de sécurité, par ex. :
  - des fichiers ou des répertoires (NTFS ou ReFS) ;
  - des processus ;
  - des clés de registres ...

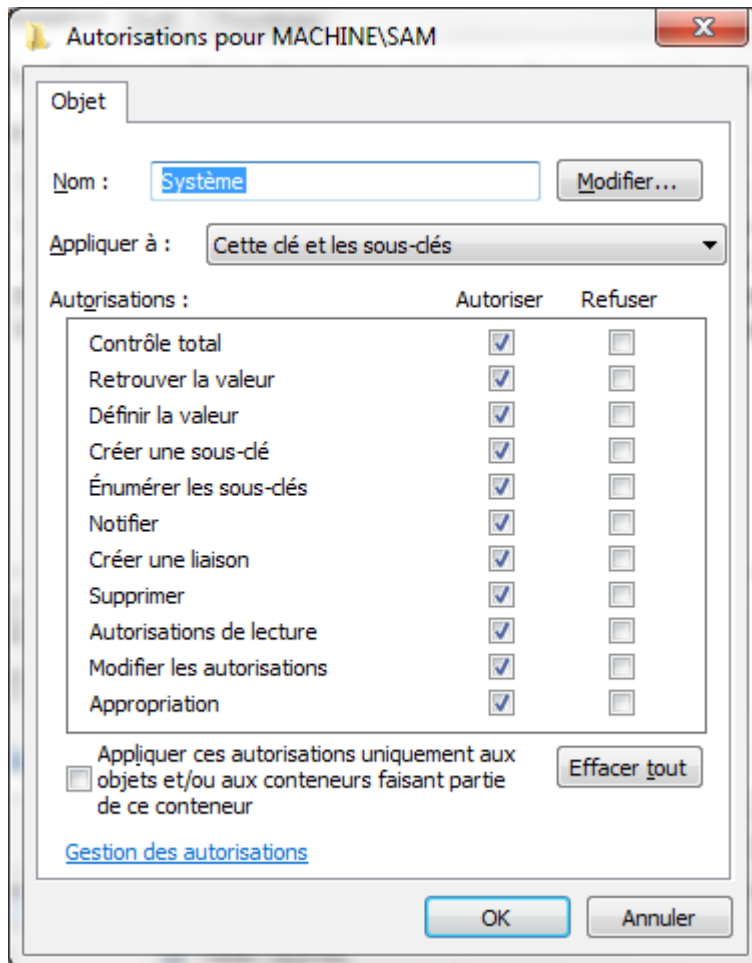
# Contrôle d'accès des systèmes Windows

- Une ACL contient une liste d'entrées de contrôle d'accès (ACE). Chaque ACE spécifie un ensemble de droits d'accès et contient un SID pour qui ces droits sont autorisés, refusés ou audités.
  - Les droits d'accès varient en fonction des objets ciblés, ils contiennent au minimum
    - DELETE
    - READ\_CONTROL
    - SYNCHRONIZE
    - WRITE\_DAC
    - WRITE\_OWNER

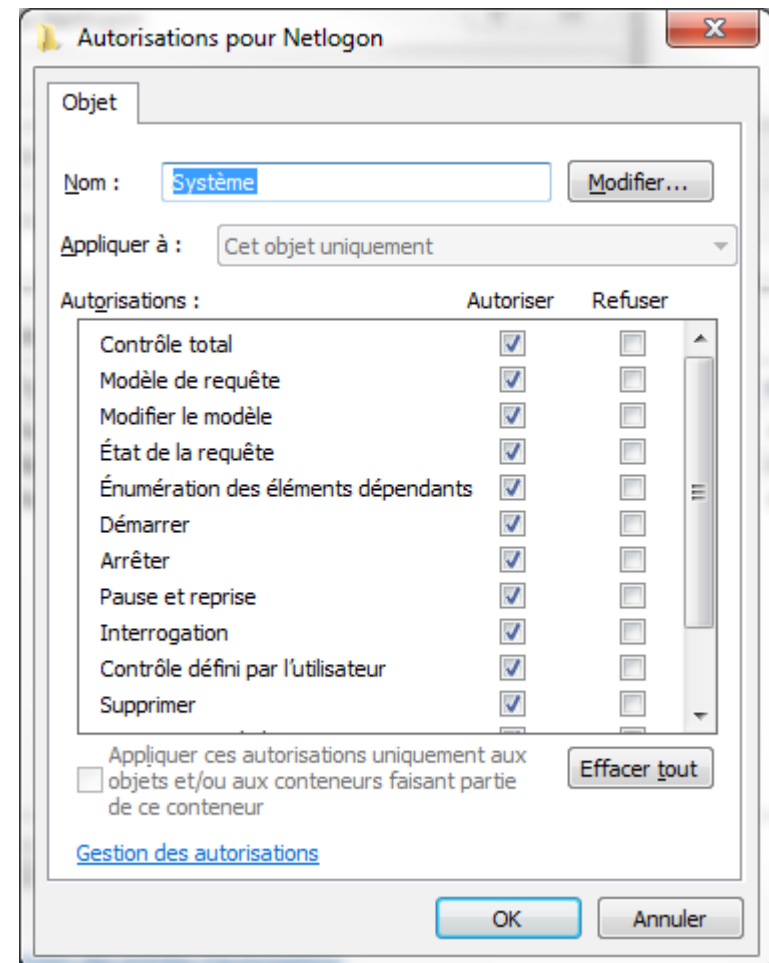
# Contrôle d'accès des systèmes Windows

- Exemple de droits d'accès

## Registre



## Service



# Contrôle d'accès des systèmes Windows

- Droits Windows
- Se composent en deux catégories
  - 10 droits d'ouverture de session :  
*SeNetworkLogonRight, SeInteractiveLogonRight, SeRemoteInteractiveLogonRight, SeDenyNetworkLogonRight, ...*
  - 34 privilèges : *SeTrustedCredManAccessPrivilege, SeTcbPrivilege, SeMachineAccountPrivilege, ...*
    - Parmi ces 34, il existe 6 privilèges très sensibles qui peuvent permettre d'obtenir des droits complets d'administrateur : *SeCreateTokenPrivilege, SeDebugPrivilege, SeLoadDriverPrivilege, SeRestorePrivilege, SeTakeOwnershipPrivilege, SeTcbPrivilege*

# SE Linux

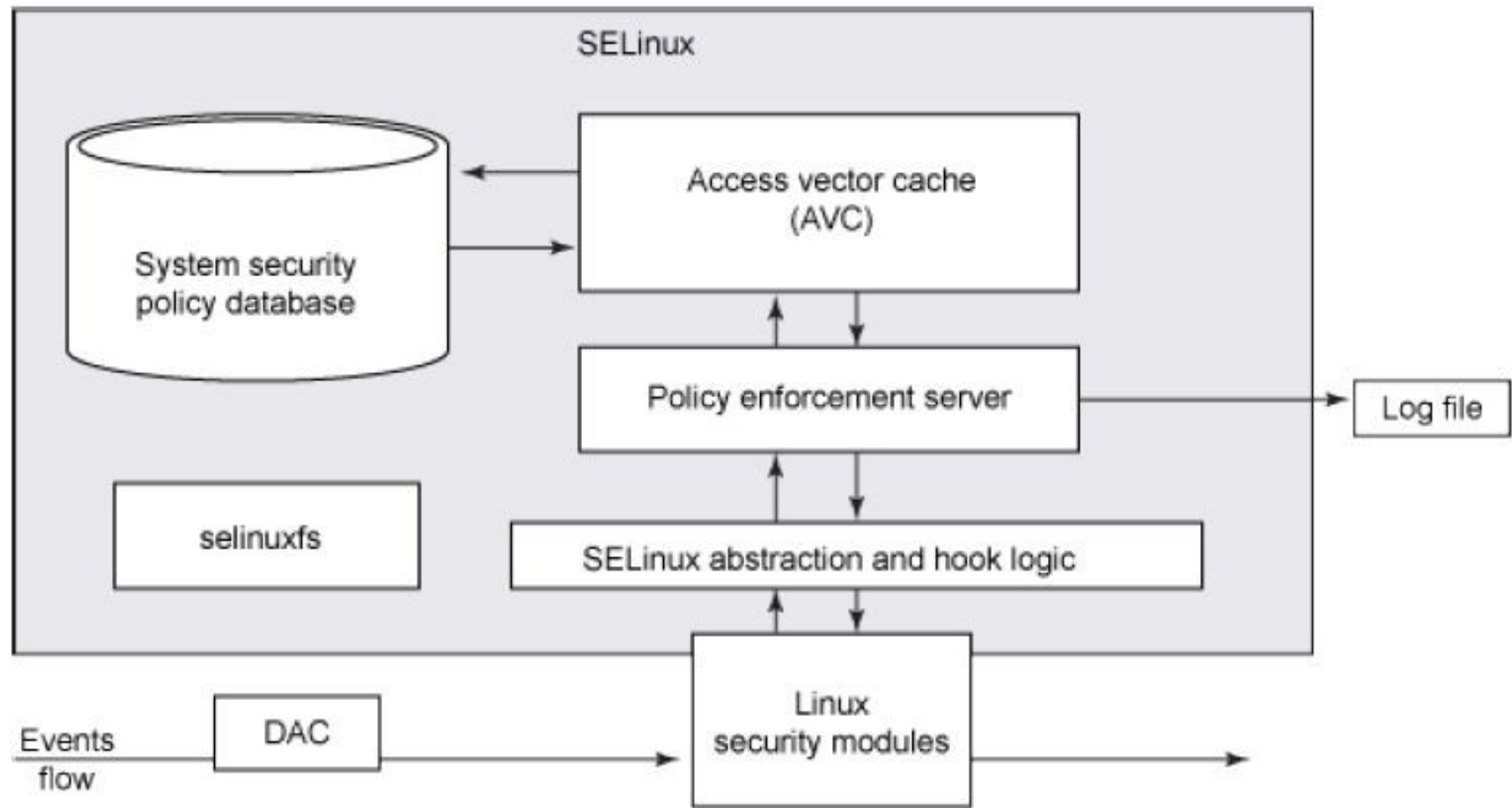
- *Security Enhanced Linux*
  - Issu d'un projet de la NSA
  - Système de contrôle d'accès obligatoire (MAC)
    - Ajoute également des mécanismes RBAC permet la mise en place de politiques de sécurité multi-niveau (MLS) et multi-catégorie (MCS)
- Présent nativement depuis le noyau 2.6 (2003) via l'interface LSM (*Linux Security Modules*)

# SE Linux

- Objectif des LSM : fournir des mécanismes de contrôle d'accès génériques pour le noyau Linux via :
  - l'intégration des nouveaux points d'ancrage dans les parties critiques du noyau (*Security Hooks*)
    - Appels systèmes de gestion des processus, fichiers, *sockets*, mémoire partagée, IPC
    - Interviennent après les vérifications liées au contrôle d'accès standard du noyau Linux => les permissions normales s'appliquent toujours
  - le support de l'empilement des modules de sécurité
    - Dépend des modules considérés car les structures noyau critiques (par ex., *task\_struct*, inode, net\_device) contiennent un champ *security* utilisé pour les attributs de sécurité propre à chaque module



# SE Linux



# SE Linux

- 2 modes de fonctionnement
  - *Enforcing* : la politique SELinux est appliquée, SELinux refuse l'accès en fonction des règles de la politique SELinux.
  - *Permissive* : la politique SELinux n'est pas appliquée, SELinux ne refuse pas l'accès
    - les refus sont enregistrés pour les actions qui auraient été refusées si elles étaient exécutées en mode *enforcing*
- Consulter le mode en cours, au choix :
  - `$ cat /etc/selinux/config`
  - `$ cat /sys/fs/selinux/enforce`
  - `$ getenforce`
  - `$ sestatus`

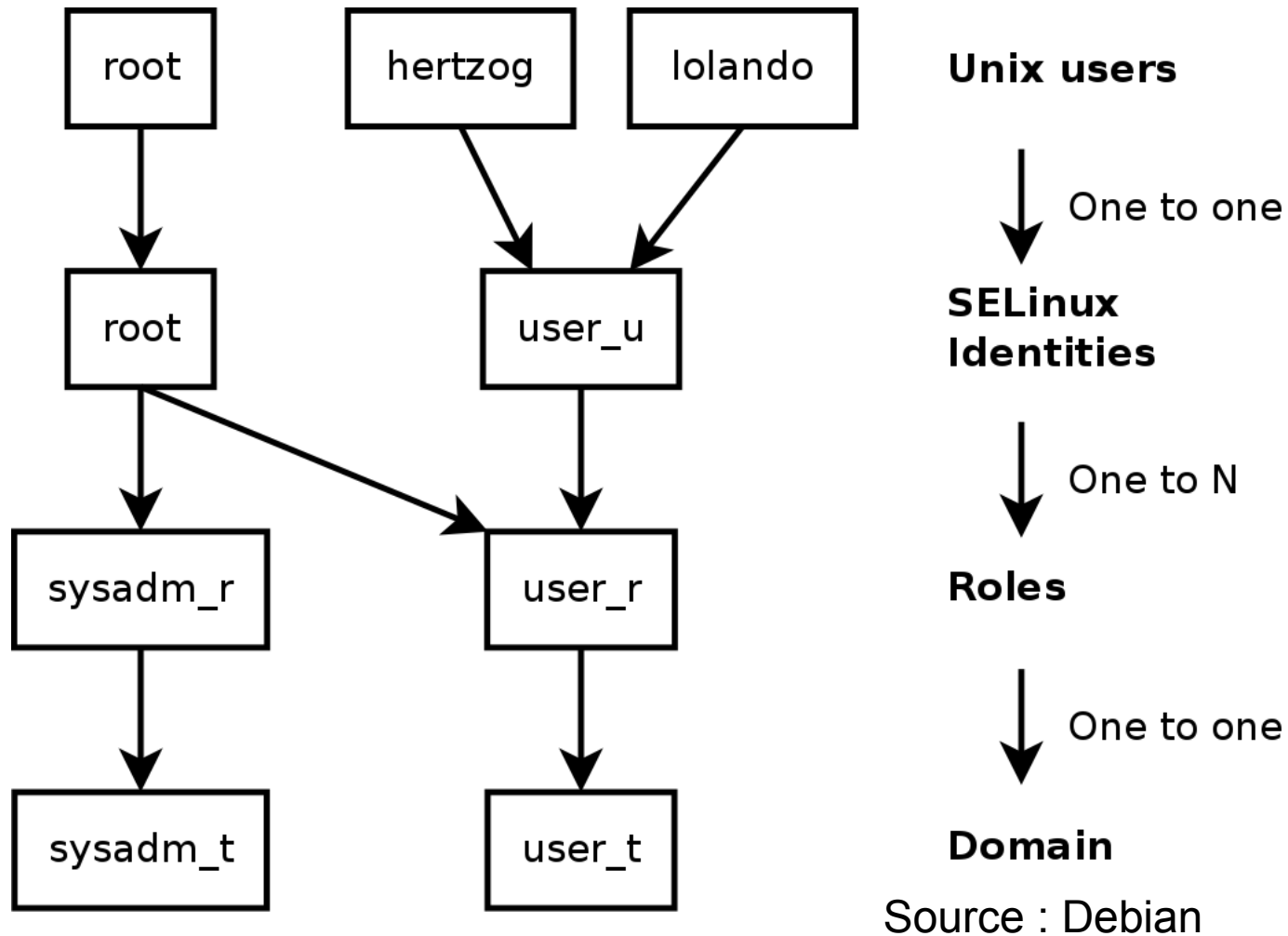
# SE Linux

- Modèle DTE (*domain and type enforcement*)
  - mécanisme générique permettant d'implanter diverses politiques de sécurité
    - chaque processus (i.e. rattaché à un sujet) est rattaché à un contexte de sécurité défini appelé domaine ;
    - à chaque objet (fichier, sockets, etc.) est associé un type.
  - Le contrôle d'accès ne se fait pas sur les sujets et les objets mais sur les domaines et les types
  - efficace pour
    - Confiner (i.e. limiter à un ensemble d'actions autorisées) des processus, notamment les processus privilégiés
    - Contrôler quels programmes ont accès aux ressources sensibles, et empêcher l'accès par tout autre programme

# SE Linux

- Format des contextes de sécurité  
`<ID>:<role>:<type>:<level>`
  - ID : identifiant spécifique à SELINUX
    - UID peut changer (par ex. bit setuid activé), mais l'ID SELINUX ne change pas, et pointe toujours vers l'utilisateur d'origine
  - les utilisateurs ont accès à un ensemble de rôles définis dans la politique SELinux
    - chaque rôle représente un ensemble de types manipulables
  - *type* définit un domaine pour les processus et un type pour les fichiers (attention à la confusion)
  - *level* est un attribut pour MLS et MCS

# SE Linux



- Identifier le contexte
  - Option -Z sur les utilitaires courants (ls, ps, id, netstat)

# SE Linux

- Contenu d'une politique SELinux
  - Éléments liés aux contextes de sécurité
    - Les identités et les rôles (RBAC)
    - Les types et les domaines (DTE)
    - Les sensibilités et les catégories (MLS/MCS)
  - Règles d'accès, d'audit et de transition (de rôle, de type, de domaine, de niveau)
  - Politiques pré-établies (famille Red Hat)
    - *minimum* : support d'un jeu restreint de démons confinés dans leur propre domaine
    - *targeted* : support d'un nombre supérieur de démons, cible également le utilisateurs
    - *mls* : politique multi-niveaux (BLP), cible des serveurs

# SE Linux

- Règles de vecteur d'accès
  - En mode *enforcing*, toutes les opérations qui ne sont pas autorisées par la politique sont interdites
- Syntaxe des règles
  - `<mot-clé> <src_t> <dest_t>:<classe> {<opération(s)>}` ;
  - Mots-clé :
    - *allow* : autorise l'opération ;
    - *auditallow* : autorise l'opération et créer une entrée dans le journal ;
    - *dontaudit* : refuse l'opération mais ne crée pas d'entrée dans le journal ;
    - *neverallow* : instruction pour le compilateur, spécifie qu'une règle *allow* ne doit jamais être générée pour l'opération

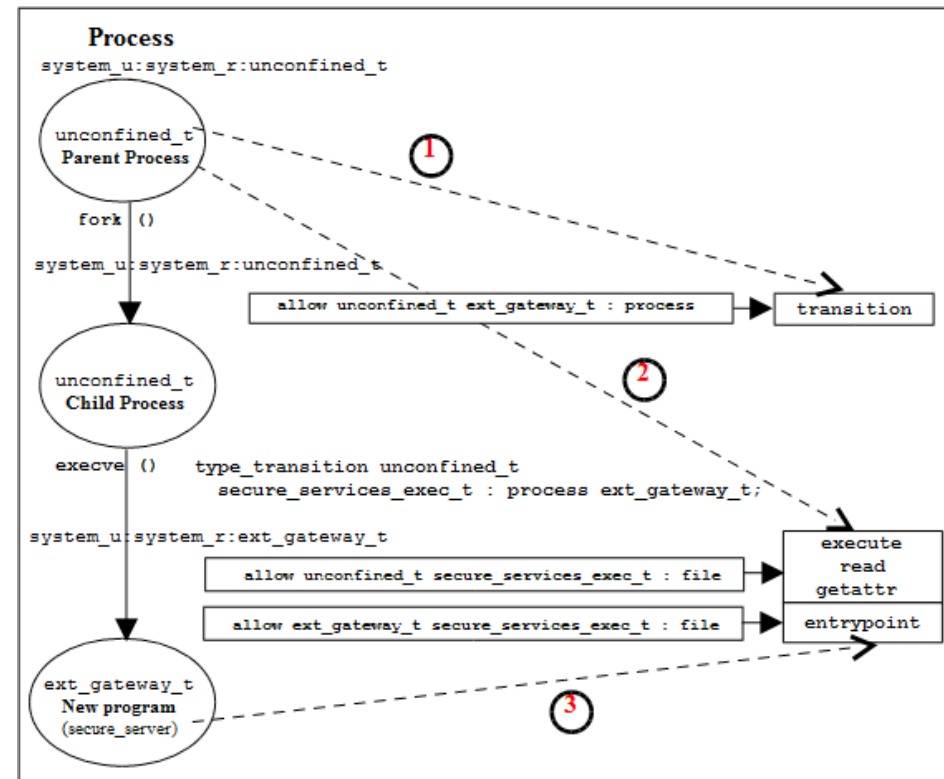
# SE Linux

- Syntaxe des règles (suite)
  - src\_t : type de la source (domaine du sujet)
  - dest\_t : type de la destination (type de l'objet)
  - classe : la classe de l'objet ;
  - opérations : les opérations concernées (dépendent de la classe de l'objet).
- Exemples
  - allow initrc\_t acct\_exec\_t:file { getattr read execute };
  - allow kernel\_t filesystem\_type:filesystem mount;
  - allow httpd\_t net\_conf\_t:file { read getattr lock ioctl };



# SE Linux

- Transitions
  - De domaine (processus)
    - Un processus dans un domaine A démarre un nouveau processus dans un domaine B avec un contexte de sécurité différent
  - De type (objet)
    - Un nouvel objet nécessite une étiquette différente de son parent
      - Par exemple, un fichier qui a besoin d'une étiquette différente de son répertoire parent



Source : SELinux project

# SE Linux

- Booléens SE Linux
  - Des modules de la politique SELinux peuvent exporter des options qui permettent de changer le comportement des règles par défaut
    - Le changement est fait à chaud, sans devoir recharger ou recompiler la politique

- État courant

```
$ getsebool -a
```

- Liste détaillée

```
$ semanage boolean -l
```

```
# An example showing a boolean and supporting if statement.

bool allow_execmem false;

# The bool allow_execmem is FALSE therefore the allow statement
# is not executed:

if (allow_execmem) {
    allow sysadm_t self:process execmem;
}
```

# SE Linux

- Il existe d'autres modules MAC implémentés sous forme de LSM
  - SMACK
  - Tomoyo
  - AppArmor

# Windows Mandatory Integrity Controls




- Depuis Windows Vista, il existe 5 niveaux d'intégrité pour les sujets et les objets :
  - « System integrity » (par ex., les services système) ;
  - « High integrity » (par ex., les processus d'administration) ;
  - « Medium integrity » (niveau par défaut) ;
  - « Low integrity » (par ex., des documents provenant d'Internet) ;
  - « Untrusted » (utilisé par les processus anonymes).
- Lors d'une opération de contrôle d'accès, le contrôle d'intégrité est effectué avant le contrôle d'accès discrétionnaire

# Windows Mandatory Integrity Controls

- Les niveaux d'intégrité sont stockés dans la SACL (*securable object* et *security principal*)
  - ACE de type **SYSTEM\_MANDATORY\_LABEL\_ACE**
    - NO\_WRITE\_UP (par défaut) : un principal avec un niveau inférieur à celui de l'objet ne peut pas écrire dans cet objet
    - NO\_READ\_UP : un principal avec un niveau inférieur à celui de l'objet ne peut pas lire cet objet
    - NO\_EXECUTE\_UP : un principal avec un niveau inférieur à celui de l'objet ne peut pas exécuter cet objet
  - Création de processus
    - Lors du lancement d'un fichier exécutable par un utilisateur, le nouveau processus est créé avec le niveau minimal entre les niveaux d'intégrité de l'utilisateur et du fichier

# Windows Mandatory Integrity Controls

- Le mode protégé d'Internet Explorer met en œuvre ce mécanisme pour chaque onglet de navigation :

 iexplore.exe	< 0.01	15292 Internet Explorer	Niveau obligatoire moyen	Microsoft Corporation
 iexplore.exe	< 0.01	12856 Internet Explorer	Niveau obligatoire faible	Microsoft Corporation
 iexplore.exe	< 0.01	17676 Internet Explorer	Niveau obligatoire faible	Microsoft Corporation

- Les processus iexplore.exe avec le niveau obligatoire faible (*Low integrity*) sont ne peuvent pas écrire dans des objets de niveaux supérieurs (« No-Write-Up ») et sont confinés à un nombre restreint d'objets :
  - par ex., C:\Users\UserName\AppData\LocalLow
    - icaccls affiche les niveaux autre que *Medium integrity* ;
    - l'outil Accesschk de Sysinternals renvoie plus de détails.

# Pour approfondir

- Sécurité des systèmes d'exploitation répartis : architecture décentralisée de méta-politique pour l'administration du contrôle d'accès obligatoire, Thèse de doctorat, Mathieu BLANC, 2006
- SELinux User's and Administrator's Guide, Red Hat
- The SELinux Notebook, 4<sup>th</sup> Edition, Richard Haines, 2014
- Windows Internals, Part 1: System architecture, processes, threads, memory management, and more, 7th Edition, 2017

# Questions ?

