Contrôler les accès aux données numériques

Alban Gabillon

Contrôler les accès aux données numériques

Article *in* Revue de l Electricité et de l Electronique · January 2013

CITATIONS
0

READS
661

1 author:

Alban Gabillon
University of French Polynesia
101 PUBLICATIONS **611** CITATIONS

SEE PROFILE

Le 16 septembre 2013

Contrôler les accès aux données numériques

Alban Gabillon
Professeur à l'Université de la Polynésie Française - Laboratoire GePaSud EA4238
alban.gabillon@upf.pf

Introduction

Un **modèle de contrôle d'accès** comprend :

- une **politique** (ou un règlement) **de contrôle d'accès** qui spécifie les accès autorisés aux données ;
- une **politique d'administration** qui indique comment la politique de contrôle d'accès peut être mise à jour.

Un **mécanisme de contrôle d'accès** est une solution logicielle ou matérielle pour appliquer une politique de contrôle d'accès.

Dans cet article, nous passons d'abord en revue les différents modèles de contrôle d'accès existants en indiquant leur domaine d'application. Ces modèles de contrôle d'accès sont répartis dans différentes catégories. Ainsi, nous considérons les modèles de contrôle d'accès discrétionnaires DAC (***D**iscretionary **A**ccess **C**ontrol*), les modèles de contrôle d'accès obligatoires MAC (***M**andatory **A**ccess **C**ontrol*), les modèles de contrôle d'accès à base de rôles RBAC (***R**ole **B**ased **A**ccess **C**ontrol*) et les modèles de contrôle d'accès contextuels CBAC (***C**ontext-**B**ased **A**ccess **C**ontrol*). Nous étudions ensuite les solutions existantes de contrôle d'accès pour les données relationnelles, XML (*e**X**tensible **M**arkup **L**anguage*) et RDF (***R**esource **D**escription **F**ramework*) en nous focalisant sur certaines particularités relatives à chacun de ces trois formats de données. Enfin, dans la dernière partie de cet article, nous évoquons les nouveaux enjeux en matière de contrôle d'accès qui sont apparus dans le cadre du réseau Internet, des nouvelles applications dans le nuage (*cloud computing*) et de l'accroissement exponentiel du volume mondiale de données numériques (*Big Data*).

Modèles de contrôle d'accès

Modèles de contrôle d'accès discrétionnaires (DAC[1])

Dans les modèles DAC tels que le modèle HRU de Harrison, Ruzzo, Ullman [1], la politique de contrôle d'accès repose sur les concepts de sujet, action et objet. Les sujets sont les entités actives du système (typiquement les utilisateurs), les objets les entités passives (typiquement les données) et les actions les accès directs que les sujets peuvent effectuer sur les objets. La politique de contrôle d'accès est représentée sous la forme d'une série de triplets (sujet, action, objet). Chaque triplet (*s,*

[1] Discretionary Access Control

a, o) signifie « le sujet *s* a la permission d'effectuer l'action *a* sur l'objet *o* »[2]. Les modèles DAC sont qualifiés de discrétionnaires dans la mesure où les permissions se réfèrent directement à l'identité de tel ou tel utilisateur. La plupart des modèles DAC supportent la notion de groupe d'utilisateurs. Les groupes simplifient la gestion des autorisations puisque une permission accordée à un groupe est alors automatiquement applicable à tous les membres du groupe. La plupart des modèles discrétionnaires incluent également la notion de droit propriétaire. Ce droit confère aux utilisateurs le privilège d'administrer le règlement de contrôle d'accès s'appliquant à leurs données (c'est-à-dire les données qu'ils ont eux-mêmes créées). Historiquement, les modèles DAC furent les premiers implantés dans les systèmes informatiques (notamment le système de fichiers d'UNIX). Malgré certaines limites que nous présentons par la suite, Ils restent, cependant, utilisés dans de nombreuses applications modernes. Le modèle de sécurité de *Facebook*, par exemple, est typiquement un modèle de sécurité discrétionnaire où les utilisateurs définissent les autorisations relatives aux données qu'ils publient.

Modèles de contrôle d'accès obligatoires (MAC[3])

Les modèles DAC ont le mérite de la simplicité. Cependant, leur principal inconvénient est qu'ils sont entièrement vulnérables aux attaques par cheval de Troie opérant par recopie de l'information. Ce fait est connu depuis les années 70. Afin de contrer ce type d'attaques, de nouveaux modèles de contrôle d'accès obligatoires ont donc été proposés. Dans les modèles MAC, un niveau de sécurité est affecté à chaque sujet et à chaque objet. Le niveau de sécurité associé à un objet s'appelle le niveau de classification alors que le niveau de sécurité associé à un sujet s'appelle le niveau d'habilitation. La politique de sécurité est obligatoire c'est-à-dire qu'elle s'impose à tous les utilisateurs et ne peut être modifiée. Si l'objectif est de garantir la confidentialité[4] des données alors la politique de sécurité obligatoire (que l'on appelle aussi la politique de sécurité multi-niveaux) est la suivante : « les utilisateurs ont l'interdiction de prendre connaissance des données ayant un niveau de classification supérieur à leur niveau d'habilitation mais ont la permission de prendre connaissance des données classifiées à un niveau égal ou inférieur à leur niveau d'habilitation ».
Bell & LaPadula [2] ont montré qu'il était nécessaire d'appliquer les deux propriétés de contrôle d'accès suivantes pour garantir la politique de sécurité multi-niveaux :

- *no read up* : cette propriété stipule qu'un sujet habilité à un certain niveau de confidentialité ne peut pas **lire** un objet classifié à un niveau supérieur ;
- *no write down* : cette propriété stipule qu'un sujet habilité à un certain niveau de confidentialité ne peut pas **écrire** dans un objet classifié à un niveau inférieur. La restriction du *no write down* est nécessaire pour empêcher qu'un cheval de Troie s'exécutant pour le compte d'un utilisateur hautement habilité ne transfère des données hautement classifiées dans un objet faiblement classifié.

Ces deux propriétés de contrôle d'accès ne sont toutefois pas suffisantes pour garantir la politique de sécurité multi-niveaux. Les modèles MAC s'inscrivent en fait dans la catégorie des modèles de contrôle de flux puisque le seul moyen de garantir totalement la sécurité multi-niveaux est de contrôler tous les flux d'informations possibles. L'information peut en effet transiter illégitimement par des canaux différents des simples opérations de lecture/écriture. Ces **canaux cachés** (*covert*

[2] Parfois ces triplets sont représentés en interne par le système sous la forme de listes de contrôle d'accès (ACL). Une ACL est une liste de paires (sujet, action) attachée à un objet.
[3] Mandatory Access Control
[4] Pour un modèle MAC visant à assurer l'intégrité des données, se référer au modèle de Biba.

channels) peuvent être de différentes sortes (canaux cachés d'inférence, temporels, etc.) et difficiles à recenser et à éliminer. Dans un modèle MAC, administrer la politique de sécurité signifie affecter des niveaux de classification et d'habilitation. Cette opération se fait en général sous la responsabilité d'un officier de sécurité unique. Les modèles MAC, de par la nature obligatoire du règlement de sécurité, sont peu flexibles mais procurent un degré de sécurité supérieur aux modèles DAC puisqu'ils essaient de prendre en compte tous les flux d'information transitant dans un système informatique. Historiquement, ces modèles ont d'abord été mis en œuvre dans des applications militaires. Dans le but de mieux confiner les applications et les processus, ils ont ensuite été implantés dans des systèmes variés dont voici une liste non exhaustive :

- Dans la carte à puce Java multi-applications, les contrôles d'accès obligatoires sont utilisés pour réguler les flux d'informations entre les différents applets Java.
- Le module noyau Security Enhanced Linux permet d'activer les contrôles d'accès obligatoires dans les systèmes d'exploitation Linux et Android (depuis la version 4.3).
- Windows Vista et Windows 7 implantent des contrôles d'accès basés sur le modèle de Biba afin, en particulier, d'éviter qu'un processus utilisateur ne corrompe un objet système.
- Il existe des versions multi-niveaux de certains SGBD (par exemple Oracle Label Security)

Modèles de contrôle d'accès à base de rôles (RBAC[5])

Les modèles DAC et MAC ne sont pas bien adaptés aux besoins des organisations commerciales. Dans ce type d'organisation les privilèges conférés aux utilisateurs dépendent du **rôle** des utilisateurs au sein de l'organisation. De ce fait les modèles RBAC sont apparus et se sont imposés comme une alternative aux modèles DAC et MAC traditionnels. En 2004, l'International Committee for Information Technology Standards de l'American National Standards Institute (ANSI/INCITS) a officiellement élevé au statut de standard la proposition de Sandhu, Ferraiolo and Khun [3]. Les principes de base du modèle RBAC sont les suivants :

- Alors que dans les modèles DAC, les permissions ont trait à des opérations de bas niveau telles que les opérations de lecture/écriture, dans les modèles RBAC elles concernent des tâches de nature organisationnelle telles que « transférer de l'argent », « acheter un billet d'avion » etc.
- Dans les modèles RBAC, le concept de rôle correspond à une fonction professionnelle. Les permissions sont accordées à des rôles et non pas à des utilisateurs. Les rôles sont ensuite distribués aux utilisateurs en fonction de leurs responsabilités au sein de l'organisation. Une même permission peut être affectée à différents rôles et différents rôles peuvent être attribués à un même utilisateur.
- Les modèles RBAC offrent une solution pour implanter des mesures de type **séparation des tâches**. Le principe de la séparation des tâches prévoit qu'un même utilisateur ne peut effectuer des tâches qui pourraient être orchestrées pour mettre œuvre des opérations frauduleuses, comme par exemple « autoriser un paiement » et « effectuer un paiement ». Ce principe peut aisément être garanti avec les modèles RBAC dans la mesure où deux rôles peuvent être déclarés comme étant mutuellement exclusifs. Deux rôles mutuellement exclusifs ne peuvent alors être affectés à un même utilisateur.

[5] Role Based Access Control

Le standard RBAC ne dit rien au sujet de l'administration du règlement de sécurité. Il suppose de manière implicite que la définition des rôles, l'affectation des permissions aux rôles et la distribution des rôles aux utilisateurs sont effectuées par une autorité centrale. Le modèle ARBAC (***A**dministrative **R**ole-**B**ased **A**ccess **C**ontrol* est un modèle à base de rôles prévoyant des rôles correspondant aux fonctions d'administration du règlement de sécurité.
Les modèles à base de rôles ont été implantés dans de nombreux systèmes et applications tels que Microsoft Active Directory, la plupart des SGBD commerciaux, FreeBSD et Wikipedia.

Modèles de contrôle d'accès contextuels (CBAC[6])

Dans un nombre d'applications de plus en plus grand, la politique de sécurité ne peut plus être définie au moyen de règles d'autorisation statiques. Dans de telles applications, les privilèges accordés aux utilisateurs dépendent de conditions contextuelles. Les modèles de contrôle d'accès qui permettent l'expression de règles dynamiques où la distribution des autorisations dépend de conditions contextuelles appartiennent à la famille des modèles CBAC. Les modèles suivants peuvent être considérés comme étant des modèles CBAC :

- Dans le modèle ABAC (***A**ttribute-**B**ased **A**ccess **C**ontrol*) [4], les autorisations dépendent de conditions booléennes s'appliquant aux attributs du sujet, de l'objet et de l'environnement.
- Certaines propositions étendent le modèle RBAC pour prendre en compte des conditions contextuelles telles que la position de l'utilisateur ou le temps. Dans la plupart de ces approches les rôles peuvent être activés en fonction de conditions spatiales ou temporelles.
- Le modèle OrBAC (***Or**ganization **B**ased **A**ccess **C**ontrol*) [5], définit une taxonomie complète de contextes (spatial, temporel, provisionnel etc.) et fournit un cadre formel fondé sur la logique du premier ordre pour exprimer des règles contextuelles. Il intègre le modèle d'administration AdOrBAC.

Une caractéristique importante des modèle CBAC réside dans le fait qu'ils permettent d'exprimer des règles d'autorisation qui ne requièrent pas d'authentifier les utilisateurs. Un utilisateur peut en effet obtenir un accès à une information simplement parce que certaines conditions contextuelles sont remplies. Cette capacité à accorder ou refuser un accès sans avoir à authentifier l'utilisateur est très utile dans le cadre d'applications Web interconnectées. Mentionnons également que de nombreux modèles de contrôle d'accès récents permettent d'exprimer des interdictions explicites. Les interdictions explicites servent à exprimer des exceptions à des permissions générales (ex : les enfants ont la permission de jouer aux jeux vidéo sauf la petite sœur). Écrire des règlements de sécurité qui contiennent à la fois des permissions et des interdictions peut toutefois donner lieu à des conflits. Il existe différentes approches pour résoudre automatiquement ces conflits. Certains modèles appliquent des principes du type « la permission l'emporte » ou « le plus spécifique l'emporte ». D'autres modèles utilisent des priorités associées aux règles.

Application du contrôle d'accès

Les mécanismes de contrôle d'accès sont des solutions matérielles ou logicielles de confiance qui appliquent la politique de contrôle d'accès. L'Internet Engineering Task Force (IETF) définit un modèle abstrait pour l'application des contrôles d'accès qui est mis en œuvre dans la plupart des implantations existantes de mécanismes de contrôle d'accès. Ce modèle fait une claire distinction entre le composant PDP (***P**olicy **D**ecision **P**oint*) et le composant PEP (***P**olicy **E**nforcement **P**oint*) :

[6] Context Based Access Control

- le composant PEP intercepte la demande d'accès et la transmet au PDP. Après avoir reçu la décision du PDP, il l'applique ;
- le composant PDP analyse la demande d'accès, évalue les conditions contextuelles, résout les éventuels conflits entre permissions et interdictions et calcule la décision finale (accès accordé ou non).

Plus de détails sur ce modèle d'application des contrôles d'accès peuvent être trouvés dans l'article « Gestion des habilitations : Modèles et Architectures » de ce même dossier.

Contrôler les accès aux données

La nature de l'application détermine l'adoption de tel ou tel modèle de contrôle d'accès. Ainsi pour une application où les utilisateurs doivent eux-mêmes gérer les droits s'appliquant à leurs propres données, on s'orientera vers un modèle discrétionnaire. Pour une application militaire, on utilisera un modèle obligatoire. Pour une application commerciale on utilisera un modèle à base de rôles et enfin pour une application mobile, on adoptera un modèle de contrôle d'accès contextuel permettant de prendre en compte le contexte spatial. On pourra également avoir besoin d'un modèle qui combine les caractéristiques des différents modèles présentés (par exemple gestion du contexte et utilisation de rôles). Dans ce cas on s'orientera vers un modèle à fort pouvoir d'expression comme le modèle OrBAC.

Une fois qu'un modèle de contrôle d'accès adapté à l'application considérée a été choisi, il reste à *instancier* ce modèle pour le type de données utilisé. Dans cette section nous nous intéressons à trois formats de données très répandus que sont le format relationnel, le format XML et le format RDF. Ces trois formats de données sont utilisés dans les bases de données se trouvant en arrière-plan de la plupart des applications Web. Dès lors que l'on instancie un modèle de contrôle d'accès pour un type de données particulier, il convient de proposer des solutions pour traiter certains aspects liés à l'application du modèle de contrôle d'accès choisi pour le format de données utilisé. Parmi ces points, nous évoquons les suivants :

- *Granularité de l'information à protéger*. Pouvoir écrire une règle de contrôle d'accès adressant une unité d'information peut dans certains cas correspondre à un besoin. Un langage permettant d'écrire des règles de contrôle d'accès adressant des données élémentaires est un langage dit *à fine granularité*.
- *Expression des vues*. La plupart du temps, un administrateur de sécurité a besoin d'écrire des règles de contrôle d'accès adressant chacune un ensemble de données élémentaires. Un tel ensemble que l'on appelle *une vue* se définit à l'aide d'un langage de requête.
- *Mécanisme de contrôle d'accès*. Grosso modo, il existe deux approches pour implanter le contrôle d'accès : le calcul des *vues autorisées* ou la *réécriture de requêtes*.

Format Relationnel

Dans le cadre du format relationnel, le plus petit élément d'information est le *tuple*[7] (voire le tuple restreint à certains de ses attributs). Le modèle de sécurité implanté dans la plupart des SGBD relationnels est le modèle de sécurité de SQL[8] défini dans la norme SQL92. Il s'agit essentiellement d'un modèle de contrôle d'accès discrétionnaire où les gestionnaires de données définissent le règlement de contrôle d'accès protégeant leurs données en deux étapes :

[7] Collection ordonnée de n objets ou n-uplet
[8] **S**tructured **Q**uery **L**anguage

1. Définition des vues à l'aide du langage SQL. Chaque vue est nommée ;
2. Affectation aux utilisateurs des droits d'accès sur les vues nommées et les tables à l'aide de la commande GRANT. Ces mêmes droits peuvent être révoqués grâce à la commande REVOKE.

Dans la mesure où il est possible de définir une vue qui ne contient qu'un seul tuple, nous pouvons dire que la partie DCL[9] de SQL, qui est le sous-ensemble de SQL dédié à la définition du règlement de contrôle d'accès, est un langage à fine granularité.

Avec la norme SQL99 il est devenu possible de créer des rôles et ainsi d'accorder des droits d'accès aux rôles qui sont alors eux-mêmes affectés aux utilisateurs. A noter toutefois, que le droit de créer des rôles est un privilège de niveau administrateur de la base de données ce qui en limite l'intérêt, puisque, en pratique, seul l'administrateur de la base de données pourra créer et utiliser des rôles. Un simple gestionnaire de données, c'est-à-dire un utilisateur qui administre quelques tables, ne pourra créer des rôles dans son schéma afin de structurer son règlement de sécurité.

L'implantation du contrôle d'accès consiste simplement à vérifier qu'un utilisateur qui accède à une vue prédéfinie ou à une table détient le ou les privilèges requis.

Format XML

XML est devenu le format standard d'échange de données sur le Web dont l'unité d'information est le *nœud* (selon le modèle ***D****ocument* ***O****bject* ***M****odel*). Plusieurs modèles de contrôle d'accès discrétionnaires pour XML ont été proposés dans la littérature [6]. Ces modèles traitent de divers problèmes tels que la granularité de l'information à protéger et la résolution des conflits entre permissions et interdictions. La politique de sécurité est écrite dans un langage basé sur XML tel que XACML[10]. Dans ces modèles, il n'y a pas, comme avec SQL, de définition préalable de vues nommées. Chaque règle de contrôle d'accès contient une expression XPath[11] adressant un ensemble de nœuds XML. L'utilisation de XPath permet d'écrire des règlements de contrôle d'accès à fine granularité. En ce qui concerne l'implantation du contrôle d'accès, ces modèles peuvent être classifiés en deux catégories : filtrage de nœuds XML [6] et réécriture de requêtes [7]

Dans la première catégorie, le PDP calcule une vue autorisée des données XML originales en fonction des droits de l'utilisateur qui a soumis la requête. Le moteur de requêtes évalue ensuite la requête de l'utilisateur sur la vue autorisée et finalement le PEP retourne le résultat à l'utilisateur. D'un point de vue technique, les auteurs dans [8] suggèrent d'implanter le PDP à partir d'un processeur XSLT[12]. Ils définissent une feuille de style pour automatiquement traduire la politique de contrôle d'accès en règles XSLT qui sont, à leur tour, utilisées par le processeur XSLT pour calculer la vue utilisateur.

Dans la deuxième catégorie, le PDP utilise les règles d'autorisation pour réécrire la requête utilisateur en une requête sûre c'est-à-dire une requête qui n'adresse pas des données non autorisées. La requête réécrite est ensuite évaluée sur les données XML originales et le résultat est retourné par le PEP à l'utilisateur. L'avantage de la deuxième approche est que le processus de réécriture de requêtes peut être implanté coté serveur ou coté client (pourvu que le client soit de confiance).

En ce qui concerne les opérations d'écriture, il existe aussi deux approches. Dans la première approche, le PDP extrait parmi les nœuds qui sont adressés par la requête, ceux pour lesquels

[9] **D**ata **C**ontrol **L**anguage
[10] e**X**tensible **A**ccess **C**ontrol **M**arkup **L**anguage
[11] **X**ML **Path** Language
[12] e**X**tensible **S**tylesheet **L**anguage **T**ransformations

l'utilisateur détient un privilège en écriture. Dans la deuxième approche, la requête originale est réécrite en une requête sûre qui n'adresse que les nœuds autorisés.

Format RDF

RDF est un modèle de graphe dont l'unité d'information est le *triplet*, utilisé pour décrire tout type de ressource sur le Web. Il est à la base du Web sémantique. Il existe plusieurs types d'approche pour réguler les accès aux données RDF. Ces approches diffèrent à la fois au niveau de la définition des vues et au niveau de la solution adoptée pour implanter la politique de contrôle d'accès.
Dans certains modèles [9], les règles de sécurité sont définies à l'aide de motifs RDF qui adressent les triplets RDF autorisés (permission) ou non autorisés (interdiction). Les triplets non autorisés sont alors simplement éliminés du résultat de la requête utilisateur par le PEP. L'inconvénient principal de cette approche est qu'à chaque évaluation de requête, les motifs RDF utilisés dans la politique de sécurité doivent être instanciés.
Dans [10], les règles d'autorisation adressent *des vues RDF* nommées, prédéfinies à l'aide de requêtes SPARQL[13]. Les règles d'autorisation sont gérées à l'aide de deux commandes de bas niveau de type GRANT/REVOKE inspirées du langage SQL. Les gestionnaires de données publient les vues sur leurs données au travers d'un proxy décentralisé qui régule les accès aux vues en fonction des droits définis par les gestionnaires eux-mêmes.
Dans d'autres modèles, les requêtes aux données RDF originales sont réécrites en fonction des droits des utilisateurs. Ainsi, dans [11], les auteurs proposent de réécrire les requêtes SPARQL en fonction des règles d'autorisation exprimées dans le cadre du modèle PrivOrBAC qui est la déclinaison du modèle OrBAC pour la gestion de la vie privée. Dans le cadre de ce modèle, les utilisateurs peuvent exprimer des règles protégeant leurs données en fonction de contextes tels que le consentement ou l'intention.
L'adoption de telle ou telle solution a un impact direct sur l'administration de la politique de sécurité. Considérons par exemple le cas concret d'une application de type réseau social. Dans le cadre de l'approche [11], les utilisateurs définissent la politique de sécurité protégeant *les différents éléments* de leur profil. A chaque fois que Bob accède au profil d'Alice, une vue autorisée du profil d'Alice est calculée dynamiquement en fonction des privilèges de Bob. Dans le cadre de l'approche [10], les utilisateurs suivent une démarche inspirée de SQL. Ils prédéfinissent d'abord *différentes versions* (vues) de leur profil (une vue pour les amis, une vue pour les amis des amis, etc.). Chaque fois que Bob souhaite accéder au profil d'Alice, une version du profil d'Alice correspondant au statut de Bob (ami, ami d'un ami, etc.) est fournie à Bob. Les actions de Bob (publication de messages sur le mur d'Alice par exemple) sont alors régulées en fonctions des règles d'autorisation protégeant la version du profil d'Alice consultée par Bob.

Nouveaux enjeux

Le contrôle d'accès est toujours un domaine actif de recherche. De nouvelles problématiques en matière de contrôle d'accès se situent dans des contextes liés à l'interconnexion des systèmes d'information et à l'explosion du volume mondial de données numériques.

[13] **S**PARQL **P**rotocol **a**nd **R**DF **Q**uery **L**anguage

Interopérabilité

Les systèmes d'information sont de plus en plus distribués et interconnectés. Il existe déjà des outils permettant d'assurer certaines interactions (par exemple Web Single Sign On). Toutefois, les modèles de contrôle d'accès existants doivent être étendus afin de pouvoir sécuriser toutes les interactions entre organisations. Plusieurs pas ont déjà été franchis dans ce sens. Parmi les travaux existants, nous pouvons citer le cas de la **négociation de la confiance** [12]. Établir la confiance entre deux entités étrangères l'une à l'autre requiert une phase de négociation qui s'accomplit par l'échange progressif de titres numériques (*digital credentials*) *de plus en plus sensibles*. La confiance est établie une fois l'échange de titres parvenu à son terme. La confiance n'est pas établie si l'une des deux parties a rompu le processus d'échanges. En cas de négociation menée avec succès, les interactions deviennent alors possibles. L'entité cliente peut, en particulier, accéder aux données ou ressources en ligne hébergées par l'entité serveur. Le modèle sous-jacent utilisé dans le processus de négociation de la confiance peut, par exemple, être le modèle ABAC. La politique de contrôle d'accès protège non seulement les données ou les ressources hébergées mais également les titres numériques échangés durant le processus d'établissement de la confiance.

Informatique dans le nuage (cloud computing)

Ce qui caractérise le plus l'informatique dans le nuage est que les données sont **externalisées.** Ceci signifie que les données ne sont pas stockées au sein de l'organisation qui est propriétaire des données mais chez un tiers hébergeur fournissant des services dans le nuage. La plupart du temps les données enregistrées dans le nuage ne sont pas chiffrées dans la mesure où le chiffrement est couteux en temps machine et rend difficiles les procédures d'indexation. Le fournisseur de service de stockage dans le nuage échappant à tout contrôle d'accès peut donc en toute liberté accéder aux données qu'il héberge. Si le tiers hébergeur est malveillant, il peut, à des fins pécuniaires, divulguer soit directement certaines données sensibles à des entités non autorisées, soit des résultats d'opérations de fouille de données sur tout ou partie des données qu'il héberge. Même s'il n'est pas malveillant, le tiers hébergeur peut être contraint de divulguer certaines données pour répondre à une injonction d'une autorité gouvernementale ou judiciaire. Ce cas est particulièrement d'actualité depuis la révélation récente de l'existence du programme de surveillance PRISM mené par la National Security Agency (NSA) impliquant la collaboration de plusieurs fournisseurs majeurs de services dans le nuage tels que Yahoo, Google et Microsoft. Des solutions d'hébergement dans le nuage utilisant la cryptographie, où le fournisseur de service de stockage ne peut accéder aux données qu'il héberge (on parle alors de *host-proof hosting*) commencent à apparaitre, que ce soit dans la littérature scientifique ou sur le marché des fournisseurs de service de stockage de données. A noter toutefois, qu'il n'existe pas, à notre connaissance, de solution convaincante de type *host-proof hosting*, autorisant le partage de données sur la base d'une politique de contrôle d'accès à fine granularité.

Big Data

Le terme *Big Data* désigne de très gros volumes de données souvent hétérogènes qui ne peuvent être traités en utilisant des outils tels que les SGBD traditionnels. Leur taille requiert de redéfinir entièrement les processus de stockage et d'exploitation. Récemment, le Cloud Security Alliance a publié 13 défis à relever en matière de sécurité dans le cadre du *Big Data*. Parmi ces 13 défis de sécurité nous mentionnerons le problème de la sécurité du traitement distribué des données, le

problème de la sécurité des bases de données NoSQL et le problème de la sécurité des opérations de fouille de données.

Dans le cadre du traitement distribué d'un grand volume de données, il faut en particulier s'assurer que chaque nœud de traitement ne divulgue pas d'informations sensibles. Des solutions à base de contrôle d'accès obligatoire ont été proposées [13], les contrôles d'accès obligatoires empêchant notamment la divulgation des données traitées par recopie (*no write down*).

Les Systèmes de Gestion de Bases de Données (SGBD) NoSQL ont été conçus pour stocker de larges volumes de données plus ou moins structurées. La sécurité n'a pas toujours été prise en compte dans le processus de développement de ces SGBD. Ainsi, ces SGBD n'offrent pas la plupart des services de sécurité traditionnellement offerts par les SGBD relationnels [14]. En particulier les SGBD NoSQL ne proposent pas l'équivalent des commandes GRANT/REVOKE présentes dans tout SGBD relationnel et qui permettent de définir des politiques de contrôle d'accès. Un préalable à l'éventuelle implantation de telles primitives consisterait à définir des modèles de contrôle d'accès pour les différents modèles de données utilisés dans les bases de données NoSQL.

Des opérations de fouille de données sont régulièrement menées sur de larges volumes de données personnelles. Ces volumes de données sont en général publiés sur Internet après anonymisation afin de respecter la vie privée des individus. Toutefois, des exemples ont montré qu'avec peu de connaissances externes, il était souvent assez aisé de retrouver des données personnelles qui avaient été masquées [15]. Il existe donc une autre approche qui consiste à autoriser l'exécution d'algorithmes de fouille de données directement sur les données originales. Des techniques de *differential privacy* [13] peuvent alors être utilisées pour permettre la publication du résultat final de l'exécution de l'algorithme de fouille de données sans rien révéler d'une quelconque information à caractère personnel. Ces techniques consistent essentiellement à ajouter du bruit au résultat final.

Contrôle d'usage

Il y a plus de 30 ans, Dorothy Dennings [16] notait « les contrôles d'accès permettent de réguler les accès aux objets mais pas ce que les sujets peuvent faire avec l'information contenue dans ces objets ». Les modèles de contrôle de flux procurent une solution partielle à ce problème. Par exemple la restriction *no write down* du modèle de Bell & LaPadula empêche les sujets de haut niveau de copier des données de haut niveau dans un fichier de bas niveau. Toutefois, les modèles de contrôle de flux existants ne sont pas toujours adaptés à un environnement ouvert comme l'Internet. En conséquence, des nouveaux modèles de contrôle d'usage ont été proposés [17]. Les politiques de contrôle d'usage spécifient des exigences qui doivent être satisfaites avant, pendant et surtout après un accès. Le contrôle d'usage est étroitement lié à l'utilisation d'**obligations** dans la politique de sécurité. Ces obligations consistent en un ensemble de contraintes qui doivent être respectées par l'utilisateur qui a l'usage de l'information. Dans le cas d'informations soumises à droit d'auteur, la politique de contrôle d'usage est souvent spécifiée dans une *licence*. Si l'on considère un morceau de musique par exemple, la licence pourra exiger que l'utilisateur paye une certaine somme pour pouvoir écouter le morceau de musique et qu'il supprime le morceau au bout d'une semaine (à moins de payer à nouveau). En ce qui concerne l'implantation, les contrôles d'usage peuvent être mis en œuvre au moyen de techniques DRM[14].

[14] **D**igital **R**ights **M**anagement : gestion des droits numériques

Conclusion

Dans la première partie de cet article, nous avons balayé les différentes approches existantes en matière de contrôle d'accès en suivant une trame plus ou moins chronologique. Dans la deuxième partie, nous avons évoqué certains aspects relatifs à l'application d'un modèle de contrôle d'accès à un type de données particulier. Enfin, dans la dernière partie, nous avons tenté d'identifier les nouveaux enjeux en matière de protection des données qui sont apparus dans le cadre de l'interconnexion des systèmes d'information et du traitement distribué de l'information.
De notre point de vue beaucoup reste à faire pour protéger les données numériques. Nous pensons, en particulier, que les deux problèmes suivants freinent le développement des applications sur l'Internet :

- Vulnérabilité des systèmes informatiques aux attaques par cheval de Troie opérant par simple recopie de l'information. Ce problème est ancien mais il subsiste dans la mesure où les modèles de contrôle d'accès implantés dans la plupart des systèmes informatiques connectés à l'Internet sont de nature discrétionnaire de type HRU. Certes des solutions, à base de pare-feu notamment, ont été proposées et mises en œuvre pour filtrer les flux d'information, mais ces solutions ne sont que des pis-aller, des sortes de rustine dont le niveau d'efficacité reste à déterminer. L'attaque par injection SQL dont sont victimes de nombreux sites Web est un exemple bien connu d'attaque par cheval de Troie opérant par recopie de l'information. Cette attaque consiste à injecter une requête SQL (le cheval de Troie) dans le processus communiquant avec le SGBD sous-jacent à l'application Web.
- Vulnérabilité des données enregistrées dans le nuage. Les récents événements divulgués par la presse nous montrent que des solutions pour protéger les données enregistrées dans le nuage restent à définir. Ces évènements ont d'ailleurs montré que certaines applications dans le nuage, tel l'e-mail, ne pouvaient, de par leur conception, être totalement sécurisées. Sécuriser l'e-mail, en particulier, nécessiterait de concevoir un nouveau protocole pour le transport et la distribution des messages en intégrant les exigences de sécurité dès le début de la conception.

Bibliographie

[1] *Harrison, Ruzzo and Ullman, "Protection in Operating Systems,"* Communication of the ACM, *vol. 19, no. 8, pp. 461-471, august 1976.*

[2] *Bell and LaPadula, "Secure Computer Systems: Unified Exposition and Multics," Technical Report ESD-TR-75-306, MTR-2997, MITRE, Bedford, Mass, 1975.*

[3] *Ferraiolo, Sandhu, Gravila, Kuhn and Chandramouli, "Proposed NIST Standard for Role-Based Access Control,"* ACM Transactions on Information and System Security, *vol. 4, no. 3, pp. 222-274, 2001.*

[4] *Yuan and Tong, "Attributed Based Access Control (ABAC) for Web Services," in* Proc. of the IEEE International Conference on Web Services (ICWS'05)*, 2005.*

[5] *Cuppens and Cuppens-Boulahia, "Modeling contextual security policies,"* International Journal of Information Security (IJIS), *vol. 7, no. 4, 2008.*

[6] *Damiani, D. C. d. Vimercati, Paraboschi and Samarati, "Securing XML documents," in* Proc. of the 2000 International Conference on Extending Database*, 2000.*

[7] *Luo, D. Lee, W. Lee and Liu, "QFilter: fine-grained run-time XML access control via NFA-based query rewriting," in* Proc. of CIKM*, 2004.*

[8] *Gabillon and Bruno, "Regulating Access to XML documents," in* Fifteenth Annual IFIP WG 11.3 Working Conference on Database Security*, Niagara on the Lake, 2001.*

[9] *Reddivari, T. Finin and A. Joshi, "Policy-based access control for an RDF store," in* Proceedings of the Policy Management for the Web workshop*, 2005.*

[10] *Gabillon and Letouzey, "A View Based Access Control Model for SPARQL," in* IEEE 4th International Conference on Network and System Security (NSS)*, Melbourne, 2010.*

[11] *Oulmakhzoune, Cuppens-Boulahia, Cuppens, Morucci, Barhamgi and Bensliman, "Privacy query rewriting algorithm instrumented by a privacy-aware access control model,"* annals of telecommunications-annales des télécommunications, *vol. 1, no. 17, 2013.*

[12] *Yu, Winslett and Seamons, "Supporting structured credentials and sensitive policies through interoperable strategies for automated trust negotiation,"* ACM Transactions on Information and System Security (TISSEC), *vol. 6, no. 1, pp. 1-42, 2003.*

[13] *Roy, Setty, Kilzer, Shmatikov and Witchel, "Airavat: Security and Privacy for MapReduce,"* NSDI, *vol. 10, 2010.*

[14] *Okman, Gal-Oz, Gonen, Gudes and Abramov, "Security Issues in NoSQL Databases," in* IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2011.*

[15] *Narayanan and Shmatikov, " Robust de-anonymization of large sparse datasets," in* IEEE Symposium on Security and Privacy*, 2008.*

[16] *D. Denning, Cryptography and Data Security, Addison-Wesley, 1982.*

[17] *Park and Sandhu, "The UCON-ABC Usage Control Model,"* ACM Transactions on Information and System Security (TISSEC), *vol. 7, no. 1, pp. 128-174, 2004.*

L'auteur (ou les auteurs)

Alban Gabillon a obtenu un doctorat en informatique (sécurité des bases de données) à L'ENSAE (Sup Aéro), Toulouse, en 1995. Il a occupé un poste d'*Assistant Professor* à l'*Eastern Mediterranean University* de Chypre de 1995 à 1998. En 1998, il est devenu Maître de Conférences en informatique à l'université de Toulon où il a soutenu son Habilitation à Diriger les Recherches en 2000. En 2001, il est devenu Professeur à l'université de Pau (IUT de Mont de Marsan). Enfin, depuis 2007, il occupe un poste de Professeur d'informatique à l'université de la Polynésie Française. Alban Gabillon est auteur ou co-auteur de plus de 70 articles sur la sécurité des systèmes d'information.

Summary

In this paper, we first review existing access control models and their application. We consider **Di**scretionary **A**ccess **C**ontrol models (DAC), **M**andatory **A**ccess **C**ontrol (MAC) models, **R**ole **B**ased **A**ccess **C**ontrol (RBAC) models and **C**ontext **B**ased **A**ccess **C**ontrol models. We then recall existing solutions for access control to relational data, e**X**tensible **M**arkup **L**anguage (XML) data and **R**esource

Description **F**ramework (RDF) data. Finally, we investigate the new security and privacy issues in the framework of web service interconnection, cloud computing, Big Data and usage control.