Spécification et analyse formelle des politiques de sécurité dans le cloud computing

Asma Guesmi[1], Patrice Clemente[2], Frédéric Loulergue[1] and Pascal Berthomé[2]

[1,2] Univ. Orléans, INSA Centre Val de Loire, LIFO EA 4022
[1]45067 Orléans, France, firstname.lastname@{univ-orleans.fr}
[2]18022 Bourges, France, firstname.lastname@{insa-cvl.fr}

June 29, 2015

Le cloud computing (informatique en nuage) représente de nos jours un enjeu économique et stratégique important. Afin que celui-ci puisse être adopté en toute sérénité par les entreprises, il est indispensable qu'il apporte des garanties en termes de sécurité. L'objectif de nos travaux est de proposer une solution globale qui permet un brokerage (courtage) des services du cloud. Cette solution doit prendre en compte les exigences de sécurité et de contrôle d'accès exprimées par le client. Ces exigences doivent être appliquées du côté des fournisseurs de service. Les travaux présentés dans ce résumé ont fait l'objet de publications dans deux conférences internationales [1, 2].

Cloud broker basé sur des critères de sécurité : Suivant ce schéma, on considère trois types d'acteurs dans l'univers des clouds. Les clients et les fournisseurs de service cloud constituent les deux premiers. Le troisième, le broker (courtier), est un intermédiaire entre les clients et les fournisseurs du cloud. Il négocie les offres et les besoins afin de satisfaire les deux autres types d'acteurs. Les brokers existants ne considèrent que les besoins fonctionnels des clients comme les capacités (CPU, mémoire et stockage) des machines virtuelles (MV). Nous proposons d'ajouter les aspects de sécurité aux critères de négociation.

Le client qui souhaite se procurer des services (ressources de calcul, de stockage ou applications) sur le cloud spécifie leurs critères fonctionnels souhaités et exprime ses besoins de sécurité tels que des propriétés de confidentialité ou d'intégrité et des propriétés d'usage telles que la collaboration, la concurrence ou l'isolation entre les ressources demandées. Le client spécifie ainsi les privilèges d'usage des ressources, qui sont traduits en termes de flux d'informations autorisés ou non entre les ressources. Parmi les propriétés de sécurité que le client peut demander on trouve : Une MV isolée, doit être allouée seule sur une grappe et aucun flux d'information ne doit transiter à l'extérieur ou à l'intérieur de cette MV. Deux MV en concurrence doivent être allouées sur différentes grappes et l'information ne peut pas transiter entre ces MV directement ou indirectement.

Le broker a deux tâches. La première consiste à vérifier la consistance du système (ressources demandées et contraintes) du client et la validité des propriétés de sécurité exigées, en utilisant des méthodes formelles. En effet, une politique de sécurité mal implémentée mène à des vulnérabilités. La deuxième tâche est le placement des ressources et services client chez les fournisseurs adéquats.

Les besoins fonctionnels et non-fonctionnels que le client a décrits sont pris en compte par le broker. Ce dernier, qui a déjà connaissance des offres fournisseurs, fait correspondre les besoins aux offres (matching). Il essaye de placer les ressources (MV par exemple) chez les fournisseurs de telle sorte que tous les besoins fonctionnels soient assurés. Le broker propose une configuration de placement de gardiens (pare-feu) entre les ressources de telle sorte que toutes les propriétés de sécurité soient respectées.
Le broker vérifie la satisfiabilité des propriétés de sécurité et d'usage exigées par le client. Par exemple, il devra vérifier qu'il n'existe pas de flux d'information direct ou transitif entre deux MV en concurrence une fois déployées chez les fournisseurs.
Le broker peut échouer à placer les ressources chez les fournisseurs soit parce que les besoins fonctionnels ne peuvent pas être assurés (par exemple manque d'espace de stockage dans la région de France), soit parce que les besoins fonctionnels sont assurés mais aucun placement ne satisfait les propriétés de sécurité. Dans ces cas, le broker retourne au client des contre-exemples pour le guider à mettre à jour ses besoins. Le broker peut trouver une configuration de placement des ressources demandées chez des fournisseurs sélectionnés tout en assurant toutes les propriétés de sécurité et d'usage. Dans ce cas, le client peut confirmer cette configuration et le broker procède ensuite au déploiement.

Le client spécifie une politique d'accès. Le broker analyse la politique et vérifie l'absence des conflits entre les règles d'autorisation. Cette politique est utilisée pour gérer l'accès aux ressources déployées dans le cloud par des utilisateurs finaux qui vont utiliser les service fournis par le client.

L'élément central du broker que nous proposons est réalisé en utilisant des techniques de vérification formelle. Nous utilisons le langage de spécification Alloy [3] qui est supporté par un outil (Alloy Analyzer). Alloy est un langage déclaratif et permet de faire des descriptions modulaires et des configurations complexes pour les systèmes à vérifier. Alloy analyzer permet une analyse syntaxique et sémantique des spécifications.
Dans ce formalisme, nous avons décrit tout un univers de cloud. En particulier, ce formalisme permet de spécifier les ressources disponibles par les différents fournisseurs et les besoins des clients. Un algorithme de matching qui permet de faire les affectations des demandes aux ressources est aussi décrit au moyen du même formalisme.
Avec Alloy Analyzer, nous analysons les contraintes du client et détectons les conflits donnés dans la description. Il retourne des contre-exemples dans le cas d'inconsistance. Alloy permet aussi de générer des placements de ressources chez les fournisseurs de sorte que toutes les contraintes soient assurées et que la configuration proposée soit consistante.

References

[1] Asma Guesmi and Patrice Clemente. Access control and security properties requirements specification for clouds' seclas. In *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*, volume 1, pages 723–729. IEEE, 2013.

[2] Asma Guesmi, Patrice Clemente, Frédéric Loulergue, and Pascal Berthomé. Cloud resources placement based on functional and non-functional requirements. In *SECRYPT 2015 - Proceedings of the 12th International Conference on Security and Cryptography, Colmar, France, 20-22 July*, 2015.

[3] Daniel Jackson. *Software Abstractions: logic, language, and analysis*. MIT press, 2012.