Démarche ascendante pour la déduction de politique de sécurité

VAP SSR

Institut Mines-Télécom

Auteurs

Durivaux Tony

Boutigny François

Tuteurs

Blanc Grégory

Garcia-Alfaro Joaquin

Outils du projet

python™

FirewallBuilder

La politique de sécurité est un concept clé de la sécurité au sein des organisations, que le standard européen de la sécurité des systèmes d'information (ITSEC) définit comme « l'ensemble de lois, règles et pratiques qui régissent la façon dont l'information et les autres ressources sont gérées, protégées et distribuées à l'intérieur d'un système. »

Enjeu : garantir la sécurité des systèmes d'information

Comment vérifier que le système répond à la politique de sécurité ?

- La définition de la **politique de sécurité** est formelle :
  - Elle vise à donner une référence pour l'ensemble de l'organisation
- Or elle porte sur des systèmes complexes :
  - Les équipements de sécurité assurent différentes fonctions, dans **différents langages**
- Des **modèles de contrôle d'accès** permettent d'aider à l'application de la politique :
  - **OrBAC** (*Organization-based Access Control*) repose par exemple **sur la notion de rôle**
  - Cette notion permet de **distinguer l'identité de l'utilisateur et son rôle dans l'organisation**
- L'implémentation de ce modèle a vu naître une toute nouvelle discipline, le *role engineering* :
  - Elle vise à définir des rôles adéquats au sein d'une entreprise
  - On parle **d'approche ascendante** : on **part de l'existant pour modéliser la globalité du système**

Notre projet : un outil d'aide à la décision

Comment connaître la politique de sécurité à partir de son implémentation ?

Partant d'un *fichier de configuration* d'un pare-feux (qui dépend du constructeur) → Nous le traduisons vers un *langage commun*, capable de décrire des équipements aux fonctionnalités similaires → Nous obtenons ainsi des *vecteurs d'attributs* → Notre programme génère différents *mappings* vers le modèle de contrôle d'accès choisi → Il est alors possible de retrouver la politique de sécurité à partir du modèle de contrôle d'accès

Perspectives : permettre la régénération de politiques

Et ainsi de valider les propriétés de sécurité du système

- Durant ce projet, nous avons implémenté essentiellement la génération des *mappings* entre les *attributs* récupérés dans les *fichiers de configuration* et les *entités concrètes*:
  - Un *attribut* est une valeur ayant attrait au contrôle d'accès, comme l'adresse IP source ou destination, le port source ou destination…
  - Une *entité* est une composante d'un modèle de contrôle d'accès. Pour OrBAC, tout contrôle accès est modélisé par trois entités concrètes: le sujet, l'action, l'objet.
  - Un *mapping* est une association possible entre ces deux notions. Chaque *mapping* représente un paradigme particulier.
  - Toutefois, tous ces paradigmes ne sont pas également pertinents. C'est pour cela que nous aidons l'administrateur réseau à restreindre les possibilités de *mappings* en prenant en compte ses *contraintes,* qui sont propres à sa philosophie de conception dudit équipement.
- Nous avons eu recours à Fwbuilder, un logiciel permettant d'émuler différents pare-feux
  - Celui-ci permettait d'effectuer la traduction dans un langage commun pour les pare-feux
- L'avantage de notre approche est l'obtention de rôles plus pertinentes que dans une démarche ascendante classique
  - Des approches concurrentes considèrent comme élément fondateur la notion de *permission*
  - Alors que nous nous fondons sur les règles présentes dans les *fichiers de configuration*, reflet en principe plus exact de la politique de sécurité déployée

Contacts tony.durivaux@telecom-sudparis.eu | francois.boutigny@telecom-sudparis.eu