Administration des architectures de sécurité réseau

Nora Cuppens-Boulahia, Frédéric Cuppens, Joaquin Garcia-Alfaro

Institut Telecom / Télécom Bretagne 35576 Cesson-Sévigné, France

mots-clés : sécurité réseau, politique de sécurité, anomalies de configuration, déploiement de la sécurité, réaction aux intrusions.

1. Introduction

Plusieurs composants permettent de construire une ASR (Architecture de Sécurité Réseau) tels que les systèmes pare-feu, les modules de détection et de prévention d'intrusion (IDS et IPS) ou les routeurs VPN. Ces composants doivent être correctement configurés et administrés. Ce processus de configuration est cependant hautement complexe et générateur d'erreurs. Il existe donc un vrai besoin de garantir la conformité du processus de gestion de la configuration d'une ASR afin de répondre efficacement aux exigences de sécurité. En effet, dans certaines grandes entreprises, l'ASR peut regrouper plus d'un millier de composants de sécurité réseau. À moins de multiplier les administrateurs de sécurité, on comprend aisément qu'il devient impossible d'administrer manuellement de telles ASR. Cependant, même des ASR réduites constituées d'un ou deux pare-feu peuvent souvent contenir des anomalies de configurations, des vulnérabilités non contrôlées, des failles dans la gestion de la surveillance, etc.

On peut classer les plateformes pour la gestion des ASR en deux grandes catégories : les plateformes mono-constructeur et les plateformes multi-constructeurs. Dans la première catégorie, les produits les plus intéressants sont le Cisco Security Manager et le Check Point Smart Center qui permettent respectivement d'administrer les composants de sécurité réseau commercialisés respectivement par les sociétés CISCO et Check Point. Les fonctionnalités de ces outils sont naturellement fortement contraintes par les caractéristiques des composants de

sécurité CISCO et Check Point que ces outils doivent respectivement administrer. Dans la seconde catégorie, on retrouve des produits tels que IBM Proventia endpoint secure control, Juniper Network and Security Manager, LogLogic Security Change Manager (ex Solsoft NetPartionner) ou Tuffin SecureTrack and SecureChange.

Les principaux objectifs d'une plateforme pour la gestion des ASR sont : une gestion centralisée et cohérente pour exprimer une politique de sécurité globale de contrôle d'accès, une gestion de la configuration des différents composants de l'ASR en conformité avec la politique globale de contrôle d'accès, une assistance pour améliorer et optimiser la politique, une assistance pour détecter et résoudre d'éventuelles erreurs ou anomalies dans la configuration des différents composants de l'ASR, une supervision centralisée des événements générés par les composants de l'ASR, et finalement, une assistance pour gérer la réaction dans l'ASR à travers des modifications de la politique et de la reconfiguration des composants de sécurité.

2. Problématique

Une plateforme pour la gestion d'une ASR (Architecture de Sécurité Réseau) doit offrir une assistance complète à la conception de l'architecture reposant sur la définition d'une politique globale de sécurité réseau et de la topologie du système. Cette fonctionnalité doit s'appuyer sur l'expression et la modélisation de ces deux éléments. Une fois ces données définies, la plateforme doit aider au déploiement technique des configurations. Ce déploiement, fondé sur la connaissance topologique du système et sur les fonctionnalités offertes par chaque composant de sécurité, peut être implémenté à travers la traduction de la politique globale. Une fois les composants déployés, il est nécessaire d'effectuer une analyse périodique des configurations, pour éviter des anomalies introduites dans le système pendant les mises à jour quotidiennes du système. Enfin, la plateforme doit offrir une assistance à l'activation de réactions pour faire face aux attaques auxquelles le système protégé par l'ASR pourrait être soumis. Ces quatre fonctions sont détaillées dans la section suivante.

3. Principales fonctions d'administration de la sécurité réseau

Nous présentons dans cette section les quatre fonctionnalités de base d'une plateforme de gestion idéale. Nous rappelons que l'objectif de cette plateforme est d'aider l'administrateur réseau à obtenir et maintenir la

configuration adéquate des composants de sécurité du réseau, conformément aux exigences de sécurité. Un bref résumé des quatre fonctions envisagées est présenté ci-dessous :

Expression organisationnelle - La plateforme doit fournir les mécanismes pour exprimer formellement la politique qui doit assurer la sécurité du système. De même, elle doit fournir à l'administrateur, dans un langage approprié, la description des informations techniques liées au système, telles que les données topologiques et les capacités des composants. En conséquence, la plateforme doit déterminer si le degré requis de sécurité mis en application par l'architecture de sécurité réseau est cohérent, c'est-à-dire réalisable avec les capacités et les fonctionnalités déployées sur cette l'architecture.

Déploiement technique - Si l'étape précédente est possible, l'administrateur peut définir avec succès la politique adaptée qui pourra gouverner le système compte tenu des éléments qu'il contient. Il doit alors être capable de transmettre automatiquement l'ensemble approprié de règles de configuration qui rendent le système de sécurité opérationnel. Un tel raffinement technique peut être efficacement mis en œuvre par une traduction descendante du modèle abstrait qui contient la politique de sécurité vers la syntaxe concrète de chaque élément contenu dans l'architecture de sécurité réseau.

Analyse des configurations - Le cycle de vie du système de sécurité doit envisager la maintenance des configurations. En d'autres termes, une fois que l'architecture de sécurité réseau est active, des mises à jour régulières peuvent être effectuées pour adapter dynamiquement la configuration des composants à des changements, des révisions des exigences ou d'autres événements connexes. Il devient alors nécessaire que la plateforme puisse offrir à l'administrateur les mécanismes appropriés afin de vérifier que les configurations en cours sont toujours conformes à la politique de sécurité globale courante. Cette fonctionnalité d'analyse doit donc fournir un processus ascendant d'audit de configurations. Celui-ci doit localiser et faire remonter toute configuration incorrecte ou incohérente.

Activation de la réaction - Enfin, il est très probable que l'activité malveillante détectée par la surveillance des éléments de l'architecture de sécurité réseau finira par activer les modules de réaction du système. Dans un tel cas, la plateforme doit fournir les mécanismes nécessaires pour assister l'activation des contre-mesures et mettre à jour la politique de sécurité globale ainsi que la configuration des composants. L'objectif est de fournir

au minimum des réactions à court terme que l'administrateur sera peut-être amené à revoir par des réactions plus efficaces et à plus long terme correspondant à un redéploiement de la politique modifiée vers les composants concernés.

Nous présentons dans la suite une analyse plus approfondie de chaque fonctionnalité, ainsi que des exemples et des solutions issues de la littérature.

3.1. Expression Organisationnelle

L'objectif de cette première fonctionnalité consiste à établir un modèle abstrait des exigences de sécurité qui doivent gouverner le système. La sémantique formelle de ce modèle abstrait doit être unique et sans ambiguïté. Afin de définir ce modèle, il est d'abord nécessaire de disposer d'une description des aspects techniques du système en entrée, tels que la topologie et les capacités fournies par les éléments de sécurité qu'il contient. Une réconciliation entre le degré de sécurité attendu par le système et l'ensemble réel des capacités contenues dans le système doit permettre de conclure si un ensemble unique et non équivoque de règles de sécurité abstraites est en mesure de couvrir toutes les exigences de sécurité identifiées et que les composants de l'ASR peuvent les mettre en œuvre. La figure 1 illustre un schéma simplifié des deux principaux processus nécessaires à la mise en œuvre de cette première fonctionnalité : découverte d'architecture et réconciliation. Nous détaillons ces deux processus dans la suite.

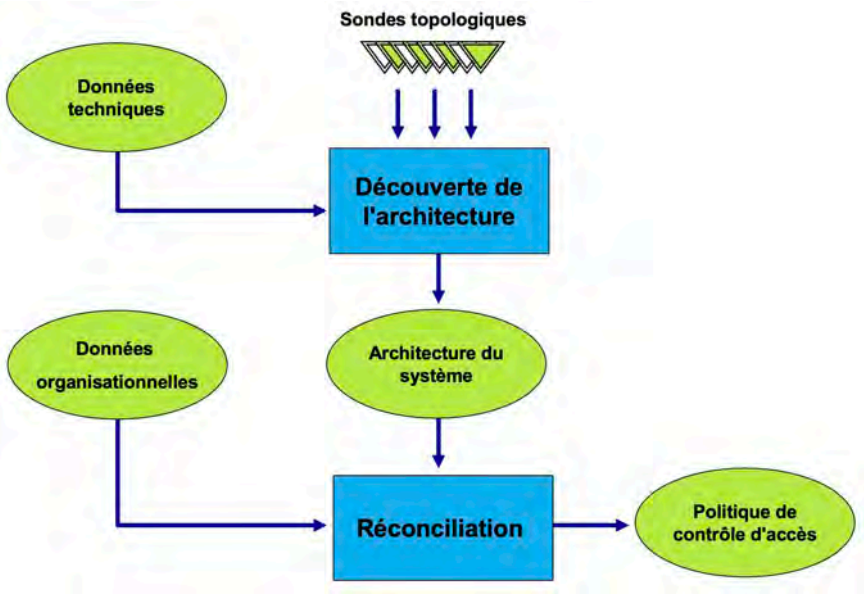

Fig. 1. Expression organisationnelle

Découverte de l'architecture - Le but de ce premier processus consiste à déterminer les données topologiques du système réseau, l'ensemble des composants de sécurité, et des informations associées, telles que les chemins minimaux, itinéraires optimaux, ou des listes pré-calculées de composants traversés par un paquet réseau donné, connaissant sa source et sa destination. Ce processus doit donc permettre l'identification des composants associés à l'ASR, et leurs fonctionnalités techniques, afin de gérer la topologie du système, ainsi que de modéliser les fonctionnalités précises de sécurité de chaque composant (telles que la possibilité de réaliser un filtrage du trafic réseau, de détecter des activités malveillantes, de construire des tunnels IPsec, etc.). Ce processus de découverte repose sur l'utilisation des outils de support réseau existants, par exemple, l'ensemble d'outils de support étudié dans la section 3.4, qui offrent des fonctionnalités pour la découverte du réseau, telles que la collecte des configurations techniques et la collecte de la cartographie topologique du réseau au niveau OSI 2 et 3. Ces outils sont suffisants pour recueillir et ensuite résumer dans un format simple et unique l'architecture du système résultant de ce premier processus.

Réconciliation - Ce deuxième processus doit permettre de spécifier, en se basant sur les caractéristiques organisationnelles et techniques du système, une politique globale de contrôle d'accès. Pour cela, la fonction de réconciliation doit offrir une assistance pour modéliser et mettre à jour les relations qui existent entre les équipements du système et les ressources. L'objectif de ce deuxième processus est donc de valider formellement si les exigences de sécurité de l'organisation peuvent être appliquées avec succès à l'architecture du système. Le cas échéant, un déploiement « best effort » pourrait être envisagé.

Ce processus devrait veiller à ce que l'application des exigences de sécurité, compte tenu de la topologie du système et des fonctionnalités offertes par les éléments de sécurité figurant dans un tel système, soit possible. Il doit faire en sorte que les composants de l'ASR mettent en œuvre des politiques exemptes de conflit et d'anomalies. Il existe, dans la littérature, plusieurs modèles permettant de décrire d'une façon unique, et sans ambiguïté, la politique résultante (voir encart numéro 1). Ainsi, les notions introduites par le modèle RBAC (*Role*

Based Access Control) **[23]** telles que le rôle et la hiérarchie de rôles sont pertinentes pour spécifier une politique de contrôle d'accès réseau. D'autres notions comme l'utilisation de la délégation et l'activation des contextes proposés dans des modèles plus récents comme le modèle OrBAC (*Organization Based Access Control*) **[1]** sont également utiles pour gérer dynamiquement des politiques réseau, notamment pour exprimer les règles de sécurité devant s'activer pour réagir en cas d'intrusion.

3.2. Déploiement technique

Cette seconde fonctionnalité porte sur le raffinement et la décomposition des règles de la politique de sécurité globale en des règles plus concrètes utilisées dans la configuration des composants de sécurité présents dans le système concerné. La figure 2 illustre un schéma simplifié des deux processus principaux de mise en œuvre de cette deuxième fonctionnalité : raffinement technique et mise en application. Nous les détaillons ci-dessous.

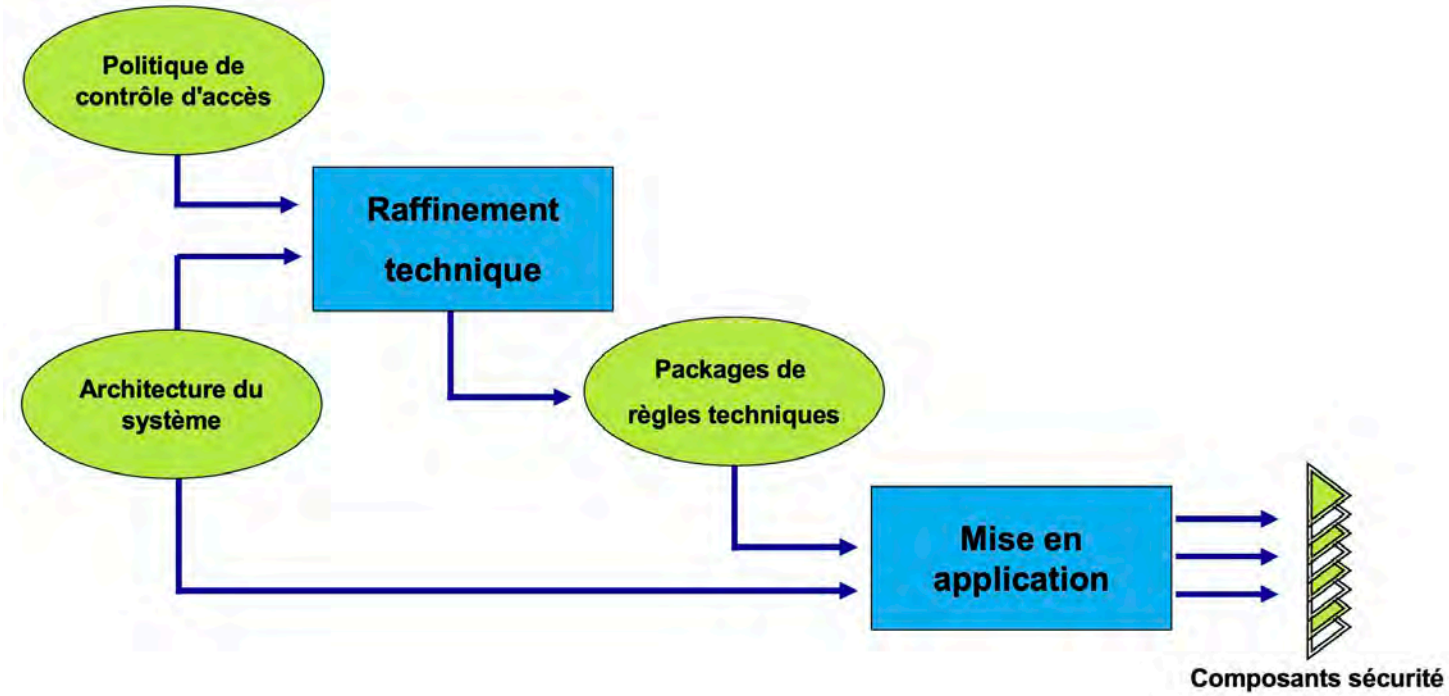

Fig. 2. Déploiement technique

Raffinement technique - Ce premier processus doit définir les mécanismes nécessaires pour traduire, à partir de la politique de contrôle d'accès et l'architecture du système, l'ensemble ou packages de règles techniques

(concrètes) pour configurer automatiquement les composants de sécurité de l'ASR. Les mécanismes de traduction reposent sur la compilation des données organisationnelles de haut niveau, formulées dans la politique de contrôle d'accès, en des règles de configuration concrètes pour chaque composant. Ces règles techniques concernent particulièrement la configuration des pare-feu (à partir des règles de filtrage exprimées dans la syntaxe de chaque pare-feu existant dans le système), des modules de détection d'intrusion (à partir de la production des signatures de détection), etc. Le processus peut être vu comme un ensemble de transformations itératives ou compilations, chacune conforme à une technologie de sécurité donnée instanciée dans le système. Chaque compilation peut être conçue comme l'instanciation de ces parties de la politique de sécurité qui doit être mise en application dans le système au moyen du composant traité.

Plusieurs solutions dans la littérature ont été proposées pour la mise en œuvre de cette fonctionnalité. La principale différence repose sur l'expressivité du modèle ou du langage utilisé pour spécifier les règles abstraites de la politique. Dans **[6]**, par exemple, on peut trouver l'une des premières propositions pour le raffinement technique des politiques de sécurité vers les configurations de pare-feu. L'approche repose sur le modèle RBAC **[23]** et utilise un modèle abstrait qui prend en compte la topologie du réseau sous forme de rôles.

L'utilisation du concept de rôle dans l'approche précédente pour définir les capacités du réseau devient assez rapidement ambiguë et amène les auteurs à faire appel à la notion de groupe pour gérer cette situation. Un groupe peut identifier un ensemble d'hôtes, mais aussi un rôle ou un ensemble de rôles. Son utilisation ne garantit pas une séparation claire entre la modélisation du réseau et l'expression de la politique de sécurité, rendant difficile l'utilisation de cette approche pour modéliser des réseaux complexes. Par ailleurs, les auteurs utilisent l'héritage des privilèges en hiérarchisant les groupes pour dériver automatiquement les autorisations. L'héritage des permissions est lié à un groupe « ouvert », artifice introduit pour éviter la fuite des privilèges, alors que les interdictions sont héritées par un groupe « fermé ». La notion de groupe introduit clairement des ambiguïtés et semble être inutile à ce niveau d'abstraction. Une approche plus complète peut être trouvée dans **[9]**. L'approche repose sur le modèle OrBAC et bénéficie donc de la sémantique inhérente à la délégation et aux contextes dans le but de remédier aux limitations précédentes. Dans **[22]**, cette approche est améliorée et permet le déploiement d'un plus grand ensemble de composants, non seulement les pare-feu, mais aussi des modules de détection d'intrusion et routeurs VPN.

Mise en application - Ce deuxième processus offre les mécanismes nécessaires pour faire la distribution de chaque package de règles sur le composant de sécurité approprié. Cela peut être fait de façon manuelle par l'administrateur ou automatiquement par l'utilisation d'un protocole de gestion des composants réseau. Plusieurs protocoles de communication peuvent répondre à ce besoin. Par exemple, le protocole SNMP (*Simple Network Management Network*), les protocoles COPS (*Common Open Policy Server*) et Netconf sont des solutions présentes dans la plupart des outils de support actuels (voir section 3.4 pour plus de détails).

Le protocole SNMP est l'un des protocoles les plus largement utilisés par les outils actuels de gestion réseau. Ce protocole repose sur un paradigme traditionnel d'agent-manager, dans lequel l'agent est en fait un programme qui s'exécute sur les équipements d'un système, et le manager est un programme centralisé au niveau d'une plateforme qui contrôle et rassemble les résultats d'exécution de chaque agent déployé dans le système. La plupart des solutions mono-constructeur de fournisseurs bien connus comme Cisco ou Check Point basent la synchronisation et la gestion de leurs outils sur SNMP. En revanche, les protocoles COPS et Netconf sont essentiellement de nature requête-réponse. Ils sont très utilisés pour échanger des informations sur les politiques entre serveurs et clients. COPS a été utilisé dans **[7]**, par exemple, pour distribuer dynamiquement des politiques de type VPN IPsec. La principale limitation soulignée dans **[15]** est le traitement du format XML par COPS, car il n'est pas supporté de manière native. Au contraire, Netconf supporte XML de manière native. Une approche récente présentée dans **[22]** propose la combinaison du formalisme OrBAC et du protocole Netconf, afin de déployer et de communiquer les configurations des composants de sécurité sur le réseau.

Expression de la politique de sécurité réseau

Le modèle le plus simple pour exprimer une politique de contrôle d'accès est le modèle DAC (*Discretionary Access Control*). Dans DAC, une politique de contrôle d'accès est exprimée sous forme de triplets <sujet, action, objet>. Chaque triplet représente une permission pour une entité active (le sujet) de réaliser une action sur une entité passive (l'objet). Dans le cas d'une politique de contrôle d'accès réseau, une interprétation possible de DAC proposée dans **[9]** est de considérer que (1) le sujet correspond à un machine hôte identifiée par son adresse IP qui émet des paquets, (2) l'action correspond à l'activation d'un protocole réseau tel que HTTP ou FTP, par exemple, et (3) l'objet correspond à une autre machine hôte également identifiée par son adresse IP

qui reçoit les paquets émis par le sujet. Par exemple, le triplet <111.222.1.1, http, 111.222.2.1> spécifie que la machine hôte d'adresse 111.222.1.1 a la permission d'activer le protocole HTTP vers la machine 111.222.2.1.

L'objectif est ensuite de déployer la politique de contrôle d'accès sur les composants de sécurité constituant une ASR. Dans le cas de DAC, il s'agirait notamment de traduire chacun des triplets dans une liste de contrôle d'accès (ACL) supportée par un composant de sécurité de l'ASR. Par exemple, si l'ASR contient un composant correspondant à un pare-feu Netfilter, on obtiendra l'ACL suivante :

iptables -A FORWARD -s 111.222.1.1 –d 111.222.2.1 -p tcp --dport 80 -j ACCEPT

À noter que la permission d'activer le protocole HTTP a été traduite par une permission d'activer TCP sur le port destination 80.

Le principal inconvénient de DAC est d'exiger la spécification des permissions pour chaque sujet et objet. Dans le cas d'un réseau, cela revient à spécifier des ACL pour chaque couple de machine hôte et destination, ce qui conduit rapidement à une explosion du nombre d'ACL à exprimer et rend impossible l'administration de la politique de contrôle d'accès.

Face aux limites de DAC, des modèles plus récents comme le modèle RBAC (*Role Based Access Control*) **[22]** ont été proposés. Ce modèle propose de raisonner non pas sur l'identité des sujets, mais sur les rôles que ce sujet joue dans l'organisation. Dans le cas d'une politique de contrôle d'accès réseau, des exemples de rôles pourraient être « Web-server », « DNS-server », « DMZ » (représentant le rôle des machines appartenant à la DMZ) ou « Private » (représentant le rôle des machines appartenant à la zone privée de l'entreprise). En utilisant RBAC, on peut par exemple spécifier la règle « la zone privée a la permission d'activer le protocole HTTP avec la DMZ » sous la forme suivante : Permission(Private, http, DMZ).

Si l'on suppose que les machines des zones 111.222.1.0/24 et 111.222.2.0/24 sont respectivement affectées aux rôles Private et DMZ, alors cette permission sera traduite en l'ACL suivante pour NetFilter :

iptables -A FORWARD -s 111.222.1.0/24 –d 111.222.2.0/24 -p tcp --dport 80 -j ACCEPT

Cependant, même si RBAC présente de nombreux avantages par rapport à DAC, l'expression d'une politique de contrôle d'accès dans ce modèle reste « statique ». Il est ainsi impossible de spécifier qu'une permission dépend

de conditions contextuelles, par exemple une condition temporelle ou de géo-localisation. Face à ces limites, le modèle de contrôle d'accès OrBAC (*Organization Based Access Control*) **[1]** introduit plusieurs concepts tels que les contextes ou la gestion de la délégation des permissions. Par exemple, la règle « la DMZ a la permission d'activer le protocole HTTP avec la zone privée entre 8 heures et 18 heures » aura en OrBAC la forme suivante : Permission(DMZ, http, Private, after(8:00) & before(18:00)). Dans cette règle, l'expression « after(8:00) & before(18:00)) » est un contexte OrBAC « composé » constitué de la conjonction de deux contextes temporels « after(8:00) » (après 8 heures) et « before(18 :00) » (avant 18 heures).

La traduction de cette permission dans la syntaxe de NetFilter correspondra à l'ajout de la contrainte suivante :

iptables -A FORWARD -m time --timestart 8:00 –timestop 18:00

On peut considérer d'autres types de contextes, par exemple « protected » (pour spécifier que le trafic n'est permis que via la construction d'un VPN), « established » (pour contrôler l'état des sessions comme dans le cas des pare-feu « stateful ») ou « flooding » (pour spécifier que la règle s'active lorsqu'une attaque par inondation est détectée). Cette dernière possibilité est intéressante pour activer les règles permettant de réagir aux intrusions (voir section 3.4).

3.3. Analyse des configurations

L'accroissement actuel du nombre d'attaques réseau amène à une mise à jour périodique du système. Cela peut affecter tant la politique de sécurité globale que les configurations locales des composants. La plateforme de gestion doit donc fournir les mécanismes nécessaires pour garantir que les modifications ne mènent pas à une mauvaise configuration du système. Autrement dit, elle doit vérifier que les règles effectivement déployées sur les composants de sécurité de l'architecture sont toujours compatibles avec la politique de sécurité globale après mise à jour. La plateforme doit aider l'administrateur à analyser, par un processus d'audit approprié, les configurations locales et reporter l'existence de mauvaises configurations ou d'incohérences avec la politique globale. L'objectif est donc la production des rapports d'audit des anomalies de configuration (détection des anomalies). La plateforme doit proposer aussi les modifications nécessaires qui permettraient de résoudre de tels problèmes - soit manuellement, en demandant l'approbation de l'administrateur, soit automatiquement, en

reconfigurant les composants affectés sans exiger de nouvelles actions.

L'analyse doit détecter et résoudre les anomalies potentielles de configuration de chaque composant de l'ASR et mettre en correspondance les configurations des différents composants pour valider de façon globale l'application correcte de la politique de sécurité, la cohérence et la non redondance des règles déjà déployées. La figure 3 décrit un schéma simplifié des processus principaux pour la mise en œuvre de cette troisième fonctionnalité : collecte, audit mono-composant, audit multi-composant et gestion des modifications. Nous les détaillons maintenant.

Collecte - Ce processus vise à rassembler les différentes règles de configurations locales déployées sur chaque composant de l'architecture de sécurité réseau. Comme dans le processus de découverte d'architecture, nous supposons ici que des outils support actuellement disponibles sur le marché, tels que ceux que nous présentons plus loin, fournissent les fonctionnalités nécessaires pour configurer l'ensemble des composants du réseau et permettre d'extraire les fichiers de configurations locales et les réunir dans un répertoire centralisé. Il s'agit ensuite de les réécrire dans un format unique pour permettre l'analyse.

Analyse mono-composant - Ce processus est destiné à détecter et résoudre des conflits entre des règles de configuration pour chaque composant de manière isolée. Le résultat dérivé par cette fonction contiendra un nouvel ensemble de règles, qui ont été vérifiées et qui devront être exemptes d'anomalies, et les informations correspondant aux éventuels problèmes qui auront été détectés.

Plusieurs solutions pour mettre en œuvre ce processus existent dans la littérature. Cependant, ces solutions s'intéressent seulement au processus d'audit. Elles ne proposent pas un choix de nouvelles règles pour remplacer les conflits. De plus, aucun des travaux mentionnés auparavant ne présente des mécanismes spécifiques pour vérifier des configurations autres que des configurations des systèmes pare-feu. À notre connaissance, seuls les travaux proposés dans **[10][19]** couvrent ces limitations. Ces travaux considèrent l'audit des configurations des systèmes pare-feu et des systèmes de détection d'intrusion, détectant et gérant des anomalies telles que la redondance et le *shadowing*. Les configurations sont traitées dans leur ensemble et pas seulement en regardant les relations entre les règles prises deux à deux. Grâce à un processus de transformation de règles, une première série de règles de configuration est dérivée, une série équivalente et

valide, qui est complètement exempte de toute anomalie. Cette réécriture de règles permet de plus la génération d'un nouvel ensemble de configurations dont les règles sont complètement disjointes. Par conséquent, l'ordre des règles n'est plus pertinent.

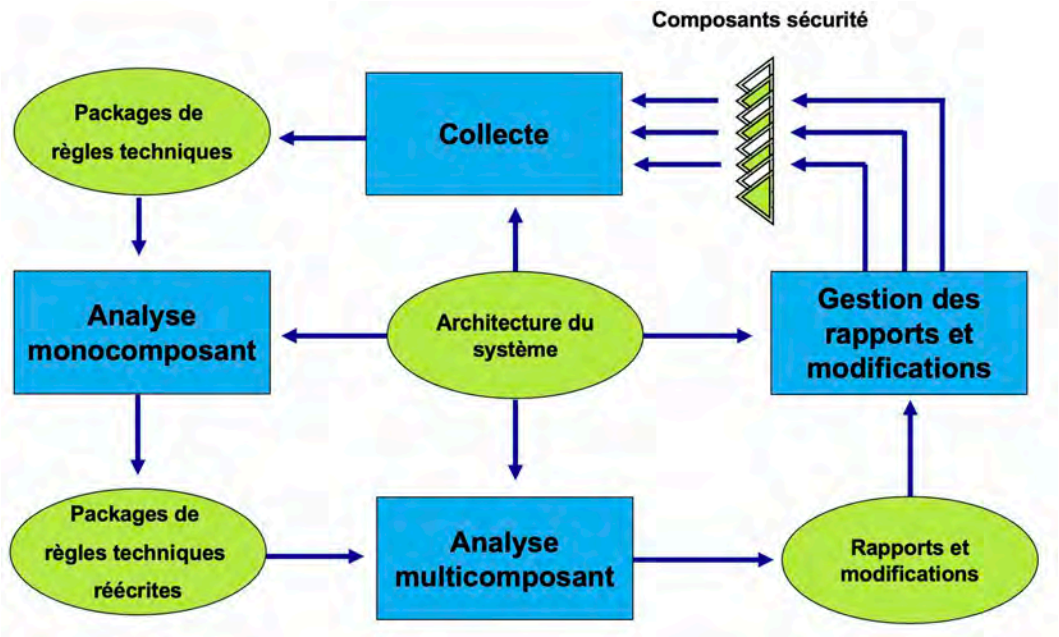

Fig. 3 : Analyse de configuration

Analyse multi-composant - Ce troisième processus, complété par la base de connaissance de l'architecture du système, vérifie que les configurations de chaque composant du réseau, d'un point de vue distribué, sont compatibles avec la politique globale. Il doit garantir par ailleurs l'interopérabilité entre les différents composants. Les propositions comme **[4][21]** étendent l'analyse mono-composant présentée dans **[3][19]**, pour détecter des déploiements multi-composants qui ne sont pas cohérents. L'approche présentée dans **[4]** fournit un processus de corrélation qui compare les règles de configuration d'architectures distribuées et en dérive les incohérences cachées dans les configurations des composants de sécurité. Le processus de détection repose de nouveau sur la comparaison de règles deux par deux. Donc les erreurs en raison de l'union de plus de deux règles ne sont pas correctement traitées. L'approche présentée dans **[21]** résout cette limitation et permet de plus l'audit d'architectures distribuées où des pare-feu et des systèmes de détection d'intrusion sont responsables de la mise en application de la politique globale. La détection vérifie s'il existe ou subsiste des erreurs dans les

configurations distribuées. Pour cela, le processus compare la configuration de chaque composant avec la politique, vérifie si elle correspond exactement aux décisions prévues par la politique et génère les rapports d'anomalies le cas échéant.

En ce qui concerne l'analyse des configurations des routeurs VPN, une approche pertinente est celle présentée dans **[16]**. Les auteurs fournissent une technique qui simule le traitement de *tunneling* VPN et qui détecte les violations des exigences de la politique de sécurité. Par exemple, dans leur approche, si une règle d'accès relative à un trafic protégé entre deux points est mise en œuvre par des configurations sans superposition de plus d'un tunnel VPN, les paquets IP peuvent, dans certaines zones du réseau, circuler sans aucune protection. Les auteurs présentent un processus pour détecter de telles situations et proposent un langage de haut niveau pour analyser les politiques VPN. Une limitation importante de cette technique de découverte d'anomalie de configuration est que, même si ce processus peut découvrir certaines violations dans certains scénarios de simulation, il n'existe aucune garantie qu'il découvre toute mauvaise configuration présente dans le système. De plus, la technique proposée détecte seulement des conflits de VPN résultant de superpositions incorrectes de tunnels, mais aucune taxonomie complète de tous les conflits possibles n'a été développée et étudiée. Une autre tentative d'audit et de réparation des mauvaises configurations de routeurs VPN est présentée dans **[5]**. Malheureusement, les algorithmes proposés restent non évalués.

Gestion des rapports et modifications - L'ensemble complet des résultats et des modifications proposées par les processus précédents et destinés à « sanitiser » les mauvaises configurations doit être centralisé et communiqué à l'administrateur. La plateforme de gestion doit aussi laisser ouverte la possibilité soit de simplement rendre compte des résultats des analyses, soit automatiquement appliquer les modifications proposées. Dans le premier cas, cela signifie que la plateforme devra seulement informer l'administrateur des résultats des analyses effectuées. L'administrateur est donc responsable de la vérification des rapports et doit activer manuellement les modifications appropriées. L'option alternative est d'accepter directement les modifications proposées et de mettre en vigueur la sécurité des composants qu'exigerait une telle reconfiguration. Dans les deux cas, un protocole de communication tel que SNMP, COPS ou Netconf, vus précédemment, est nécessaire pour appliquer les modifications.

3.4. Activation des réactions

L'objectif de cette dernière fonctionnalité est l'activation de la réaction. Nous supposons ici l'utilisation de composants appropriés, tels que des systèmes de détection et de prévention d'intrusions (IDS ou IPS) ou des systèmes de gestion des événements et informations de sécurité (SIEM), pour la gestion d'événements et d'alertes. La réaction repose, par exemple, sur l'analyse de descriptions d'attaques définies comme des actions qui violent la politique globale. Nous supposons également que ces composants peuvent anticiper l'occurrence des scénarios d'attaque possibles, et fournissent le diagnostic approprié pour mettre en place ou déclencher des contre-mesures et neutraliser les attaques. Par exemple, si les attaques ont été spécifiées dans la politique comme des interdictions, l'accomplissement d'une attaque doit être interprété comme une violation de la politique de sécurité et détecté par la surveillance de l'ensemble des événements collectés par les composants de sécurité de l'ASR. Les événements et les alertes sont donc traités par un système de surveillance. La figure 4 représente un schéma simplifié des principaux processus pour la mise en œuvre de cette quatrième fonctionnalité : surveillance, mise à jour de la politique et redéploiement des configurations des composants de l'ASR. Nous détaillons ces processus dans ce qui suit.

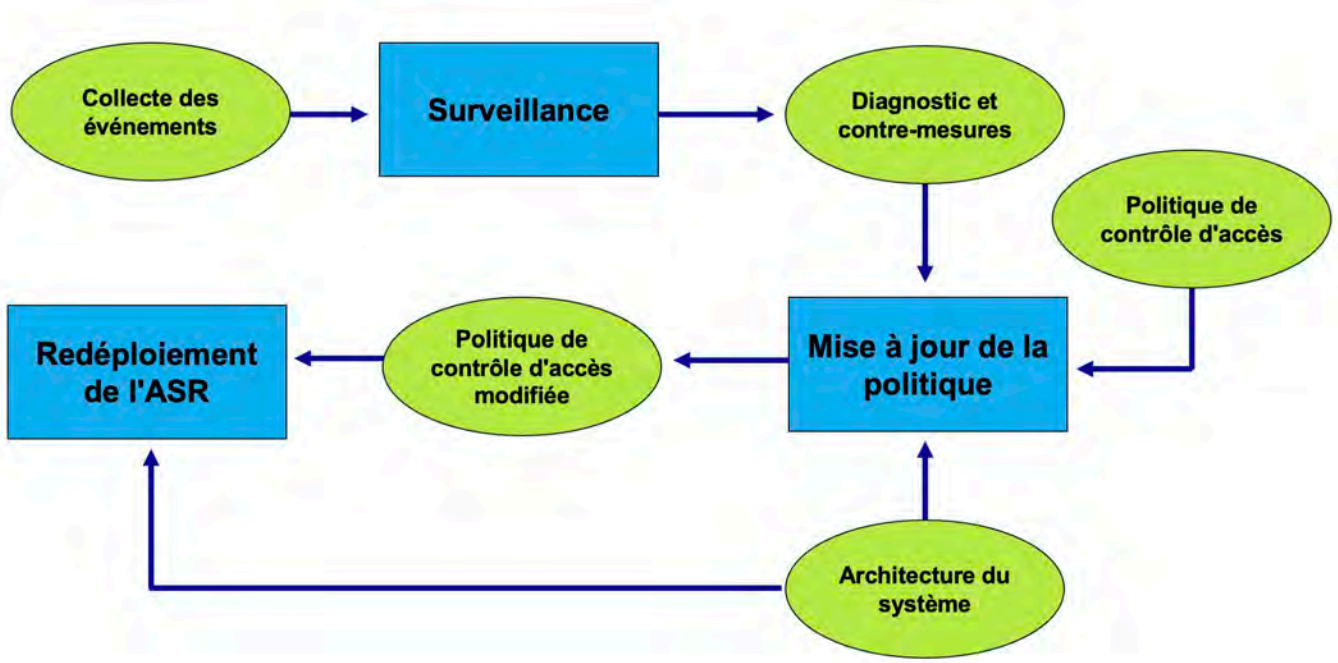

Fig. 4. Activation des réactions

Surveillance - Ce processus doit fournir les mécanismes nécessaires pour produire des diagnostics et choisir des contre-mesures appropriées lorsque des activités malveillantes sont détectées par les composants de sécurité de l'ASR. Le but est de fournir à l'administrateur ou à la plateforme de gestion un ensemble approprié de contre-mesures concrètes. Ces contre-mesures ont été conçues sur la base d'un mécanisme de réponse qui met à jour la politique de sécurité globale en activant, par exemple, des conditions déjà définies dans la politique initiale. Il existe plusieurs propositions dans la littérature pour la mise en œuvre d'un tel processus. Nous recommandons au lecteur les références **[8][12][13]** pour plus d'informations.

Mise à jour de la politique - Ce deuxième processus doit offrir les mécanismes nécessaires pour faire la mise à jour de la politique de contrôle d'accès. Cette fonction établit le lien entre la politique de sécurité effectivement déployée et les contre-mesures déclenchées par la fonction de surveillance, afin de réagir aux violations détectées. Cette fonction permet de gérer de façon dynamique une mise à jour de la politique. Celle-ci assure la mise en place d'une réaction à court terme. Par exemple, fermer l'accès aux services et aux ressources en fonction des attaques détectées, en attendant que les administrateurs de sécurité activent ensuite des réactions à long terme.

Des solutions existent dans la littérature, qui traitent de la mise en œuvre de ce processus. Par exemple, les propositions présentées dans **[17][24]** fournissent une solution de mise à jour des politiques de filtrage réseau (c'est-à-dire des configurations de systèmes pare-feu) activée à partir d'un traitement des alertes par des modules de détection d'intrusion. Un serveur central, qui contrôle la politique et reçoit les alertes d'intrusion, est chargé de décider des modifications nécessaires afin de réagir aux menaces détectées. Malheureusement, ces solutions ne fournissent pas une indication claire entre la stratégie globale de réponse et l'activation des contre-mesures concrètes. Deux autres solutions pertinentes ont été présentées dans **[12][13]**. Dans **[12]**, le lien entre alertes et contre-mesures est défini comme un *mapping* entre des événements exprimés dans le modèle IDMEF **[11]** et des contextes OrBAC **[1]**. Les contextes OrBAC représentent la réaction appropriée qui doit être mise en application quand les événements IDMEF sont détectés par le système de surveillance. Dans **[12]**, les auteurs fournissent de plus le processus nécessaire d'instanciation spécifique pour mettre à jour la politique globale suite

aux alertes fournies par le processus de surveillance.

Redéploiement de l'ASR - Ce dernier processus doit offrir à l'administrateur les mécanismes nécessaires pour mettre en vigueur la configuration des composants de sécurité, selon la politique de sécurité précédemment mise à jour. Nous supposons que cette fonction est équivalente à la fonction de déploiement présentée dans la section concernant le déploiement technique.

Anomalie de configuration

Dans l'encadré précédent, nous avons montré comment spécifier une politique de contrôle d'accès réseau sous forme d'un ensemble de permissions qui sont ensuite traduites dans la syntaxe des différents composants de l'ASR. Dans la pratique, il est commode d'exprimer la politique en combinant des permissions et des interdictions correspondant respectivement à ces ACL « accept » et « deny ». Par exemple :

- iptables -A FORWARD -s 111.222.1.1 –d 111.222.2.1 -p tcp --dport 80 -j DENY
- iptables -A FORWARD -s 111.222.1.0/24 –d 111.222.2.0/24 -p tcp --dport 80 -j ACCEPT

Dans ce cas, un conflit entre les règles apparaît, puisque les paquets analysés par la première règle peuvent être analysés par la seconde. Dans la plupart des pare-feu, le conflit est résolu en ordonnant les règles et en considérant que la première règle est plus prioritaire que la seconde (stratégie dite de « first matching »). Cependant, l'application de la stratégie de *first matching* peut introduire des anomalies. Ainsi, si l'on change l'ordre des deux règles ci-dessus, alors la règle « deny » ne pourra jamais s'appliquer. On parle alors d'anomalie de masquage (« shadowing »). Un autre exemple d'anomalie est la redondance, c'est-à-dire l'existence d'une ACL qui peut être supprimée sans changer la politique de contrôle d'accès. Par exemple, un problème de redondance apparaît si l'on modifie la première ACL de l'exemple ci-dessus en changeant la décision « deny » en « accept ».

Dans **[2]**, on peut trouver l'une des premières solutions pour détecter les anomalies de configuration dans les pare-feu. Dans **[3]**, les auteurs fournissent une taxonomie très complète d'anomalies et un jeu d'algorithmes pour les détecter en analysant, comme proposé dans **[2]**, les relations entre couples de règles. La limitation de

ces deux approches est que les anomalies dues à l'union de plus de deux règles ne sont pas explicitement prises en compte. La proposition présentée dans **[25]** utilise une solution basée sur le *model checking* pour détecter des incohérences et des redondances dans des configurations de pare-feu. Cette nouvelle proposition traite le processus de détection en abordant directement la façon dont les trafics sont assurés par les composants.

4. Synthèse de l'offre commerciale

Le marché actuel d'outils supports pour l'administration d'une ASR (Architecture de Sécurité Réseau) est dominé par les deux leaders industriels Cisco et Check Point. Leurs produits les plus intéressants sont le Cisco Security Manager et le Check Point Smart Center. Ces outils permettent l'expression de l'architecture et de la politique globale du système dans un langage de niveau relativement bas (trop proche de celui des outils à gérer). En revanche, ces deux outils offrent un ensemble très complet de solutions pour la gestion d'événements et d'alertes d'intrusions, l'automatisation du déploiement des configurations des composants et la gestion des routeurs VPN. Ces outils offrent des fonctions complémentaires pour la découverte de l'architecture réseau et la collecte des configurations techniques (par exemple, collecte de la cartographie du réseau au niveau OSI 2 et 3). Ils fournissent également, en plus des activités traditionnelles de gestion de la sécurité, le stockage de l'historique des logiciels, ainsi que la mise à jour des logiciels et des modifications des configurations techniques. Enfin, ces outils offrent certaines fonctions pour la gestion de la tolérance aux pannes, telles que la centralisation des opérations associées et la gestion des *workflows*.

Nous pouvons toutefois souligner certaines limitations dans l'utilisation de ces outils. Tout d'abord, et le plus important, ces solutions sont mono-constructeur. Elles ne fournissent que les opérations mentionnées ci-dessus afin d'aider des administrateurs clients de Cisco et Check Point uniquement. Par ailleurs, elles n'offrent pas de modèle sémantique assez riche pour exprimer une politique de sécurité globale complète qui pourrait intégrer des notions comme les contextes et les hiérarchies. Bien qu'il soit possible de définir certaines variables et exprimer ainsi des règles impliquant de telles variables, les tâches de l'administrateur n'en seront pas beaucoup plus simplifiées. L'administrateur doit toujours avoir une vue globale de la topologie afin d'affecter correctement chaque règle à des composants réseau. En effet, ils ne disposent d'aucun mécanisme de découverte

automatique des dispositifs de sécurité qui mettent en application de façon optimale le lien entre les règles d'accès et le trafic. En outre, ces outils ne sont pas dotés de fonctionnalités permettant une vraie démarche d'analyse ascendante comme celle proposée dans la section 3.2 portant sur le déploiement technique vu précédemment. Celle-ci peut être partiellement remplacée par d'autres instruments (par exemple certains outils de découverte de conflits développés par Cisco) qui nécessitent l'assistance de l'administrateur chargé de la sécurité et qui garantissent seulement la résolution de conflit dans les configurations locales.

Les fournisseurs IBM, Juniper Networks et Tufin offrent également des solutions intéressantes. Ils fournissent des outils d'administration un peu plus généraux, comme IBM Proventia Desktop Endpoint Security, Juniper Network and Security Manager et Tufin SecureTrack and SecureChange. Le principal avantage de ces solutions est qu'elles ne sont pas de type mono-construteur. En effet, elles peuvent interagir avec des composants d'autres fournisseurs (par exemple des pare-feu, des routeurs VPN et des systèmes de détection d'intrusion fabriqués par d'autres sociétés que Cisco et Check Point). Par rapport aux solutions de Cisco et Check Point, ces outils tentent d'intégrer des normes plus ouvertes, spécialement des normes reposant sur XML et SOAP, afin de faciliter l'intégration des futures solutions. Alors que la définition des politiques par les solutions d'IBM et Juniper repose sur RBAC **[23]**, la solution de Tufin utilise encore un modèle de niveau relativement bas, c'est-à-dire un modèle trop proche de la syntaxe et de la sémantique d'origine des composants destinés à être administrés. D'autres extensions offertes par ces solutions sont : le traitement des systèmes d'exploitation des composants, la gestion de l'analyse des menaces et du workflow des opérations de *rollback*. Cette dernière fonctionnalité permet de restaurer des configurations de l'architecture précédente en cas de pannes ou d'attaques.

Au niveau français, l'éditeur principal de solutions pour l'administration d'architectures de sécurité réseau était la société Solsoft, avec les outils Solsoft Policy Server et Solsoft Network Security Policy Server. Suite à la fusion de Solsoft avec la société Exaprotect qui a été ensuite rachetée par la société américaine LogLogic en 2009, il n'existe aujourd'hui aucun leader français pour offrir des solutions d'administration d'architectures de sécurité de réseau.

Enfin, il convient de mentionner l'existence de solutions pertinentes open source comme le produit Firewall

Builder. Cette solution est fournie avec une licence GPL open source pour son utilisation sur des systèmes Unix tels que GNU/Linux et FreeBSD et une licence commerciale pour son utilisation sur des systèmes commerciaux tels que MS-Windows et Apple Mac OS. Bien que cette solution permette seulement la gestion des configurations des systèmes pare-feu, elle offre néanmoins une assistance très complète pour la gestion de solutions de différents fournisseurs et le déploiement de grandes configurations sur des réseaux hétérogènes. La principale limitation de cette solution est, cependant, la définition d'une politique centralisée de la sécurité dans un langage de définition de politique relativement pauvre.

5. Limites des outils actuels et perspectives

Nous avons déjà évoqué dans la section précédente quelques problèmes liés à la mise en œuvre des outils actuels pour la gestion des architectures réseau. Nous examinons, dans cette section, certaines de ces limites en termes de pertinence et nous résumons certains domaines d'études importants pour aller vers une plateforme de gestion « idéale ».

Avant de clore cette section, nous considérons deux perspectives intéressantes qui méritent d'être explorées afin d'améliorer significativement l'efficacité des solutions commerciales que nous avons présentées. Un domaine peu exploré dans la plupart des solutions étudiées est l'intégration d'un processus guidé par l'analyse ascendante des configurations techniques avec une détection automatique des rôles associés aux différents composants de sécurité déjà déployés dans le système. L'utilisation, par exemple, de techniques de *role mining* **[18]** faciliterait les tâches d'administration en découvrant les rôles de contrôle d'accès associés aux composants réseau pour obtenir, après l'analyse, les règles constituant la politique globale. Une autre perspective intéressante est l'inclusion de la gestion du workflow des opérations associées à l'ASR. En effet, les architectures à grande échelle pourraient largement bénéficier de ce type de gestion automatique pour permettre une mise en œuvre plus efficace des activités d'administration quotidienne, qui devrait comprendre des actions telles que l'inclusion de nouvelles règles et la suppression des conditions inutiles. Le défi est de trouver un lien effectif entre la gestion du workflow et le déploiement des politiques, afin d'optimiser l'exécution des opérations d'administration dans leur ensemble. Dans ce contexte, on peut mentionner le module Tufin SecureChange Workflow de gestion des demandes de mise à jour des politiques de sécurité. Ce module propose de gérer le

cycle de vie d'une requête, de la soumission à l'audit, en passant par la conception, l'analyse des risques induits par la mise à jour, l'approbation et la mise en œuvre. Enfin, des prototypes opérationnels issus de solutions académiques, tels que MIRAGE **[20]**, offrent plusieurs fonctionnalités à la fois : l'expression formelle cohérente et sans conflits des exigences de sécurité réseau, son déploiement sur l'ASR et l'analyse des anomalies de configurations, avec l'appui d'un processus de *mining* sur les données de configuration pour contrôler la politique de sécurité déployée et effectuer la réconciliation avec la politique initiale.

Conclusion

L'objectif d'une plateforme pour la gestion d'une ASR est de simplifier le travail d'un administrateur ayant à assurer la gestion d'un système d'information, et spécialement ses composants de sécurité. La plateforme doit fournir les fonctions suivantes : (1) assistance à la conception de l'ASR, (2) assistance à la configuration automatique des composants de l'ASR, (3) analyse et production des rapports d'audit des anomalies de configurations et (4) assistance à l'activation de la réaction de l'ASR en cas de malveillance ou d'anomalie. L'étude de l'offre commerciale actuelle nous a conduits à signaler les limites suivantes : le manque de sémantique formelle pour la description des politiques, le manque d'analyse des configurations de composants avec état (*stateful*) **[14]** et la dépendance élevée des solutions mono-constructeur. Nous avons identifié essentiellement deux perspectives : l'intégration d'un processus de role mining pour compléter l'analyse ascendante des configurations techniques et l'inclusion de la gestion du workflow des opérations pour permettre une application plus efficace des activités quotidiennes d'administration de l'ASR.

Remerciements

Nous remercions la Direction Générale de l'Armement pour le soutien qu'elle nous a apporté dans la réalisation de cette étude, menée dans le cadre du PEA SuperSec sur les problématiques relatives à la spécification et à la définition d'un dispositif de supervision et d'administration de la sécurité d'un système, Marché N° 2008-99-0908 pour la DGA/UM ESIO.

Bibliographie

[1] Anas Abou El Kalam, Rania El Baida, Philippe Balbiani, Salem Benferhat, Frédéric Cuppens, Yves Deswarte, Alexandre Miège, Claire Saurel, Gilles Trouessin : Organization based access control. POLICY 2003.

[2] H. Adiseshu, S. Suri, and G. Parulkar. Detecting and resolving packet filter conflicts. Joint Conference of the IEEE Computer and Communications Societies, pp. 1203–1212, 2000.

[3] E. S. Al-Shaer and H. H. Hamed. Firewall Policy Advisor for Anomaly Discovery and Rule Editing. In Proceedings of the 8th IFIP/IEEE International Symposium on Integrated Network Management (IM 2003), pp. 17–30, 2003.

[4] E. S. Al-Shaer and H. H. Hamed. Taxonomy of Conflicts in Network Security Policies. In IEEE Communications Magazine, 44(3), March, 2006.

[5] S. Baek, M. Jeong, J. Park, T. Chung. Policy based Hybrid Management Architecture for IP-based VPN. In Network Operations and Management Symposium, NOMS 2000.

[6] Y. Bartal, A. Mayer, K. Nissim, A. Wool. Firmato: A novel firewall management toolkit. In 20th IEEE Symposium on Security and Privacy, Oakland, California, 1999.

[7] F. Clemente, G. Lopez, G. Martinez, and A. Gomez-Skarmeta. Deployment of a Policy-Based Management System for the Dynamic Provision of IPsec-Based VPNs in IPv6 Networks. In 2005 Symposium on Applications and the Internet Workshops, pp.10–13, 2005.

[8] F. Cuppens, F. Autrel, Y. Bouzida, J. Garcia-Alfaro, S. Gombault, and T. Sans. Anti-correlation as a criterion to select appropriate counter-measures in an intrusion detection framework. Annals of Telecommunications, 61(1-2):192–217, 2006.

[9] F. Cuppens, N. Cuppens, T. Sans, and A. Miège. A formal approach to specify and deploy a network security policy Second Workshop on Formal Aspects in Security and Trust, Toulouse, France, August 2004, pp. 203–218.

[10] F. Cuppens, N. Cuppens, and J. Garcia-Alfaro. Misconfiguration management of network security components. 7th International Symposium on System and Information Security (SSI 2005), Sao Paulo, Brazil, November 2005, pp. 1–10.

[11] H. Debar, D. Curry, and B. Feinstein. Intrusion detection message exchange format data model and extensible markup language. Request for Comments 4765, March 2007.

[12] H. Debar, Y. Thomas, F. Cuppens, and N. Cuppens-Boulahia. Enabling Automated Threat Response through the Use of a Dynamic Security Policy. In Journal in Computer Virology (JCV), 3(3):195- 210, 2007.

[13] H. Debar, Y. Thomas, F. Cuppens, and N. Cuppens-Boulahia. Using contextual security policies for threat response. In Third GI International Conference on Detection of Intrusions & Malware, and Vulnerability Assessment (DIMVA), Berlin, Germany, 2006.

[14] W. M. Fitzgerald, M. O. Foghlu, and S. N. Foley. Network access control configuration management using semantic web techniques. Journal of Research and Practice in Information Technology, 41(2):99–118, 2009.

[15] T. Franco, W. Lima, G. Silvestrin, R. Pereira, M. Almeida, L. Tarouco, L. Granville, A. Beller, E. Jamhour, and M. Fonseca. Substituting COPS-PR: An Evaluation of NETCONF and SOAP for Policy Provisioning. In 7th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'06), pp. 195–204, USA, 2006.

[16] Z. Fu, et al. IPSec/VPN Security Policy: Correctness, Conflict Detection and Resolution. International Policy Workshop. January 2001.

[17] K. Hwang and M. Gangadhran. Micro-Firewalls for Dynamic Network Security with Distributed Intrusion Detection. In International Symp. on Network Computing and Applications, 2001.

[18] M. Kuhlmann, D. Shohat, and G. Schimpf. Role mining-revealing business roles for security administration using data mining technology In Eighth ACM symposium on Access control models and technologies, 2003.

[19] J. Garcia-Alfaro, F. Cuppens, and N. Cuppens-Boulahia. Towards Filtering and Alerting Rule Rewriting on Single-Component Policies. In Intl. Conference on Computer Safety, Reliability, and Security (Safecomp 2006), pp. 182–194, Gdansk, Poland, 2006.

[20] Joaquín García-Alfaro, Frédéric Cuppens, Nora Cuppens-Boulahia, Stere Preda. MIRAGE: A Management Tool for the Analysis and Deployment of Network Security Policies. DPM/SETOP 2010: 203-215

[21] J. Garcia-Alfaro, F. Cuppens, and N. Cuppens-Boulahia. Complete Analysis of Configuration Rules to Guarantee Reliable Network Security Policies. International Journal of Information Security, Springer, 7(2):103–122, April 2008.

[22] S. Preda, F. Cuppens, N. Cuppens-Boulahia, J. Garcia-Alfaro, L. Toutain, and Y. Elrakaiby. A Semantic Context Aware Security Policy Deployment. ACM Symposium on Information, Computer and Communications Security, pp. 251–261, Sydney, Australia, March 2009.

[23] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman. Role-Based Access Control Models. IEEE Computer, 29(2):38–47, 1996.

[24] F. Xian, H. Jin, K. Liu, and Z. Han. A Mobile-Agent based Distributed Dynamic microFirewall Architecture. In 9th International Conf. on Parallel and Distributed Systems, pp. 431-436, 2002.

[25] L. Yuan, J. Mai, S. Su, H. Chen, C. Chuah, and P. Mohapatra. FIREMAN: a toolkit for FIREwall Modeling and ANalysis. In IEEE Symposium on Security and Privacy, pp. 199–213, Oakland, California, 2006.