

Rapport final du projet POLITESS

ANR-05-RNRT-01301

Mémoire de présentation du projet POLITESS

ANR-05-RNRT-01301

Projet soutenu par l'Agence Nationale de la Recherche (France), programme Télécommunications, de mars 2006 à décembre 2008.

Partenaires :

- Grenoble INP (coordinateur) LIG équipe Vasco et Vérimag équipe DCS,
- Institut Télécom avec Télécom Bretagne équipe SERES et Télécom Sud Paris département LOR UMR Samovar,
- INRIA Bretagne Rennes Atlantique équipes Vertecs et Distribcom,
- Orange Labs laboratoire MAPS/NSS à Caen,
- Smartesting à Besançon, SAP Research à Sophia Antipolis et
- Orange Business Services (Silicomp-AQL) à Rennes.

1. Enjeux et problématique, état de l'art

Le projet POLITESS visait à étudier la démarche méthodologique et les outils logiciels associés permettant d'appréhender de manière formelle le continuum intégrant l'expression des politiques de sécurité de haut niveau, l'analyse de leur déploiement, le test de conformité et la surveillance de l'implémentation du système par rapport aux politiques de sécurité.

L'enjeu est celui de la sécurité des systèmes d'information et des services, répartis sur des ensembles de machines et des domaines d'exploitation. Cette sécurité doit être assurée par la mise en œuvre correcte de politiques de sécurité bien définies, capables d'interagir lorsqu'elles appliquent des règles différentes selon les domaines.

Les modèles « classiques » de description de politiques de sécurité permettent seulement de contrôler l'accès aux informations conformément à un ensemble de règles d'autorisation *statiques*. Ces modèles s'avèrent inadaptes pour répondre aux besoins d'interopérabilité d'infrastructures informatiques ouvertes et mobiles. Il faut trouver des moyens d'exprimer des règles dynamiques, en fonction du contexte du contexte. Il faut aussi permettre une mise en relation de règles entre plusieurs domaines, à travers des mécanismes de traduction et de négociation. Compte tenu de la complexité induite par cette dimension des politiques de sécurité, il faut aussi être capable de valider la cohérence des règles, de gérer les anomalies.

Ensuite, il faut trouver des méthodes permettant de garantir le respect de ces politiques en pratique, dans la mise en œuvre qui en est faite par les mécanismes de sécurité déployés dans le réseau. Grâce à la formalisation des règles de la politique de sécurité, on veut pouvoir déployer automatiquement le contrôle, surveiller la bonne application des règles, détecter les violations. Enfin, on s'intéresse aussi au test des systèmes comme moyen de vérifier le respect d'une politique, pour y détecter des failles avant que celles-ci ne puissent être exploitées.

Au démarrage du projet POLITESS, les modèles classiques existant pour la description des politiques de sécurité comme DAC, ou MAC comme RBAC ne permettaient qu'une expression limitée à des règles statiques. Le formalisme OrBAC (Organization based access control) proposé par Télécom Bretagne permettait déjà une prise en compte du contexte et les

travaux menés dans le projet ont permis d'étendre OrBAC pour traiter les problèmes mentionnés.

En ce qui concerne le test, actif ou passif d'une politique, certains travaux permettaient de dériver des moniteurs de contrôle. Le test proprement dit de politiques de sécurité est un domaine assez neuf qui s'est développé pendant le projet, et a donné lieu au premier séminaire international sur le sujet en 2008 (SECTEST).

2. Matériel et méthodes

Le projet POLITESS a permis de fédérer un certain nombre de directions de recherche. En termes de méthodes, on n'a pas cherché à développer une méthode universelle, mais une collection d'approches abordant les différentes dimensions du projet (à travers les sous-projets correspondants - SP): expression des politiques (SP1), techniques de surveillance (SP4), négociation de politiques entre domaines (SP2) et test (SP3).

a) Études de cas (SP5)

Afin d'alimenter les méthodes et d'évaluer leur pertinence, les partenaires France Télécom et SAP fournissaient chacune un cas d'étude. Par rapport au montage et en cours de projet, il y a eu des évolutions.

En début de projet, nous avons pu disposer d'une spécification assez riche et précise d'une application « Travel » correspondant au système interne de gestion des missions à France Télécom. Celle-ci a constitué un bon cas d'étude pour la définition des propriétés à prendre en compte, le partage entre les aspects fonctionnels et la politique de sécurité et l'architecture des domaines. Malheureusement, elle n'était plus disponible lorsqu'il s'est agi de mener des expérimentations, ce qui a empêché une évaluation réelle sur cible des méthodes. SAP avait aussi fourni une application issue du domaine de la défense (gestion de stock militaire) en début de projet, mais il était difficile d'exploiter celle-ci dont les spécifications de sécurité étaient trop limitées (l'essentiel de la sécurité étant pris en charge par un portail extérieur à l'application). C'est pourquoi en cours de projet SAP a fourni une application sur la gestion de crédits (« bank loan origination process ») réalisé en technologie « services web ». Ce cas d'étude présente un scénario bancaire où les différentes entités sont autant de services. Cet environnement distribué représentait donc un véritable défi aussi bien pour la modélisation que pour l'exécution des tests, tout en restant proche des nouvelles générations de produits SAP orientés-services *ByDesign*. L'application est arrivée trop tardivement pour que l'on puisse reprendre l'ensemble des démarches de modélisation (en particulier, la politique de sécurité n'était pas clairement définie), mais on a au moins pu procéder au test à base de modèle (basé sur les aspects fonctionnels) selon l'approche mise en œuvre par Smartesting avec les techniques à base d'UML/OCL. Par ailleurs, une autre application de ce type fournie par SAP a permis d'expérimenter des approches de test passif.

Globalement cependant, les études de cas n'auront pas permis de mener des campagnes d'évaluation comme nous l'aurions souhaité, et cela aura constitué un handicap du projet.

b) Modélisation (SP1)

L'ensemble des partenaires ont contribué à la modélisation, en particulier pour modéliser les cas d'études, exprimer des propriétés et adapter au traitement automatique des modèles. D'abord, les deux études initiales ont été définies grâce à l'expertise du partenaire Silicomp-AQL, sous la forme de cibles de sécurité selon l'approche « critères communs ». Deux axes de modélisation principaux ont été apportés par les partenaires TELECOM Bretagne (modèles de sécurité reposant sur OrBAC et NOMAD) et Smartesting (sur une base UML/OCL).

Dans la première approche, on développe un modèle de sécurité capable d'exprimer des politiques de sécurité contextuelles adaptées au dynamisme des applications de type Web service auxquelles le projet s'intéressait.

Dans la deuxième approche, on vise à intégrer les exigences de sécurité à un modèle UML utilisé pour la spécification du système. On le fait en utilisant une formalisation des règles de sécurité sous formes de contraintes OCL, ce qui permet de fonder la production de tests par l'exploration et la résolution de ces contraintes.

c) Négociation de politiques (SP2)

Le projet POLITESS s'intéressait en particulier aux applications de type service web, dans lesquelles différents services, relevant de domaines différents, doivent collaborer. Les applications Travel de FT et l'application bancaire de SAP correspondaient bien à cette logique. On s'est donc intéressé principalement à identifier des méthodes de négociation entre domaines ayant des politiques de sécurité différentes. Nous nous sommes pour cela placés dans le contexte de services Web où les informations de sécurité sont échangées en XACML. Une difficulté à prendre en compte est le cas où la politique de sécurité inclus à la fois des permissions et des interdictions.

d) Test de conformité à une politique (SP3)

Le projet POLITESS réunissait des partenaires ayant des bases techniques différentes pour aborder la génération de tests de conformité : calculs d'instanciations principalement basés sur la résolution de contraintes sur les paramètres d'un scénario pour Smartesting, génération de tests liées aux modèles logiques de la sécurité (comme OrBAC, Nomad) pour Grenoble INP et Télécom Sud Paris, recherche de violation de propriétés par les techniques de diagnostic pour l'IRISA... Plutôt qu'une méthode de test uniforme, on a donc essentiellement défini dans le L3.2a des éléments de méthode, et une architecture générique pour les outils associés dans le L3.5. La méthodologie, l'identification des concepts et l'architecture se sont appuyés sur l'état de l'art en audits de sécurité tels que pratiqués par Silicomp-AQL.

e) Surveillance (SP4)

Le *test passif* consiste à observer le comportement d'un système pendant son exécution -sans contrôler les entrées- de détecter les déviations de comportement par rapport à une propriété de sécurité et de prendre les mesures nécessaires pour garantir son bon fonctionnement. La force majeure de ces techniques est leur capacité à être utilisées sur un système alors qu'il est en train de fonctionner, en limitant la perturbation induite par l'observation. On s'est intéressé à deux problèmes :

- La construction de moniteurs analysant des traces d'un système et les confrontant aux règles d'une politique exprimée en OrBAC ou Nomad. On a également étudié une approche partant de la spécification des comportements normaux d'un système définis sous forme d'ordres partiels.
- La détection de fuite d'information confidentielle pour des systèmes finis partiellement observables, sans partir de règles de contrôle d'accès, mais d'une simple identification des actions (ou données) confidentielles.

3. Résultats

a) Études de cas (SP5)

Sur l'application « Travel », un travail important a été mené au niveau de la modélisation. Cette application a servi de fil conducteur pour définir et illustrer les méthodes de négociation, de test, de surveillance. Compte tenu de la non-disponibilité de l'implantation de cette application on n'a pu mener les expérimentations jusqu'au bout. Des partenaires ont proposé des solutions de remplacement, mais celles-ci étaient trop limitées pour permettre une évaluation correcte des méthodes, c'est pourquoi elles n'ont pas été intégrées au livrable 5.4. En revanche, il a été possible de mener à terme une évaluation de l'approche SBT (de génération de test) sur le cas fourni en fin de projet par SAP.

Validation de l'approche SBT au sein du projet dans le cadre de l'étude de cas avec SAP

La validation de la démarche de test SBT a été réalisée sur l'application SAP de processus métier de prêt bancaire. Les résultats sont disponibles dans le livrable L5.4. La figure suivante montre le processus modélisé et testé.

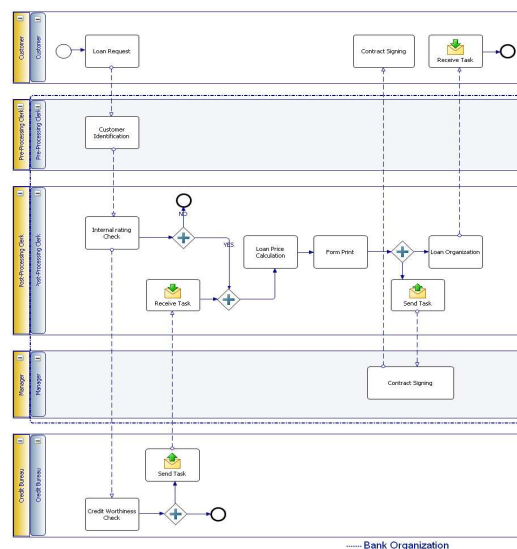


Figure 5 : Business process de l'étude de cas SAP

Le processus complet a été déroulé, avec la formalisation avec UML4ST de l'application SAP, la génération des tests pilotés par des scénarios à partir des propriétés de sécurité et leur concrétisation en scripts exécutables. Cette application a permis de mettre au point différents composants du démonstrateur : la vérification du modèle, le pilotage par scénarios, et le module de concrétisation des tests.

Évaluation de l'outil de test passif

L'outil de test passif (SP4) a été appliqué à une étude de cas différente (SAP R/3) et les résultats étaient satisfaisants. En effet, cette évaluation a été menée avant que l'application de crédit bancaire ne soit disponible. Il a été aussi appliqué dans le cadre dans un réseau mobile ad hoc basée sur le protocole de routage Ad hoc pour détecter d'éventuels nœuds malveillants.

b) Modélisation (SP1)

Extensions au modèle OrBAC

Dans le modèle OrBAC, chaque règle de sécurité (une permission, une interdiction, une obligation ou une dispense) ne s'applique que lorsqu'un contexte particulier est actif. Un contexte est vu comme une condition qui doit être satisfaite pour activer la (ou les) règle(s) de sécurité qu'elle contraint. Le travail réalisé a consisté à définir une taxonomie des différents

types de contextes et à analyser les données devant être fournies par le système d'information pour gérer ces différents contextes. Nous avons montré ensuite comment les exprimer dans le modèle OrBAC et défini une stratégie d'évaluation des règles de sécurité calculable en temps polynomial.

TELECOM Bretagne a défini une démarche formelle pour détecter et gérer les conflits dans une politique de contrôle d'accès. Un conflit apparaît dans une politique de contrôle d'accès lorsqu'un sujet a à la fois la permission et l'interdiction de réaliser une action. Nous avons tout d'abord montré les limites d'un modèle à base de règles (Rule-BAC), le problème de la gestion des conflits devenant indécidable dans un tel modèle. Nous avons ensuite étudié ce problème dans le cadre du modèle OrBAC (Organization Based Access Control) et montré comment la gestion des conflits et des règles redondantes devient calculable en temps polynomial.

D'autre part, pour spécifier des exigences de contrôle d'usage répondant notamment aux besoins de sécurité des applications de services web, nous avons proposé le modèle Nomad. Nous avons ensuite étudié le problème de la mise en œuvre des exigences de contrôle d'usage dans un système d'informations. L'approche proposée est originale car nous considérons qu'un système d'informations peut être compatible avec sa politique de sécurité même si certaines exigences de sécurité sont violées. Dans ce cas, nous définissons formellement les conditions que le système d'informations doit satisfaire pour être compatible avec sa politique et présentons une approche permettant de vérifier si ces conditions sont satisfaites. Nous avons appliqué cette approche pour traiter les besoins de contrôle d'usage dans l'application de services web « Travel » utilisée dans le projet POLITESS.

TELECOM Bretagne a développé une seconde version de MotOrBAC, un outil qui permet de spécifier et d'administrer la politique de sécurité d'un système dans un modèle unique. MotOrBAC peut simuler et analyser une politique de sécurité spécifiée en utilisant le modèle OrBAC.

Définition d'un sous-ensemble formel de la notation UML/OCL pour la génération de test de politique de sécurité

Pour la génération automatique de tests de politique de sécurité à partir de modèles, il est nécessaire que le modèle de test comporte les comportements attendus du système sous test au regard des fonctions de sécurité à tester. Ce modèle de test doit être supporté par une notation formelle qui au sein du projet POLITESS s'appuie sur un sous-ensemble de UML/OCL. Nous désignons **UML4ST** (UML for Security Testing) ce sous-ensemble proposé de la notation UML.

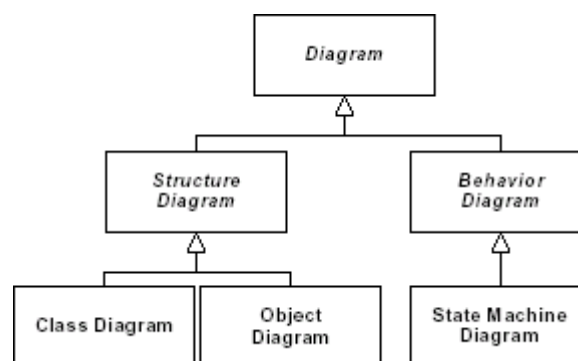


Figure 1. Méta-modèle UML du sous-ensemble UML4ST

UML offre un ensemble de diagrammes pour représenter les aspects statiques et dynamiques du système sous test. Voir le rapport 1.6 pour une description complète. Parmi ceux-ci, notre sous-ensemble UML4ST se compose des trois diagrammes suivants : diagramme de classes, diagramme d'objets et diagramme états-transitions, selon la hiérarchie donnée en Figure 1.

Les diagrammes de classes et d'objets sont utilisés pour représenter les données du système, leurs types et leurs liens. Le diagramme états-transitions (ou state machine) contient les comportements du système sous test, exprimés à partir de ces données.

Un sous-ensemble du langage OCL – Object Constraint Language -, et une sémantique associée, ont été définis pour caractériser formellement la partie comportementale au sein de UML4ST. Le sous-ensemble OCL supporté par UML4ST est nommé **OCL4ST** – OCL for Security Testing. Il s'appuie sur une double interprétation du langage suivant le contexte d'expression. On parle de contexte de contraintes pour les expressions suivantes :

- précondition d'opération,
- garde de transition,
- toute expression conditionnelle : condition dans une structure conditionnelle (*if-then-else*) par exemple.

Par ailleurs, on parle de contexte d'action pour les expressions suivantes :

- postcondition d'opération,
- action de transition,
- comportements d'entrée et de sortie d'état.

c) Négociation de politiques (SP2)

France Télécom et TELECOM Bretagne ont défini un modèle de négociation fondé sur une méthodologie de classification des ressources à négocier. Trois classes de négociation ont ainsi été définies : (1) Accès sans besoin de négociation, (2) Accès avec négociation et la politique de sécurité contrôlant la négociation est publique et (3) Accès avec négociation et la politique est secrète. Nous avons également étudié les problèmes de négociation lorsque la politique de sécurité inclut des règles de permission et d'interdiction. Dans ce cas, il n'est pas souhaitable de négocier des informations pour établir qu'une interdiction est applicable. Nous avons donc défini une approche formelle pour réécrire une politique de sécurité incluant des permissions et des interdictions en une politique équivalente ne contenant que des permissions. Comme cette dernière politique peut nécessiter de négocier des informations négatives, nous avons défini une approche pour négocier de telles informations.

Une architecture de négociation dénommée XeNA (XML Negotiation Architecture) reposant sur ces différents principes a également été définie, implémentée et intégrée dans une infrastructure de services web. Cette architecture repose sur l'échange d'attributs (credentials) XaCML. Les différents cas d'exception (échec, boucle) sont pris en compte dans cette architecture.

d) Test de conformité à une politique (SP3)

Plusieurs instanciations du cadre méthodologique proposé dans le L3.2a ont été définies et outillées selon le schéma d'architecture défini dans le L3.5. On donne ici les principales approches outillées et expérimentées dans le projet.

Méthodes

Méthode pour le test de sécurité fondée sur une approche de type Scénario-based testing pour le pilotage du test de sécurité

Le pilotage de la génération de tests pour la politique de sécurité élaboré par Smartesting au sein du projet est défini suivant un processus synthétisé en figure 3.

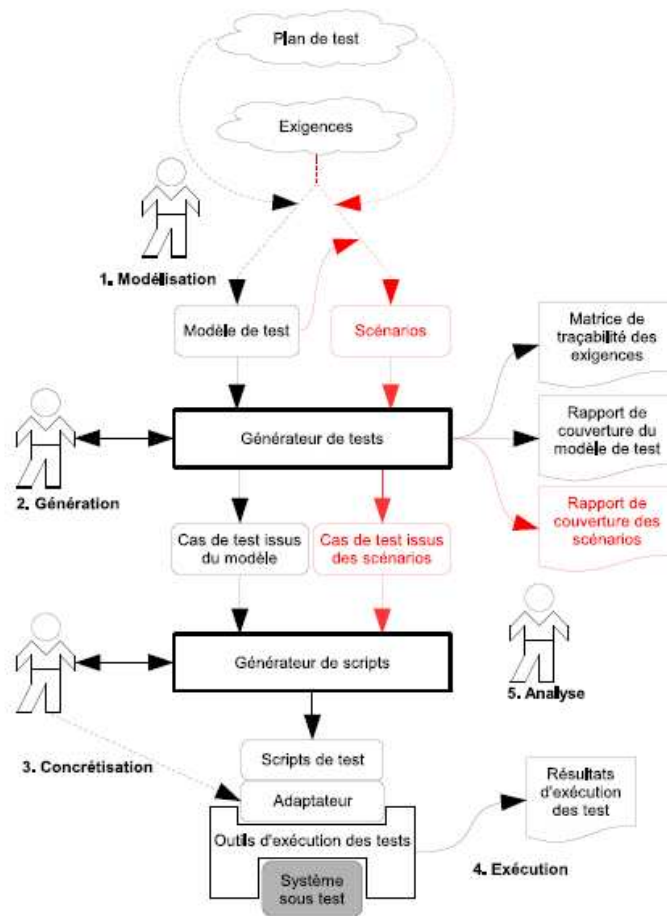


Figure 3 : Processus de génération de tests

Comme le montre la figure 3, la stratégie de génération SBT comporte les caractéristiques suivantes (voir rapports 3.6a & b pour plus de détails) :

- les scénarii sont formalisés en UML/OCL au même niveau d'abstraction que le modèle fonctionnel sécuritaire qui permet ainsi de déterminer le résultat attendu des tests ;
- la génération de tests s'appuie sur un parcours du modèle fonctionnel pour couvrir les objectifs de test (scénarii de test) ;
- la traçabilité des exigences et propriétés de sécurité vers les cas de test générés à partir des liens établis dans le modèle ce qui permet d'assurer la liaison entre propriétés de sécurité et tests correspondants ;
- la publication des cas de test générés dans un référentiel de test ;
- la concrétisation au travers de scripts exécutables dépendant de la structure d'exécution de tests utilisée pour l'application sous test.

Méthode de test basée sur une expression des règles de sécurité (property-based testing) et un pilotage par objectifs de test, outil j-POST.

Dans cette approche, le test est construit à partir des exigences de sécurité exprimées sous formes de formules de logiques temporelles ou d'expressions régulières étendues (ERE)

combinant des prédicats correspondant à des modules de test (selon la terminologie habituelle du test de conformité, norme IS9646). Les tests sont construits selon un mécanisme de combinaison des modules, selon un calcul associé aux formules décrivant la sécurité requise, et des objectifs de test permettant de guider le choix des instanciations.

Autres approches étudiées

Les livrables L3.2a, L3.3 et L3.7 présentent également d'autres approches qui ont été étudiées dans le projet.

- Intégration des règles de la politique (en OrBAC) à une spécification (sous forme de machines à états étendus, type SDL ou Statecharts) et dérivation de tests à partir du modèle combiné en ciblant les transitions modifiées par l'intégration des règles de sécurité.
- Génération d'objectifs de test à partir d'une politique exprimée en OrBAC.
- Dérivation de moniteurs de contrôle d'accès à partir de propriétés de sécurité (intégrité, confidentialité) et de testeurs associés permettant de vérifier si l'implantation du contrôle d'accès est bien conforme au modèle (calculé automatiquement), cf. L3.3.
- Détection, test et mesure de flux d'information cachés (ou canaux cachés) dans un système. Cette approche comporte plusieurs particularités par rapport au test classique de conformité : les canaux cachés peuvent être distribués, et la mesure d'une bande passante est essentielle pour connaître les mesures à mettre en œuvre : fermeture du canal, surveillance, bruitage... Nous avons proposé une méthodologie permettant de mettre en œuvre une attaque de type canal caché (lorsque celle-ci est effectivement calculable) à partir d'une description de haut niveau du déploiement d'un système (L3.7).

Outils

Prototype de mécanisme de fonction de Scénario-based testing au sein de l'outil de génération de tests à partir de modèles Test Designer™

Le démonstrateur développé par Smartesting au sein du projet s'appuie sur la technologie Test Designer™. Les caractéristiques de ce démonstrateur sont les suivantes :

- il prend en entrée des modèles formels de test en UML/OCL qui synthétisent les comportements fonctionnels et de sécurité attendus pour le système sous test
- la génération de tests est pilotée par une approche de type Scénario-based testing à partir des exigences de sécurité
- la génération de tests s'appuie sur un composant d'évaluation symbolique qui permet la simulation du modèle formel
- les tests abstraits générés sont ensuite concrétisés en scripts exécutables sur le système sous test.

Le démonstrateur a été développé en s'interfaçant avec un modeleur fondé sur la plate-forme Eclipse/UML, soit Borland Together, soit IBM Rational Software Modeler.

Selon un schéma classique, la structure d'un test généré est constituée de trois parties (cf. figure 4) : un préambule, permettant d'atteindre l'état d'exécution du test, le corps de test permettant de tester les comportements ciblés, et un postambule pour remettre le système dans l'état initial.

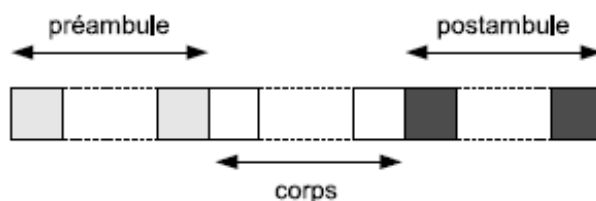


Figure 4 : Structure d'un test généré

Outil jPOST pour la génération de tests à partir de propriétés pour des systèmes répartis

j-POST est une chaîne d'outils associant :

- Design : c'est à la fois un éditeur d'exigences et un éditeur de modules de tests
- Génération : produit les cas de test abstraits à partir des formules et de la bibliothèque de modules de tests.
- Exécution : j-POST teste des programmes Java répartis en s'interfaçant aux méthodes par RMI. Le moteur de j-POT interprète les cas de tests abstraits construits par la génération, et opère à la volée la sélection et la concrétisation des tests.

La sélection des tests est guidée par les objectifs qui restreignent les entrelacements d'actions : ces objectifs sont décrits par des automates dont les transitions sont étiquetées par les interactions (entre les modules de test et le système sous test, ou les actions de synchronisation entre les modules de test).

L'outil a été développé dans le cadre du projet POLITESS en partant de l'approche de génération de test à partir d'une expression logique des propriétés de sécurité, et en développant un calcul de composition des modules de test élémentaires.

Il a été expérimenté pour le cas d'étude « Travel » de FT. En effet disposait pour ce cas d'une formalisation de la politique de sécurité (L5.2), et jPOST est précisément adapté à la génération de test pour des systèmes pour lequel on ne dispose pas d'une spécification complète, mais simplement d'exigences partielles (ici les exigences de sécurité).

Comme l'implémentation de Travel n'était pas disponible, nous avons seulement pu expérimenter sur une petite maquette développée en interne, et dont les fonctionnalités étaient restreintes.

e) Surveillance (SP4)

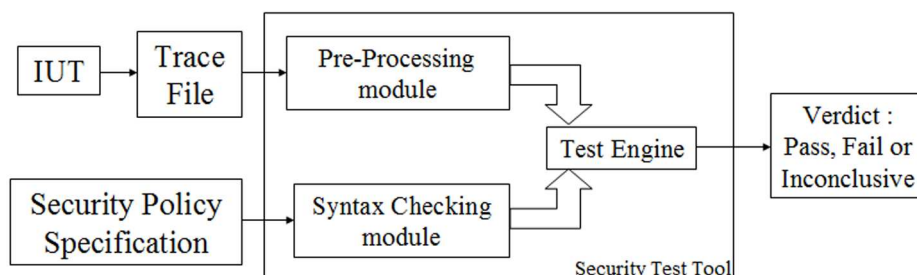


Figure 5 : Outil de Test Passif

Pour le premier des problèmes définis en 2)-e), une architecture de test passif a été proposée et un outil de test passif a été implémenté. L'outil de surveillance a 2 entrées principales. (1) *La trace d'exécution du système* : cette trace est collectée au fur et à mesure que le système sous test fonctionne. L'une des difficultés pour analyser une trace est sa longueur. En effet, analyser une trace longue peut avoir une incidence directe sur les performances du testeur. Un choix pourrait être pris par l'administrateur du système suivant l'étude de cas pour délimiter

la longueur adéquate de la trace collectée. (2) *Les règles de sécurité à vérifier sur le système sous test* : ces règles sont décrites en utilisant une instanciation du modèle Nomad. Lors de cette instanciation on a défini précisément la notion d'action en tant qu'échange de messages entre les entités du système et l'utilisateur (émission ou réception). L'utilisation du modèle Nomad est motivée par la puissance de langage qui combine la logique déontique avec la logique temporelle pour permettre de décrire des règles de sécurité assez complexes. Ces règles sont fournies sous forme textuelle.

En se basant sur ces deux principales entrées, l'outil de surveillance a comme objectif de déduire un verdict relatif à la conformité de la trace vis à vis de sa politique de sécurité. Dans le cas de violations, les séquences erronées de la trace sont fournies ainsi que les règles non respectées.

Pour le deuxième problème, le L4.6 montre comment on peut appliquer les techniques de diagnostic sur des systèmes à événement discrets. On a explicité des conditions nécessaires et suffisantes sur le système pour permettre la détection et/ou la prédiction de cette fuite d'information et comment construire un moniteur permettant à un administrateur d'assurer cette détection.

4. Discussion sur le degré de réalisation des objectifs initiaux, les verrous restant à franchir, les ruptures, les élargissements possibles, les perspectives ouvertes par le projet, l'impact scientifique, industriel ou sociétal des résultats.

Le projet était exploratoire, il cherchait donc à étudier différents aspects liés à la modélisation et à l'outillage pour mettre en place et valider automatiquement des politiques de sécurité. En même temps, pour fixer un but, on se plaçait dans la perspective d'une méthode universelle avec une chaîne d'outil unique pour assurer l'expression des politiques de sécurité, leur déploiement, le test de conformité et la surveillance du système.

De fait, le projet aura rempli sa fonction en développant de nouvelles techniques au delà de l'état de l'art sur toutes les dimensions étudiées : modélisation, déploiement, test, surveillance. En revanche, il n'a pas été possible dans le cadre du projet de développer une seule approche unifiée : rétrospectivement, il valait sans doute mieux que l'on explore comme on l'a fait plusieurs pistes que de chercher à faire rentrer toutes les études dans le même moule. Les deux principaux verrous évoqués lors de la soumission (cf. le §1.3 de la soumission) ont été au moins partiellement levés :

- On a développé autour d'OrBAC un formalisme suffisamment riche pour aborder la diversité des contraintes des politiques pour des systèmes hétérogènes, qui inclut les possibilités de négociation, de contrôle d'usage, de gestion des anomalies dans un modèle unifié.
- On a pu poser les fondements du test de propriétés non fonctionnelles spécifiques de la sécurité, comme l'opacité et développer des approches de validation sur différents types de représentation des politiques de sécurité, à des niveaux de représentation et de mise en œuvre variés (combinaison avec les aspects fonctionnels décrits en SDL ou UML...).

En revanche, le verrou « humain » sur l'applicabilité dans le cadre du travail des acteurs de terrain en sécurité n'a pas pu être abordé, bien que le projet ait pu bénéficier de l'expertise des auditeurs de Silicomp-AQL, du fait des difficultés avec les études de cas et des restrictions des prototypes. Toutefois, la confrontation en fin de projet des approches étudiées dans le cadre de POLITESS et des méthodes utilisées en audit soulèvent le problème des méthodes de détection des vulnérabilités. Ces aspects ne sont pas couverts par le projet POLITESS,

néanmoins ils ont fait l'objet d'une réflexion particulière et pourront faire l'objet de travaux ultérieurs.

Plus largement, plusieurs pistes de recherche novatrices ont ainsi été suggérées par le projet. On peut citer en particulier les suivantes sur lesquelles les partenaires académiques ont commencé à réfléchir en fin de projet :

- La modélisation pour synthétiser des testeurs, contrôleurs ou moniteurs de propriétés d'intégrité et de confidentialité s'est appuyé sur des systèmes de transition finis ou des HMSC. Il serait souhaitable d'étendre l'expressivité des modèles en considérant des modèles temporisés / probabilistes et des modèles manipulant des données en s'appuyant par exemple sur des techniques d'abstraction. Ce serait particulièrement utile pour étendre aux propriétés de disponibilité ; une autre piste pour cela pourrait être d'avoir des automates modaux pour la modélisation des systèmes (IRISA, Institut Télécom).
- On s'intéresse aussi à la définition de critères de couvertures des modèles et des propriétés.
- On a abordé grâce au projet la modélisation des fuites d'information comme des stratégies distribuées dans des jeux : cette piste mériterait d'être poursuivie avec des jeux quantitatifs (IRISA).
- Concernant la détection de vulnérabilités soulevées par la pratique des auditeurs, on envisage de combiner des techniques de test et d'inférences de machine (sur lesquelles une collaboration a déjà eu lieu entre Grenoble INP et SAP).

5. Conclusions et recommandations sur l'exploitation et la dissémination des résultats.

Les résultats obtenus par ce projet exploratoire devront être consolidés sur plusieurs points par des études pour une réelle applicabilité. Cependant, plusieurs des travaux ont conduit à des prototypes qui pourront être exploités.

En modélisation, les extensions apportées par le projet ont pour la plupart été incorporées dans la dernière version MotOrBAC v2 qui déjà fait l'objet de plusieurs présentations et peut être chargé depuis SourceForge. MotOrBAC implante également AdOrBAC qui sert de modèle d'administration pour OrBAC. Dans cette nouvelle version de MotOrBAC, une API (Application Programming Interface) de OrBAC a été développée. Cette API peut être intégrée dans des applications pour mettre en œuvre une politique de sécurité OrBAC.

Smartesting, outilleur au sein du consortium, compte mener trois types d'actions dans la suite du projet POLITESS :

- l'industrialisation au sein du produit Test Designer™ de l'approche du pilotage de la génération de tests à partir de scénarios
- la poursuite de l'élaboration de la méthodologie de test de politique de sécurité (démarche de formalisation du système sous test et de ses propriétés associées, démarche de production des tests, ...)
- étude du marché du test de la politique de sécurité de grands systèmes d'information et de systèmes en réseau.

Ces travaux conduiront Smartesting à proposer une offre outillée, originale et innovante pour le test de politique de sécurité.

6. Liste des publications.

Revue Internationale

- C. Constant, T. Jéron, H. Marchand, V. Rusu, Integrating formal verification and conformance testing for reactive systems, IEEE Transactions on Software Engineering, 33(8):558-574, August 2007.

- F. Cuppens and N. Cuppens-Boulahia and M. Ben Ghorbel. High-level conflict management strategies in advanced access control models. *Electronic Notes in Theoretical Computer Science (ENTCS)*, Vol. 186, pp. 3-26, July 2007.
- F. Cuppens and N. Cuppens-Boulahia. Modeling contextual Security Policies. *International Journal of Information Security*, 7(4), August 2008.
- D. Abi Haidar, N. Cuppens-Boulahia, F. Cuppens, H. Debar. XeNA: An access Control Framework Using XaCML. *Annals of TELECOM*, October, 2008.

Conférences internationales

- Darmaillacq V., Fernandez J.-C., Groz R., Mounier L., Richier J.-L., « Test Generation for Network Security Rules. », *TestCom*, p. 341-356, 2006.
- Falcone Y., Fernandez J.-C., Mounier L., Richier J.-L., « A Test Calculus Framework Applied to Network Security Policies. », *FATES/RV*, p. 55-69, 2006.
- Falcone Y., Fernandez J.-C., Mounier L., Richier J.-L., « A Compositional Testing Framework Driven by Partial Specifications. », *TESTCOM/FATES*, 2007a.
- K. Li, L. Mounier, R. Groz: Test Generation from Security Policies Specified in OrBAC- ; *COMPSAC – IWSSE Workshop (IEEE International Workshop on Security in Software Engineering)* Beijing, July 2007.
- Shahbaz M., Groz R. : Using Invariant Detection Mechanism in Black Box Inference ; *ISoLA Workshop on Leveraging Applications of Formal Methods*, Poitiers, December 2007.
- Falcone Y., Mounier L., Fernandez J.-C., Richier J.-L., « j-POST: a Java Toolchain for Property-Oriented Software Testing », *Model-Based Testing (MBT)*, 2008.
- V. Darmaillacq, J.-L. Richier, R. Groz; Test generation and execution for security rules in temporal logic; *1st IEEE Workshop on Security Testing*, Lillehammer, April 2008.
- V. Darmaillacq. *Security policy testing using vulnerability exploits chaining*. In *Proceedings of the 1st International ICST Workshop on Security Testing - Sectest'08*. Lillehammer, Norway. April 9th, 2008.
- Falcone Y., Fernandez J.-C., Mounier L., « Synthesizing Enforcement Monitors wrt. the Safety-Progress Properties », *International Conference on Information Systems Security (ICISS)*, Hyderabad, (India), December, 2008 (to appear).
- J. Dubreil, Ph. Darondeau, H. Marchand, Opacity Enforcing Control Synthesis, in *Workshop on Discrete Event Systems, WODES'08*, Gothenburg, Sweden, March 2008.
- M. Oostdijk, V. Rusu, J. Tretmans, R. de Vries, T. Willemse, Integrating verification, testing, and learning for cryptographic protocols, in *Integrated Formal Methods (IFM'07)*, 2007
- T. Jéron, H. Marchand, S. Genc, S. Lafortune, Predictability of Sequence Patterns in Discrete Event Systems, in *IFAC World Congress*, Seoul, Korea, July 2008
- T. Jéron, H. Marchand, S. Pinchinat, M-O. Cordier, Supervision Patterns in Discrete Event Systems Diagnosis, in *Workshop on Discrete Event Systems, WODES'06*, Ann Arbor (MI, USA), July 2006.
- Wissam Mallouli, Jean-Marie Orset, Ana Cavalli, Nora Cuppens et Frédéric Cuppens. *A Formal Approach for Testing Security Rules*, the 12th ACM symposium on access control models and technologies (*SACMAT'07*), SAP Labs, Sophia Antipolis, France, June 20-22, 2007.
- D. Abi Haidar, N. Cuppens-Boulahia, F. Cuppens, H. Debar. An extended RBAC profile for XACML. *SWS'06: 3rd ACM workshop on Secure Web Services*, November 3, Fairfax VA, USA, 2006, pp. 13-22.

- Frédéric Cuppens, Nora Cuppens-Boulahia et Meriam Ben Ghorbel. *High level conflict management strategies in advanced access control models*. Workshop on Information and Computer Security (ICS), Timisoara, Roumanie, Septembre 2006.
- Frédéric Cuppens, Nora Cuppens-Boulahia et Céline Coma. *O2O: Managing Security Policy Interoperability with Virtual Private Organizations*. HP Open View Workshop. Presqu'île de Gien, France. Juin 2006.
- Frédéric Cuppens, Nora Cuppens-Boulahia, Céline Coma: *O2O: Virtual Private Organizations to Manage Security Policy Interoperability*. ICISS 2006, Calcutta, Inde, Décembre 2006.
- D. Abi Haidar, N. Cuppens-Boulahia, F. Cuppens, H. Debar. Resource Classification Based Negotiation in Web Services . The Third International Symposium on Information Assurance and Security (IAS), Manchester, United Kingdom, August 29-31, 2007.
- J. Brunel, F. Cuppens, N. Cuppens-Boulahia, T. Sans, J.-P. Bodeveix. Security Policy Compliance with Violation Management. 5th ACM Workshop on Formal Methods in Security Engineering: From Specifications to Code (FMSE), Alexandria, VA, USA, 2 November, 2007.
- J. G. Alfaro, F. Cuppens, and N. Cuppens-Boulahia. *Aggregating and Deploying Network Access Control Policies*. In 2nd International Conference on Availability, Reliability and Security (ARES 2007), April 2007.
- Céline Coma, Nora Cuppens-Boulahia, Frédéric Cuppens, Ana R. Cavalli, *Context Ontology for Secure Interoperability*, Third IEEE International Conference on Availability, Reliability and Security ARES 2008, March 4-7, Barcelona, Spain.
- N. Cuppens-Boulahia, F. Cuppens, D. Abi Haidar, H. Debar. *Negotiation of Prohibition: An Approach Based on Policy Rewriting*. 23rd International Information Security Conference (SEC 2008). Milan, Italy. September 2008.
- B. Alcalde, and A. Cavalli, Parallel Passive Testing of System Protocols - Towards a Real-time Exhaustive Approach, *ICN'06*, Mauritius, June 2006
- W. Mallouli, F. Bessayah, A. Cavalli and A. Benameur, *Security Rules Specification and Analysis Based on Passive Testing*, The IEEE Global Communications Conference (GLOBECOM 2008), New Orleans, USA, November 30 - December 04, 2008.
- W. Mallouli, B. Wehbi, A. Cavalli, *Distributed Monitoring in Ad Hoc Networks: Conformance and Security Checking*, The 7th International Conference on AD-HOC Networks & Wireless (ADHOC-Now 2008), Sophia Antipolis, France, September 10-12, 2008.
- W. Mallouli, G. Morales and A. Cavalli, *Testing Security Policies for Web Applications*, the 1st International ICST workshop on Security Testing (SECTEST'08), Lillehammer, Norway, April 09, 2008.
- R. Cavalli, E. Montes De Oca, W. Mallouli, M. Lallali, *Two Complementary Tools for the Formal Testing of Distributed Systems with Time Constraints*, The 12-th IEEE International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2008), Vancouver, Canada, October 27-29, 2008.
- W. Mallouli, M. Lallali, G. Morales, A. R. Cavalli, *Modeling and Testing Secure Web-Based Systems: Application to an Industrial Case Study*, The fourth International Conference on Signal-Image technology & Internet-Based Systems (SITIS 2008), Bali, Indonesia, November 30 - December 03, 2008
- Benameur, F. Abdul Kadir, S. Fenet. XML Rewriting Attacks on SOAP Messages: Existing Solutions and their Limitations. Dans IADIS Applied Computing 2008, Algarve Portugal.

- F. Bouquet, C. Grandpierre, B. Legeard, and F. Peureux. A test generation solution to automate software testing. In AST'08, 3rd Int. workshop on Automation of Software Test, Leipzig, Germany, pages 45--48, May 2008. ACM Press
- F. Bouquet, C. Grandpierre, B. Legeard, F. Peureux, N. Vacelet, and M. Utting. A subset of precise UML for model-based testing. In A-MOST'07, 3rd int. Workshop on Advances in Model Based Testing, London, UK, pages 95--104, July 2007. ACM Press.

Revue française

- V. Darmaillacq, J.C. Fernandez, R. Groz, L. Mounier, J-L. Richier : Tester la conformité d'un réseau à une politique de sécurité; REE (Revue de l'Électricité et de l'Électronique) Juin-Juillet 2006, pp 33-43.

Conférences francophones

- Groz R., Shahbaz M., Li K. : Une approche incrémentale de test par extraction de modèles ; AFADL'07 (Approches Formelles dans l'Assistance au Développement de Logiciels, 10^{ème} anniversaire), Namur, Juin 2007.
- Falcone Y., « Combiner Test et Monitoring pour la Sécurité », MajeSTIC, 2007.
- Falcone Y., Jaber M., « Vers l'Intégration Automatique d'une Politique de Sécurité OrBAC », MajeSTIC, 2007.
- J. Dubreil, T. Jéron, H. Marchand, Construction de moniteurs pour la surveillance de propriétés de sécurité, in 6ème Colloque Francophone sur la Modélisation des Systèmes Réactifs, Lyon, France, October 2007.
- Diala Abi Haidar, Nora Cuppens-Boulahia, Frédéric Cuppens et Hervé Debar. *Access Negotiation within XACML Architecture*. The 2nd Joint Conference on Security in Network Architectures and Information Systems (SAR-SSI). Annecy, France. June 2007.
- F. Autrel, F. Cuppens, N. Cuppens-Boulahia, C. Coma. MotOrBAC 2: a security policy tool. Third Joint Conference on Security in Networks Architectures and Security of Information Systems (SARSSI). Loctudy, France, 13-17 October 2008.

Chapitre de livres

- C. Constant, T. Jéron, H. Marchand, V. Rusu, Validation of Reactive Systems, in Modeling and Verification of Real-TIME Systems - Formalisms and software Tools, S. Merz, N. Navet (eds.), Chapter 2, Pages 51-76, Hermès Science, January 2008.

Autres communications

- Présentation du projet au colloque STIC (Lyon).
- J. Dubreil, T. Jéron, H. Marchand, Monitoring Information flow by Diagnosis Techniques, Research Report IRISA, No 1901, August 2008.
- J. Dubreil, Ph. Darondeau, H. Marchand, Opacity Enforcing Control Synthesis, Research Report IRISA, No 1887, March 2008.
- T. Jéron, H. Marchand, S. Genc, S. Lafortune, Predictability of Sequence Patterns in Discrete Event Systems, Research Report IRISA, No 1834, March 2007.
- (Conférence de vulgarisation) : présentation par Smartesting et SAP à la conférence SQC'07 sur la base de l'étude de cas du sous-projet 5
- Bruno Legeard (Smartesting), Azzedine Benameur and Maarten Rits (SAP) – *Model-based testing of SAP systems* - Software and Systems Quality Conferences – Zurich, Octobre 2007.
- Démonstration de MotOrBAC à SARSSI 2008.

- Présentation de OrBAC au SIB (Syndicat Inter-hospitalier de Brest)
- TAROT Summer School
- Fêtes de la science (TMSP)
- STIC AMSUD

Éléments de valorisation.

- MotOrBAC v2 a été déposé à l'APP sous la référence : IDDN.FR.001.500008.001.R.C.2006.000.10700.
- Le logiciel MotOrBAC v2 est accessible sur le site www.sourceforge.org (voir aussi le site de OrBAC : <http://www.orbac.org>)

Liste des livrables réalisés par le projet POLITESS

Les livrables publics figurent **en gras**.

Ils sont accessibles à l'URL http://www.rnrt-politess.info/rubrique.php3?id_rubrique=50.

- L1.1** **Modèle de sécurité avec règles contextuelles répondant aux besoins de contrôle d'accès à gestion centralisée. Rapport Public**
- L1.2** **Modélisation des besoins de contrôle d'usage des services web. Rapport Public**
- L1.3** **Modélisation et traitement des anomalies d'une politique de sécurité. Rapport Public**
- L1.4-L2.1 Modèle d'interopérabilité et protocoles de négociation de politiques de sécurité. Rapport Interne
- L1.5** **Modèle de référence pour la prise en compte des exigences de sécurité des services web. Rapport Public**
- L1.6** **Règles de modélisation UML/OCL pour l'intégration des exigences fonctionnelles et de sécurité. Rapport Public**
- L2.3 Protocole, spécifications et logiciel implémentant les fonctions de négociation. Rapport et prototype Internes
- L3.1** **Cahier des charges pour la suite du SP3 en fonction des applications choisies dans le SP5. Rapport Public**
- L3.2** **Version intermédiaire de la méthode de génération de test. Rapport Public**
- L3.3** **Towards automatic test generation for opacity properties. Rapport Public**
- L3.5** **Spécification de l'outil. Rapport Public**
- L3.6a Première version de l'environnement de test. Prototype Interne
- L3.6b Version finale de l'environnement de test. Prototype Interne
- L3.7 Outil de test de canal caché. Rapport Interne
- L4.1** **Définition d'une architecture pour la surveillance et de la configuration du système. Rapport Public**
- L4.2** **Spécification des politiques de sécurité compatibles avec celles proposées dans le SP1 et adaptées aux cas d'études du SP5. Rapport Public**
- L4.3** **Définition d'un gestionnaire du réseau. Rapport Public**
- L4.4 Adaptation de l'outil de surveillance à l'analyse de politiques de sécurité. Prototype Interne
- L4.5 Application des techniques de surveillance aux cas d'étude. Rapport Interne
- L4.6** **Étude de faisabilité de la construction de détecteurs d'intrusion. Rapport Public**
- L5.1 Analyse de besoins de sécurité pour les études de cas. Rapport Interne
- L5.2 Modélisation formelle de la politique de sécurité - Étude de cas France Télécom. Modèle Interne
- L5.3 Modélisation formelle de la politique de sécurité - Étude de cas SAP. Modèle Interne
- L5.4 Rapport de synthèse, étude de cas. Rapport Interne
- L5.1bis Complément à l'expression des besoins – Sécurité des architectures Web Services. Rapport Interne

Liste des thèses soutenues dans le cadre (et dans le courant) du projet

- Baptiste ALCALDE, 20 décembre 2006, INT-ParisVI
Titre : Techniques avancées pour le test passif de protocoles de communication
- Jean-Marie ORSET, 06 février 2007, INT-Paris VI
Titre : Une architecture de test passif appliquée à la détection des attaques dans les réseaux ad hoc
- Vianney Darmaillacq, 7 décembre 2007, Université Joseph Fourier

Titre : Génération de tests de sécurité pour les systèmes répartis

- Diala Abi Haidar, 27 novembre 2008, Institut Télécom Bretagne
Titre : Web services access negotiation
- Wissam Mallouli, 8 décembre 2008, Institut Télécom SudParis et Université d'Evry Val d'Essonne
Titre : Une approche formelle pour le test des politiques de sécurité
- Muhammad Muzammil Shahbaz, 12 décembre 2008, Institut Polytechnique de Grenoble
Titre : Reverse Engineering Enhanced State Models of Black Box Software Components to support Integration Testing

D'autres doctorants qui sont intervenus sur le projet soutiendront au-delà de la fin du projet. On peut citer en particulier Céline Coma, Jérémy Dubreil, Ylies Falcone, Bachar Wehbi, Samiha Ayed, Meriam Ben Ghorbel, Azzedine Benameur dont les thèses sont bien avancées.