

Politique de contrôle d'accès multi-niveaux : test de conformité vis à vis des flots avec l'outil FoCaL

Journée SSURF

Matthieu Carlier Catherine Dubois
Lionel Habib Mathieu Jaume

CPR - CEDRIC / SPI - LIP6

7 Novembre 2008

Introduction - Motivations

- Détecter automatiquement des erreurs dans la spécification
 - ▶ permet d'éviter d'essayer de prouver des propriétés fausses
- Détecter automatiquement des erreurs dans l'implantation
 - ▶ permet de trouver des bugs plus tôt dans la vie du logiciel
- Mise en œuvre sur une politique de contrôle d'accès multi-niveaux
- Utilisation de l'outil FoCaLTest de l'atelier FoCaL

FoCaLTest - Test de propriétés (1)

- Test de propriétés :

- ▶ *propriété* $\overset{?}{\iff}$ *implantation*

- ▶ *propriété* \Rightarrow *créer/soumettre* des jeux de test

- Propriétés de la forme :

$$\forall X_1 \dots X_n, \underbrace{A_1 \Rightarrow \dots \Rightarrow A_m}_{\text{Précondition}} \Rightarrow \underbrace{B_1 \vee \dots \vee B_m}_{\text{Conclusion}}$$

A_i et B_i : appels de *fonctions*

- Procédure de test :
 - ▶ jeu de test potentiel \equiv valuation des X_i
 - ▶ A_i sont évaluées à *true* \Rightarrow jeu de test *valide*
 - ▶ évaluation de $B_1 \vee \dots \vee B_m =$ *verdict* du test

FoCaLTest - Propriétés testées

- Ensemble des formules :

$$\forall X_1 \in T_1 \cdots X_n \in T_n, \alpha_1 \Rightarrow \cdots \Rightarrow \alpha_n \Rightarrow (A_1^1 \vee \cdots \vee A_{n_1}^1) \wedge \cdots \wedge (A_1^m \vee \cdots \vee A_{n_m}^m)$$

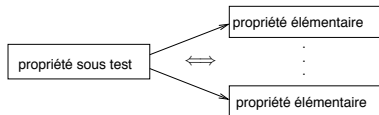
avec

$$\alpha ::= \alpha \vee \alpha \mid \alpha \wedge \alpha \mid A$$

- Propriété *réécrite* en un ensemble de *propriétés élémentaires* :

- ▶ *testées séparément*
- ▶ conjonction propriétés élémentaires

\iff
propriété initiale



FoCaLTest - Synthèse des jeux de tests

- Génération aléatoire :
 - ▶ valeurs des X_i générées aléatoirement
 - ▶ jeu de test gardé si précondition validée
- Approche par contraintes :
 - ▶ programme FoCaL traduit en contraintes
 - ▶ 1 solution du système \equiv 1 jeu de test *valide*

Contrôle d'accès

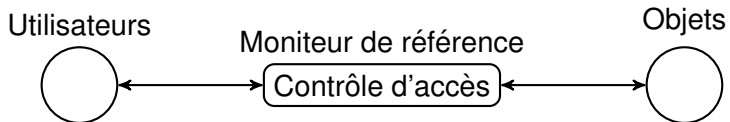
Utilisateurs



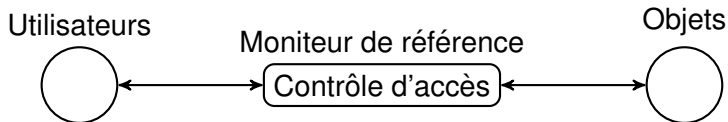
Objets



Contrôle d'accès



Contrôle d'accès



- Un système de contrôle d'accès repose sur deux concepts principaux :
 - ▶ La politique de contrôle d'accès
 - ▶ Le moniteur de référence

Contrôle d'accès multi-niveaux

Basé sur un treillis de niveaux de sécurité (Top Secret > Classifié ...)

Chaque sujet et objet est associé à un niveau de sécurité

Contrôle d'accès multi-niveaux

Basé sur un treillis de niveaux de sécurité (Top Secret > Classifié ...)

Chaque sujet et objet est associé à un niveau de sécurité

No read up : un sujet autorisé au niveau Classifié ne peut pas lire un objet Top Secret

Contrôle d'accès multi-niveaux

Basé sur un treillis de niveaux de sécurité (Top Secret > Classifié ...)

Chaque sujet et objet est associé à un niveau de sécurité

Propriété (MAC)

$$\forall s \in \mathcal{S} \quad \forall o \in \mathcal{O} \quad (s, o, \text{read}) \in m \Rightarrow f_o(o) \preceq f_s(s)$$

Contrôle d'accès multi-niveaux

Basé sur un treillis de niveaux de sécurité (Top Secret > Classifié ...)

Chaque sujet et objet est associé à un niveau de sécurité

Propriété (MAC)

$$\forall s \in \mathcal{S} \quad \forall o \in \mathcal{O} \quad (s, o, \text{read}) \in m \Rightarrow f_o(o) \preceq f_s(s)$$

No write down : un sujet ne peut pas écrire de l'information Top Secret dans un objet Classifié

Contrôle d'accès multi-niveaux

Basé sur un treillis de niveaux de sécurité (Top Secret > Classifié ...)

Chaque sujet et objet est associé à un niveau de sécurité

Propriété (MAC)

$$\forall s \in \mathcal{S} \quad \forall o \in \mathcal{O} \quad (s, o, \text{read}) \in m \Rightarrow f_o(o) \preceq f_s(s)$$

Propriété (MAC*)

$$\forall s \in \mathcal{S} \quad \forall o_1, o_2 \in \mathcal{O} \\ (s, o_1, \text{read}) \in m \wedge (s, o_2, \text{write}) \in m \Rightarrow f_o(o_1) \preceq f_o(o_2)$$

Politique de flots (1)

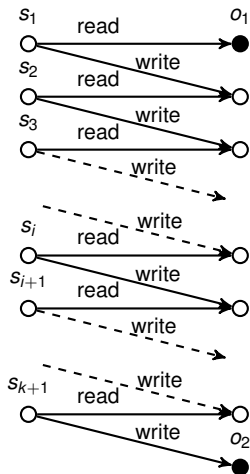
- Spécification des flots d'information autorisés
- Correction de la politique de confidentialité vis-à-vis des flots

Propriété

$$\forall \sigma = (m, f_s, f_o) \in \Sigma \quad \forall o_1, o_2 \in \mathcal{O}$$
$$(\text{MAC}(\sigma) \wedge \text{MAC}^*(\sigma) \wedge o_1 \xrightarrow{\sigma} o_2) \Rightarrow f_o(o_1) \preceq f_o(o_2)$$

Politique de flots (2)

$$o_1 \xrightarrow[\sigma]{OO} o_2$$



Erreur dans la spécification (1)

- Définition de la propriété MAC* selon McLean

Propriété (MAC* McLean)

$$\forall s \in \mathcal{S} \forall o_1, o_2 \in \mathcal{O}$$
$$((s, o_1, \text{read}) \in m \wedge (s, o_2, \text{write}) \in m) \Rightarrow \neg(f_o(o_2) \prec f_o(o_1))$$

Erreur dans la spécification (1)

- Définition de la propriété MAC* selon McLean

Propriété (MAC* original)

$$\forall s \in \mathcal{S} \forall o_1, o_2 \in \mathcal{O}$$
$$((s, o_1, \text{read}) \in m \wedge (s, o_2, \text{write}) \in m) \Rightarrow f_o(o_1) \preceq f_o(o_2)$$

Erreur dans la spécification (1)

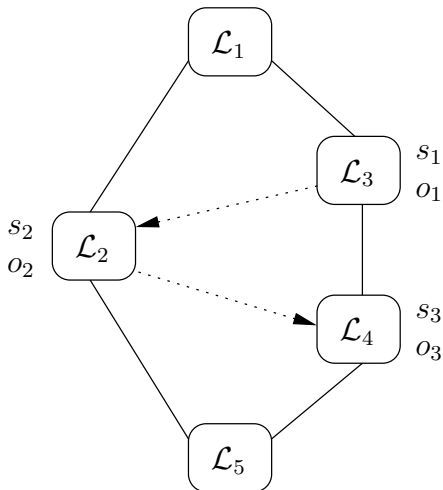
- Définition de la propriété MAC* selon McLean

Propriété (MAC* original)

$$\forall s \in \mathcal{S} \forall o_1, o_2 \in \mathcal{O}$$
$$((s, o_1, \text{read}) \in m \wedge (s, o_2, \text{write}) \in m) \Rightarrow f_o(o_1) \preceq f_o(o_2)$$

- ▶ Dans un ordre partiel $a \preceq b \not\Rightarrow \neg(b < a)$

Erreur dans la spécification (2)



Erreur dans l'implantation

- Introduction d'erreurs dans le moniteur de référence (fonction de transition)
- Correction de la fonction de transition vis-à-vis de la politique : une transition à partir d'un état sûr mène à un état sûr
 - ▶ Erreur dans la fonction de transition \Rightarrow état non-sûr atteignable

Contexte de test

- 3 sujets et 3 objets
- Modes d'accès considérés : lecture et écriture
- Treillis fixé
- Traduction sous forme de fonctions des propriétés MAC et MAC*
 - ▶ MAC \rightarrow `mac_fun`
 - ▶ MAC*(McLean) \rightarrow `mac_star_fun`
 - ▶ MAC*(correcte) \rightarrow `mac_star_fun_correct`

Remarque : peut être fait automatiquement

Test de la spécification

- Spécification testée :

Propriété

$$\forall \sigma = (m, f_s, f_o) \in \Sigma \quad \forall o_1, o_2 \in \mathcal{O} \\ (\text{MAC}(\sigma) \wedge \text{MAC}^*(\sigma) \wedge o_1 \xrightarrow[\sigma]{OO} o_2) \Rightarrow f_o(o_1) \preceq f_o(o_2)$$

- 2 implantations :

- ▶ $\forall \sigma \in \Sigma \quad \forall o_1, o_2 \in \mathcal{O}$
 $(\text{mac_fun}(\sigma) \wedge \text{mac_star_fun}(\sigma) \wedge \text{flow}(\sigma, o_1, o_2)) \Rightarrow$
 $f_o(o_1) \preceq f_o(o_2)$
- ▶ $\forall \sigma \in \Sigma$
 $(\text{mac_fun}(\sigma) \wedge \text{mac_star_fun}(\sigma)) \Rightarrow \text{all_flows_correct}(\sigma)$

Test de la spécification - Résultats

- Résultats implantation 1 :
 - ▶ Nombre de jeux de test valides : 5000
 - ▶ Nombre de flots « transversaux » : 71 %
 - ▶ Nombre de flots descendants : 0,8 %
 - ▶ Exécution : 75 sec

- Résultats implantation 2 :
 - ▶ Nombre de jeux de test valides : 5000
 - ▶ Nombre de flots « transversaux » : 1 %
 - ▶ Nombre de flots descendants : 0,04 %
 - ▶ Exécution : 3 sec

Test mutationnel

- Mutant : programme où l'on a changé 1 instruction
- Mutant par rapport à un jeu de test :
 - ▶ comportement *différent* à l'original : mutant *tué*
- But :
 - ▶ évaluer la qualité d'un ensemble de jeu de test
 - ▶ *nombre de tué / nombre total de mutants*

Test de l'implantation

- Propriété testée :

$$\forall \sigma \in \Sigma \quad \forall R \in \mathcal{R}$$
$$(\text{mac_fun}(\sigma) \wedge \text{mac_star_fun_correct}(\sigma)) \Rightarrow$$
$$\left(\begin{array}{l} \text{mac_fun}(\text{snd}(\tau_f(R, \sigma))) \wedge \\ \text{mac_star_fun_correct}(\text{snd}(\tau_f(R, \sigma))) \end{array} \right)$$

- Mutants considérés : $a \preceq b \longrightarrow b \preceq a$
- Mutant *tué* si propriété non *vérifiée*
- Création de 3 mutants
- Exécution d'un même jeu de test sur les 3 mutants

Test de l'implantation : Résultats

- Génération de 10 000 jeux de test par mutant
- Tous les mutants sont tués :
 - ▶ mutant 1 : 1 ‰ tué
 - ▶ mutant 2 : 2 ‰ tué
 - ▶ mutant 3 : 39 ‰ tué

Conclusion - Travaux futurs

- Erreurs dans la spécification et l'implantation automatiquement détectées
- Différentes approches dans l'écriture d'une propriété mène à des résultats différents
- Transformation automatique des prédicats (MAC et MAC^{*})