Modélisation du comportement hybride d'une application distribuée pour la détection d'intrusion

David Lanoë [1,2] Eric Totel [2] Michel Hurfin [1] Carlos Maziero [3]

16 Mai 2019

[1]Univ Rennes, Inria, CNRS, IRISA

[2]CentraleSupélec, Inria, CNRS, IRISA

[3]Univ Federal Paraná, Curitiba, Brazil

Application distribuée

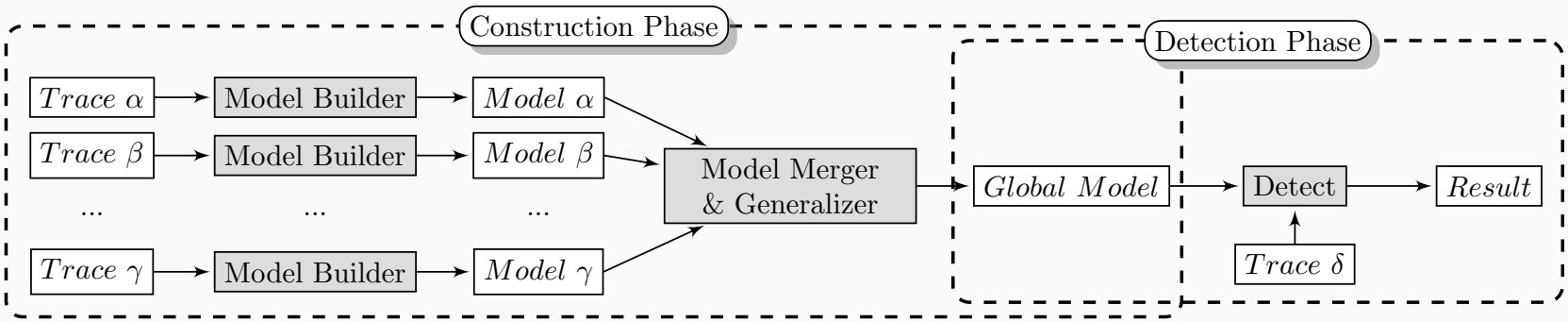
- Plusieurs nœuds qui interagissent
- Concurrence entre les différents nœuds
- Difficile de mettre en place un mécanisme d'**horloge globale**

Détection d'intrusion (corrélation d'alertes)

1. Surveillance de chaque processus à l'aide de sondes locales
2. Ordonnancement des alertes à l'aide d'une horloge globale et envoi à un manager
3. Corrélation d'alertes par le manager

L'approche de modélisation et de détection

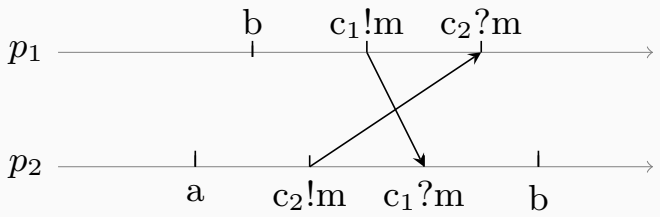
- Détection d'intrusion selon une approche **comportementale**
- Apprentissage d'un **modèle de comportement** dual (automate et invariants) qui ne repose pas sur un mécanisme d'horloge globale.



Trace d'exécution

Une trace est une collection de fichiers de logs (un par processus).

Exécution α :



Trace :

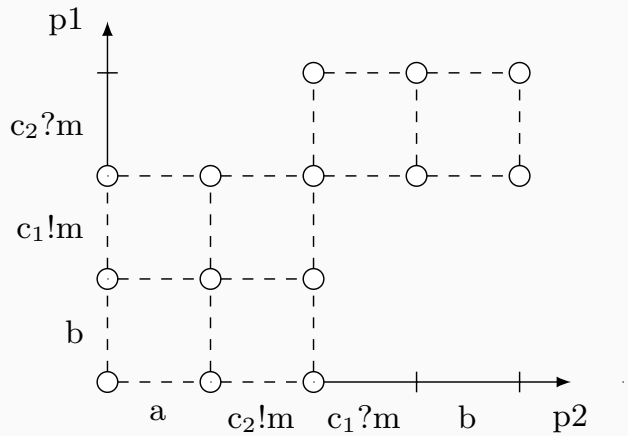
p_1	p_2
b	a
$c_1!m$	$c_2!m$
$c_2?m$	$c_1?m$
	b

L'ordre total ne peut etre observé à partir des logs.

Ordonnancement

Utilisation d'une relation d'ordre partiel *happened-before* [Lamport, 1978] sur les différents événements des logs.

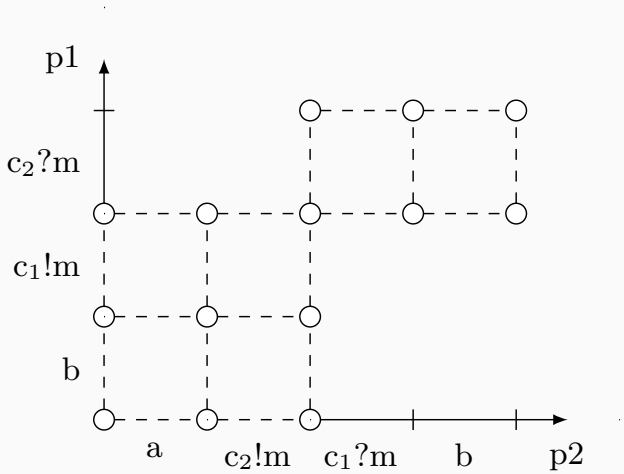
p_1	p_2
b	a
c_1!m	c_2!m
c_2?m	c_1?m
	b

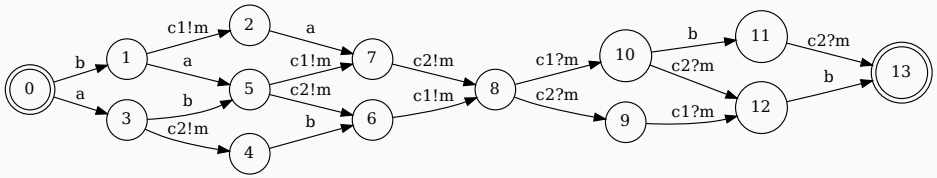


Grâce au treillis des états globaux cohérents, on est capable d'énumérer de l'ensemble des ordres totaux possibles.

Construction du modèle intermédiaire - Automate

Construction d'un automate lors du calcul du treillis.





Les automates reconnaissent exactement les traces utilisées pour les construire. Ils seront généralisés par la suite pour introduire d'autres comportements.

Construction du modèle intermédiaire - Invariants probables

Des invariants temporels peuvent être calculés à partir du treillis des états globaux cohérents [Ernst et al., 2001].

- $x \rightarrow y$: x toujours suivi par y
- $x \leftarrow y$: y toujours précédé par x
- $y \nrightarrow x$: y jamais suivi par x

Le scope correspond à l'ensemble de l'exécution.

Fusion et généralisation de modèles intermédiaires

Invariants

- Conservation des invariants qui restent vrais pour l'ensemble des exécutions.

Automates

- Fusion des états initiaux de tous les automates.
- Généralisation de l'automate (Ktail [Biermann and Feldman, 1972]) pour introduire de nouveaux comportements.

Détection

Automates
Parcours en profondeur pour vérifier qu'un chemin est bien présent dans l'automate.

Invariants

- En cours d'évaluation de trace : $\leftarrow$, $\nrightarrow$
- À la fin de la trace : $\rightarrow$, $\leftarrow$, $\nrightarrow$

Invariants de communication
Vérification de l'envoi d'un message sur un canal avant sa réception.

Évaluation de l'approche

- Un système de fichier distribué tolérant aux fautes : XtreemFS
- Plateforme : 5 raspberry PI (1 Client, 1 DIR, 1 MRC et 2 OSD)
- 5 attaques sur l'intégrité du système de fichier : Newfile, Deletefile, Osdchange, Chmod, Chown
- 4 contextes d'attaque : aucun client n'est actif (c1), avant les actions du client (c2), après les actions du client (c3) and depuis une autre source que le client (c4)

Une évaluation possible : complémentarité des modèles

Attack	*NewFile*				*DeleteFile*				*OsdChange*				*Chmod*				*Chown*			
Context	c1	c2	c3	c4	c1	c2	c3	c4	c1	c2	c3	c4	c1	c2	c3	c4	c1	c2	c3	c4
`Aut-1`	-	2/5	-	5/5	-	3/5	1/5	5/5	2/5	5/5	4/5	5/5	-	1/5	-	5/5	-	1/5	-	-
`Inv`	5/5	5/5	-	-	5/5	5/5	-	-	5/5	5/5	-	-	5/5	5/5	-	-	-	-	-	-
`InvCom`	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-

Questions?

References i

Biermann, A. W. and Feldman, J. A. (1972).
On the synthesis of finite-state machines from samples of their behavior.
IEEE transactions on Computers, 100(6):592–597.

Ernst, M. D., Cockrell, J., Griswold, W. G., and Notkin, D. (2001).
Dynamically discovering likely program invariants to support program evolution.
IEEE Transactions on Software Engineering, 27(2):99–123.

Lamport, L. (1978).
Time, clocks, and the ordering of events in a distributed system.
Communications of the ACM, 21(7):558–565.