LSR

IRISA

# Comparaison de divers formalismes pour la modélisation et le test de politiques de sécurité

***Vianney Darmaillacq***
***Equipe VASCO (LSR)***
***[Potestat] - 16/06/05***

# Sommaire

LSR

Avancement des travaux de la partie "réseaux" de Potestat.

*1 - Objectifs*

IRISA

*2* - Démarche

3 - Comparaison

4 - Conclusion

# Objectifs

LSR

- *Objectifs initiaux :*
  - *Déterminer les concepts nécessaires au test de la sécurité des réseaux*
  - *Trouver un formalisme existant approprié*

IRISA

- *Objectif de ce travail préliminaire :*
  - *Etudier des techniques de formalisation pour la sécurité des réseaux à travers une étude de cas*

LSR

# Démarche

- Choix d'un Exemple :
  - Réseau (Internet), caractéristique, réaliste
- Choix des Formalismes :
  - Formalismes proposés spécifiquement pour modéliser des politiques de sécurité de réseaux
- Choix des Critères de Comparaison :
  - Facilité à traduire la description informelle de la politique
  - Critères utilisés en Génie Logiciel pour décrire et comparer des formalismes

IRISA

LSR

IRISA

# Exemple
## Contexte

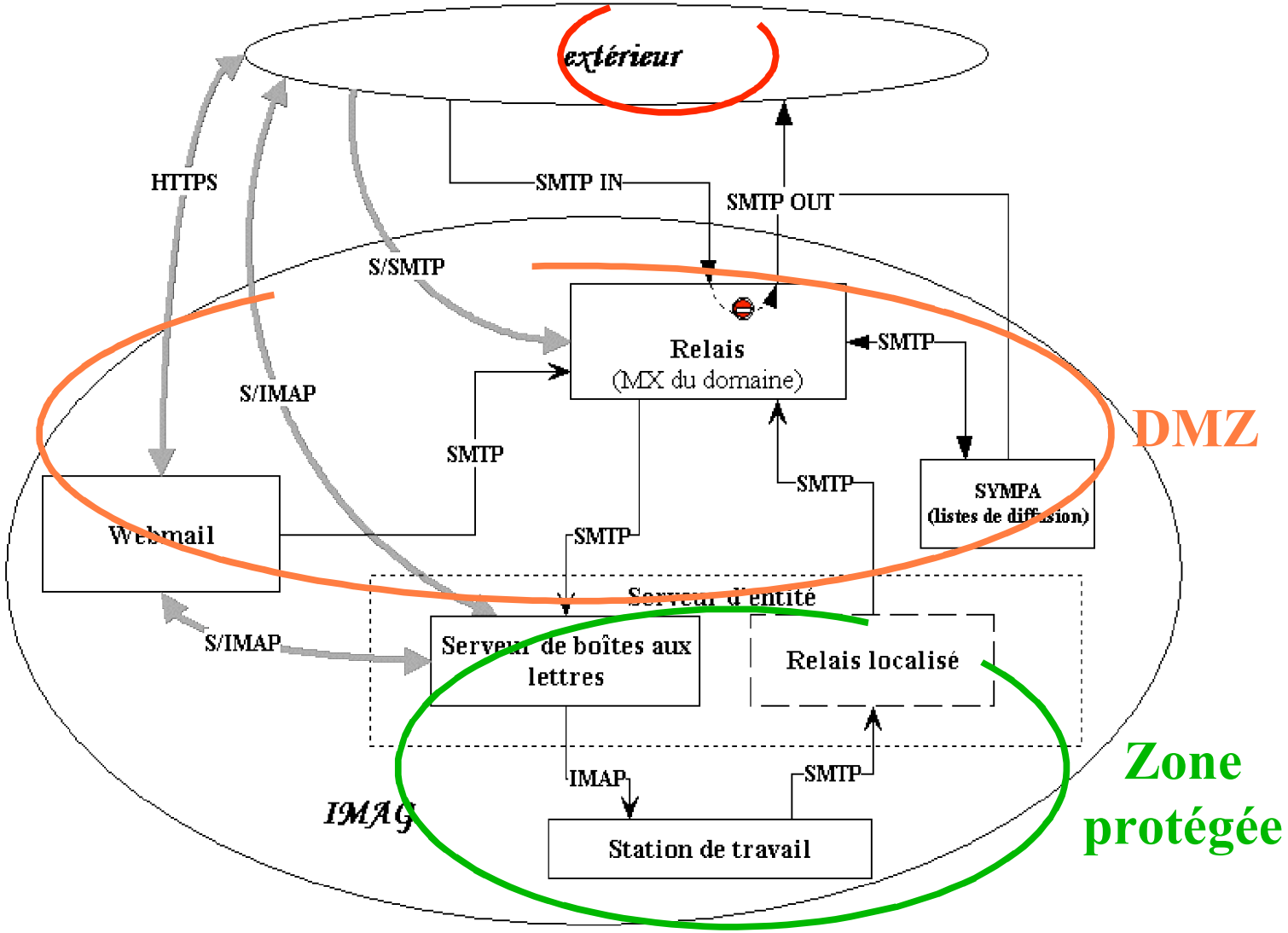
- Tiré de l'étude de cas "les politiques de sécurité de l'IMAG", développée selon l'axe messagerie électronique
  - Documents IMAG
  - Règles sur la messagerie
  - Etat de l'art technique
- Basé sur une découpe architecturale du réseau en zones :
  - Extérieur
  - Intérieur
  - DMZ

LSR

IRISA

# Exemple

## Schéma des zones du réseau



# Politique de sécurité de messagerie

## Extrait de l'étude IMAG

LSR

IRISA

| *n°* | *Règle* |
|---|---|
| 1 | Les relais de messagerie ouverts sur l'extérieur doivent être placés dans la DMZ ; ils seront par la suite appelés "relais principal". |
| 2 | Il n'y aura pas de compte utilisateur sur les relais principaux. |
| 3 | Les serveurs de boîtes aux lettres qui contiennent les comptes des utilisateurs seront dans la zone protégée. |
| 4 | Les relais principaux sont les seules machines habilitées au dialogue SMTP avec le monde extérieur : le relais du courrier entrant vers les serveurs de boîtes aux lettres et du courrier sortant vers l'extérieur s'effectue par les relais principaux. |
| 5 | Les serveurs de boîte aux lettres peuvent être utilisés comme relais internes de messagerie. |
| 6 | En entrée de site la politique de filtrage par défaut est que tout ce qui n'est pas explicitement autorisé est interdit. |
| 7 | Tous les messages de l'extérieur arrivant sur une machine qui n'est pas un relais principal doivent être redirigés sur un relais principal. |
| 8 | Il est interdit de relayer des courriers de l'extérieur vers l'extérieur. |

LSR

# Politique de sécurité (2)

## Extrait de l'étude IMAG

IRISA

9 [blacklisting] Refus de communiquer avec les relais appartenant à une liste quotidiennement mise à jour par notre fournisseur MAPS (Mail Abuse Prevention System).

10 [greylisting] La première fois qu'un relais extérieur inconnu du réseau essaye de transmettre un message à un relais externe, un code d'erreur de transmission doit être envoyé à ce relais extérieur. Si le relais extérieur rééssaie d'envoyer le courrier après un certain délai (maximum 10 minutes), le courrier pourra être accepté. Ce processus recommence si le relais extérieur n'a plus de contact avec les relais principaux pendant plus de 15 jours (fenêtre glissante).

11 Des logiciels antivirus et antispams sont installés au niveau des relais de messageries ou au niveau des boîtes aux lettres. Il est possible de mettre en place une protection au niveau des postes clients.

12 Toute pièce jointe à un courrier électronique doit être contrôlé par un logiciel anti-virus mis à jour au moins une fois avant d'être ouverte (virus, troyens, malwares divers).

13 Tout courrier électronique doit être modifié s'il contient un virus. Le virus est supprimé du courrier et le destinataire en est prévenu.

14 Tout courrier électronique doit être modifié s'il contient un fichier attaché potentiellement dangereux. Le fichier attaché est supprimé. L'émetteur et le destinataire en sont prévenus, on fournit au destinataire une adresse pour obtenir l'attachement conservé par le système.
Les fichiers attachés « à risque » qui ont été détachés (et eux seuls) sont conservés pendant 15 jours.

15 Tout courrier électronique doit être modifié s'il est analysé par un relais ou une autre machine comme un spam. On ajoute dans les en-têtes une indication qu'il s'agit probablement d'un spam.

16 Les triplets {émetteur, destinataire, heure} sont conservées pendant une durée inférieure à l'année, tels qu'enregistrés par le mécanisme « syslog » standard.

# Analyse des règles de la politique

LSR

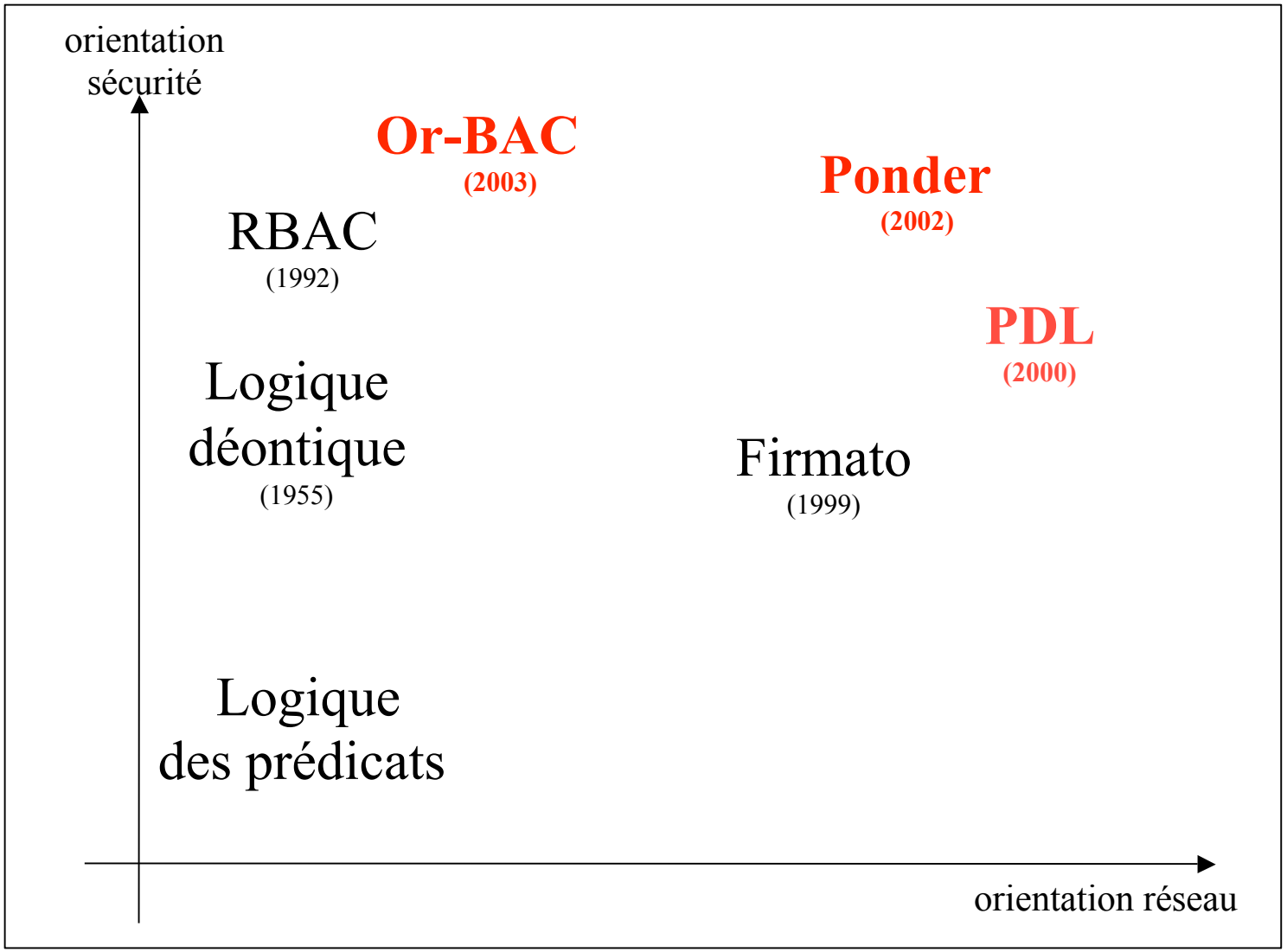
IRISA

| *n°* | *non comportementale* | *sujet imprécis* | *déonticité* | *temporalité* | *transition* |
|---|---|---|---|---|---|
| 1 | X | | | | |
| 2 | X | | | | |
| 3 | X | | | | |
| 4 | | | | | |
| 5 | | | X | | |
| 6 | | | X | | |
| 7 | | X | | | X |
| 8 | | | | | |
| 9 | | | | | |
| 10 | | | | X | X |
| 11 | X | | X | | |
| 12 | | | | | X |
| 13 | | | | | X |
| 14 | | | | | X |
| 15 | | | | | X |
| 16 | | X | | X | X |

# Les Formalismes

LSR

IRISA



# PDL

Présentation [Lobo 99]

LSR

- 2 types de règles :

  event ***causes*** action ***if*** condition

  policy-event ***triggers*** event ***if*** condition

- L'ensemble des *event* est une algèbre de processus. A chaque *event* sont attachés différents *attribute* (lieu/instant de génération, ...)
- Les *action* sont des commandes du système et les *condition* des fonctions booléennes. Les 2 types de fonctions prennent comme paramètres des *attribute*.

IRISA

- Les règles 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16 se modélisent bien en PDL.
- Règle 4 : *"Les relais externes sont les seules machines habilitées au dialogue SMTP avec le monde extérieur".*

*transfer (x, x', m)* ***causes*** *accept_transfer (x', x, m)* ***if*** *exterior (x), inDMZ (x'), relay (x')*
*transfer (x, x', m)* ***causes*** *accept_transfer (x', x, m)* ***if*** *exterior (x'), inDMZ (x), relay (x)*
*transfer (x, x', m)* ***causes*** *reject_transfer (x', x, m)* ***if*** *exterior (x),* □ *(inDMZ (x'), relay (x'))*
*transfer (x, x', m)* ***causes*** *reject_transfer (x', x, m)* ***if*** *exterior (x'),* □ *(inDMZ (x), relay (x))*

# PDL

## Modélisation statique

LSR

IRISA

- Règle 1: *"Les relais de messagerie ouverts sur l'extérieur (relais externes) doivent être placés dans la DMZ".*
- Cette règle ne peut être traduite directement en PDL : la structure d'un règle PDL est centrée autour d'une action déclenchée par un événement. Cette règle ne référence ni événement ni action.
- Il faut modifier la sémantique de la règle pour la modéliser comme un comportement.
- On réécrit la règle ainsi : *"Une machine agissant comme un relais externe doit être dans la DMZ".*
- Règles PDL:

*transfer (x, x', m)* ***causes*** *accept_transfer (x', x, m)* ***if*** *exterior (x), inDMZ (x'), relay (x')*
*transfer (x, x', m)* ***causes*** *accept_transfer (x', x, m)* ***if*** *exterior (x'), inDMZ (x), relay (x)*
*transfer (x, x', m)* ***causes*** *reject_transfer (x', x, m)* ***if*** *exterior (x),  (inDMZ (x'), relay (x'))*
*transfer (x, x', m)* ***causes*** *reject_transfer (x', x, m)* ***if*** *exterior (x'),  (inDMZ (x), relay (x))*

- La structure événement-action modélise un comportement normal du système, pas une réponse sécurité.
- Mêmes règles PDL que pour le besoin n°4.
- Ce problème se retrouve pour la modélisation des règles 2, 3 et 11.

# PDL

## Gestion des conflits

LSR

IRISA

- Conflit :

  R1: *transfer (r, r', m) causes accept_transfer if dmz(r') and relay(r') and ext(r)*

  R6: *transfer (r, r', m) causes reject_transfer if ext(r) and int(r')*

  Si un transfert est effectuée par une machine extérieure sur une relais de la DMZ, les deux règles se contredisent.

- En PDL un conflit est spécifié manuellement par une contrainte [Chomicki 01]
  - ***never*** $a_1$ ***and ... and*** $a_n$ $a_1, \ldots, a_n$ ***actions***
  - C = ***never*** *accept_transfer (r, r', m)* ***and*** *reject_transfer (r, r', m)*

- Résolution des conflits :

  Un moniteur édite les traces d'actions ou d'événements pour les rendre conformes à une contrainte.

  $M_C$ *(accept_transfer (r, r', m)* ***and*** *reject_transfer (r, r', m)) = accept_transfer (r, r', m)*

# Ponder

## Présentation

LSR

- Règles représentant l'*autorisation*, l'*interdiction*, l'*obligation*, la *délégation* ou la *contention*.
- Règle 4 : *"Les relais externes sont les seules machines habilitées au dialogue SMTP avec le monde extérieur".*

***inst auth+*** *{*
***subject*** */Machine/Interior/DMZ/Relay ;*
***target*** */Machine/Exterior/ ;*
***action*** *transfer (/Mail), get (/Mail) ; }*

IRISA

***inst auth-*** *{*
***subject*** */Machine/Interior/ ;*
***target*** */Machine/Exterior/ ;*
***action*** *transfer (/Mail), get (Mail) ; }*

- Règle 11 : "*La première fois qu'un relais extérieur inconnu du réseau essaye de transmettre un message à un relais externe, un code d'erreur de transmission doit être envoyé à ce relais*".

***inst oblig*** *{*
***on*** *transfer (R=/Exterior, R'=/DMZ/Relay, /Mail) ;*
***subject*** ***r'*** *;*
***target*** *r;*
***do*** *reject_transfer (R, R', m), return_transfer_error (R, R', m) ;*
***when*** *unknown_relay (R) ; }*

# Ponder

## Modélisation statique

LSR

IRISA

- En Ponder on type un objet en utilisant la notion de domaine. Les domaines sont ordonnés partiellement. La relation est notée "A/B" et signifie "*l'ensemble des objets du domaine B appartenant aussi au domaine A*".
- Règle 1 : "*Les relais de messagerie ouverts sur l'extérieur doivent être placés dans la DMZ*".
- A l'instar de PDL,
  - non modélisable dans cet état.
  - réécriture de la règle
  - ***inst auth+*** *{*
    ***subject*** */Machine/Interior/DMZ/Relay ;*
    ***target*** */Machine/Exterior/ ;*
    ***action*** *transfer (/Mail), get (/Mail) ; }*
- Autre solution : la structuation en domaines des types permet d'introduire des limitations sur la construction des domaines (à faire !)

# Ponder

## Gestion des conflits

LSR

- Règle 6 : *"En entrée de site la politique de filtrage par défaut est que tout ce qui n'est pas explicitement autorisé est interdit".*

***inst auth-*** *{*
***subject*** */Machine/Interior;*
***action*** *transfer (/Mail), get (/Mail) ;*
***target*** */Machine/Exterior; }*

- Conflit : Règle 4 et Règle 6
- Gestion du conflit par méta-politique :

IRISA

***inst meta raises*** *R.action {*
***exists*** *(R|R.type=/Rule* ***and*** *R'|R'.type=/Rule* ***and*** *R.subject = R'.subject* ***and*** *R.action = R'.action = transfer* ***and*** *R.target = R'.target* ***and*** *R.modality != R'.modality );*
*R.modality = '"auth+"; }*

- La méta-politique utilise une relation de précédence sur les règles selon les modalités.

# Or-BAC
## Présentation

LSR

IRISA

- Structure d'une règle Or-BAC :
  ***permission*** *(orgExpr, roleExpr, activityExpr, viewExpr)*
- Signification : dans l'organisation *orgExpr* les sujets assumant le rôle *roleExpr* peuvent effectuer une action contenue dans *activityExpr* sur un object de *viewExpr*.
- Pour cette modélisation nous définissons :
  - Les organisations comme les zones du réseau : Internal, DMZ, Private
  - Les machines du réseau comme sujets
    - Rôles : relay, mailbox, station, extern_relay, intern_relay
  - Les machines du réseau et de l'extérieur comme les objets
    - Vues : external
  - Les transmissions de messages comme actions : send_mail, relay_mail, stock_mail, get_mail, read_mail
    - Activités : relay_activity, mailbox_activity, station_activity

# Or-BAC

## Modélisation des contraintes statiques

LSR

- *R1: "Les relais de messagerie ouverts sur l'extérieur doivent être placés dans la DMZ".*

***permission** (DMZ, relay, relay_activity, exterior)*

- *R2: **Il n'y aura pas de compte utilisateur sur les relais externes.***

***prohibition** (DMZ, relay, mailbox_activity, any)*

IRISA

- *R3: Les serveurs de boîtes aux lettres qui contiennent les comptes des utilisateurs seront dans la zone protégée.*

***permission** (Private, mailbox, mailbox_activity, internal)*

***prohibition** (Interior, any_machine, mailbox_activity, any_zone)*

# Or-BAC

## Gestion des conflits

LSR

- Détection syntaxique des conflits
- Modélisation précédente  CONFLIT

***R3: prohibition*** *(Interior, any_machine, mailbox_activity, any_zone)*

***R3': permission*** *(Private, mailbox, mailbox_activity, interior)*

- Détection syntaxique des conflits :organisations, rôles, activités et vues des deux règles précédentes se recoupent alors que leurs modalités s'opposent  CONFLIT !

IRISA

- Résolution des conflits :
  - Traduction en logique du premier ordre
  - Priorisation des règles : R3' prioritaire sur R3

# Critères de Comparaison

LSR

- Atomes :
  - Typage, structuration
- Compositionalité :
  - Cohérence (conflits), completude (règle par défaut)
- Pouvoir d'Expression :
  - Modalités, classe de langage, réflexivité, états

IRISA

- Modèle d'Exécution :
  - Déclenchement, dirigé par les données ou les buts
- Methodologie :
  - Phase du dévelopement, rafinement, management scenario, cycle de vie type de la cible, fonctionalité de la cible

LSR

IRISA

# Analyse

## Table comparative

| | *PDL* | *Ponder* | *Or-BAC* |
|---|---|---|---|
| Typing | in conditions | in domain definition | in abstraction, definitions and contexts |
| Structuring | complex events | relation on domains<br>relation on roles | relation on organizations, roles, views, activities |
| Default rule (completeness) | no | should be specified | restrictive policy by default |
| Inter-rules conflicts (consistence) | manual specification of conflicts<br>manual resolution of conflicts | syntactic detection of conflicts<br>precedence relation to resolve conflicts | syntactic detection of conflicts<br>precedence relation to resolve conflicts |
| Modalities | triggered obligation | permission/interdiction<br>triggered obligation | permission/interdiction<br>obligation |
| Reflexivity | no | yes | administration model |
| Statefulness | yes | no | no |
| Triggering | easy apart in case of use of pseudo-permission | requires a mapping | requires a mapping (realized with abstractions?) |
| Data/goal driven | data driven | data driven | data driven |

# Bilan

LSR

IRISA

- Chaque formalisme permet de représenter les besoins informels de la politique :
  - PDL, orienté politiques de bas niveau (protocoles ou gestion du matériel), doit être augmenté pour modéliser des propriétés de sécurité abstraites
  - Ponder est complet mais parfois trop riche (et confus)
  - Or-BAC, orienté sécurité des organisations, présente des règles très structurées qui nécessitent un important travail de conceptualisation
- Les 3 formalismes insistent sur le caractère déclaratif et incohérent d'une politique de sécurité
  - Méthodes de spécification, détection et résolution de conflits

# Retour sur les objectifs

LSR

IRISA

- La comparaison des trois formalismes a permis d'étudier
  - Les modalités,
  - Le vocabulaire utilisé, ...
- Par contre, pas de prise en compte spécifique
  - Des problèmes d'adéquation avec le système réel décrit et le test de ce système
  - Le choix d'un formalisme est assez indifférent aux problèmatiques du test :
    - Différents niveaux d'abstraction des politiques par rapport au système
    - Spécificités techniques du test

# Conclusion

LSR

IRISA

- Test : Conformité d'un système vis à vis d'une politique de sécurité
  - Relations de conformité et modalités
  - Signification des verdicts en sécurité
  - Localisation du testeur et des sondes
  - Fiabilité des informations

  → cf. exposé de Laurent Mounier demain
- Raffinement
  - Politique de haut niveau → Politique de bas niveau
  - Identification des niveaux pertinents

  → cf. exposé de Nicolas Stouls et de Jean-Claude Fernandez

LSR

# Annexe A : Modèle PDL

IRISA

| *R* | *PDL rule set* |
|---|---|
| 1 | request c=(r, transfer, r')<br>**causes** grant(c)<br>**if** dmz(r) **and** relay(r) **and** exterior(r')<br>request c=(r, transfer, r')<br>**causes** grant(c)<br>**if** dmz(r') **and** relay(r') **and** exterior(r) |
| 2 | request c=(u, addAccount, m)<br>**causes** deny(c) **if** dmz(m) **and** relay(m)<br>request c=(u, connect(c), m)<br>**causes** deny(c) **if** dmz(m) **and** relay(m) |
| 3 | request c=(u, addMailbox, n)<br>**causes** deny(c) **if** dmz(n)<br>request c=(r, transfer(m), r')<br>**causes** grant(c) **if** relay(r) **and** interior(r)<br>**and** mailbox(r') **and not** private(r')<br>request c=(r, transfer(m), r')<br>**causes** deny(c)<br>**if** relay(r) **and** mailbox(r') **and not** private(r') |
| 4 | request c=(r, transfer, r')<br>**causes** deny(c)<br>**if not** (relay(r) **and** dmz(r)) **and** exterior(r')<br>request c=(r, transfer, r')<br>**causes** deny(c)<br>**if not** (relay(r') **and** dmz(r')) **and** exterior(r) |

| *R* | *PDL rule set* |
|---|---|
| 5 | request c=(r, transfer, r')<br>**causes** grant(c)<br>**if** mailbox(r) **and** private(r) **and** interior(r')<br>request c=(r, transfer, r')<br>**causes** grant(c)<br>**if** mailbox(r') **and** private(r') **and** interior(r) |
| 6 | request c=(r, transfer, r')<br>**causes** deny(c) **if** exterior(r) **and** interior(r')<br>**never** grant c=(r, transfer(m), r') **and** deny(c)<br>**monitor** ( grant(c) **and** deny (c) ) = grant(c) |
| 7 | request c=(m, transfer(e), m')<br>**causes** deny(c)<br>**if** interior(m) **and** exterior(e.src)<br>**and** exterior(e.dst) |
| 8 | request c=(r, transfer(m), r')<br>**triggers** redirect-mode(m)<br>**if** exterior(r) **and** interior(r')<br>**and not** (relay(r') **and** dmz(r'))<br>redirect-mode(m), request c=(r, transfer(m),r')<br>**causes** grant(c)<br>**if** relay(r') **and** dmz(r') |

# Annexe B : Modèle Ponder

LSR

IRISA

*Rqrt*

1
```
inst auth+  P1A1 {
  subject /Machine/DMZ/Relay/Extern ;
  action   transfer(/Mail) ;
  target   /Machine/Exterior ; }
```

2
```
inst auth- P2A1 {
  subject /User;
  action   addAccount (/Account);
  target   /Machine/DMZ/Relay; }
```

3
```
inst auth+ P3A1 {
  subject /User;
  action   addMailbox (/Mailbox);
  target   /Machine/Private; }
```

4
```
inst auth- P4A1 {
  subject /Machine/Interior;
  action   transfer(/Mail);
  target   /Machine/Exterior; }
```

5
```
inst auth+ P5A1  {
  subject /Machine/Private/Mailbox;
  action   transfer(/Mail);
  target   /Machine/Interior;}
```

6
```
inst auth- P6A {
  subject /Machine/Exterior;
  action   transfer (/Mail);
  target   /Machine/Interior; }
```

7|8
```
inst auth- P7A {
  subject /Machine/Interior;
  action   transfer (m=/Mail);
  target   /Machine/Exterior;
  when    m.src.isExt() and m.dest.isExt(); }
```

*Corresponding Ponder rule set*

```
inst auth+  P1A2 {
  subject /Machine/Exterior;
  action   transfer(/Mail)
  target   /Machine/DMZ/Relay; }
```

```
inst auth- P2A2 {
  subject /User;
  action   connect ();
  target   /Machine/DMZ/Relay/Account; }
```

```
inst auth+ P3A2 {
  subject  /Machine/Private/Mailbox;
  action   transfer(/Mail);
  target   /Machine/Private/Station; }
```

```
inst auth+ P3A3 {
  subject /Machine/Interior/Relay;
  action   transfer(/Mail);
  target   /Machine/Private/Mailbox;
```

```
inst auth- P4A2 {
  subject /Machine/Exterior;
  action   transfer(/Mail);
  target   /Machine/Interior; }
```

```
inst auth+ P5A2  {
  subject /Machine/Interior;
  action   transfer(/Mail);
  target   /Machine/Private/Mailbox;}
```

```
inst meta P6M raises R.action {
  exists(R | R.type=/Rule and R.subject==/Machine/Exterior
  and R.action==transfer(/Mail) and R.target==/Machine/Interior );
  R.modality == '"auth+"; }
```

```
inst oblig P8O {
  on  R.transfer(m=/Mail) to R'=/Machine/Interior);
  subject    /Machine;
  do          transfer(m) to /Machine/DMZ/Relay;
  when       R == /Machine/Exterior and R' != /Machine/DMZ/Relay
```

# Annexe C : Modèle Or-BAC

LSR

IRISA

| *Rqrt* | *Or-BAC rule set* |
|---|---|
| 1 | permission ( dmz, relay, relaying, any-machine ) |
| 2 | interdiction ( dmz, relay, mail-boxing, any-machine ) |
| 3 | permission ( private, mailbox, mail-boxing, workstation )<br>permission ( private, mailbox, mail-boxing, relay )<br>interdiction ( dmz, int-machine, mail-boxing, any-machine ) |
| 4 | interdiction ( interior, int-machine, relaying, ext-machine, not relay-in-dmz ) |
| 5 | permission ( interior, mailbox, relaying, int-machine ) |
| 6 | interdiction ( dmz, int-machine, any-activity, any-machine ) |
| 7 | interdiction ( interior, int-machine, ext-relaying, any-machine ) |
| 8 | obligation ( interior, any-machine, redirecting, relay, not-relay-and-receive) |