Prelude Manager

Prelude Manager est le point central de l'architecture Prelude. C'est lui qui va récupérer les alertes parsées par le module LML de Prelude, servir de point central dans les échanges avec le Prelude Correlator, récupérer directement les informations envoyées par les sondes compatibles IDMEF, envoyer les informations pertinentes à la base de données et filtrer les alertes sur tous les critères IDMEF possibles et imaginables.

Filtres IDMEF

Dans certains cas, il est nécessaire de retirer certains éléments récurrents présents dans le bac à alertes Prelude. Dans l'exemple suivant, il s'agira de retirer toutes les alertes provenants de sondes Nessus un peu trop bruyantes en terme de génération d'événements de sécurité. Les sondes sont au nombre de 3 :

- 10.54.2.98 : dctscnvlp01.prod.gemplus.com.
- 10.9.171.66 : dctscnvlp02.gemalto.com.
- 10.9.80.2 : dctscnvlp03.cbp.dct.sdc.gemalto.com.

Le filtrage va se faire au sein du Prelude manager, de façon à intercepter les alertes avant leur entrée dans la base de données Prelude.

Les filtres s'appliquent sous */etc/prelude-manager/prelude-manager.conf*
La syntaxe de base d'un filtre IDMEF est la suivante :

```
[idmef-criteria]
rule = <Filtre IDMEF>
hook = db[default]
```

La bannière *[idmef-criteria]* indique le début du filtre.
Le préfixe *rule* va être suivi de la règle de filtrage.
Enfin, le hook redirige les informations correspondant au filtre. les options du hook sont au nombre de deux :

- db[default] : Ce qui correspond à la rule n'est pas forwardé vers la base de données. les éléments sont donc abandonnés.

- relaying[default] : ce qui correspond au fait que la rule est forwardé vers la base de données.

Les filtres IDMEF, comme leurs noms l'indiquent, s'écrivent avec la même syntaxe que les messages IDMEF classiques. Quelques petits exemples :
Pour filtrer via une adresse IP : *alert.source(0).node.address(0).address.*
Pour filtrer via un FQDN : *alert.source(0).node.name.*
Enfin, pour filtrer via l'intitulé de l'alerte : *alert.classification.text.*

Les éléments IDMEF sont compatibles avec les opérateurs de comparaisons classiques.
il est également possible d'empiler les filtres, ou de rendre les règles plus complexes, avec les opérateurs 'ET' et 'OU', respectivements représentés par '&&' et par '||'.

Le règle permettant de bloquer les alertes remontées par une sonde NESSUS est donc la suivante :

```
[idmef-criteria]
rule = alert.source(0).node.address(0).address != '10.54.2.98' &&
alert.source(0).node.address(0).address != 'dctscnvlp01.gemalto.com'
hook = db[default]
```

ATTENTION : La configuration actuel du Prelude permet de filtrer un nom d'hôte au travers d'un filtre 'address' uniquement puisque le Prelude LML à été configuré de façon à ne pas résoudre les adresses. Celui-ci doit traiter des informations provenant de plusieurs zones DNS distinctes. Il est donc nécessaire que les machines ne tentent pas de résoudre des noms qu'elles ne connaissent pas. L'option de retrait de la résolution de nom se fait sur le fichier */etc/init.d/prelude-lml* :

```
[...]
        # Start daemon.
        daemon $NICELEVEL $progpath -d -P $pidfile --no-resolve
[...]
```

Dans le cas ou le filtre doit être placé en fonction du nom de l'alerte dans le Prelude, un filtre texte peut donc être mis en place. Pour retirer les alertes inconnues remontées régulièrement dans le Prelude, la règle suivante peut être mise en place :

```
[idmef-criteria]
rule = alert.classification.text != "Unknown problem somewhere in the system."
hook = db[default]
```

Si l'on veut maintenant coupler les règles créées précédemment, il ne sera malheureusement pas possible de créer plusieurs filtres IDMEF, mais il faudra fusionner les règles sur une seule ligne avec l'opérateur d'addition '&&'.
Les règles ne sont donc pas très esthétiques, ni très lisibles une fois accumulées, mais il n'existe pas encore de fonctionnalités permettant de faire autrement pour le moment.