IUT NICE CÔTE D'AZUR – DÉPARTEMENT INFORMATIQUE – S3T

Projet optionnel - _EspionGD

Lisez-moi

16 janvier 2013

Répartition des responsabilités :

- *EspionP et réseau EspionP :* Sylvain Levasseur
- *Réseau EspionP/EspionGD et EspionGD/EspionG :* Gauthier Cibert—Volpé
- *Système de filtrage EspionGD/EspionG :* Martini Didier

_EspionGD

1. Instructions d'installation

Plusieurs démonstrateurs ont été imaginés et mis en place. En effet, des éléments vus dans le cadre des TP/TD tels que PanneauG, MosaiqueG, TrafalgarMap et TrafalgarPlayer ont été adaptées pour illustrer des utilisations possible du package _EspionGD.

Par conséquent :

- Deux launchers sont présents pour l'exécution du *.jar* qui contient le serveur EspionP, un *.bat* (Windows) et un *.sh* (Linux)
- Deux launchers sont présents pour l'exécution du *.jar* qui contient le serveur EspionGD, un *.bat* (Windows) et un *.sh* (Linux)
- Deux launchers sont présents pour l'exécution du *.jar* qui contient TrafalgarPlayer observé par EspionG, un *.bat* (Windows) et un *.sh* (Linux)
- Deux launchers sont présents pour l'exécution du *.jar* qui contient TrafalgarMap observé par EspionG, un *.bat* (Windows) et un *.sh* (Linux)
- Deux launchers sont présents pour l'exécution du *.jar* qui contient PanneauG observé par EspionG, un *.bat* (Windows) et un *.sh* (Linux)
- Deux launchers sont présents pour l'exécution du .jar qui contient MosaiqueG observé par EspionG, un *.bat* (Windows) et un *.sh* (Linux)

2. Fonctionnalités et spécifications techniques

Rôles :

- EspionG (Espion hameçon)
  - Récupère de manière non-intrusive un maximum d'événements, applique le système de filtrage et envoie les événements qui ont réussi à passer le filtrage vers son EspionGD
  - Retourne le résultat de commandes qu'EspionP peut lui demander via EspionGD
- EspionGD (Espion délégué)
  - Serveur pour les EspionG de la machine
  - Connecté à l'EspionP
  - Récupère les événements renvoyé par les EspionG, applique le système de filtrage et envoie les événements qui ont réussi à passer le filtrage vers EspionP
  - Transfère les données de l'EspionP vers les EspionG concernés si nécessaire

- EspionGP (Espion principal)
  - Serveur qui va centraliser tous les événements (qui ont passé les filtres)
  - Etablir en mettre en place les règle de filtrage des espions délégués et des espions hameçons (par le biais des espions délégués)

- Une IHM qui permet la visualisation des événements et résultats de commandes, l'envoie de commandes, visualisation des clients connecté, envoyer des fichiers de config en vue du filtrage vers EspionGD

Système de filtrage :

Tout d'abord, les filtres appliqués dans EspionG implémentent toute l'interface *Validator* qui oblige l'implémentation de la méthode « `public Boolean validate(final Object évent, final Long time) throws Exception` »

Le booléen renvoyé correspond au passage avec succès du filtre ou pas :

- `True` : l'événement répond aux critères du filtre
- `False` : l'événement ne répond pas aux critères du filtre

Chaque filtre (*Validator*) a un nom associé dans un *HashMap*.

L'assemblage de filtre est fait par une expression booléenne :

Exemple : « (*Validateur1* AND *Validateur2*) OR *Validateur3* »

- Si l'événement passe (TRUE) on remplace le nom du validateur par '`true`'
- Si l'événement ne passe pas (FALSE) on remplace le nom du validateur par '`false`'

L'événement passe le filtre '*Validateur1*' et le filtre '*Validateur3*' mais pas '*Validateur2*' on obtiendra : « (`true` AND `false`) OR `true`».

On utilisera le Groovy Shell pour analyser l'expression booléenne, ce qui nous permettra d'obtenir le booléen final qui nous indiquera si oui ou non on doit envoyer l'événement à EspionGD (grâce à la méthode `validate`).

3. Problèmes connus

- Le décalage temporel possible entre deux machines d'un réseau tel qu'exposé n'est pas pris en charge par la version actuelle de _EspionGD.

4. Idées d'améliorations possibles

Si ce n'est une remédiation des problèmes exposés plus haut, aucune amélioration n'a encore été imaginée pour _EspionGD.

5. Notes et informations

- Utilisation de XML et donc de libraire pour le faire. Nous avons choisi : JDOM : http://www.jdom.org/
- Utilisation de Groovy : http://groovy.codehaus.org/