Fonctionnement d'OSSIM

Dans le cadre de SIMS - Security Intrusion Management System

(http ://www.tcom.ch/Tcom/Projets/SIMS/sims.html)

Auteur : Joël Winteregg (joel.winteregg@eivd.ch)
Chef de projet : Cyril Friche
Professeur : Stefano Ventura
École : Swiss University of Applied Sciences (EIVD)
Tcom institut (http ://www.tcom.ch)
Date : 31 mai 2005

Table des matières

Chapitre 1

Fonctionnement logiciel d'OSSIM

Des informations relatives à l'installation sont disponibles dans les annexes ou sur le site officiel d'Ossim (http ://www.ossim.net/docs.php)

1.1 Les applications Ossim-server et Ossim-agent

Ossim-agent récupère simplement les informations des fichiers de logs des plugins (fichier fast.log pour Snort) et les envoie directement au serveur Ossim permettant ainsi le traitement temps réel de celles-ci. De plus, l'agent Ossim s'occupera de la mise en marche et de l'arrêt des différentes sondes qui lui sont connectées. Il ne sera ainsi pas nécessaire de démarrer la sonde Snort "à la main" puisque son activation sera effectuée depuis la console de management offerte par Ossim-server.

Ossim-server constitue le noyau de l'architecture. En effet, celui-ci comporte les modules d'analyse et de corrélation des données ainsi qu'un serveur Web permettant l'interaction avec l'utilisateur (administrateur réseau).

1.2 Fonctionnement de l'architecture avec une sonde Snort

Le principe de communication d'une sonde Snort avec le serveur OSSIM est illustré par la figure 1.1. Nous remarquons que l'IDS[1] Snort est indépendant du programme client d'OSSIM (nommé : ossim-agent) et que deux flux d'informations sont émis en direction du serveur.

1.2.1 Pourquoi deux flux d'informations ?

Le flux nommé "Requêtes SQL pour dépos des alertes" est utilisé afin de déposer directement les alertes dans la base de donnée "Snort BD" du serveur. Ceci permettra l'archivage de celles-ci et

[1] Intrusion Detection System

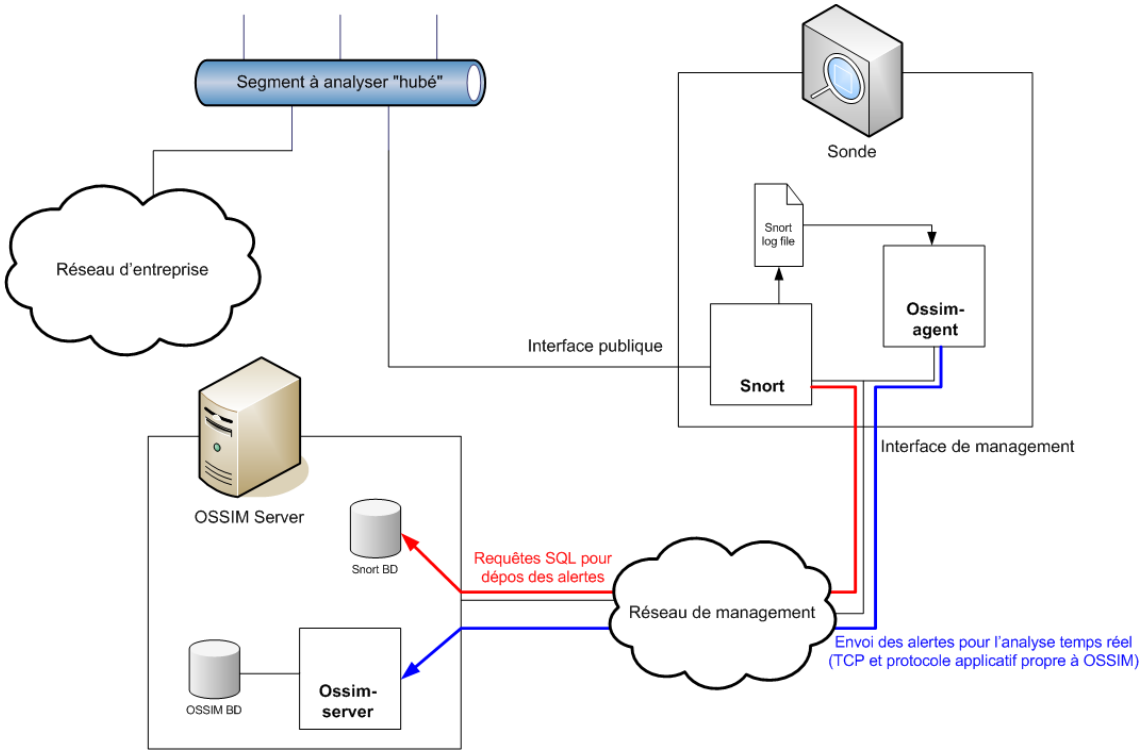

FIG. 1.1 – Principe de communication entre une sonde Snort et le serveur d'OSSIM

leur conslutation via ACID[2], permettant ainsi l'analyse précise de chaque alertes.

Le flux nommé "Envoi des alertes pour l'anayse temps réel (TCP et protocole applicatif propre à OSSIM" est quant à lui nécessaire pour le procédé d'analye et de corrélation temps réel opéré sur le serveur d'OSSIM.

Ces deux flux d'informations redondants sont indispensables si l'on ne veut pas redéfinir le protocole d'envoi des informations dans la base de donnée "Snort BD". En effet, le plugin de sortie Mysql ne serait pas suffisant pour un traitement temps réel puisque le stockage des informations dans une base de donnés "casse" le procédé temps réel. Un tel fonctionnement impliquerait l'interrogation continuelle de la base de donnée afin de découvrir les nouvelles données insérées. Les concepteurs d'OSSIM ont donc préféré utiliser deux flux d'informations plutôt que de créer un nouveau plugin de sortie pour Snort permettant d'envoyer les alertes dans un seul flux structuré au serveur. Dans ce mode de fonctionnement, c'est le serveur qui se chargerait ensuite du traitement temps réel et de l'insertion des informations dans une base de donnée.
Pour l'analyse et la corrélation, le serveur Ossim utilise uniquement les alertes provenant de l'agent Ossim, alors que les alertes directement stockées dans la base de données sont uniquement utilisées pour la

[2] Analysis Console for Intrusion Databases, interface Web intégrée à OSSIM permettant l'interrogation d'une base de donné Snort (développé dans le cadre du projet Open Source Snort)

consultation.

1.3 Fonctionnement de l'architecture avec une sonde Ntop

Le principe de communication d'une sonde Ntop avec le serveur OSSIM est illustré par la figure 1.2.

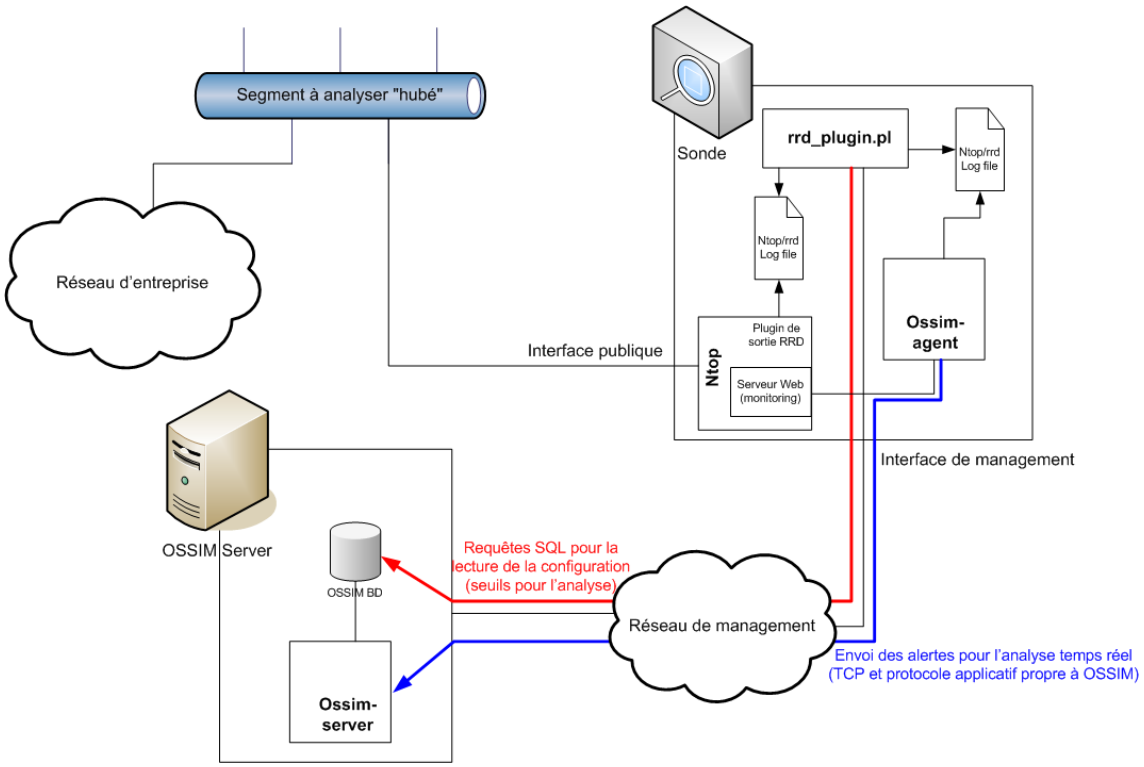

FIG. 1.2 – Principe de communication entre une sonde Ntop et le serveur d'OSSIM

1.3.1 Ntop c'est quoi ?

Ce logiciel analyse de manière temps réel le trafic réseau et met à disposition une liste de compteurs (par exemple : IP_DNSBytes, IP_HTTPBytes), permettant le monitoring ainsi que le calcul de statistiques. La sonde Ntop met en place un serveur Web permettant le monitoring ainsi que la configuration de celle-ci à distance.
Le plugin de sortie RRD[3] est nécessaire pour l'intégration de Ntop dans Ossim. Celui-ci permet l'enregistrement des données sous forme de tourniquet (les plus vieilles données sont écrasées par les nouvelles).

[3]Round Robin Database

L'interrogation de la base de donnée RRD est ensuite facilité à l'aide de l'outil RRDtool[4].

1.3.2 Fonctionnement

Le script Perl *rrd_plugin.pl* effectue la liaison entre Ntop et l'agent Ossim. Celui-ci interroge périodiquement la "base de donnée" RRD (illustrée par Ntop/rrd log file) à l'aide de l'outil RRDtool. Il récupère (via des requêtes SQL sur le serveur) les seuils des compteurs défini par l'administrateur réseau à l'aide du framework de configuration[5] et les compare aux des données précédemment récupérées (à l'aide de RRDtool).
Les éventuels dépassement des seuils sont ensuitre stocké dans un fichier de log (/var/log/ossim/rrd_plugin.log) qui sera récupéré par l'agent Ossim afin de permettre l'envoi temps réel des informations au serveur.
La corrélation des ces informations peut ensuite être effectuée sur le serveur.

1.4 Fonctionnement de l'architecture avec une sonde P0f

Le principe de communication d'une sonde P0f avec le serveur OSSIM est illustré par la figure 1.3.

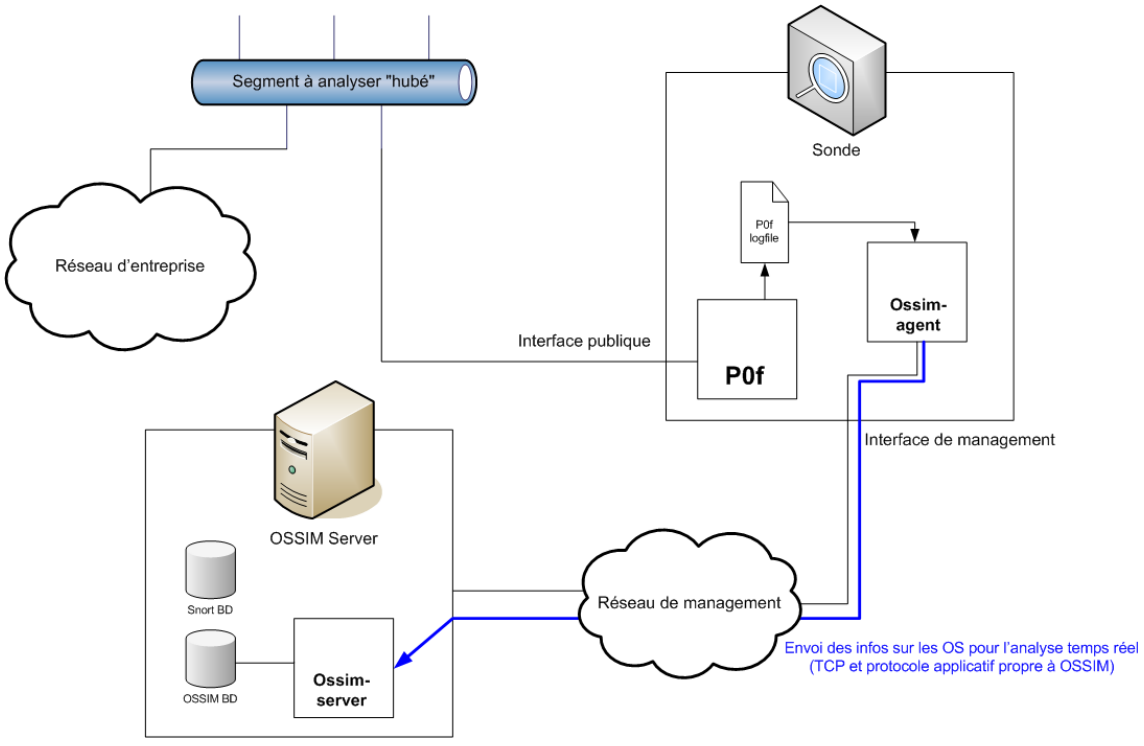

FIG. 1.3 – Principe de communication entre une sonde P0f et le serveur d'OSSIM

[4] http ://people.ee.ethz.ch/ oetiker/webtools/rrdtool/
[5] interface Web offerte par le serveur. Menu : Configuration - RRD config

1.4.1 P0f c'est quoi ?

P0f est un logiciel de détection de systèmes d'exploitations[6] passif. Il analyse les trames transitant sur le réseau (le segment analysé) et les compare avec une base de données des caractéristiques de chaque OS (prise d'empreintes) afin d'en retrouver l'OS correspondant. P0f est capable d'autres choses (non utilisée dans le cadre d'OSSIM) :

1. détection de la présence d'un firewall et NAT
2. détection d'un load balancer (répartiteur de charge réseau)
3. détection de la distance de la machine distante ainsi que le nombre d'heures de depuis le boot de celle-ci

P0f est totalement passif. Il ne génère aucun traffic réseau supplémentaire !

1.4.2 Fonctionnement

Celui-ci est assez simple. P0f écrit ses logs dans le fichier */var/log/ossim/p0f.log* (chemin fournit à P0f par l'agent Ossim lors du lancement de P0f). Ce chemin se trouve donc dans la configuration du plugin P0f (/etc/ossim/agent/plugins/p0f.xml) de l'agent. Le daemon agent (ossim-agent) s'occupera ensuite de les récupérer afin de les envoyer au serveur Ossim pour une analyse temps réel.

1.5 Fonctionnement de l'architecture avec une sonde TCPTrack

Le principe de communication d'une sonde TCPTrack avec le serveur OSSIM est illustré par la figure 1.4.

1.5.1 TCPTrack c'est quoi ?

TCPTrack est un sniffer affichant des informations sur les connexions TCP qu'il rencontre sur une interface. Il détecte passivement les connexions TCP sur l'interface à analyser et affiche les informations de la même manière que la commande Unix *top*. Il permet l'affichage des adresses source et destination, de l'état de la connexion, du temps de connexion ainsi que de la bande passante utilisée.

1.5.2 Fonctionnement

TCPTrack fonctionne d'une manière similaire à l'affichage Web des informations de Ntop. En effet, aucune information n'est spontanément envoyée vers le serveur Ossim. TcpTrack ouvre simplement un port serveur[7] sur la loopback de l'agent. C'est ensuite le serveur Ossim, qui lors du procédé de corrélation, interrogera si nécessaire l'agent afin qu'il interroge à son tour TCPTrack (via la loopback). Une fois les

[6] Operating System (OS)
[7] port 40003 par défaut

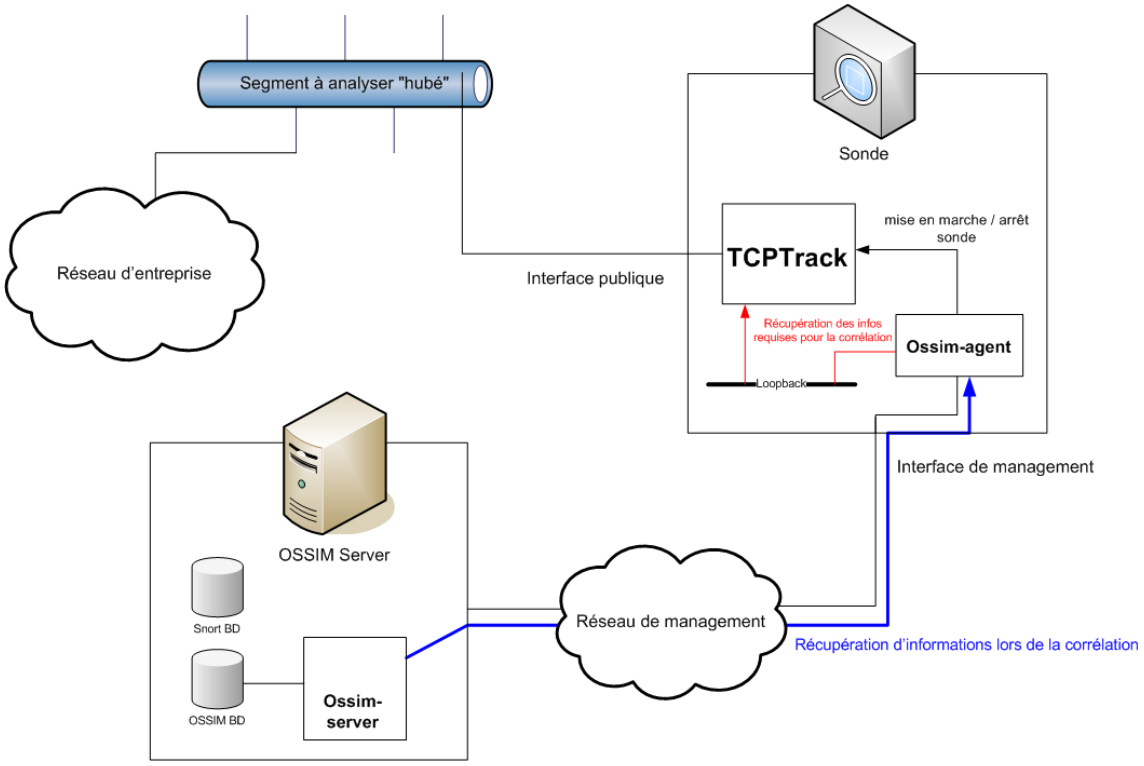

FIG. 1.4 – Principe de communication entre une sonde TCPTrack et le serveur d'OSSIM

informations récoltées par l'agent, celui-ci ce chargera de les remttres au serveur qui les utilisera pour la corrélation. L'agent joue donc un rôle d'intermédiaire entre le serveur et le sonde TCPTrack.

1.6 Fonctionnement de l'architecture avec une sonde PADS

Le principe de communication d'une sonde PADS[8] avec le serveur OSSIM est illustré par la figure 1.5.

1.6.1 PADS c'est quoi ?

PADS va permettre d'identifier les machines (adresses IP et MAC) ainsi que leurs services uniquement en sniffant le réseau. Il permettra l'affichage des services d'une machine sans avoir à opérer un scan actif comme nmap[9]. Il permettra l'affichage des services d'une machine configurée sur Ossim sans opérer un scan actif.

[8] Passive Asset Detection System

[9] Outil intégré à Ossim, permettant, entre autre, des scans de ports

1.6.2 Fonctionnement

Le logiciel PADS reportera simplement toutes les informations récoltées dans le fichier de log */var/log/ossim/pads.csv* (indiqué dans la configuration du plugin, fichier */etc/ossim/agent/plugins/pad.xml*). L'agent Ossim se chargera ensuite de les récolter et de les envoyer de manière temps réel au serveur.

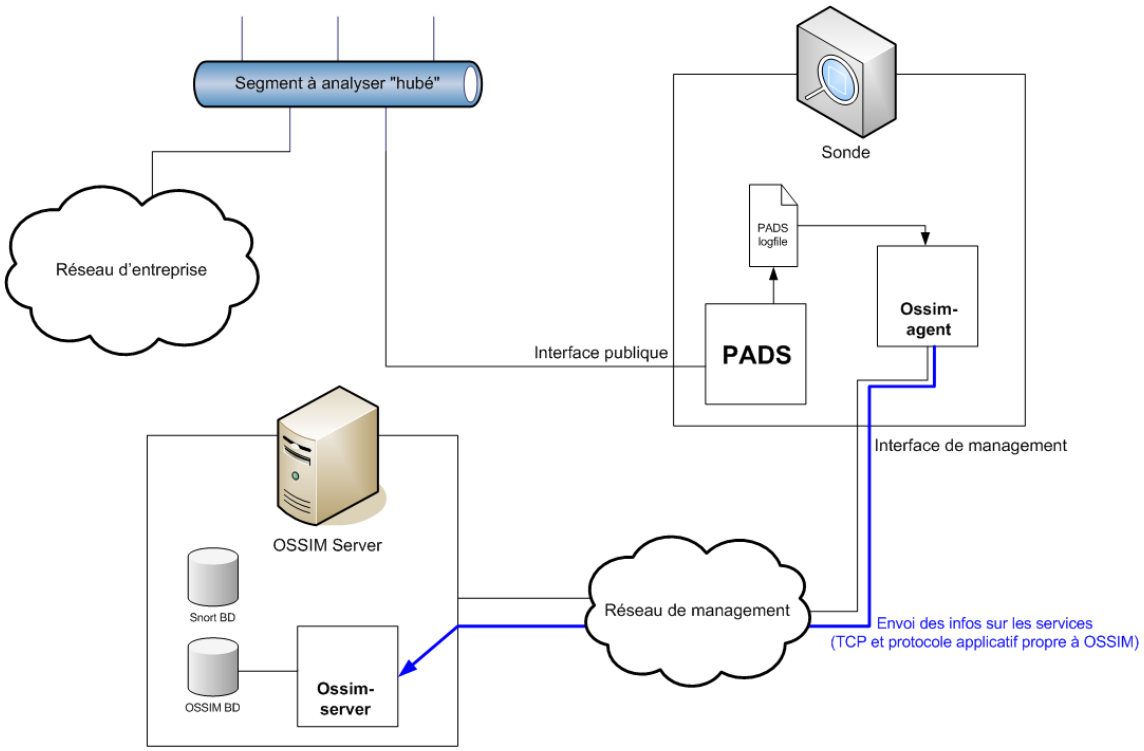

FIG. 1.5 – Principe de communication entre une sonde PADS et le serveur d'OSSIM

1.7 Fonctionnement de l'architecture avec une sonde Syslog

Le principe de communication d'une sonde HIDS[10] Syslog avec le serveur OSSIM est illustré par la figure 1.6.

1.7.1 Qu'est-ce qu'une sonde HIDS

Ces sondes ont pour but de rechercher des patterns spécifiques dans des fichiers de logs. Dès qu'une suite de caractère (pattern) appartenant à une règle de la blacklist est trouvée, une alerte est générée. Il est ainsi possible de remonter des événements spécifiques (critiques) vers la console de monitoring. De

[10] Host Intrusion Detection System

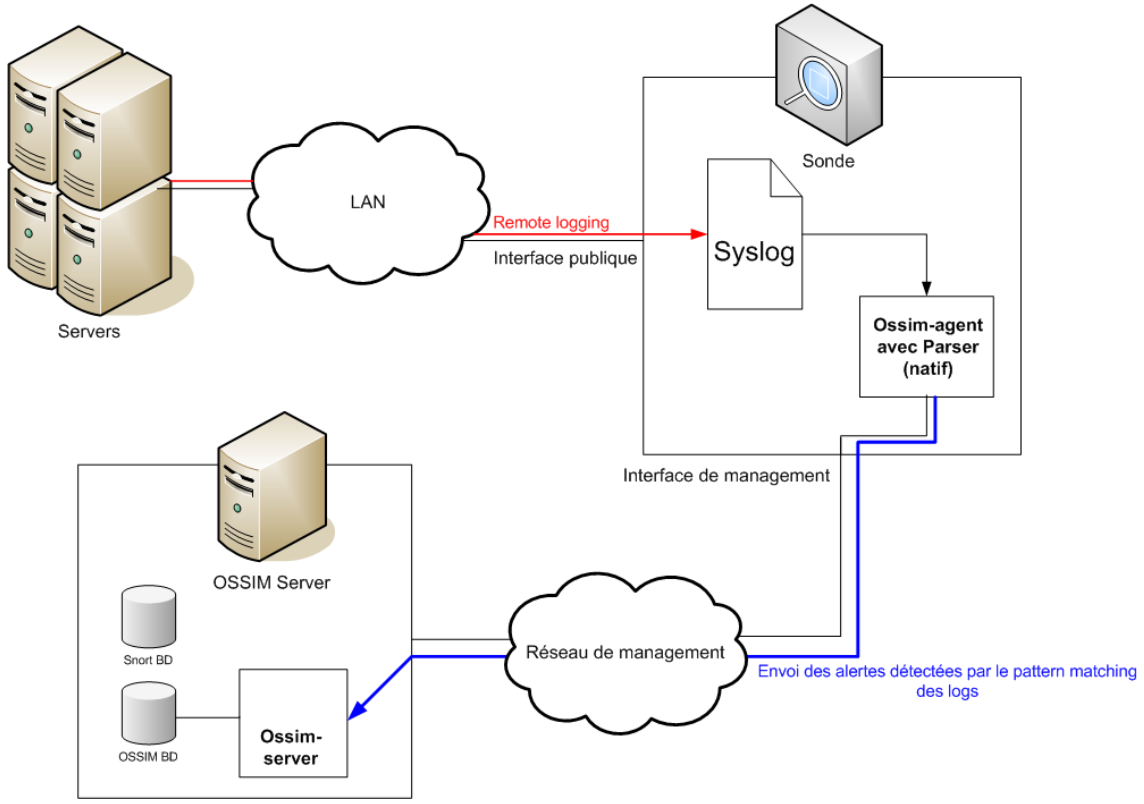

FIG. 1.6 – Principe de communication entre une sonde HIDS Syslog et le serveur d'OSSIM

plus, à l'aide du *sid* de chaque règle, il est possible de les utiliser dans des règles de corrélation sur le serveur.

1.7.2 Fonctionnement

Le parser analysant le fichier Syslog d'une machine Linux est directement présent sous forme native sur les agents Ossim. Son code et ses règles sont présents dans un seul et unique fichier */usr/share/ossim-agent/pyossim/ParserSyslog.py*. Il analysera le fichier de logs de manière temps réel et informera le serveur Ossim, via la connexion présente entre l'agent et le serveur, dès qu'un pattern présent dans la blacklist aura été détecté.

Annexe A

Installation et configuration d'Ossim-server

Ce chapitre décrit toutes les étapes d'installation d'un serveur Ossim[1] sur une plateforme GNU Linux/Debian. Celui-ci est tiré de http ://www.ossim.net/docs/INSTALL.Debian.quick.html

A.1 Prérequis

Disposer d'une machine Linux Debian ayant un noyau 2.6.xx. Avoir configuré le manager de paquets Debian (aptitude) afin qu'il récupère ceux-ci sur le site Web d'Ossim (cf. Section B.1.1).

A.2 Installation

Installation de la base de donnée MySql-Ossim :

```
# apt-get install ossim-mysql
```

Création du compte root et mise en place de son mot de passe :

```
# mysqladmin -u root password your_secret_password
```

Vous pouvez ensuite éditer le fichier */etc/mysql/my.cnf* afin de choisir l'adresse IP (bind-address) associée au serveur MySql.

La création des bases de données s'opère simplement à l'aide des commandes suivantes :

```
# mysql -u root -p
```

[1] Serveur des gestion des sondes ET framework de gestion (= interface Web)

```
mysql> create database ossim;
mysql> create database ossim_acl;
mysql> create database snort;
mysql> exit;
```

Le chargement des tables peut ensuite s'effectuer à l'aide des scriptes fourni par mysql-ossim :

```
# zcat /usr/share/doc/ossim-mysql/contrib/create_mysql.sql.gz \
    /usr/share/doc/ossim-mysql/contrib/ossim_config.sql.gz \
    /usr/share/doc/ossim-mysql/contrib/ossim_data.sql.gz \
    /usr/share/doc/ossim-mysql/contrib/realsecure.sql.gz | \
    mysql -u root ossim -p

# zcat /usr/share/doc/ossim-mysql/contrib/create_snort_tbls_mysql.sql.gz \
    /usr/share/doc/ossim-mysql/contrib/create_acid_tbls_mysql.sql.gz \
    | mysql -u root snort -p
```

Installation du serveur (daemon de corrélation et récupération des données des agents) :

```
 # apt-get install ossim-server
```

Installation du framework (interface Web) ainsi que du paquetage de la gestion des droits d'accès au serveur Web :

```
# apt-get install phpgacl

# apt-get install ossim-framework
```

Installation du paquetage *utils* permettant la gestion des connexions à la base de donnée Ossim :

```
# apt-get install ossim-utils
```

Installation de Nessus (pour la détection des vulnérabilités). Installation du serveur Nessus :

```
# apt-get install nessusd
```

Installation du client (utilisé par le script */usr/share/ossim/script/do_nessus.pl*, exécuté par le framework lors d'une demande de scan) pour l'exécution des scan nessus :

```
# apt-get install nessus
```

A.3 Configuration

Tous les paquetages intallés ci-dessus offrent une configuration garphique *ncruses* à l'installation. Celle-ci peut être effectuée à la demande via la commande :

```
# dpkg-reconfigure <nomDuPaquetage>
```

L'adresse IP du serveur ainsi que les nom d'utilisateurs et mots de passes des bases de données seront nécessaire. Il convient donc ensuite de créer ces utilisateurs. Si nous choissons de nous connecter aux bases de données à l'aide d'un utilisateur nommé *ossim* ayant comme mot de pase *ossim_pass*, il sera nécessaire d'opérer ainsi afin d'ajouter cet uilisateur sur les différentes bases de données :

```
# mysql -u root -p

mysql> GRANT ALL PRIVILEGES ON snort.* TO 'ossim'@'localhost'
        ->     IDENTIFIED BY 'ossim_pass' WITH GRANT OPTION;

mysql> GRANT ALL PRIVILEGES ON ossim.* TO 'ossim'@'localhost'
        ->     IDENTIFIED BY 'ossim_pass' WITH GRANT OPTION;

mysql> GRANT ALL PRIVILEGES ON ossim_acl.* TO 'ossim'@'localhost'
        ->     IDENTIFIED BY 'ossim_pass' WITH GRANT OPTION;

mysql> FLUSH PRIVILEGES;
```

Ici, nous offrons tous les privilèges à l'utilisateur ossim sur les bases de données nécessaires.

Pour une configuration graphique, il suffira d'installer le paquetage suivant sur le serveur :

```
# apt-get install phpmyadmin
```

A.3.1 Configuration du serveur Web

Celle-ci se situe dans */etc/apache/httpd.conf*. Il est en premier lieu indispensable de charger le module de l'interpréteur PHP afin que le site d'Ossim puisse fonctionner. Ceci s'opère via la commande suivante :

```
# apache-modconf apache active mod_php4
```

La configuration d'Ossim se situe dans */etc/apache/httpd.conf/conf.d/* qui est importé dans le fichier de configuration principal d'Apache. Le fichier de configuration d'Apache pour Acid[2] n'est par contre pas directement fourni dans ce répertoire. Il est donc nécessaire de le copier afin qu'Apache soit capable de servir les pages web d'Acid :

```
# cp /etc/acidlab/apache.conf /etc/apache/conf.d/acid.conf
```

Il est ensuite nécessaire de modifier la configuration par défaut d'Apache pour Ossim, afin que celui-ci soit capable de suivre les liens symboliques. En effet, un lien symbolique est utilisé pour l'affichage des vulnérabilités découvertes à l'aide de Nessus. Il faut donc ajout l'option suivante dans le fichier */etc/apache/conf.d/ossim.conf* (à la suite des options déjà définies) :

```
Options FollowSymLinks
```

[2] Viewer pour les alertes Snort et Ossim

Annexe B

Ajout et configuration d'une sonde Snort

Les manipulations décrites ci-dessous requièrent un serveur Ossim opérationnel (installation décrite dans la section A).

B.1 Installation

B.1.1 Snort

Son installation sur une distribution Debian (via apt-get) implique l'ajout d'une source de download afin de récupérer l'application Snort directement patchée. Il est nécessaire d'ajouter la ligne suivant dans */etc/apt/source.list* afin que le gestionnaire de paquets de Debian (aptitude) soit capable de récupérer les paquetages d'Ossim :

```
deb http://www.ossim.net/download/ debian/
```

Il est ensuite nécessaire de créer un fichier de préférences */etc/apt/preferences* afin que Debian aille en premier lieu rechercher les paquetages disponibles sur Ossim plutôt que ceux disponibles sur d'autres serveurs. Nous serons ainsi certain d'obtenir la version patchée de Snort :

```
Package: *
Pin: release o=ossim
Pin-Priority: 995
```

Après la mise à jour des paquetages disponibles, il est possible de downloader Snort :

```
# apt-get update
# apt-get install snort-mysql
```

Pour plus d'informations à ce sujet, consultez : http ://www.ossim.net/docs/INSTALL.Debian.html#d0e783

B.1.2 Ossim-agent

Maintenant qu'*aptitude* est capable de récupérer des paquets sur les serveurs d'OSSIM, il suffit de taper la commande suivante afin d'installer Ossim-agent :

```
# apt-get install ossim-agent
```

L'installeur de paquets de Debian va maintenant vous questionner afin de créer une configuration de base. Reportez-vous à la section suivante (section B.2.2) pour plus de détails.

B.2 Configuration

B.2.1 Snort

Comme mentionné plus haut, le logiciel de détection d'intrusion Snort utilise son plugin de sortie Mysql afin de transmettre directement ses alertes en direction d'une des bases de données d'Ossim-server (Snort BD). Celui-ci disposera alors de tous les détails de chaque alertes levée par la sonde.
Le second plugin de sortie utilisé est "logfile=fast.log". Il s'agît d'un plugin développé par OSSIM très proche de "alert_full"[1] directement intégré à Snort. Fast.log place simplement les différentes alertes dans un fichier de logs qui sera ensuite lu par l'agent Ossim qui s'occupera de transmettre ces informations vers le serveur Ossim (permettant ainsi l'analyse temps réel).

La configuration des plugins de sorties de Snort ressemble donc à ceci (fichier : /etc/snort/snort.conf) :

```
output database: alert, mysql, user=snort password=myPass
dbname=snort host=sgbdServerIP sensor_name=MonSensor logfile=fast.log
```

Les champs *user*, *password*, *dbname* et *host* correspondent aux informations relative à la base de donnée Snort distante. Il est donc nécessaire de créer un nouvel utilisateur sur celle-ci afin que la sonde puisse s'y connecter à l'aide du mot de passe indiqué. L'ajout d'un utilisateur sur la base de donnée Snort sera détaillé ci-dessous (section B.3). De plus, il est indispensable de retirer le script de démarrage automatique de Snort afin que la sonde puisse être directement activée par le Serveur (via Ossim-agent). Ceci s'opère à l'aide de la commande suivante :

```
# update-rc.d -f snort remove
```

Mise à jour des règles sur le serveur Ossim

La mise à jour des règles Snort permet l'utilisation de celles-ci dans les scénarii de corrélation. En effet, le menu de "Correlation - Directives" du framework permet l'utilisation des alertes Snort dans

[1] *fast.log* ajout simplement deux paramètres supplémentaire à *alert_full* afin d'augmenter les performances du serveur. Info sur : https ://sourceforge.net/forum/message.php ?msg_id=2627915

la définition de scénarii de corrélation. L'utilisation du *SID* des alarmes permet de les référencer dans les règles de corrélation. Celui-ci doit donc être unique pour chaque plugin. Il convient donc lors de création de règles Snort de ne pas les dupliquer (un contrôle est quand même effectué par le script */usr/share/ossim/scripts/create_sidmap.pl* de mise à jour des règles dans la base de donnée du serveur Ossim). La mise à jour des règles sur le serveur Ossim requiert le client *Mysql* puisque le script procède au contrôle de la duplication des règles et fournit les commandes SQL à entrer dans le client. Ce script doit évidemment être exécuté sur l'agent hébergeant la sonde Snort. Installation du client *mysql* :

```
# apt-get install mysql-client
```

Lancement du script de mise à jour à l'aide d'un simple pipe vers le client mysql configuré pour un connexion vers la base de donnée du serveur Ossim (à écrire sur une ligne) :

```
# /usr/share/ossim/scripts/create_sidmap.pl /etc/snort/rules | mysql
     --host=10.192.73.171 -u ossim ossim -p
```

Le user *ossim* utilisé pour la connexion du client mysql doit bénéficier des droits suffisants pour la commande Sql *INSERT*.

B.2.2 Ossim-agent

Sa configuration s'effecture directement à l'aide du menu de configuration des paquets Debian et peut être reconfiguré à volonté à l'aide de la commande :

```
# dpkg-reconfigure ossim-agent
```

La configuration requiert (dans l'ordre d'apparition) :

1. L'adresse IP de l'agent
2. L'interface réseau à utiliser (pour la communication avec le serveur)
3. L'adresse IP du serveur OSSIM
4. Les plugins qui vont être connecté à cet agent. Dans notre cas, uniquement Snort (illustré à la figure B.1)

L'exécutable de l'agent Ossim est ensuite directement lié dans les runlevels appropriés afin qu'un démarrage automatique s'effectue à chaque boot de la machine.

B.3 Configuration d'Ossim-server pour une nouvelle sonde Snort

La procédure d'ajout d'un nouveau sensor doit être effectué via l'interface Web du serveur Ossim. Celle-ci est décrite dans le manuel disponnible à l'URL suivante : http ://www.ossim.net/docs/User-Manual.pdf
Lors de l'ajout d'un sensor local (agent placé sur le serveur Ossim), il faudra bien prendre garde de spécifier correctement l'adresse du serveur Ossim dans la configuration de l'agent (/etc/ossim/agent/config.xml). En effet, pour une bonne génération des liens HTML du framework, il sera nécessaire de ne **pas** spécifier l'adresse de loopback du serveur comme *serverip*.

FIG. B.1 – Interface de configuration (dpkg) des plugins à utiliser sur un agent Ossim

```
<serverip>Adr_ip_non_loopback</serverip>
```

L'ajout d'une nouvelle sonde Snort implique l'ajout de droit d'accès dans la base de donnée snort afin que cette nouvelle sonde (=nouvel utilisateur) soit capable d'y déposer directement ses alertes (comme illustré sur la figure 1.1). Il est donc nécessaire de se trouver sur la machine serveur afin d'y entrer les requêtes SQL pour l'ajout d'un utilisateur.
Lancement du client Mysql local et utilisation de la base de donnée snort :

```
# mysql -u root
mysql> use snort;
```

Requêtes SQL à entrer pour l'ajout d'un nouvel utilisateur et pour la mise en place de son mot de passe :

```
mysql> GRANT ALL ON snort.* TO snort@sensorIP;
mysql> UPDATE user SET Password = PASSWORD('passwordDeSnort@sensorIP')
        -> WHERE Host = 'sensorIP' AND User = 'snort';
```

L'utilisateur *snort* provenant de *sensorIP* a maintenant un accès total à la base de donnée *snort* du serveur. Son mot de passe (à indiquer dans la configuration de Snort, section B.2.1) est maintenant *passwordDeSnort*.

B.4 Test de fonctionnement

Maintenant que l'agent Ossim est configuré et que la base de donnée snort est accessible depuis celui-ci, nous pouvons tester le fonctionnement de l'agent (le serveur doit être en marche).
Il est maintenant possible de démarrer l'agent Ossim (en mode de debug afin de contrôler son bon fonctionnement) à l'aide de la commande suivante :

```
# ossim-agent -v -f
```

Les logs suivants doivent alors apparaître sur la console :

```
  (->)  pyossim.Agent (2005-04-27 11:38:04):     Waiting for server...
  (->)  pyossim.Agent (2005-04-27 11:38:04):     Waiting for server...
  (<-)  pyossim.Agent (2005-04-27 11:38:04):     Server connected

  (<-)  pyossim.Agent (2005-04-27 11:38:04):     Server connected

  (=>)  pyossim.Agent (2005-04-27 11:38:04):     Apending plugins...
  (--)  pyossim.Watchdog (2005-04-27 11:38:04):  monitor started
 Starting Network Intrusion Detection System: snort(eth0)No /etc/snort/snort.eth0.conf,
  (--)  pyossim.ParserSnort (2005-04-27 11:38:04):       plugin started (fast)...
 .
  (=>)  pyossim.Agent (2005-04-27 11:38:05):     plugin-start plugin_id="1001"
```

Vous pouvez maintenant exécuter l'agent Ossim en tâche de fond :

```
# ossim-agent -d
```

Votre nouvelle sonde Snort est prête à l'emploi !

B.4.1 Erreur de démarrage de l'agent Ossim

Il se peut que l'erreur suivante apparaisse à la console :

```
[Errno 2] No such file or directory: '/var/log/snort/alert'
```

Celle-ci signifie que le fichier de log de Snort que récupère l'agent (pour l'envoi temps réel des informations au serveur), n'a pas été trouvé. Il est alors nécessaire de modifier la configuration du plugin Snort afin d'y indiquer le bon chemin (le bon fichier de log). Celle-ci est décrite dans le fichier XML /etc/ossim/agent/plugins/snort.xml. Il est indispensable de modifier la balise *location* afin d'y indiquer l'emplacement réel du fichier de log de Snort (généré par le plugin *fast.log* de Snort), comme illustré ci-dessous :

```
<plugin id="1001" process="snort" type="detector" start="yes" enable="yes">
      .......... balises de configuration ...........
    <location>/var/log/snort/fast.log</location>
</plugin>
```

Annexe C

Ajout et configuration de Ntop

Les manipulations décrites ci-dessous requièrent un serveur Ossim opérationnel, ainsi qu'un agent Ossim installé sur la machine qui hébergera Ntop (l'installation de l'agent est décrite dans la section B.1.2 puisqu'il s'agît, dans notre cas, du même agent que pour la sonde Snort).

C.1 Installation

C.1.1 Ntop

Installation de Ntop et des librairies nécessaires :

```
# apt-get install librrd0 ntop
```

Installation du paquetage *ossim-utils* fournissant le script d'analyse des informations de Ntop (/usr/share/ossim/scripts/rrd_plugin.pl) ainsi que le fichier de configuration nécessaire pour l'interrogation de la base de donnée Ossim par rrd_plugin.pl (permettant la récupération de la configuration des seuils défini par l'administrateur réseau sur le framework) :

```
# apt-get install ossim-utils
```

Pour les détails de la configuration de ce paquetage consultez la section C.2.2.

Installation des outils utilisés par le script *rrd_plugin.pl* pour la récupération des informations dans la base de donnée RRD :

```
# apt-get install rrdtool librrd0 librrd0-dev librrdp-perl librrds-perl
```

C.2 Configuration

C.2.1 Ntop

Configuration du mot de passe *admin* pour Ntop :

```
# ntop -u ntop
>> Please enter the password for the admin user:
#
```

Comme mentionné dans la section 1.3.1, Ntop dispose d'une interface Web pour le monitoring ainsi que pour sa configuration. Le serveur Web s'installe par défaut sur le port 3000. Il est maintenant nécessaire de configurer le format des logs de sortie (plugin RRD) afin que le script *rrd_plugin.pl* soit capable de les récupérer via l'outil RRDtool. Pour ce faire il suffit de se rendre à l'URL suivante : *http ://yourhost :3000/* et d'activer le rrdPlugin dans Admin - plugins. En cliquant sur *rrdPlugin* le menu de configuration de celui-ci devient éditable. Vous pouvez maintenant cliquer sur *Host* dans le menu *Data to Dump* puis entrer votre masque de sous-réseau dans *Hosts Filter*. Il est encore nécessaire de contrôler que le *RRD Files Path* soit le même que celui fournit dans le framework de configuration (Configuration - Main - rrdpath_ntop). En effet le script Perl *rrd_plugin.pl* récupère la configuration des seuils sur le framework, ainsi que les chemins d'accès aux fichiers de logs.

Il est ensuite nécessaire de modifier le fichier */etc/default/ntop* afin d'y mettre *–no-mac* comme *GETOPT* :

```
GETOPT="--no-mac"
```

Le script d'analye et de comparaison des seuils (rrd_plugin) sera maintenant capable de récupérer les données de la base de donnée tourniquet afin de les analyser.

C.2.2 L'agent Ossim

Ossim-utils

Ce paquetage contenant le script *rrd_plugin.pl* met à disposition le module perl */usr/lib/perl5/config.pm* permettant de récupérer la configuration établie dans le framework (configuration établie dans Configuration - main). Il est nécessaire de configurer correctement ce paquetage afin que ce module soit capable de se connecter à la base de donnée distante. Il faudra lui indiquer un utilisateur Mysql, capable de se connecter depuis l'agent Ossim. Dans notre cas, nous indiquerons l'utilisateur *snort* précédemment configuré (cf. section B.3). Éditée à la main (/etc/ossim/framework/ossim.conf) cette configuration ressemble à ceci :

```
################
# OSSIM db configuration
################
```

```
ossim_type=mysql
ossim_base=ossim
ossim_user=snort
ossim_pass=E1ephant
ossim_host=10.192.72.172
ossim_port=3306
```

Celle-ci peut aussi être aisément éditée à l'aide de dpkg via la commande suivante :

```
# dpkg-reconfigure ossim-utils
```

Ossim-agent

Le script *rrd_plugin.pl* récupère de lui même (sans appel à config.pm) les informations relatives à la configuration des seuils dans la base de donnée Ossim distante. Il est donc nécessaire de lui indiquer correctement les mêmes informations que ci-dessus. Celles-ci sont visibles comme "variables globales" dans la configuration de l'agent (/etc/ossim/agent/config.xml) et sont ensuite utilisées par certains plugins (dont le plugin RRD). La définition de l'ENTITY suivant avec les bon paramètres de connexion est nécessaire :

```
<!ENTITY ossim_db "mysql:10.192.72.172:ossim:snort:E1ephant" >
```

Il faut ensuite reconfiguer l'agent Ossim afin de préciser que Ntop est activé sur cet agent. Étant donné que *rrd_plugin.pl* est utilisé pour l'analyse des données, il est nécessaire d'indiquer que le plugin RRD est aussi en fonction. La figure C.1 illustre cette nouvelle configuration exécutée à l'aide de la commande suivante :

```
# dpkg-reconfigure ossim-agent
```

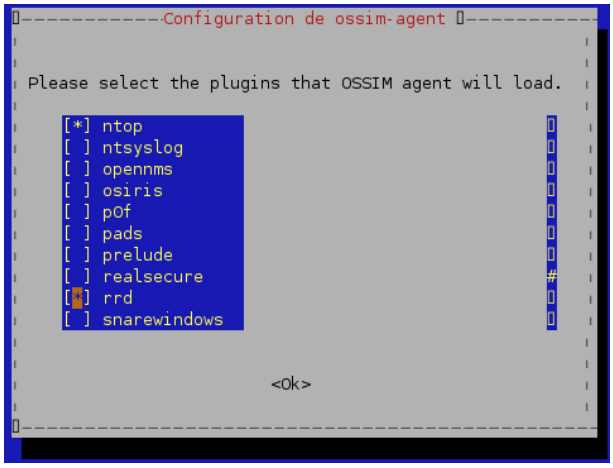

FIG. C.1 – Interface de configuration (dpkg) des plugins à activer sur l'agent Ossim

C.3 Configuration d'Ossim-server pour une nouvelle sonde Ntop

Si l'ajout de cette sonde a été effectuée sur un sensor existant (comme dans notre cas), il ne sera pas nécessaire d'enregistrer un nouvel agent sur l'interface Web du serveur. L'agent existant transmettra de lui même l'existance d'une nouvelle sonde au serveur. Dans le cas contraire (mise en place de cette sonde sur un nouvel agent), il sera nécessaire de procéder à l'enregistrement du nouvel agent, comme expliqué dans le manuel disponnible à l'URL suivante : http ://www.ossim.net/docs/User-Manual.pdf

Il est aussi indispensable d'ajouter les droits suffisants à l'utilisateur utilisé lors de la connexion à la base de donnée, afin que celui-ci soit capable de récupérer la configuration sur le serveur :

```
# mysql -u root
mysql> use ossim;
```

Requêtes SQL à entrer pour l'ajout d'un nouvel utilisateur et pour la mise en place de son mot de passe :

```
mysql> GRANT ALL ON ossim.* TO snort@sensorIP;
mysql> UPDATE user SET Password = PASSWORD('passwordDeSnort@sensorIP')
        -> WHERE Host = 'sensorIP' AND User = 'snort';
```

Ceci peut aussi être effectué via phpMyAdmin.

Annexe D

Ajout et configuration de P0f

Les manipulations décrites ci-dessous requièrent un serveur Ossim opérationnel (installation décrite dans la section A), ainsi qu'un agent Ossim installé sur la machine qui hébergera P0f (l'installation de l'agent est décrite dans la section B.1.2 puisqu'il s'agît, dans notre cas, du même agent que pour la sonde Snort).

D.1 Installation

Son installation nécessite uniquement l'installation du paquetage P0f :

```
# apt-get install p0f
```

D.2 Configuration

D.2.1 P0f

P0f ne nécessite aucune configuration supplémentaire puisque le chemin de son fichier de log lui est fourni par l'agent Ossim lors de son exécution.

D.2.2 L'agent Ossim

Il est nécessaire de reconfiguer l'agent Ossim afin de préciser que P0f est activé sur cet agent. Celle-ci s'opère à l'aide de la commande suivante :

```
# dpkg-reconfigure ossim-agent
```

Annexe E

Ajout et configuration de TCPTrack

Les manipulations décrites ci-dessous requièrent un serveur Ossim opérationnel (installation décrite dans la section A), ainsi qu'un agent Ossim installé sur la machine qui hébergera TCPTrack (l'installation de l'agent est décrite dans la section B.1.2 puisqu'il s'agît, dans notre cas, du même agent que pour la sonde Snort).

E.1 Installation

Celle-ci est vraiment très simple puisqu'il suffira de récupérer le paquetage Debian de TCPTrack sur le serveur Web d'Ossim[1] via la commande suivante :

```
# apt-get install tcptrack
```

E.2 Configuration

Aucune configuration spécifique n'est nécessaire pour TCPTrack puisque c'est le serveur Ossim qui s'occupera d'interoger TCPTrack. Il sera par contre indispensable d'indiquer à l'agent qu'une nouvelle sonde lui a été ajoutée. Ceci s'opèrera via le menu de configuration offert par la commande suivante :

```
# dpkg-reconfigure ossim-agent
```

[1] Il est donc indispensable d'avoir configuré *aptitude* afin qu'il récupère directement les paquetages chez Ossim (cf. Section B.1.1). En effet, la version de TCPTrack utilisée dans l'architecture d'Ossim est une version modifiée de la version originale.

Annexe F

Ajout et configuration d'une sonde HIDS Syslog

Les manipulations décrites ci-dessous requièrent un serveur Ossim opérationnel (installation décrite dans la section A), ainsi qu'un agent Ossim installé sur la machine qui hébergera cette sonde HIDS Syslog (l'installation de l'agent est décrite dans la section B.1.2 puisqu'il s'agît, dans notre cas, du même agent que pour la sonde Snort).

F.1 Installation

Aucune installation n'est requise puisque le parser de fichiers de log Syslog et directement présent sur tous les agents (fichier */usr/share/ossim-agent/pyossim/ParserSyslog.py*).

F.2 Configuration

Il suffira d'activer le plugin voulu afin que celui-ci soit automatiquement exécuté par l'agent. Pour ce faire, il vous suffira de l'activer à l'aide du menu de configuration offert par la commande suivante :

```
# dpkg-reconfigure ossim-agent
```

La configuration du fichier à contrôler peut être directement faite en modifiant la balise *location* du fichier */etc/ossim/agent/plugins/syslog.xml*. Par défaut celui-ci est */var/log/auth.log*.
Sous Debian il sera en plus nécessaire de modifier (dans le même fichier de configuration) le nom du daemon de logging puisque celui-ci se nomme *sysklogd* et non pas *syslog*.

L'ajout de nouvelles règles nécessitera malheureusement la modification du code du parseur */usr/share/ossim-agent/pyossim/ParserSyslog.py* puisque celles-ci s'y trouvent directement intégrées. Aucun fichier de configuration des règles n'est pour le moment disponnible.

Annexe G

Ajout et configuration de PADS

Les manipulations décrites ci-dessous requièrent un serveur Ossim opérationnel (installation décrite dans la section A), ainsi qu'un agent Ossim installé sur la machine qui hébergera PADS (l'installation de l'agent est décrite dans la section B.1.2 puisqu'il s'agît, dans notre cas, du même agent que pour la sonde Snort).

G.1 Installation

Celle-ci est vraiment très simple puisqu'il suffira de récupérer le paquetage Debian de PADS, via la commande suivante :

```
# apt-get install pads
```

G.2 Configuration

Aucune configuration spécifique n'est nécessaire pour PADS puisque le chemin de ses logs de sortie est directement fourni par l'agent Ossim (paramètre lors de son exécution). Il sera par contre indispensable d'indiquer à l'agent qu'une nouvelle sonde lui a été ajoutée. Ceci s'opèrera via le menu de configuration offert par la commande suivante :

```
# dpkg-reconfigure ossim-agent
```