PPE : Installation et configuration de Wazuh.

Documentation n°1 : Installation d'un outil de visualisation des alertes de sécurités d'un parc de serveurs.

Auteur : Guillaume Orlando

Date : 19/03/2017

I. Introduction.

Puisque en stage, j'ai utilisé une solution propriétaire trop lourde pour être mise en place sur le labo du BTS (Le SIEM "Prelude"), j'ai décidé d'implémenter une solution équivalente open-source, basée sur la suite ELK (ElasticSearch, Logstash et Kibana).
Les fonctionnalités finales ne sont donc pas aussi abouties ou complètes que dans le cas d'une solution professionnelle, mais il est tout de même possible d'en tirer une plateforme de centralisation d'éléments de sécurités. Le but final de l'architecture sera de superviser l'intégrité de certains fichiers de configurations sensibles, que l'utilisateur pourra définir manuellement. Il sera aussi nécessaire de paramétrer une fonctionnalitée de surveillance d'appels systèmes jugés sensibles par l'administrateur de la plateforme.

L'outil de surveillance d'intégrité de fichier sera Ossec, et celui de surveillance d'appel système Auditd.

Pour mettre en place cette architecture, j'avais comme idée de créer mon propre plugin et mes propres filtres Kibana pour Ossec et Auditd. Après avoir écrit plusieurs des filtres, je me suis rendu compte qu'une solution similaire à ce que je cherche avait déjà été imaginé par la communauté du la solution ELK. Cette solution, relativement nouvelle, et uniquement supporté par la communauté se nomme "Wazuh". La totalitée des configurations sont disponibles sous Github à cette adresse : https://github.com/wazuh/wazuh

Cette documentation décrit donc l'installation et la prise en main de cette solution, sous Ubuntu-Server 16.04.

II. Installation du manager Wazuh.

Afin d'installer les paquets nécessaires à cette architecture, il sera nécessaire d'installer certains outils (notamment des outils de compilations, des langages nécessaires à l'interprétation des événements, etc …) :

```
apt-get install gcc make git libc6-dev
apt-get install libssl-dev
curl -sL https://deb.nodesource.com/setup_6.x | sudo -E bash -
apt-get install -y nodejs
```

Nous pouvons désormais télécharger l'archive contenant le manager Wazuh, et la décompresser :

```
curl -Ls https://github.com/wazuh/wazuh/archive/v2.0.1.tar.gz | tar zx
```

Un script bash d'installation automatique est compris dans l'archive. Il suffira de lancer le script en question :

```
cd wazuh-*
./install.sh
```

Le script nous demandera le type d'installation que nous souhaitons effectuer. L'installation "local" ne permet que de monitorer le serveur sur lequel la solution Wazuh est installée, alors qu'une installation de type "server" va permettre de créer une architecture client serveur, où plusieurs machines vont êtres capables d'envoyer des alertes au serveur. Nous choisirons l'installation de type "server" :

```
What kind of installation do you want (server, agent, local, hybrid or help)? server
```

Nous pourrions démarrer le service Wazuh-manager avec la commande suivante :

```
/var/ossec/bin/ossec-control start
```

III. Installation de l'API Wazuh.

Pour que Wazuh puisse fonctionner en tant que plugin pour Kibana, celui-ci nécessite une API spécifique. Cette API nécessite quelques dépendances :

```
curl -sL https://deb.nodesource.com/setup_6.x | sudo -E bash -
apt-get install -y nodejs
```

Nous pouvons désormais installer et exécuter le script d'installation de l'API Wazuh :

```
curl -s -o install_api.sh
https://raw.githubusercontent.com/wazuh/wazuh-api/v2.0.1/install_api.sh && bash
./install_api.sh download
```

Il sera aussi nécessaire d'utiliser une version de python supérieur ou égale à 2.7.
Celle-ci est normalement nativement disponible sur la plupart des distributions Linux.

IV. Installation de Filebeat.

Filebeat permet aux agents de forwarder de manière sécurisée les alertes aux stack Elasticsearch, Logstash, Kibana. Nous allons installer Filebeat depuis les dépôts ELK. Nous commencerons par ajouter la clés GPG du dépôt, puis par ajouter les sources et de les mettres à jour :

```
curl -s https://artifacts.elastic.co/GPG-KEY-elasticsearch | apt-key add -
echo "deb https://artifacts.elastic.co/packages/5.x/apt stable main" | tee
/etc/apt/sources.list.d/elastic-5.x.list
apt-get update
```

Nous devrions être désormais capables de trouver le paquet Filebeat via le gestionnaire de paquet :

```
apt-get install filebeat
```

Il sera nécessaire de télécharger le fichier de configuration officiel de filebeat :

```
curl -so /etc/filebeat/filebeat.yml
https://raw.githubusercontent.com/wazuh/wazuh/2.0/extensions/filebeat/filebeat.yml
```

Il faudra éditer celui-ci pour l'adapter à notre configuration réseau (/etc/filebeat/filebeat.yml)

```
output:
  logstash:
    hosts: ["ELASTIC_SERVER_IP:5000"]
```

Il sera ensuite nécessaire d'activer le service Filebeat, et de le démarrer :

```
systemctl daemon-reload
systemctl enable filebeat.service
systemctl start filebeat.service
```

V. Prérequis pour stack Logstash, Elasticsearch et Kibana.

Le stack ELK nécessite l'installation de Java 8, via les dépôts :

```
add-apt-repository ppa:webupd8team/java
apt-get update
apt-get install oracle-java8-installer
```

Nous installerons ensuite les clés GPG nécessaire à l'installation du stack ELK :

```
apt-get install curl apt-transport-https
curl -s https://artifacts.elastic.co/GPG-KEY-elasticsearch | apt-key add -
echo "deb https://artifacts.elastic.co/packages/5.x/apt stable main" | tee
/etc/apt/sources.list.d/elastic-5.x.list
apt-get update
```

VI. Installation d'ElasticSearch.

Maintenant que nous disposons des paquets ELK à portée de main, nous allons commencer par installer ElasticSearch :

```
apt-get install elasticsearch
```

Nous allons ensuite activer et démarrer le service :

```
systemctl daemon-reload
systemctl enable elasticsearch.service
systemctl start elasticsearch.service
```

Nous allons ensuite télécharger le fichier de configuration Wazuh relatif à Wazuh :

```
curl
https://raw.githubusercontent.com/wazuh/wazuh-kibana-app/master/server/startup/
integration_files/template_file.json | curl -XPUT
'http://localhost:9200/_template/wazuh' -H 'Content-Type: application/json' -d @-
```

Enfin, nous ajoutons un sample d'alerte ElasticSearch :

```
curl
https://raw.githubusercontent.com/wazuh/wazuh-kibana-app/master/server/startup/
integration_files/alert_sample.json | curl -XPUT
"http://localhost:9200/wazuh-alerts-"`date +%Y.%m.%d`"/wazuh/sample" -H
'Content-Type: application/json' -d @-
```

VII. Installation de Logstash.

Tout comme ElasticSearch, les paquets Logstash sont disponibles dans les dépôts :

```
apt-get install logstash
```

Nous pouvons donc télécharger et placer les fichiers de configurations relatifs à Wazuh :

```
curl -so /etc/logstash/conf.d/01-wazuh.conf
https://raw.githubusercontent.com/wazuh/wazuh/2.0/extensions/logstash/01-wazuh
.conf

curl -so /etc/logstash/wazuh-elastic5-template.json
https://raw.githubusercontent.com/wazuh/wazuh/2.0/extensions/elasticsearch/waz
uh-elastic5-template.json
```

Nous activons et démarrons le service :

```
systemctl daemon-reload
systemctl enable logstash.service
systemctl start logstash.service
```

VIII. Installation de Kibana.

Enfin, il ne reste qu'à installer Kibana, l'IHM web qui va porter l'instance de Wazuh. Si les éléments du stash ELK ne vous sont pas familiers, référez à mes précédentes documentations concernant l'installation de cette même architecture pour Pfsense.

```
apt-get install kibana
```

Il ne restera qu'à installer le plugin Wazuh pour Kibana :

```
/usr/share/kibana/bin/kibana-plugin install
https://packages.wazuh.com/wazuhapp/wazuhapp.zip
```

Enfin, s'il est nécessaire que l'instance de Kibana écoute sur toutes les interfaces de la machine, il faudra ajouter la ligne suivante dans le fichier de configuration situé sous */etc/kibana/kibana.yml* :

```
server.host: "0.0.0.0"
```

Il ne restera qu'à activer et démarrer le service Kibana :

```
systemctl daemon-reload
systemctl enable kibana.service
systemctl start kibana.service
```

L'interface Web kibana devrait désormais être disponible à l'adresse du serveur, sur le port 5601.
Cependant, il est nécessaire de connecter l'API Wazuh à Kibana pour disposer des fonctionnalitée de Wazuh.

IX. Connexion à l'API Wazuh.

Les options graphiques de connexion à l'API Wazuh sont disponible dans l'onglet de gauche suivant :

Avant de renseigner les options de connexion à l'API, il sera nécessaire de créer des identifiants sécurisé, directement sur le serveur :

```
cd /var/ossec/api/configuration/auth
sudo node htpasswd -c user myUserName
```

Il faudra redémarrer le service pour que ce nouvelle utilisateur soit effectif :

```
systemctl restart wazuh-api
service wazuh-api restart
```

Nous allons désormais pouvoir initier la connexion entre Kibana et l'APi Wazuh, et indiquant les identifiants de connexion créés précédemments, et en renseignant l'URL de connexion au service web du serveur en question, et le port de connexion au service (à savoir le port 55000) :

Nous avons désormais accès aux Dashboards Wazuh. Ceux-ci sont assez intuitif, et il est possible de prendre une pause pour commencer à les manipuler, et à jouer avec les fonctionnalitées offertes. Nous allons tout de même finaliser les configurations des outils Ossec et Auditd, pour que ceux-ci remontent les informations qui nous intéressent.

X. Configuration de la surveillance d'intégrité de fichiers.

Maintenant que la structure de visualisation des logs est en place, commençons à customiser la solution, en l'utilisant à des fins biens précises.
Dans un premier temps, nous allons surveiller l'intégrité des fichiers Linux.

Le module Ossec sera utilisé tout le long de ce chapitre. Il sera nécessaire d'éditer le fichier situé par défaut sous "*/var/ossec/etc/ossec.conf*" dans la rubrique "*syscheck*". Syscheck est le nom du processus qui va régulièrement effectuer des scans sur les fichiers indiqués comme "sensibles". Ce même processus va également construire la base de données d'intégrité de fichiers, afin d'y effectuer des comparaisons et de détecter de potentielles modifications d'intégrités.
Les options pour surveiller

```
<syscheck>
        <disabled>no</disabled>
        <frequency>120</frequency>
        <scan_on_start>yes</scan_on_start>
        <alert_new_files>yes</alert_new_files>
        <auto_ignore>no</auto_ignore>
        <directories check_all="yes">/etc/exemple.txt</directories>
        <directories check_all="yes">/etc/dir/</directories>
</syscheck>
```

Détaillons ces lignes :
<*disabled*> : Précise si le module est actif
<*frequency*> : Indique la fréquence des scans d'intégrités de fichiers
<*scan_on_start*> : Indique si un scan est effectuée à chaques démarrage
<*alert_new_file*> : Indique si le dossier/fichier doit être surveillé de manière récursive
<*auto-ignore*> : Indique si un fichier modifié trop souvent doit continuer de générer des alertes de sécurités.
<*directories check_all="yes"*> : Précède un fichier ou dossier à monitorer.

Il sera nécessaire de redémarrer le service Ossec avec la commande :

```
/var/ossec/bin/ossec-control restart
```

Désormais, toutes les alertes relatives à ces fichiers sont enregistrés sous */var/ossec/log/alerts/alerts.log*. Elles sont également visibles de façon graphique dans l'interface Kibana, au sein de l'onglet "File Integrity" :

Un ensemble de graphiques s'offrent ensuite à nous :

L'ensemble des éléments de sécurités relatifs à la modification d'intégrité de fichier de la machine sont désormais tous centralisés en un point, sur l'interface web.

XI. Ajout d'un agent Windows Server.

Nous allons désormais ajouter un agent Windows Server, dans le but de monitorer les tentatives de connexions sur la machine, les logs de sécurités relatifs aux différents services de la machine, l'état de la machine, etc …

Avant de commencer, il faudra s'assurer que les deux machines peuvent correctement communiquer. Dans le cas présent, elles sont dans le même sous-réseau.

Il faudra, sur le manager Wazuh, créer le profil de notre futur agent. Pour ce faire, l'outil "*manage_agent*" va nous assister tout le long de cette tâche.

```
cd /var/ossec/bin
./manage_agent
```

Pour créer le profil de l'agent, nous choisissons l'option "*Add an agent*" :

```
* Wazuh v3.2.1 Agent manager.         *
* The following options are available: *
****************************************
   (A)dd an agent (A).
   (E)xtract key for an agent (E).
   (L)ist already added agents (L).
   (R)emove an agent (R).
   (Q)uit.
```

```
Choose your action: A,E,L,R or Q: A
```

Il suffira de renseigner un nom pour le serveur à superviser, une adresse IP, et un identifiant unique allant de 001 à 255 :

```
* A name for the new agent: SCA1-Windows_Server
  * The IP Address of the new agent: 172.16.55.1
  * An ID for the new agent[001]: 001
```

Côté agent Windows, il faudra télécharger un éxécutable Wazuh disponible ici : https://documentation.wazuh.com/3.x/installation-guide/packages-list/index.html

Il suffira de renseigner l'adresse du Manager et la clé correspondant à l'agent :

Pour obtenir la clé, il faudra lancer de nouveau l'exécutable "*manage_agent*"