Documentation de stage

Documentation : Installation et configuration de l'HIDS Ossec

Auteur : Guillaume Orlando
Date : 10/01/2018

OSSEC est un HIDS compatible Unix et Windows, permettant notamment d'effectuer un contrôle d'intégrités sur des fichiers et des dossiers. Nous allons essayer d'implémenter uniquement cette fonctionnalité de façon à générer des logs dans le cas où certains fichiers sensibles seraient modifiés à notre insus.
OSSEC fonctionne en architecture Client - Serveur. Le Serveur sera ici une machine RedHat 6. Les clients seront émulés par un parc de machines virtuelles RedHat 6 et Windows.

I. Architecture OSSEC local

Ce premier POC comprendra une unique machine, sous RedHat 6, faisant office de serveur et de client. Ce mode d'implémentation est qualifiée d'installation "local" par OSSEC.
Dans un premier temps, les paquets d'installation unix sont disponibles via l'adresse : *http://www.ossec.net/files/ossec-hids-2.8.1.tar.gz*.

```
wget -U ossec http://www.ossec.net/files/ossec-hids-2.8.1.tar.gz
tar xf ossec-hids-2.8.1.tar.gz
./install.sh
```

L'assistant d'installation nous permettra de personnaliser l'installation d'OSSEC en fonction des besoins. Les options peuvent êtres laissées par défaut, à l'exception du type d'installation :

```
1- What kind of installation do you want ? local
```

Une fois l'installation correctement effectuée, nous pouvons entamer la configuration des différents fichiers. L'ensemble des fichiers de configurations sont accessibles sous le répertoire /var/ossec.
Le premier fichier à éditer sera le fichier de configuration général */var/ossec/etc/ossec.conf.*
Celui-ci est découpé en plusieurs sections. La première est relative à l'envoie automatique d'e-mail en cas d'alertes sévères. Dans le cas de cet environnement de test, nous n'utilisons pas cette fonctionnalité :

```
<global>
        <email_notification>no</email_notification>
</global>
```

La deuxième section correspond aux règles d'alertes effectives. Nous ne les modifierons pas pour l'instant.
La section suivante nous intéresse tout particulièrement, puisqu'il s'agit des paramètres de l'outil "syscheck", le vérificateur d'intégrité de l'outil OSSEC. Les paramètres mis en place sont les suivants :

```
<syscheck>
        <frequency>120</frequency>
        <auto_ignore>no</auto_ignore>
        <alert_new_files>yes</alert_new_files>
        <scan_on_start>yes</scan_on_start>
        <directories                                    report_changes="yes"
check_all="yes">/etc/DirectoryTest/test2</directories>
        <directories          realtime="yes"          report_changes="yes"
check_all="yes">/etc/DirectoryTest</directories>
 </syscheck>
```

- L'argument <frequency> est relatif à la durée d'actualisation de l'agent.
- L'argument <auto_ignore> indique que même si le fichier est modifié un grand nombre de fois sur une courte période temporelle, celui-ci sera quand même traité, ce qui n'est pas le cas par défaut.
- L'argument <alert_new_files> indique que la création de nouveaux fichiers au sein du répertoire donné doit être rapportée par les logs.
- L'argument <scan_on_start> spécifie qu'un scan syscheck sera fait à tous les démarrages du service.
- L'argument <report_changes> indique que la différence liée à la modification du fichier est sauvegardée (fonctionne uniquement sur les fichiers txt).
- L'argument <check_all> indique que la recherche de différences sur les fichiers indiqués se fait au fonction du hash md5 du fichier en question, du hash sha1, de la taille du fichier, du propriétaire du fichier, du groupe

auquel appartient le propriétaire du fichier et des permissions Linux accordées au fichier.
- L'argument <realtime> va permettre de scanner en temps réel les modifications d'un dossier (et uniquement d'un dossier).

Pour terminer avec ce fichier de configuration, nous désactivons le module de recherches de menaces "*rootcheck*" :

```
<rootcheck>
        <disabled>yes</disabled>
  </rootcheck>
```

Ce premier fichier de configuration est maintenant terminé, nous allons pouvoir passer à la création d'une règle sous */var/ossec/rules/local_rules.xml.*
La règle est la suivante :

```
<rule id="554" level="7" overwrite="yes">
<category>ossec</category>
<decoded_as>syscheck_new_entry</decoded_as>
<description>File added to the system.</description>
<group>syscheck</group>
</rule>
```

Les autres règles relatives à la modification d'un fichier sont nativement intégrées à l'outil. Il s'agit principalement de la règle "*550*" présente sous */var/ossec/rules/ossec_rules.xml*. Il est donc possible de l'éditer manuellement en cas de besoins additionnels.
Il est maintenant possible de lancer la configuration ossec ainsi que le service :

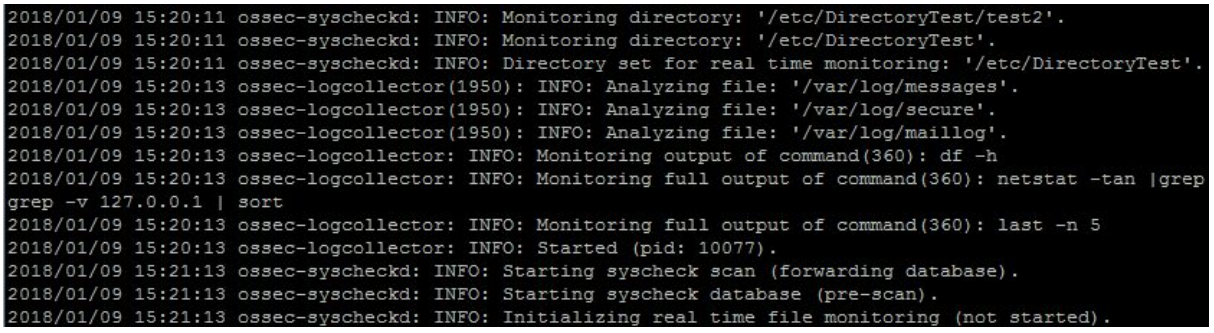
```
/var/ossec/bin/ossec-control [start-restart-stop]
```

Pour tester le bon fonctionnement de notre configuration, ouvrons une fenêtre nous montrant l'actualisation en direct des logs du service, une

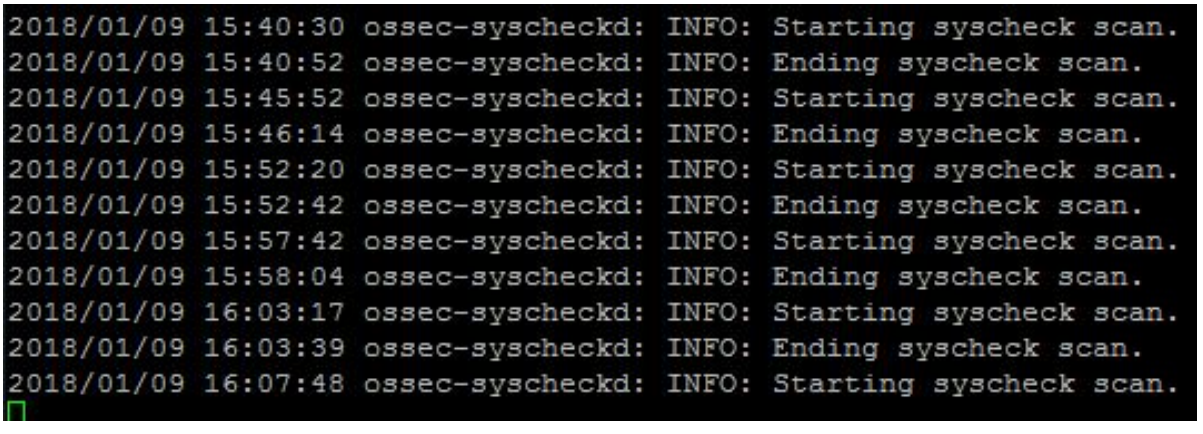
fenêtre nous remontant les alertes générées, et une dernière fenêtre pour modifier le fichier sur lequel le contrôle d'intégrité s'effectue.

tail -f /var/ossec/logs/ossec.log
tail -f /var/ossec/logs/alerts/2018/Jan/ossec_alerts-09.log
echo "test" >> /etc/DirectoryTest/test2

L'initialisation du service va prendre un certain temps, et une première analyse du/des fichiers va s'effectuer automatiquement :



Une fois cette première analyse effectuée, nous allons pouvoir observer le renouvellement automatique des scans du fichier :



Si nous modifions le fichier dont il est question, une alerte sera automatiquement, et instantanément remontée via le fichier de logs d'alertes :

```
Size changed from '50' to '55'
Old md5sum was: 'f8de96a2ddb5c995c054af69661bd4b3'
New md5sum is : '34ff53cd966f0fde4ea0b8b94580302f'
Old sha1sum was: 'cb18c626a8928eb6ec220c6117fd8b20a16bb304'
New sha1sum is : 'b38cc21ec25966f549a07d5d697ec536a990d1c6'
What changed:
17a18
> test
```

Les modifications du fichier indiquées génèrent ainsi automatiquement un message d'erreur comportant les modification effectuées. Il s'agit ici de la meilleur implémentation possible du système de vérification d'intégrité de fichier. En effet, un système de contrôle d'intégrité via ssh et sans installation côté client est aussi disponible sous OSSEC, mais celui-ci ne prend pas en compte les modifications en temps réel.

II. Architecture OSSEC Client-Serveur Unix

La mise en place d'OSSEC en architecture client-serveur unix reste très similaire à la mise en place de l'architecture locale. Le résultat est identique, mais est actif sur un plus grand nombre de machines (moins de 256), et permet de centraliser le logs pour les exploiter plus facilement.

II.I - Installation côté serveur

Le paquet est le même que précédemment, et l'installation se lance de la même manière :

```
wget -U ossec http://www.ossec.net/files/ossec-hids-2.8.1.tar.gz
tar xf ossec-hids-2.8.1.tar.gz
./install.sh
```

Les différences résident dans le choix du rôle :

```
1- What kind of installation do you want ? server
```

En fonction du besoin, les plugins peuvent êtres activés, mais puisque nous sommes uniquement concentrés sur la vérification d'intégrité de fichiers, il serait plus judicieux de désactiver les fonctionnalités indésirables (rootcheck, active-response, etc …).

II.II - Installation côté client

Côté client, le paquet d'installation est exactement le même que pour le serveur, et la démarche est identique. Il faudra simplement spécifier que la machine dispose d'un rôle d'agent :

```
1- What kind of installation do you want ? agent
```

L'installation se déroule ensuite d'elle même. Attention tout de même à bien activer le module de vérification d'intégrité, et à désactiver les modules indésirables.

II.III - Configuration côté serveur

Dans un premier temps, il est nécessaire de configurer le comportement du service au travers du fichier */var/ossec/etc/ossec.conf.*
Celui-ci est identique au fichier créé plus tôt, mais dispose de quelques arguments en plus :

```
<remote>
    <connection>secure</connection>
    <allowed-ips>10.9.97.25</allowed-ips>
    <local_ip>10.9.97.24</local_ip>
    <port>1514</port>
</remote>
```

Cette section <remote> va être capable d'activer le service *ossec-remoted,* et donc d'activer le listener de connexions entrantes.
La connexion doit être définie comme "secure".
L'adresse IP de l'agent doit être autorisée avec *<allowed-ips>.*
L'IP du serveur doit être renseignée sous *<local_ip>.*

Et enfin le port sous les balises *<port>.*
Les règles paramétrées sous les balises <rules> peuvent être toutes retirées, à l'exception de ces deux règles obligatoires au fonctionnement d'O pour effectuer des contrôles d'intégrités :

```
<rules>
    <include>rules_config.xml</include>
    <include>ossec_rules.xml</include>
</rules>
```

Il faudra également ouvrir les ports correspondants aux échanges agent-serveur sur la machine serveur :

```
iptables -I INPUT -p udp -m udp --dport 1514 -j ACCEPT
iptables -I INPUT -p udp -m tcp --dport 1514 -j ACCEPT
```

Nous allons ensuite intégrer l'agent distant à la configuration, via l'utilitaire *manage_agents* sous */var/ossec/bin/ :*

```
/var/ossec/bin/manage_agents
```

L'utilitaire nous permet de facilement ajouter des agents à la configuration :

```
****************************************
* OSSEC HIDS v2.8 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 sera ensuite nécessaire de fournir quelques informations sur l'agent OSSEC, à savoir un nom unique, une adresse IP et un numéro d'identification unique relatif au serveur.

```
Please provide the following:
  * A name for the new agent: UnixAgent
  * The IP Address of the new agent: 10.9.97.25
  * An ID for the new agent[001]: 001
```

Il s'agira ensuite de confirmer l'entrée effectuée pour ajouter l'agent OSSEC.

Une clé d'authentification est nécessaire pour l'appareillage entre le serveur et le client. Cette clé est récupérable uniquement sur le serveur, au sein de ce même utilitaire, via l'option *"(E)xtract key for an agent" :*

```
Available agents:
   ID: 001, Name: UnixAgent, IP: 10.9.97.25
Provide the ID of the agent to extract the key (or '\q' to quit): 001

Agent key information for '001' is:
x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x
```

Il sera nécessaire de la copier au sein d'un fichier puis de l'envoyer sur la machine agent avant de quitter l'utilitaire :

```
scp -r -p :/root/ossec-key.txt root@10.9.97.25:/root/key.txt
```

Le service peut maintenant être démarré :

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

II.IV - Configuration côté client

La configuration côté client nécessite les mêmes étapes que pour le serveur, à savoir la création du fichier de configuration *ossec.conf* ainsi que l'importation de la clé.

Le fichier de configuration est situé au même endroit, et comporte les mêmes informations que le serveur (sauf la localisation du / des fichiers à surveiller), et dispose d'une balise en plus, identifiant l'adresse du serveur :

```
<client>
<server-ip>x.x.x.x</server-ip>
</client>
```

Nous pouvons maintenant importer la clé précédemment exportée au sein de l'agent. Le module est le même que sur le serveur : *manage-agent,* mais ne dispose que de la possibilité d'importer une clée serveur :

```
**************************************
* OSSEC HIDS v2.8 Agent manager.     *
* The following options are available: *
**************************************
   (I)mport key from the server (I).
   (Q)uit.
Choose your action: I or Q: I
```

Il suffira de coller la clé brute au prochain prompt :

```
Paste it here (or '\q' to quit): x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x
```

La clé contient les identifiants de la machine agent reportée plus tôt (nom, adresse IP et numéro d'identification). Le tout est hashé en base64. Une fois la clé entrée, il est possible de démarrer le service client OSSEC :

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

II.V - Debug

Pour vérifier la communication entre les deux machines, nous disposons de plusieurs options :
Premièrement, vérifions que les services tournent correctement :

```
/var/ossec/bin/ossec-control status
```

La commande doit notamment nous montrer que le service “ossec-remoted” tourne correctememnt. Celui-ci se comporte comme un listener de connexions et d’informations entrantes sur le serveur, en provenance des agents. Si les fichiers relatifs aux communications entre les machines comportent des erreurs, celui-ci risque de ne pas démarrer.
Deuxièmement, la visualisation des logs ossec sont indicateur d’un grand nombre d’informations sur le lancement des services :

```
tail -f /var/ossec/logs/ossec.log
```

La séquence de boot du service doit comporter plusieurs lignes importantes :

```
ossec-remoted(1410): INFO: Reading authentication keys file.
```

Celle-ci indique que la clé échangée entre le serveur et le client est valide.

```
ossec-syscheckd: INFO: Monitoring directory: '/etc/FileTest'.
```

Celle-ci indique que le fichier en question est bien indiqué comme surveillé avec un système de contrôle d’intégrité… Une base de donnée va ensuite s’actualiser avec le contenu du fichier au lancement du

service. Les étapes sont là aussi détaillés, de façon à identifier les éventuelles erreurs.

Troisièmement, un dump tcp sur le port d'écoute du service va permettre d'observer le fonctionnement des échanges entre les machines :

tcpdump -i eth0 port 1514

Enfin, la liste des ports ouverts devrait permettre de confirmer l'origine du problème si le dump tcp ne donne rien :

iptables -nL

II.VI - Vérification

Le test final peut être réalisée de la façon suivante :
Nous ouvrons une fenêtre actualisant les logs d'alertes OSSEC :

tail -f /var/ossec/logs/alerts/alerts.log

Nous pouvons donc dès maintenant effectuer une modification sur l'agent et sur le serveur :

echo “from server” >> /etc/FileTest
echo “from agent” >> /etc/FileTest

Les deux modifications sont immédiatement et automatiquement remontées au serveur :

```
Integrity checksum changed for: '/etc/DirectoryTest/test2'
Size changed from '31' to '43'
Old md5sum was: '66765523ac13b392dcb6154d36dd9f49'
New md5sum is : 'b6ebd527248e3b35302f2ee0fe87a5f1'
Old sha1sum was: 'f4ff6073b6eda46e66f9a33e4bd5185d9107b632'
New sha1sum is : '441be74a4f91c8f6ee28c3f7a6be73a6038ef00d'
What changed:
8a9
> from server
```

```
Integrity checksum changed for: '/etc/DirectoryTest/test2'
Size changed from '45' to '56'
Old md5sum was: '8b7ca25f37e7c3883d2b840423de298f'
New md5sum is : 'd8b52820ee7d7358ddf1a4fdfdfcbb69'
Old sha1sum was: '04bf250ca850cba3de52551c8178f4fee901d004'
New sha1sum is : '7e24ac3ba1043f4cc5ced192e7af9c2a47aa4c4e'
What changed:
9a10
> from agent
```

Nous allons ensuite passer à l'ajout d'un agent Windows au setup actuel.

II. Architecture OSSEC Client-Serveur Windows

Côté serveur, la mise en place d'un agent Windows OSSEC se déroule de la même manière que pour les autre clients.
Il faudra ajouter l'agent via l'utilitaire *manage_agents* sous */var/ossec/bin/manage_agents* :

```
****************************************
* OSSEC HIDS v2.8 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 sera nécessaire de renseigner le nom de la machine, son adresse IP et un identifiant unique :

```
Please provide the following:
   * A name for the new agent: WindowsAgent
   * The IP Address of the new agent: 10.9.96.53
   * An ID for the new agent[002]: 002
```

Après avoir confirmé les informations, nous pourrons observer ce nouvel agent grâce à la commande */var/ossec/bin/agent_control -l :*

```
OSSEC HIDS agent_control. List of available agents:
   ID: 001, Name: UnixAgent, IP: 10.9.97.25, Active
   ID: 002, Name: WindowsAgent, IP: 10.9.96.53, Never connected
```

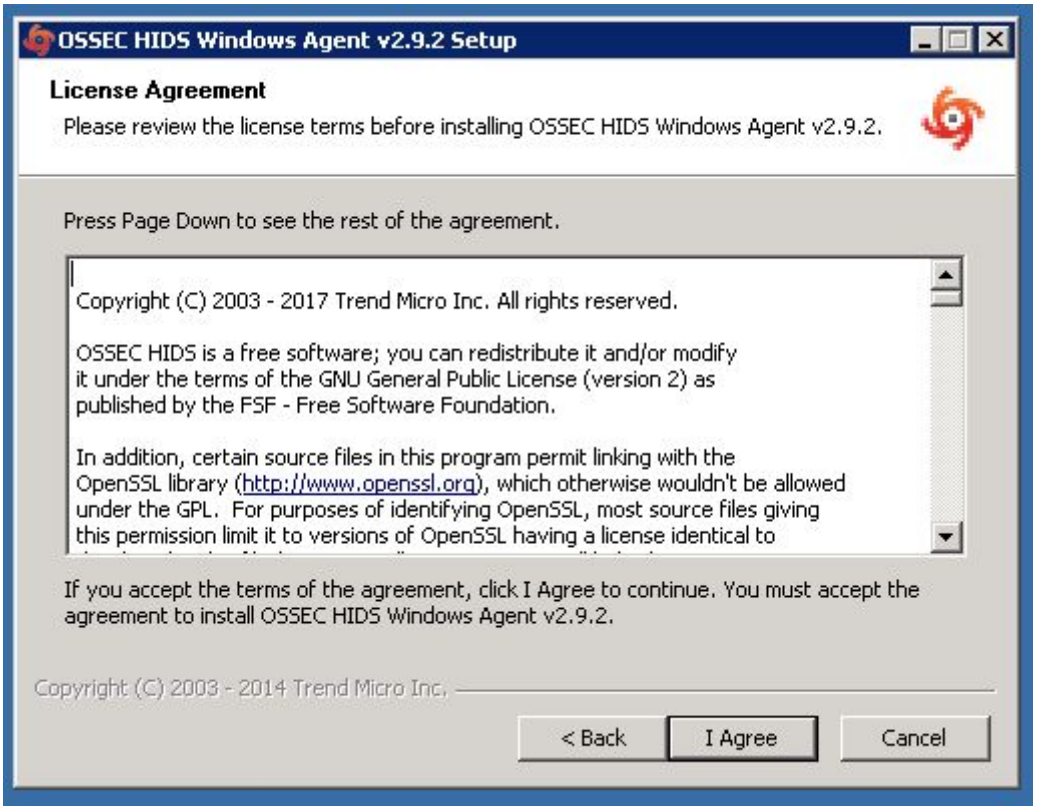
Maintenant que notre agent à été ajouté au serveur, il est possible d'en extraire la clé, puis de la transférer vers la machine agent, exactement de la même manière que précédemment.

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

Available agents:
   ID: 001, Name: UnixAgent, IP: 10.9.97.25
   ID: 002, Name: WindowsAgent, IP: 10.9.96.53
Provide the ID of the agent to extract the key (or '\q' to quit): 002

Agent key information for '002' is:
x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x
```

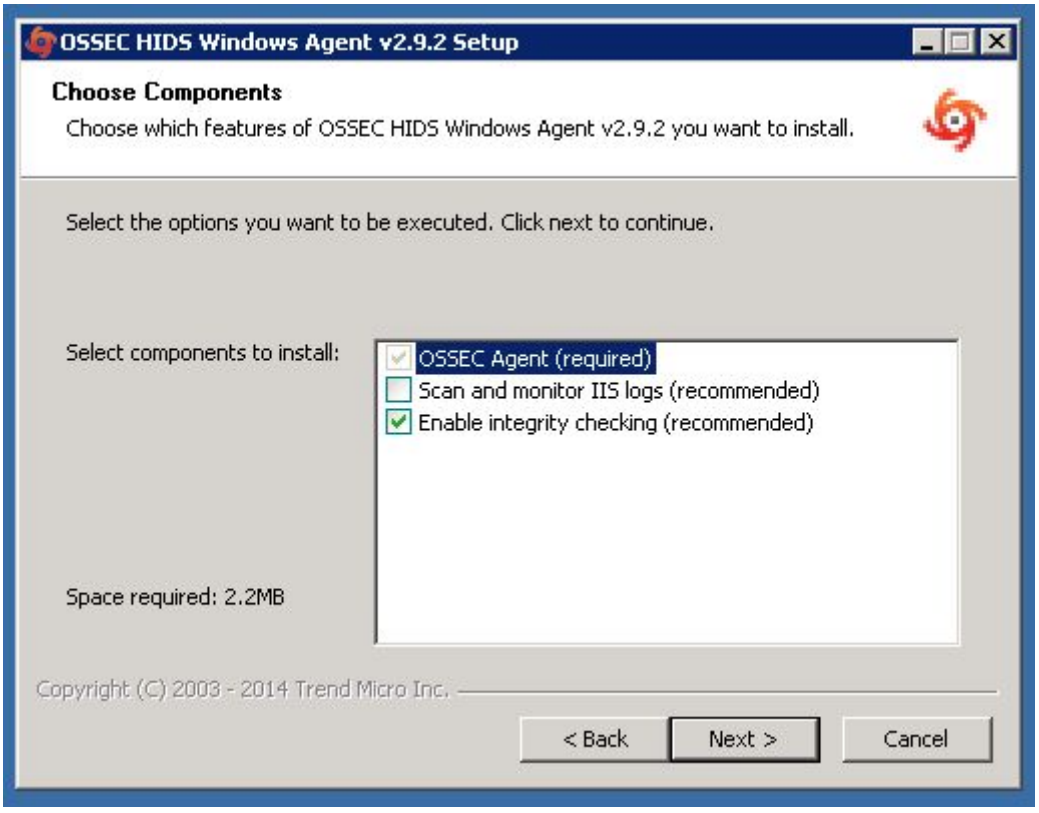
L'installation du client OSSEC windows peut maintenant débuter. L'exécutable est téléchargeable à l'adresse : https://osec.github.io/download.html.

Agent Windows	ossec-agent-win32-2.9.3.exe

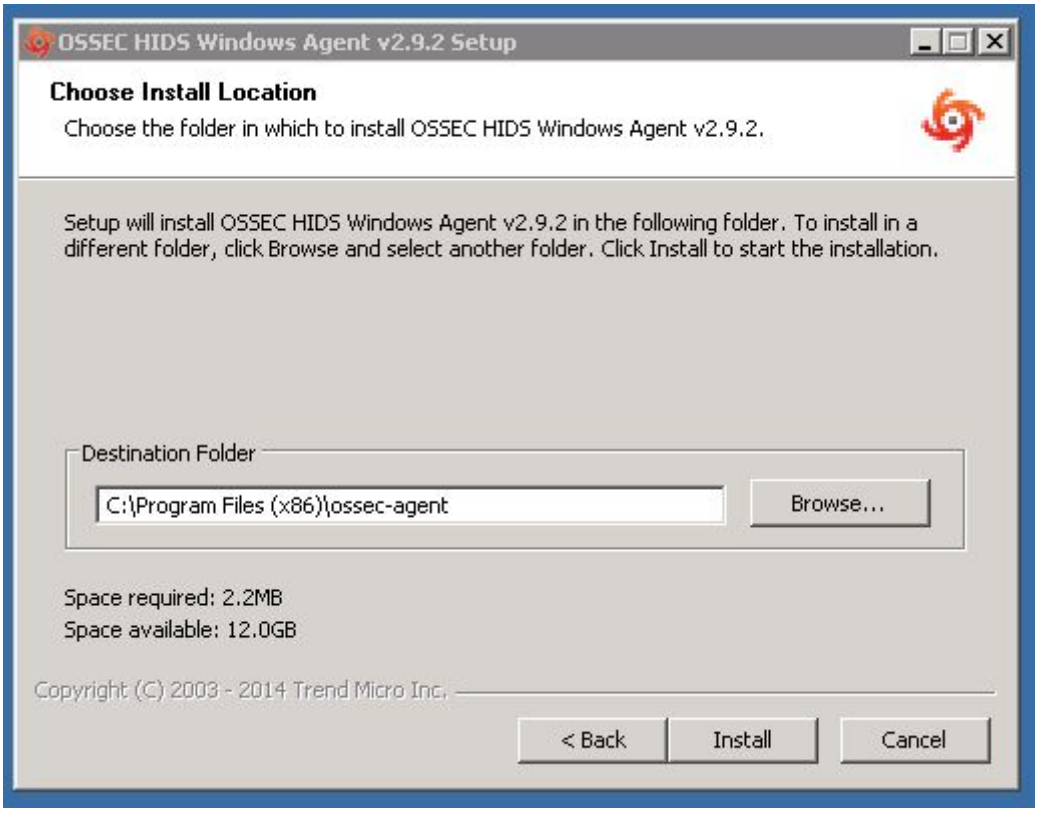
Le déroulement de l’installation reste classique :



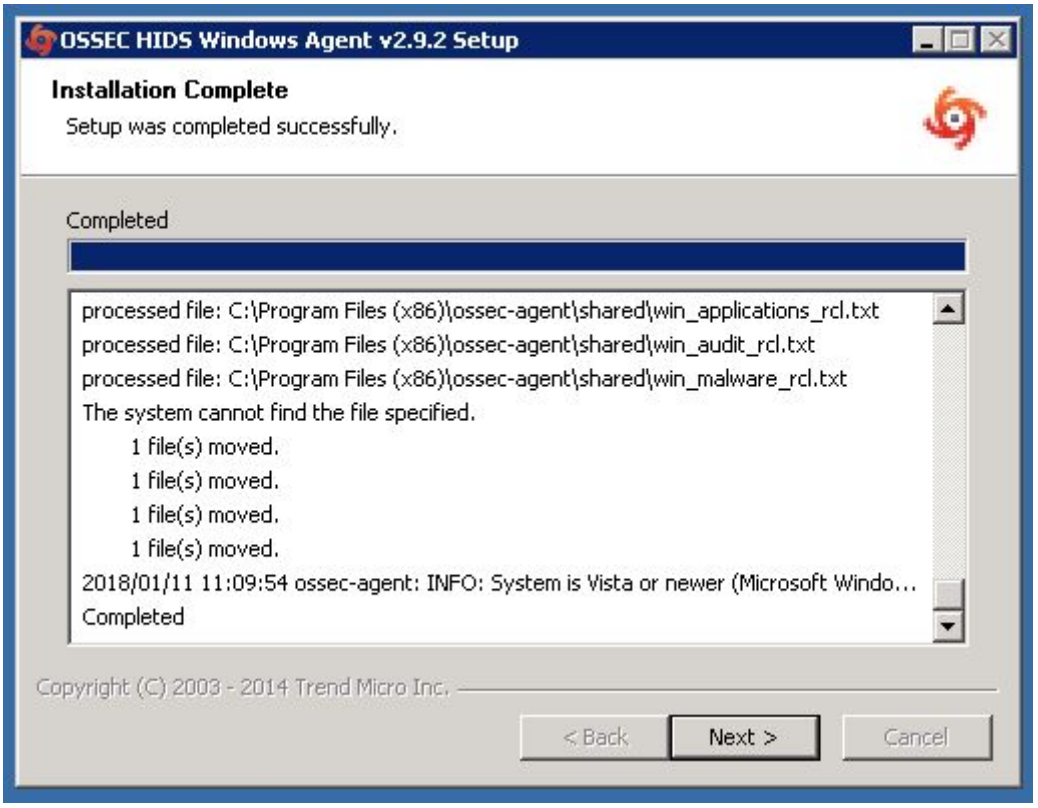
Nous allons tout de même désactiver le contrôle des logs IIS, et installer le module de contrôle d’intégrité :



La destination des fichiers installés peut rester celle par défaut :



L'installation va ensuite se faire d'elle même, et ne durer que quelques secondes :



Le manager OSSEC pour windows peut maintenant être ouvert pour débuter la phase de configuration de l'agent. L'adresse IP du serveur OSSEC devra être renseigné, ainsi que la clé de l'agent extraite à l'étape précédente :

En cliquant sur “Save”, il suffira de vérifier les informations sur l’agent :

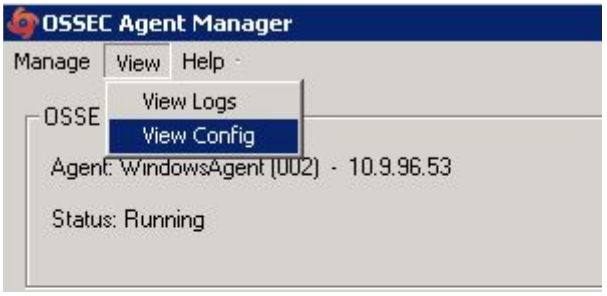
Il est maintenant possible de redémarrer l’agent pour appliquer les modifications, sous “Manage” > “Restart” :

La configuration initiale de l’agent est maintenant terminée, nous sommes donc capables de le voir comme “actif” dans la liste des agents présente sur le serveur, au travers de la commande */var/ossec/bin/agent_control -l* :

```
OSSEC HIDS agent_control. List of available agents:
   ID: 001, Name: UnixAgent, IP: 10.9.97.25, Active
```

```
ID: 002, Name: WindowsAgent, IP: 10.9.96.53, Active
```

Nous pouvons désormais commencer à modifier le fichier de configuration de l'agent. Le fichier est accessible via "View" > "View Config" :



Le fichier doit être remplacé par le même fichier de configuration agent Unix utilisé tout au long de cette documentation.
Attention à simplement renseigner les chemins relatifs jusqu'aux fichiers et dossiers à surveiller au format Windows :

```
<directories report_changes="yes" check_all="yes">C:\Program Files (x86)\ossec-agent\DirectoryTest\test1.txt</directories>
```

Il est également possible de surveiller l'intégrité des clés du registres :

```
<windows_registry>HKEY_LOCAL_MACHINE\Software\Classe\batfile</windows_registry>
```

Ou d'en retirer certaines de notre rayon d'action :

```
<registry_ignore>HKEY_LOCAL_MACHINE\Security\Policy\Secrets</registry_ignore>
```

Attention à redémarrer l'agent lors des modifications de la configuration.

Ainsi, la modification d'un des fichiers renseignés nous remontera instantanément une alerte via les logs */var/ossec/logs/alerts/alerts.log* :

```
2018 Jan 11 11:41:36 (WindowsAgent) 10.9.96.53->syscheck
Rule: 550 (level 16) -> 'Integrity checksum changed.'
Integrity checksum changed for: 'C:\Program Files (x86)\ossec-agent\DirectoryTest/test1.txt'
Size changed from '22' to '37'
Old md5sum was: '1e233901a561cc711aa02a30497f8d67'
New md5sum is : '59d71eda9e7a798838d419efd468b602'
Old sha1sum was: '1f3d100ec3d7bc1079199da5933e4df7ca671143'
New sha1sum is : 'b4d43181cd4c4faddc2d79a9d7a10b6afd9a16a9'
```

La configuration de l'agent Windows est maintenant terminée. Toute modification des fichiers indiquées dans la configuration retournera une alerte sur le serveur OSSEC.