# Architecture de sécurité

# Architecture de sécurité

## Architecture de sécurité

L'objectif de ce document est de décrire les éléments de sécurité de chaque composant de Pandora FMS, afin que l'administrateur les connaisse et sache comment les utiliser pour mettre en œuvre une architecture plus sécurisée, conformément aux réglementations telles que PCI/DSS, ISO 27001, ENS, LOPD ou similaires. En plus d'une description spécifique des mécanismes de sécurité de chaque élément de Pandora FMS, les risques possibles et la manière de les atténuer, en utilisant les outils dont dispose Pandora FMS ou d'autres mécanismes possibles.


## Mise en œuvre de la sécurité (généralités)

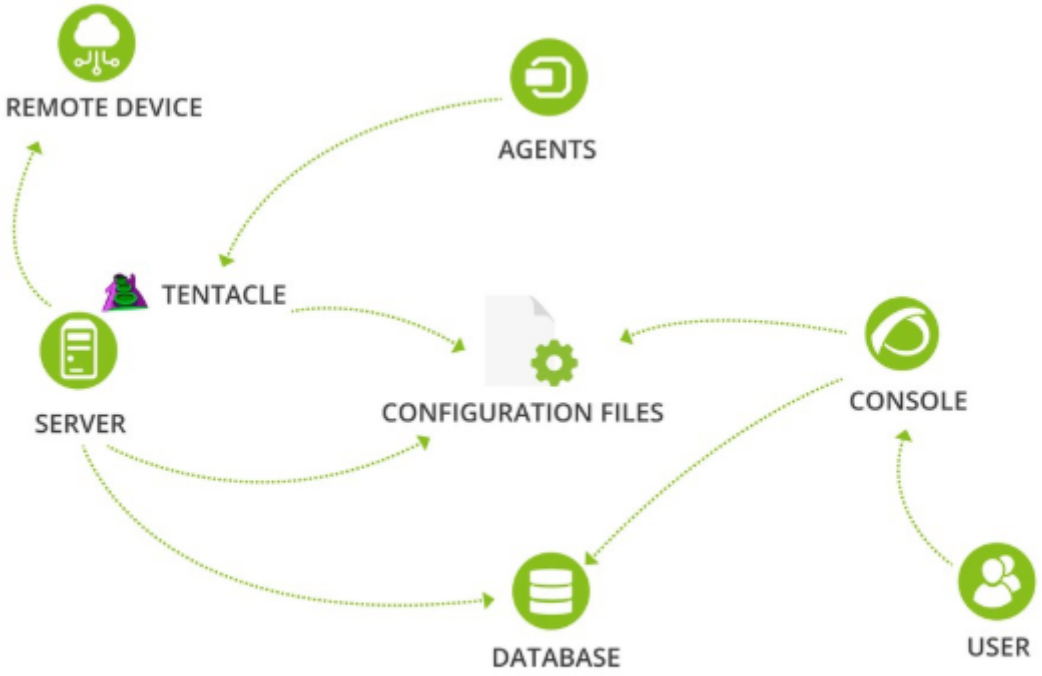
Ces points s'appliquent aux normes internationales telles que PCI/DSS, ISO 27001, National Security Scheme, LOPD, entre autres. Ils peuvent vous aider à réaliser une mise en œuvre sécurisée de Pandora FMS dans votre environnement.

- Les ports d'entrée et de sortie des composants de Pandora FMS sont documentés, de sorte qu'il est possible de sécuriser par des pare-feu (*f irewalls*) tous les accès vers et depuis ses composants.
- Sécurisation du trafic grâce au cryptage et aux certificats : Pandora FMS, à tous les niveaux (fonctionnement de l'utilisateur, communication entre les composants), prend en charge le cryptage SSL/TLS et les certificats aux deux extrémités.
- Système d'authentification à double accès : Un système d'authentification double peut être mis en œuvre. La première, au niveau de l'accès (HTTP) s'intègre à n'importe quel système de jeton Opensource ou commercial.
- Système d'authentification tiers : Au niveau de l'application, il est géré par Pandora FMS, qui peut s'authentifier contre LDAP ou Active Directory.
- SSO (Single Sign-On), avec SAML.
- Les politiques de sécurité dans la gestion des utilisateurs : La gestion des utilisateurs est délimitée par des politiques tant au niveau du profil de l'utilisateur qu'au niveau du profil de visibilité des opérations, défini comme le système ACL étendu.
- Possibilité d'audits sur les actions des éléments supervisés : Pandora FMS audite toutes les actions

des utilisateurs, y compris les informations sur les champs modifiés ou supprimés. Il comprend également un système de validation par signature de ces enregistrements.
- Transfert des données d'audit vers des gestionnaires de journaux externes : Les journaux d'audit sont disponibles pour l'exportation via SQL et peuvent être intégrés dans une 3ème source pour une plus grande sécurité, en quasi temps réel.
- Séparation physique des composants : Fournir une interface entre l'utilisateur et les conteneurs d'informations (système de fichiers). Les données stockées dans la base de données et les systèmes de fichiers qui stockent les informations de configuration du monitorage peuvent se trouver sur des machines physiques distinctes, dans des réseaux différents, protégés par des systèmes de périmètre.
- Politique active de mots de passe : Elle permet d'appliquer une politique stricte de gestion des mots de passe pour l'accès aux utilisateurs de l'application (console).
- Cryptage des données sensibles : Le système permet de stocker les données les plus sensibles, telles que les identifiants d'accès, de manière cryptée et sécurisée.

## Sécurité par les composants de l'architecture

L'architecture de Pandora FMS, de manière très simplifiée, peut être résumée comme suit :




### Serveur

- Le serveur nécessite des droits root, mais peut être installé (avec des limitations) avec des droits non-root (uniquement sur les systèmes GNU/Linux).
- Le serveur a besoin d'un accès direct (lecture et écriture) aux fichiers de configuration distants des agents, qui sont propagés lorsque les agents contactent le serveur, de façon périodique. Ces fichiers sont protégés par le système de fichiers (*filesystem*), avec des permissions standard.

- Le serveur en tant que tel n'écoute aucun port. C'est le serveur Tentacle qui écoute sur un port, le serveur n'accède qu'aux fichiers qu'il laisse sur le disque.
- Le serveur a son propre journal des événements (*log*), très détaillé.
- Le serveur se connecte à la base de données principale via une connexion MySQL/TCP standard.
- Une partie du code peut être commandée sous des conditions contractuelles spécifiques (pour les clients uniquement).

### Vulnérabilités et sauvegardes possibles

- Accès non autorisé aux fichiers de configuration de l'agent. Solution :
  1. Implémenter un conteneur externe sécurisé pour les fichiers de configuration externes, via NFS.
- Injection de commandes dans des agents distants par la manipulation de fichiers de configuration stockés dans le conteneur de configuration. Solution :
  1. Désactivez la configuration à distance sur les agents particulièrement sensibles après la configuration et laissez-les fonctionner sans pouvoir modifier quoi que ce soit à distance, pour une sécurité absolue.
  2. Surveillance à distance - sans agent - des dispositifs les plus sensibles.
- Vulnérable aux attaques de falsification d'informations, simulant des agents que nous n'avons pas dans le système ou supplantant leur identité. Plusieurs mécanismes peuvent être utilisés pour éviter cela :
  1. Mécanisme de protection par mot de passe (qui fonctionne par groupe).
  2. Limiter l'auto-création des agents, et les créer manuellement.
  3. Limiter la capacité d'auto-détection des changements dans l'agent et ne pas prendre les nouvelles informations du XML dans l'existant.
- Capture malveillante de la communication entre le serveur et la console (capture du trafic réseau). Solution :
  1. Activez la communication TLS entre le serveur et la base de données MySQL.

## Tentacle

- Tentacle est un service Internet officiel, documenté comme tel par l'IANA. Cela signifie qu'il peut être facilement protégé avec n'importe quel outil de sécurité du périmètre.
- Il ne nécessite pas de root ou de privilèges spéciaux.
- Il dispose de quatre niveaux de sécurité : aucun cryptage (par défaut), SSL/TLS de base, SSL/TLS avec certificat aux deux extrémités et SSL/TLS avec validation du certificat et de l'AC.
- Conçu spécifiquement pour ne donner aucun indice aux intrus potentiels dans les messages d'erreur et avec des délais d'attente spécifiques pour décourager les attaques par force brute.
- Possède son propre journal (*log*) d'audit.
- 100 % du code est accessible (sous licence Opensource GPL2).

### Vulnérabilités et sauvegardes possibles

- Attaques sur le système de fichiers (*filesystem*). Vous devez accéder au conteneur de configuration. Solution :
  1. Il est protégé de la même manière que le serveur, par un système NFS externe sécurisé.

- Attaques DoS dues à la surcharge. Solutions :
  1. Montez une solution HA au-dessus du service TCP qu'elle fournit pour l'équilibrage, ou un cluster actif/actif. Toute solution matérielle ou logicielle disponible convient, car il s'agit d'un service TCP standard.

## Console

- Elle ne nécessite pas d'être root, elle s'installe avec un utilisateur non privilégié.
- Elle doit avoir accès au référentiel de configuration de l'agent (*filesystem*).
- Elle écoute sur les ports HTTP ou HTTPS standard.
- Consigne toutes les demandes via la consignation des demandes HTTP.
- Elle fournit une API publique via HTTP/HTTPS, sécurisée par des informations d'identification.
- Il existe un audit spécifique à l'application, qui consigne l'activité de chaque utilisateur sur chaque objet du système.
- Il est possible de limiter l'accès de chaque utilisateur à n'importe quelle section de l'application, et même de créer des administrateurs aux droits restreints.
- L'application intègre un système de double authentification.
- L'application intègre un système d'authentification déléguée (LDAP, AD).
- Un système en lecture seule peut être mis en place. Sans accès aux configurations des appareils.
- Les informations sensibles (mots de passe) peuvent être stockées de manière cryptée dans la base de données.
- L'application se connecte à la base de données principale via une connexion MySQL/TCP standard.
- Une partie du code peut être commandée sous des conditions contractuelles spécifiques (uniquement pour les clients).
- Il existe une forte mise en œuvre des politiques de sécurité concernant les mots de passe (longueur, changement forcé, historique, type de caractères valides, etc.).

### Vulnérabilités et sauvegardes possibles

- Attaques sur le système de fichiers (*filesystem)*. Doit accéder au conteneur de configuration. Solution:
  1. Il est protégé de la même manière que le serveur, par un système NFS externe sécurisé.
- Attaques par force brute ou par dictionnaire contre l'authentification des utilisateurs. Solutions :
  1. Mettre en place une politique de mot de passe difficile (point 14).
  2. Mettre en œuvre un mécanisme d'authentification à deux facteurs (point 8).
- Capture du trafic (*eavesdropping*) du trafic vers la console. Solution:
  1. Mettre en œuvre SSL/TLS.
- Capture du trafic (*eavesdropping*) du trafic vers la base de données. Solution:
  1. Mettre en œuvre SSL/TLS.
- Les attaques par injection SQL pour obtenir des informations confidentielles de la base de données de l'application. Solution :
  1. Mettre en œuvre le stockage de données cryptées.
- Mauvaise utilisation (intentionnelle ou non) par les utilisateurs de l'application. Solutions:
  1. Activer le journal d'audit (*log*) et montrez aux utilisateurs qu'il existe et qu'il est exact.
  2. Activer le système ACL étendu pour restreindre au maximum les rôles de chaque utilisateur.

3. Exporter régulièrement le journal (*log*) d'audit vers un système externe.

- Exécution de code malveillant dans les outils de la console locale, en remplaçant les fichiers binaires. Solution :
  1. Sécurisation ou durcissement (*hardening*) du serveur contenant l'application.

## Agents

- Ils peuvent être exécuté sans les droits de superutilisateur (avec des fonctionnalités limitées).
- La gestion à distance de l'agent peut être désactivée (localement et à distance), de sorte que l'impact d'une intrusion dans le système central peut être minimisé.
- L'agent n'écoute pas les ports du réseau, c'est lui qui se connecte au serveur Pandora FMS.
- Il existe un journal de chaque exécution.
- Les fichiers de configuration sont protégés par défaut, au moyen des permissions du système de fichiers. Seul un utilisateur disposant de droits de super administrateur peut les modifier.
- 100 % du code est accessible (sous licence Opensource GPL2).

### Vulnérabilités et sauvegardes possibles

- Intrusion dans le système central qui permet de distribuer l'exécution de commandes malveillantes aux agents. Solutions :
  1. Limiter les utilisateurs qui peuvent effectuer ces modifications de politique ou de configuration (via une ACL de console ordinaire ou une ACL étendue).
  2. Activer le mode lecture seule pour les agents (ils n'autorisent pas les modifications de configuration à distance), pour les systèmes particulièrement sensibles.

- Faiblesse du système de fichiers (*filesystem)* qui permet de modifier des fichiers. Solution :
  1. Configuration correcte des permissions.
- Exécution de *plugins* ou de commandes malveillantes. Solution :
  1. Limiter les utilisateurs qui peuvent télécharger des exécutables (par le biais d'une ACL de console ordinaire ou d'une ACL étendue).
  2. Effectuez un audit des nouveaux *plugins*.

## Base de données

- Il s'agit d'un produit standard (MySQL).

### Vulnérabilités et sauvegardes potentielles

- *Eavesdropping* (capture du trafic réseau). Solution :
  1. Mise en œuvre d'une connexion TLS sécurisée. MySQL le prend en charge.

- Permissions incorrectes. Solution :
  1. Configuration correcte des autorisations d'accès.

- Vulnérabilités connues de MySQL. Il est conseillé d'établir un plan de mise à jour du serveur MySQL afin de le maintenir le plus à jour possible et de se débarrasser ainsi des éventuelles vulnérabilités des anciennes versions.

# Durcissement du système de base

Le durcissement (*hardening*) ou la sécurisation des systèmes est un point clé de la stratégie de sécurité globale d'une entreprise. En tant que fabricants, nous émettons une série de recommandations pour une installation sécurisée de tous les composants de Pandora FMS, basée sur une plateforme standard RHEL7 ou son équivalent Centos7. Ces mêmes recommandations sont valables pour tout autre système de supervision basé sur GNU/Linux.

## Accréditations d'accès

Pour accéder au système, des utilisateurs à accès nominatif seront créés, sans privilèges et avec un accès restreint à leurs besoins. Idéalement, l'authentification de chaque utilisateur devrait être intégrée à un système de double authentification, basé sur des jetons. Des alternatives gratuites et sécurisées telles que Google Authenticator® sont disponibles et peuvent être intégrées à GNU/Linux, mais elles dépassent le cadre de ce guide. Envisagez sérieusement de les utiliser.

S'il est nécessaire de créer d'autres utilisateurs pour les applications, il doit s'agir d'utilisateurs sans accès à distance (pour ce faire, désactivez leur shell ou une méthode équivalente).

## Accès super-utilisateur

Dans le cas où certains utilisateurs doivent avoir des droits d'administrateur, la commande sudo est utilisée.

## Maintenez votre système à jour

Il vous suffit d'être connecté au réseau ou de configurer votre système yum pour utiliser un serveur *proxy*.

Cette commande peut causer des problèmes potentiels en cas de modification des bibliothèques, des configurations, etc. Il est important de ne pas sauter la mise à niveau du système, surtout lors de la mise en production du système. Si vous vérifiez un système de production déjà actif, vous pouvez avoir besoin de mettre à jour uniquement les composants critiques, par exemple ceux qui présentent une vulnérabilité. Par exemple, si vous voulez mettre à jour uniquement mysql-server, utilisez la commande avec le nom du paquet que vous voulez mettre à jour.

```
yum update mysql-server
```

La mise à niveau du système est un processus qui doit être effectué périodiquement. En utilisant l'inventaire des paquets système, vous pouvez vérifier la présence de versions vulnérables et

exécuter des mises à jour d'urgence.

### Audit d'accès

Vous devez avoir le journal de sécurité `/var/log/secure` actif et superviser ces journaux (*logs*) avec le monitoring (que nous verrons plus tard).

Par défaut, CentOS est livré avec cette option activée. Sinon, vérifiez `/etc/rsyslog.conf` ou `/etc/syslog.conf` .

Nous vous recommandons de prendre les logs du système d'audit et de les collecter avec un système de gestion de *logs* externe, Pandora FMS peut le faire facilement et il sera utile d'établir des alertes ou de les revoir de manière centralisée en cas de besoin.

### Sécuriser le serveur SSH

Le serveur SSH vous permet de vous connecter à distance à vos systèmes GNU/Linux pour exécuter des commandes, il s'agit donc d'un point critique qui doit être sécurisé en faisant attention aux points suivants (pour ce faire, modifiez le fichier `/etc/ssh/sshd_config` et redémarrez ensuite le service).

- Modifier le port par défaut (par exemple en 31122)

```
#Port22     ->     Port 31122
```

- Désactiver le root login

```
#PermitRootLogin yes        ->    PermitRootLogin no
```

- Désactiver le transfert de port port forwarding

```
#AllowTcpForwarding yes        ->    AllowTcpForwarding no
```

- Désactiver tunneling

```
#PermitTunnel no        ->    PermitTunnel no
```

- Suppression des clés SSH pour l'accès root à distance. Supposons qu'il n'y ait qu'un seul utilisateur valide pour l'accès à distance (par exemple, artica). S'il y en a d'autres, nous devrions les vérifier également. Pour ce faire, regardez le contenu du fichier `/home/artica/.ssh/authorized_keys` et voyez de quelles machines vous êtes issus. Supprimez-la si vous pensez qu'il ne devrait pas y en avoir.
- Définissez un avertissement standard d'accès à distance qui explique que le serveur est privé et que toute personne ne possédant pas d'informations d'identification doit se déconnecter.

```
Banner /etc/issue.net
```

## Sécuriser le serveur MySQL

Port d'écoute. Si le serveur MySQL doit desservir l'extérieur, vérifiez simplement que les informations d'identification de root sont sécurisées. Si MySQL ne sert qu'un élément interne, assurez-vous qu'il n'écoute que sur *localhost* :

```
netstat -an | grep 3306 | grep LIST
tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN
```

Dans ce cas, il s'agit d'une écoute pour tous, vous devez vous en assurer. Pour ce faire, éditez le fichier `/etc/my.cnf` et dans la section `[mysqld]` ajoutez la ligne suivante :

```
bind-address = 127.0.0.1
```

Et vérifiez à nouveau qu'il écoute, mais maintenant seulement sur *localhost* après avoir redémarré le service :

```
netstat -an | grep 3306 | grep LIST
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
```

Mot de passe MySQL

Connectez-vous à la console MySQL avec un utilisateur privilégié :

```
mysql –h host –u root –p
```

Vérifiez que le mot de passe est sécurisé et qu'on vous a demandé un mot de passe. Si ce n'est pas le cas, réglez-le avec la commande :

```
mysqladmin password
```

Cette mesure de sécurité est essentielle pour protéger vos bases de données non seulement contre les attaques extérieures, mais aussi contre les utilisations abusives par des utilisateurs internes.

## Sécurisation d'Apache

```
ServerTokens Prod
```

Ajoutez la ligne ci-dessus au fichier:

- `/etc/httpd/conf/httpd.conf` (RHEL).
- `/etc/apache2/conf-enabled/pandora_security.conf` (Ubuntu server)

pour cacher la version du serveur web (Apache, Nginx) dans les en-têtes d'information du serveur.

### Moteur d'application PHP

Afin de *sécuriser* le moteur d'application sur lequel fonctionne Pandora FMS, il peut être nécessaire, dans certains environnements particulièrement sensibles à la sécurité, de *sécuriser* l'accès à l'application afin que les *cookies* de session ne soient transmis qu'en SSL.

Pour ce faire, incluez la configuration *tokens* suivante dans le fichier `php.ini` :

```
session.cookie_httponly = 1
session.cookie_secure = 1
```

L'application ne fonctionnera pas si elle est utilisée sur
HTTP (sans cryptage).

### Minimisation des services dans le système

Cette technique, qui peut être très exhaustive, consiste simplement à éliminer tout ce qui n'est pas nécessaire dans le système. De cette manière, vous évitez d'éventuels problèmes à l'avenir avec des applications mal configurées dont vous n'avez pas vraiment besoin. Pour simplifier l'approche de cette pratique, considérerez uniquement les applications qui ont un port ouvert sur la machine, pour ce faire, exécutez la commande suivante :

```
netstat -tulpn
```

Il devrait retourner un résultat pour chaque port d'écoute, quelque chose de similaire à ceci, mais pas le même :

```
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN
996/master
tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN
75171/httpd
tcp        0      0 0.0.0.0:31122           0.0.0.0:*               LISTEN
872/sshd
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
75097/mysqld
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
75171/httpd
tcp        0      0 0.0.0.0:6099            0.0.0.0:*               LISTEN
7721/Xvfb
tcp6       0      0 :::4444                 :::*                    LISTEN
7726/java
```

```
tcp6       0      0 :::34599                :::*                    LISTEN
7726/java
tcp6       0      0 :::6099                 :::*                    LISTEN
7721/Xvfb
```

Si vous avez désactivé IPv6, vous ne devriez pas voir de lignes tcp6, à moins qu'il ne s'agisse de services qui ont été démarrés en mode ipv6 et qui ont été laissés non démarrés après avoir effectué un changement de système avec sysctl.

Vous devez étudier chaque port et connaître l'application qui se cache derrière. Dans ce cas, 443, 80 semblent provenir du service `http`, mais pour en être sûr, nous allons analyser quels processus système utilisent chaque port. Pour ce faire, nous allons utiliser la commande lsof, qui devra être installée avec yum car elle n'est pas installée par défaut.

Les services qui écoutent sur *localhost* (`127.0.0.1`) sont plus sûrs que ceux qui écoutent sur toutes les adresses IP (`0.0.0.0`) et certains d'entre eux, s'ils écoutent en clair, vous devriez essayer de les sécuriser pour qu'ils n'écoutent que *localhost*. Dans cet exemple d'écran, cela a été fait par exemple pour MySQL (3306).

Par exemple, vous voyez que le processus principal de postfix est en cours d'exécution. Comme vous n'avez pas besoin de ce service, désinstallez-le du système :

```
yum remove postfix
```

En examinant le PID de chacun des processus en question (voir l'étape précédente), vous verrez le processus qui se trouve sur ce port :

```
ps aux | grep 7726 root 7726 0.1 8.5 3258724 248608 ? Sl Mar09 60:01
/usr/bin/java -jar /usr/lib/pwr/selenium-server-standalone-2.53.1.jar -host
185.247.117.28 -port 4444 -firefoxProfileTemplate /opt/firefox_profile root
79041 0.0 0.0 112716 960 pts/4 S+ 11:54 0:00 grep --color=auto 7726
```

Et si vous n'utilisez pas ce service, vous pouvez le supprimer.

Ce processus “ d'investigation ” des processus doit être exhaustif et répétitif dans le temps. Au moyen du système d'inventaire des processus de Pandora FMS, vous devez vérifier qu'aucun nouveau processus n'est lancé au fil du temps. Un port d'écoute dans un serveur est quelque chose de très important du point de vue de la sécurité, c'est comme une fenêtre à l'avant du bâtiment. Vous pouvez penser qu'elle est fermée et sûre, mais une fenêtre sera toujours un point d'entrée possible pour un intrus qualifié et motivé.

## Configuration supplémentaire

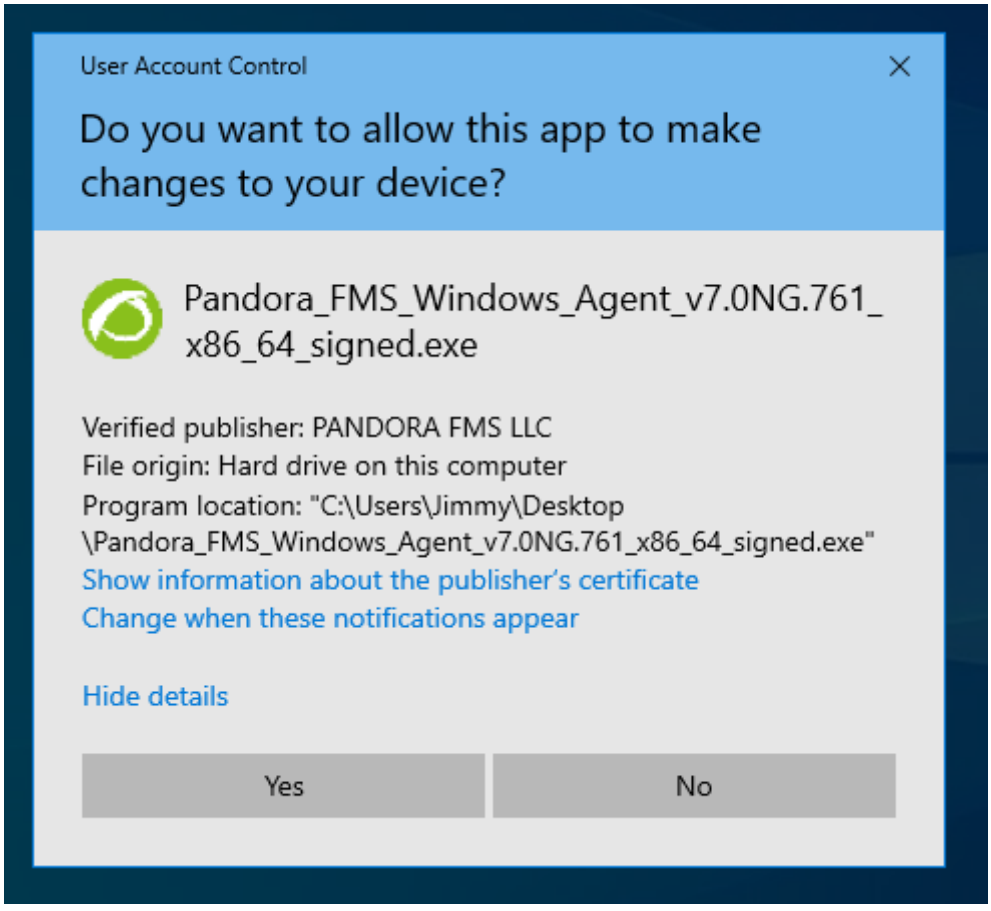
### Synchronisation du temps NTP

Il est recommandé de configurer la synchronisation horaire du système :

```
yum install ntpdate
echo "ntpdate 0.us.pool.ntp.org"> /etc/cron.daily/ntp
chmod 755 /etc/cron.daily/ntp
```

## Surveillance locale

Le système doit avoir un agent logiciel Pandora FMS installé et lancé contre notre serveur. Pour le système d'exploitation MS Windows®, à partir de la version 761, les exécutables d'installation sont signés.




Les contrôles actifs suivants sont recommandés en plus des contrôles standard :

- *Plugin* de sécurité actif.
- Inventaire complet du système (notamment les utilisateurs et les paquets installés).
- Collecte des journaux (logs) du système et de la sécurité :

```
module_plugin grep_log_module /var/log/messages Syslog \.\*
module_plugin grep_log_module /var/log/secure Secure \.\*
```

Une fois l'agent logiciel installé, au moins les informations suivantes doivent être définies manuellement dans l'onglet de l'agent :

- Description.
- Adresse IP (si vous en avez plusieurs, mettez-les toutes).
- Groupe.
- Département, responsable et lieu physique (*custom fields*).

## Surveillance de la sécurité dans GNU/Linux

Le plugin officiel permet de surveiller de manière proactive la sécurité de l'agent, à chaque exécution, presque en temps réel, en proposant des vérifications qui peuvent nous alerter de certains événements pertinents de manière proactive.

Ce plugin est destiné à fonctionner uniquement sur des machines GNU/Linux modernes. Il est prêt à fonctionner sur 64 bits et 32 bits.

Il contient une version personnalisée de John the ripper 1.8 + des correctifs Contrib avec des binaires 32 et 64 statiques. Le concept principal du *plugin* est d'être monolithique, de détecter ce qui peut être renforcé et d'essayer de résoudre les différences entre les distros sans rien demander à l'administrateur, de sorte que le déploiement pourrait être le même pour n'importe quel système, sans tenir compte des versions, des distros ou de l'architecture.

Ce *plugin* vérifiera:

- Vérification du mot de passe de l'utilisateur, en utilisant le dictionnaire (fourni) avec les 500 mots de passe les plus courants. Cela ne prend généralement pas plus de quelques secondes. Si vous avez des centaines d'utilisateurs, vous devrez probablement personnaliser le plugin pour qu'il ne fonctionne que toutes les 2 à 6 heures. Vous pouvez personnaliser le dictionnaire de mots de passe en ajoutant simplement le mot de passe type de votre organisation au fichier “basic_security/password-list”.
- Vérifiez que SSH ne fonctionne pas sur le port par défaut.
- Vérifiez que le FTP ne fonctionne pas sur le port par défaut.
- Vérifiez que SSH n'autorise pas l'accès root.
- Vérifiez s'il y a un MySQL en cours d'exécution sans le mot de passe root défini.
- Autres contrôles.

Retour à l'index de documentation du Pandora FMS