Satellite Server

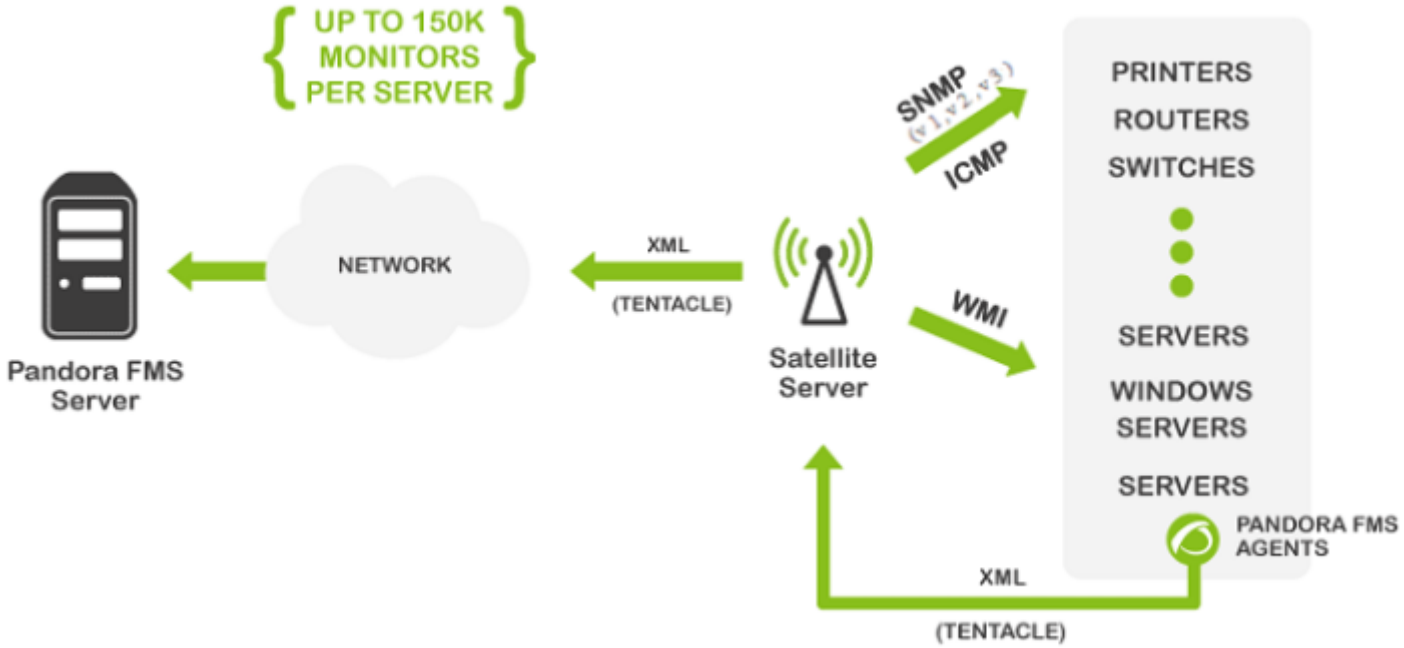
Print:
https://pandorafms.com/manual/!775/
Permanent link:
https://pandorafms.com/manual/!775/fr/documentation/pandorafms/complex_environments_and_optimization/05_satellite
2024/03/18 21:03

Satellite Server

Nous travaillons sur la traduction de la documentation du FMS Pandora. Nous sommes désolés pour tout désagrément.

Introduction

Le Satellite Server est utilisé pour découvrir et surveiller des réseaux et des équipements distants, soit des éléments de réseau (routeurs, commutateurs, etc.) via SNMP ou ICMP, soit des serveurs Windows® (via WMI) ou Linux® (via SNMP). Ce n'est pas un serveur “ ordinaire ”, mais il peut être considéré comme un Agent en mode broker avec des fonctionnalités étendues. Il est particulièrement utile pour surveiller les réseaux distants qui sont inaccessibles depuis le serveur Pandora FMS, et où nous ne pouvons pas non plus installer d'agents.



Le Serveur Satellite fonctionne sous Windows® et Linux® (système d'exploitation recommandé), et possède quelques caractéristiques qui le rendent spécial, plus que recommandé dans certains environnements.

- Il peut effectuer des tests réseau (ICMP, Latence et SNMP v1 et v2) à un rythme extrêmement élevé (500 vérifications par seconde). Pour SNMP v3 configurez les identifiants d'accès et en raison du chiffrement des donnés la vérification sera plus lente.
- Il n'envoie des données au serveur que toutes les X secondes (par défaut 300), mais il peut exécuter

les tests de latence, ICMP et SNMP avec un intervalle plus petit (par exemple 30 secondes), de sorte que, lorsqu'il détecte des changements d'état, il en informe immédiatement le serveur. Ces changements d'état doivent être définis au préalable si le type de module n'est pas un `*_proc` (par exemple, les interfaces réseau ou la connectivité réseau générale).

- C'est un serveur autonome, il ne nécessite pas de connexion à la base de données. Il envoie toutes les données en XML pour qu'il fonctionne comme un serveur indépendant, comme le fait un agent en mode courtier ou vers un Export server.
- Il dispose d'un mécanisme d'autodiscovery pour SNMP et WMI, il crée donc les agents détectés (par IP), détecte les éléments dynamiques (interfaces réseau, stockage) et les surveille automatiquement.
- Dans les systèmes Windows®, il détecte les disques, le CPU et la mémoire.
- Dans les systèmes réseau avec SNMP, il détecte l'état des interfaces, le trafic d'entrée et de sortie pour chaque interface et le nom du système.
- Les modules autogénérés peuvent être modifiés, comme un autre module, en gérant l'agent depuis la console, comme s'il s'agissait d'un agent ordinaire (dans la section des Opérations massives → Satellite).
- Il est possible de créer des agents manuellement, en créant un fichier de configuration d'agent dans le répertoire de configuration du serveur satellite (expliqué plus loin).

Version NG 759 ou ultérieure.

- Depuis la version 759 NG, le serveur satellite et le serveur réseau d'entreprise prennent en charge IPv6 dans toutes les fonctionnalités avancées. Le code haute performance qui n'était auparavant pris en charge qu'en IPv4 s'applique désormais aussi à IPv6, ce qui améliore les *polling* capacités existantes.

Capacité

Il est difficile de spécifier la capacité maximale du Satellite Server, car elle dépend entièrement du serveur où il est exécuté, et du type de vérifications que vous voulez effectuer. Dans notre environnement de test, nous avons réussi à faire 500 vérifications ICMP et SNMP par seconde, mais cela dépend beaucoup des temps de réponse de l'appareil distant (ce n'est pas le même qui répond en 0,5ms que celui qui prend 2sec pour répondre). Dans des conditions théoriques idéales, on peut parler d'une surveillance d'environ 150.000 moniteurs avec un seul Satellite Server. En conditions réelles, nous avons testé dans des environnements plus ou moins contrôlés (réseaux locaux) environ 50 000 modules avec un serveur satellite dans un ordinateur matériel très discret (Intel i5, 2GHz, 4GB RAM).

S'il y a beaucoup de modules critiques, les performances peuvent être très affectées. Le timeout configuré doit également être pris en compte, car un seul contrôle critique est effectué par timeout. Si vous avez 1 000 modules critiques et que le timeout est configuré à 4

secondes, il faudrait 4 000 secondes pour exécuter toutes ces vérifications avec un seul thread.

Installation

Le Serveur Satellite est distribué sous forme de tarball (GNU/Linux®) ou de `.exe` (Windows®), il n'est donc pas nécessaire d'installer Perl ou toute autre bibliothèque supplémentaire. Le fonctionnement dans les versions Windows® ou Linux® est identique. Dans le cas de Windows®, il est installé comme un service, et dans le cas de Linux®, il est installé comme un démon système. Le fichier de configuration et les spécifications des deux sont identiques.

La version Linuxə du Serveur Satellite dépend de paquetages externes qui sont spécifiés dans la section correspondante de cette documentation.

Outil d'installation en ligne

E C'est un caractéristique spéciel de Pandora FMS. Vous avez besoin d'une licence Enterprise pour son utilisation. En tout cas, le paramètre d'installation obligtoire est l'adresse IP ou FQDN d'un server Pandora FMS Enterprise. Veuillez contacter l'équipe commercialle, demandez un devis ou vos questions sur les licences dans ce lien.

Cet outil est pris en charge par Rocky Linux 8.x, AlmaLinux 8.x et RHEL 8.x.

Exigences pour l'utilisation de l'outil d'installation en ligne (*online*) :

- Avoir connexion à internet.
- Avoir curl installé (il est inclus par défaut dans la plupart des distributions).
- Se conformer aux exigences minimales materièlles.
- Être utilisateur administrateur root.
- Avoir un système d'exploitation compatible.
- Si vous utilisez RHEL 8 il sera nécessaire de l'avoir activé préalablement avec une licence et abboné

aur répertoires standard.

Afin d'utiliser l'outil d'installation *online* accèdez tout simplement à la ligne de commande de votre fournisseur Cloud en tant qu'utilisateur administrateur root et exécutez :

```
export PANDORA_SERVER_IP='<PandoraServer IP or FQDN>' && curl -Ls
https://pfms.me/satellite-ent-deploy| bash
```

Installation personnalisée en utilisant l'outil d'installation *online :*

- `PANDORA_SERVER_IP` : Adresse IP ou FQDN du serveur Pandora FMS Enterprise vers lequel le Satellite server pointera. Paramètre obligatoire.
- `TZ` : Fuseau horaire du Satellite server. Paramètre optionnel.
- `SATELLITE_SERVER_PACKAGE` : URL personnalisée du package tarball d'installation du Satellite server. Paramètre optionnel.
- `SATELLITE_KEY` : Licence Satellite server pour activer automatiquement. Paramètre optionnel.
- `REMOTE_CONFIG` : Configuration distante. Paramètre optionnel, activé par défaut (valeur 1).
- `INSTALL_AGENT` : Paramètre optionnel, activé par défaut (valeur 1), il permet d'installer l'agent logiciel (toutes les variables de configuration de l'installateur en ligne de l'agent peuvent être utilisées).
- `VMWARE_DEPENDENCIES` : Optionnel, il permet d'installer les dépendences du *plugin* de VMwware®, désactivé par défaut (`0`).
- `ORACLE_DEPENDENCIES` : Optionnel, il permet d'installer des dépendences du plugin Oracle®, désactivé par défaut (`0`).
- `MSSQL_DEPENDENCIES` : Optionnel, il permet d'installer des dépendences du plugin MS SQL Server®, désactivé par défaut (`0`).
- `SKIP_KERNEL_OPTIMIZATIONS` : Désactiver l'optimisation du *kernel* recommandée, avancée, désactivée par défaut (`0`).

Exemple :

```
env TZ='Europe/Madrid' \
SATELLITE_KEY='SOPORTEDEV00RS0REB3M2T7ZHISO51IIQH52JISJ47VGHIRM...'\
  PANDORA_SERVER_IP='192.168.10.10' \
  REMOTE_CONFIG=1 \
  INSTALL_AGENT=1 \
  VMWARE_DEPENDENCIES=1 \
  ORACLE_DEPENDENCIES=1 \
  MSSQL_DEPENDENCIES=1 \
  SKIP_KERNEL_OPTIMIZATIONS=0 \
sh -c "$(curl -fsSL https://pfms.me/satellite-ent-deploy)"
```

Installation du serveur satellite sous Linux

Le système d'exploitation GNU/Linux recommandé est RedHat Enterprise (RHEL) 8 / Rocky Linux 8.

Vous devez installer Fping, Nmap et libnsl indépendamment et vous devez d'abord configurer le

dépôt EPEL, visitez le lien suivant:

```
https://docs.fedoraproject.org/en-US/epel/#_quickstart
```

et sélectionnez le système d'exploitation. Si vous utilisez Rocky Linux 8:

```
dnf config-manager --set-enabled powertools dnf install epel-release
```

Installez Perl à l'aide de la commande suivante:

```
dnf install perl
```

Dépendances de base du serveur satellite : PandoraWMIC (version 762 et ultérieure), Fping, Nmap et libnsl. Les dépendances pour Braa et PandoraWMIC sont jointes à l'installateur.

```
dnf install fping nmap libnsl
```

Une fois que le paquet contenant le Serveur Satellite a été téléchargé, il serait nécessaire d'aller dans le dossier de téléchargement avec les privilèges root et de décompresser le binaire :

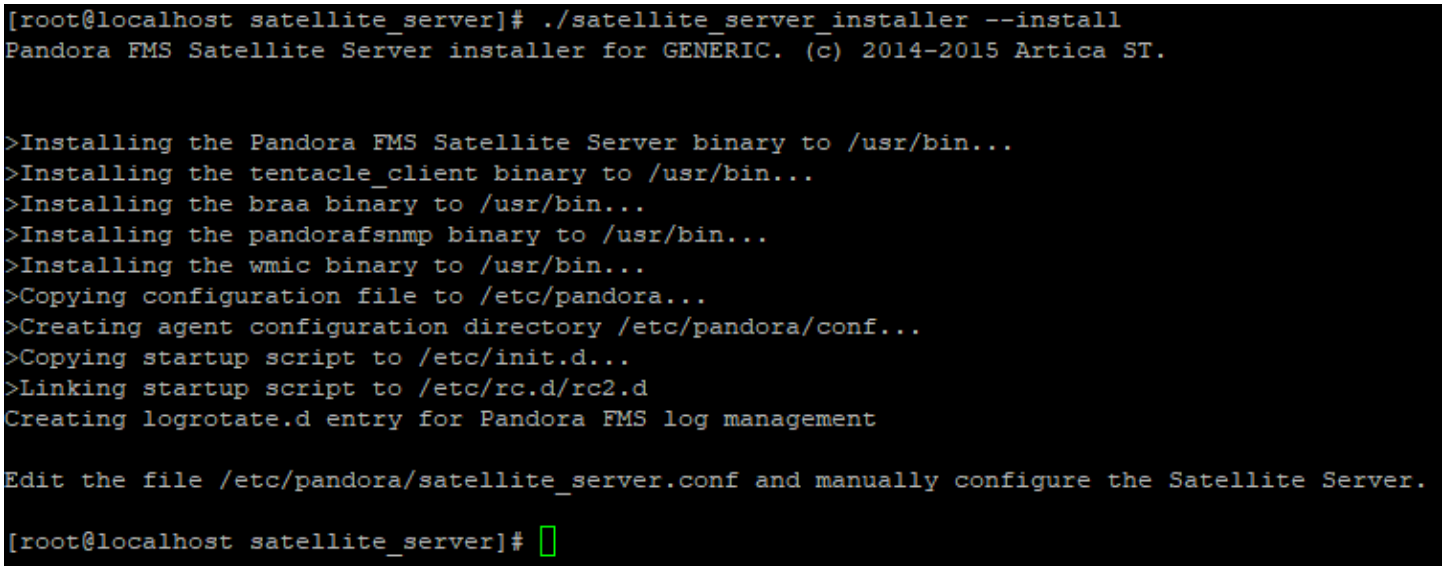
```
tar -xvzf pandorafms_satellite_server_X.XNG.XXX_x86_64.tar.gz
```

Un dossier appelé `satellite_server` sera alors généré. On doit entrer en tapant :

```
cd satellite_server/
```

Ensuite, pour installer le serveur satellite, il suffira d'exécuter la commande d'installation :

```
./satellite_server_installer --install
```



Une fois le processus terminé, il sera nécessaire d'éditer le fichier de configuration du satellite

situé dans :

```
/etc/pandora/satellite_server.conf
```

Dans des versions précéntes à la version 761, la licence doit être entrée manuellement dans l'option `pandora_licence`. Avec un éditeur de texte tel que VIM éditez le fichier, recherchez le jeton et décommentez. Entrez la licence du serveur Pandora FMS Enterprise.

Recherchez le jeton *server_ip*, et entrez l'adresse IP ou domaine du serveur Pandora FMS qui se connectera au serveur Satellite.

Après cela, sauvegardez le fichier et initiez le service, en exécutant ceci :

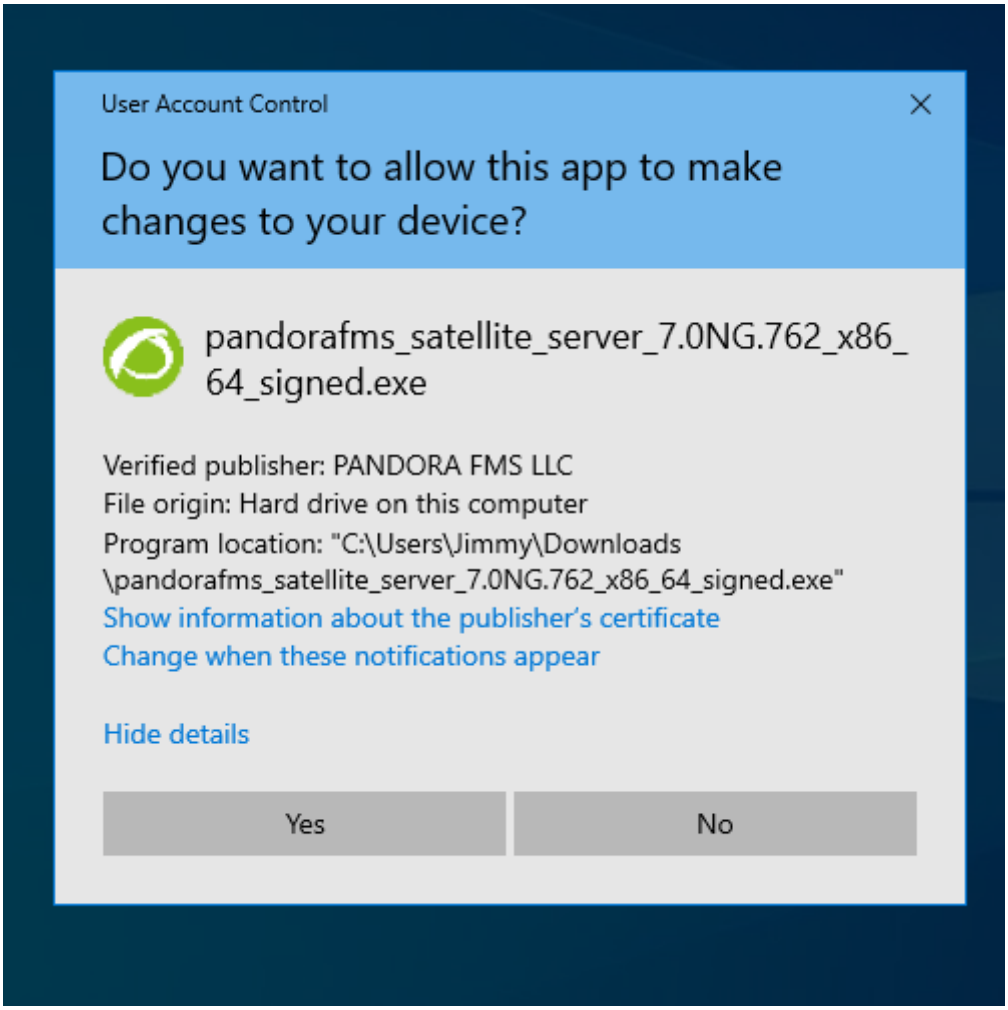
```
sudo /etc/init.d/satellite_serverd start
```

En cas d'erreur ou de dysfonctionnement, vous pouvez consulter le fichier de registre dans :

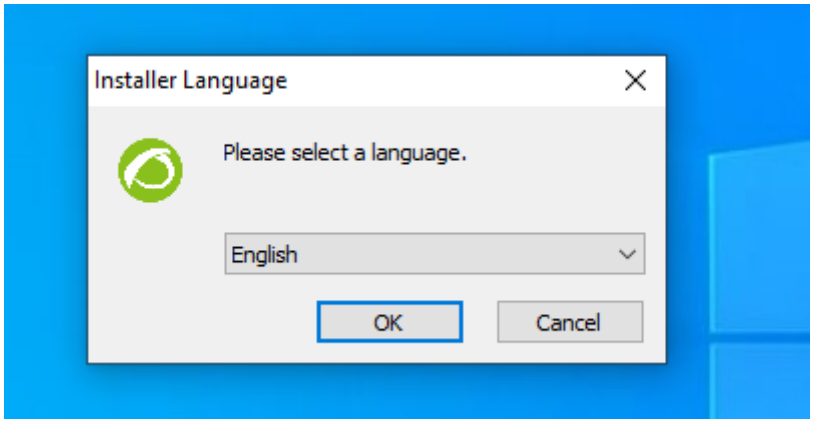
```
/var/log/satellite_server.log
```

Installation dans Windows

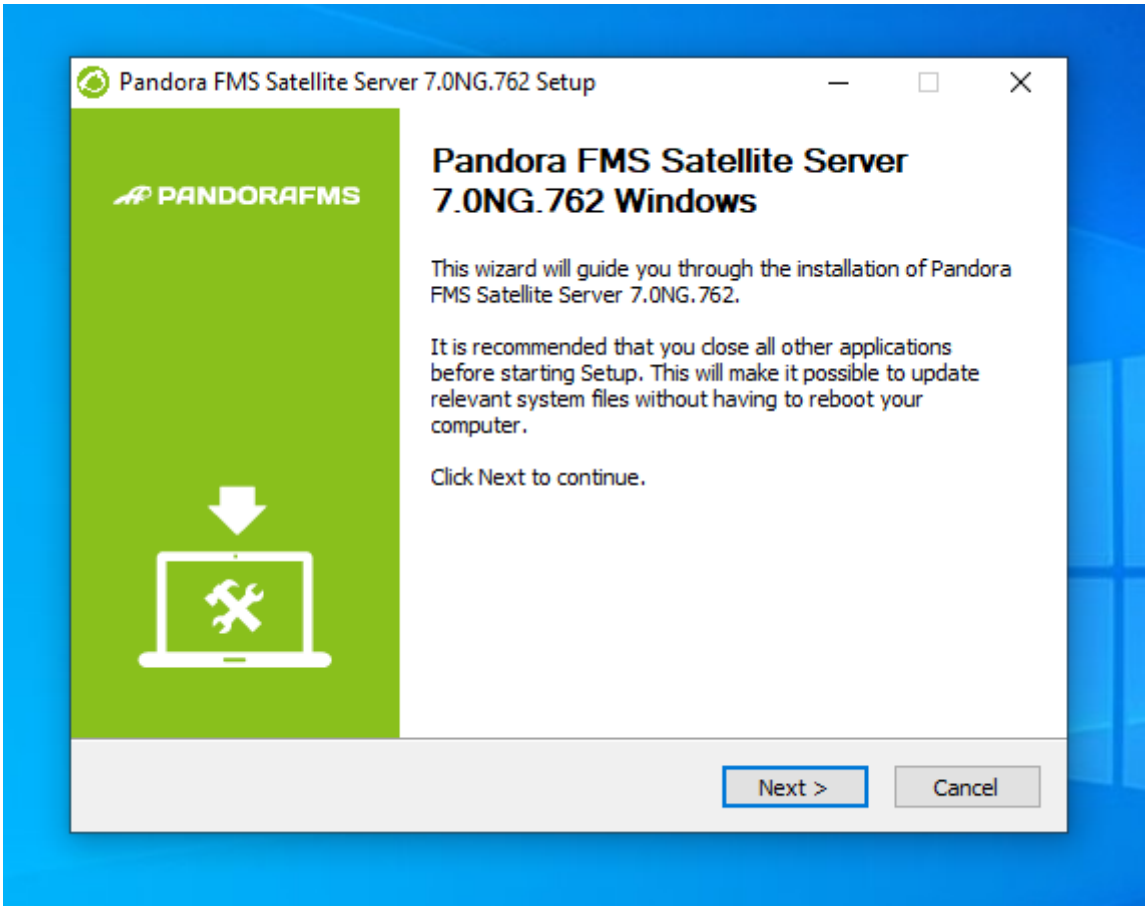
Exécutez le programme d'installation signé numériquement (version 762 et ultérieure), cliquez Yes (Oui) :



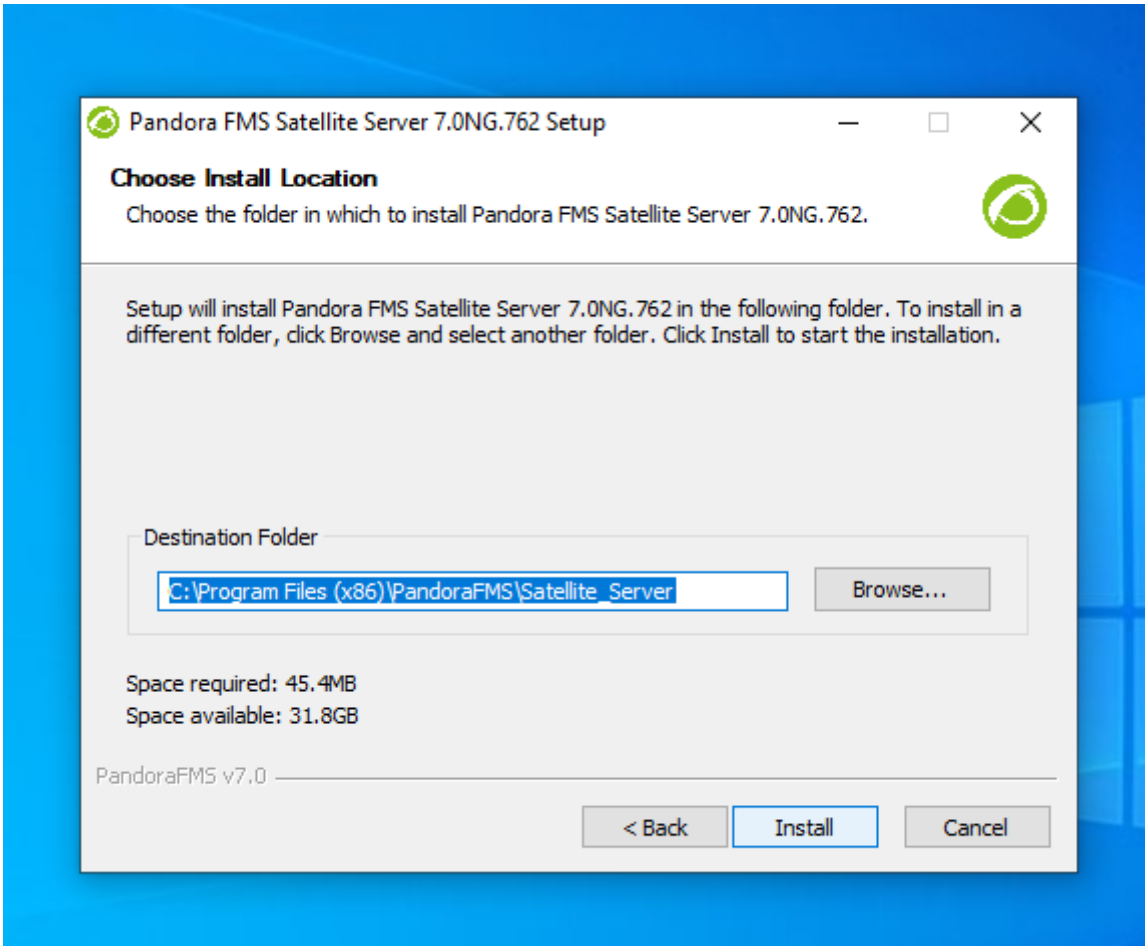
Choisissez la langue d'installation :



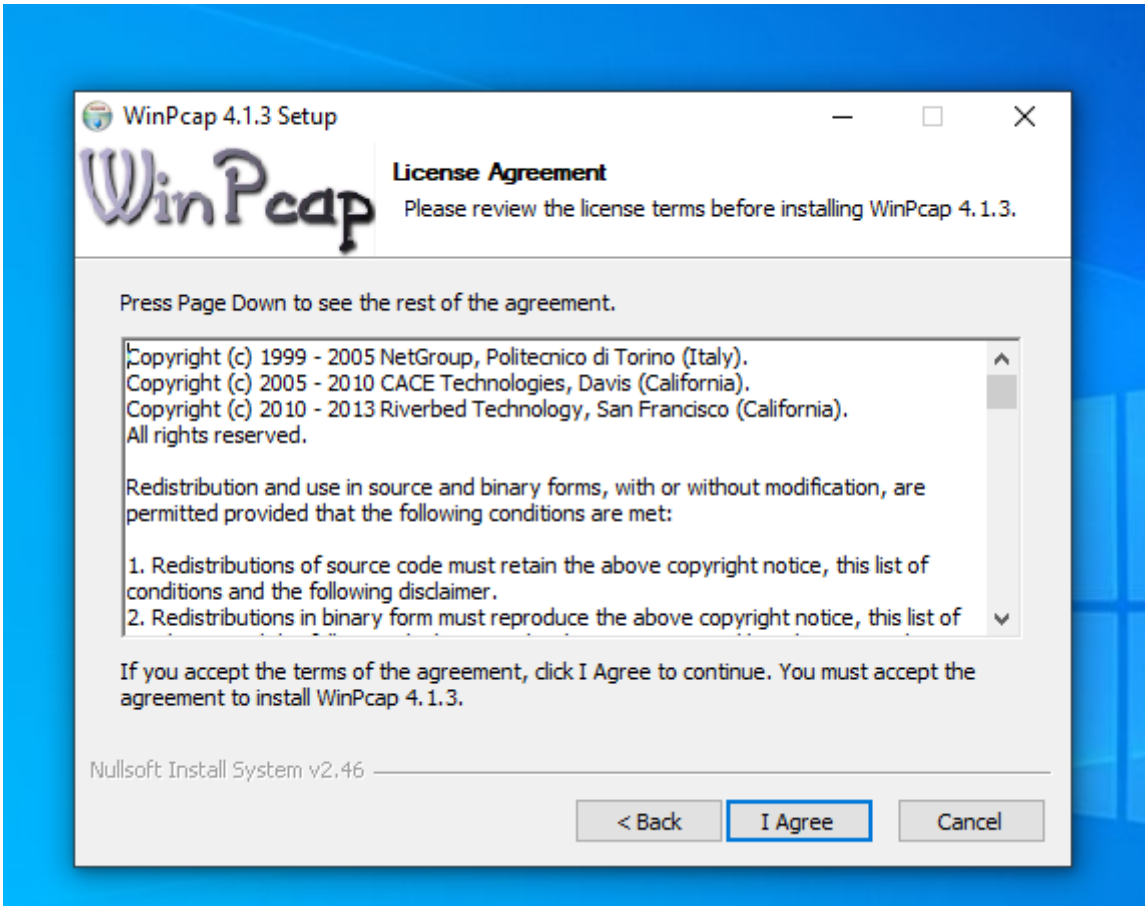
Cliquez sur Next (Suivant) :



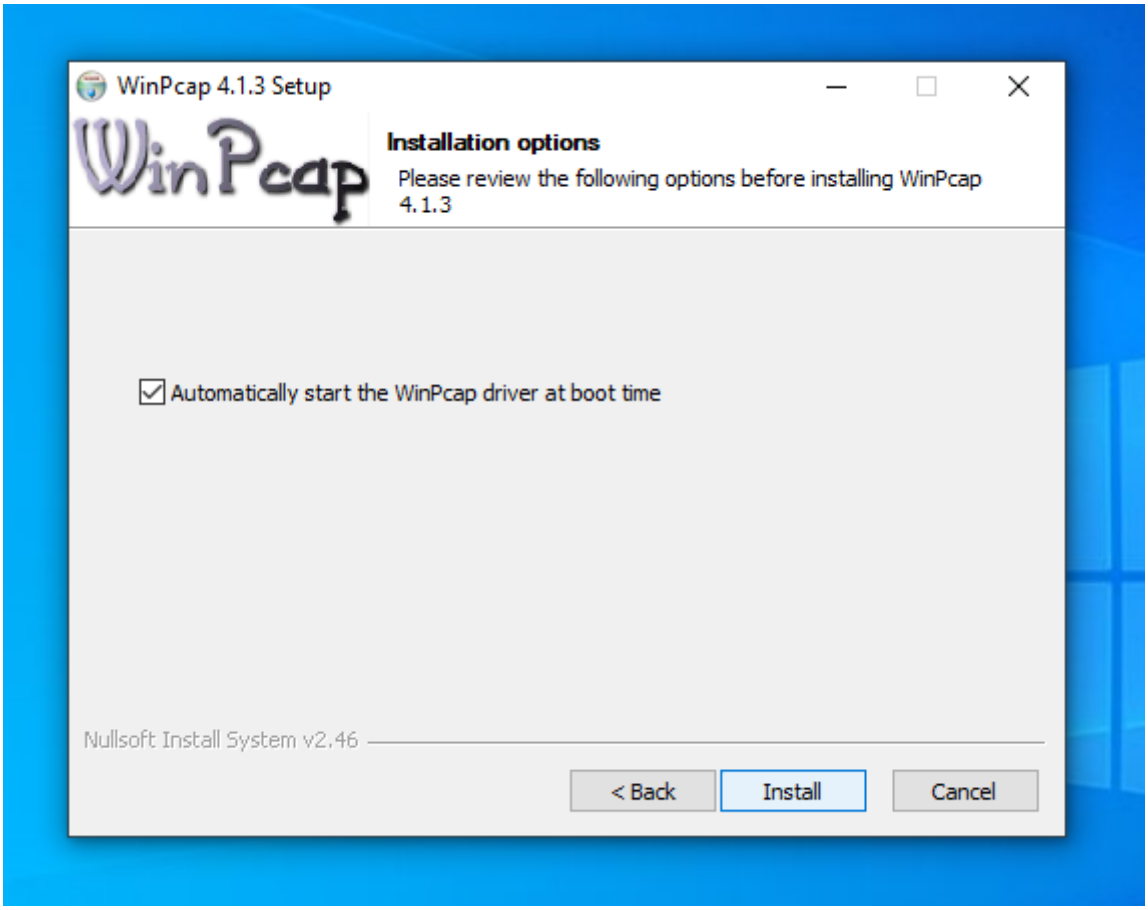
Choisissez où vous voulez installer le programme :



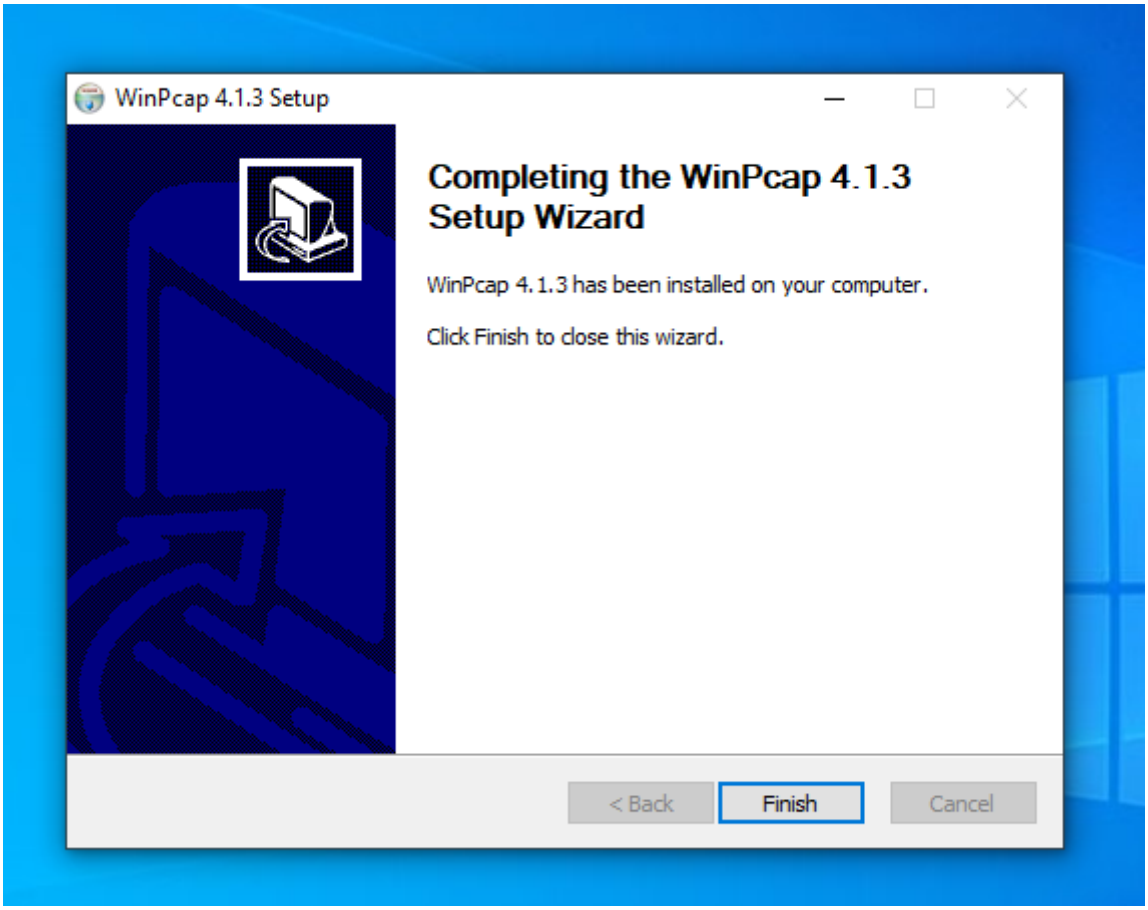
Vous devrez également installer WinPCap, la fenêtre d'installation apparaîtra à cette étape de l'installation.



Configurez le démarreur de WinPCap au démarrage de la machine :

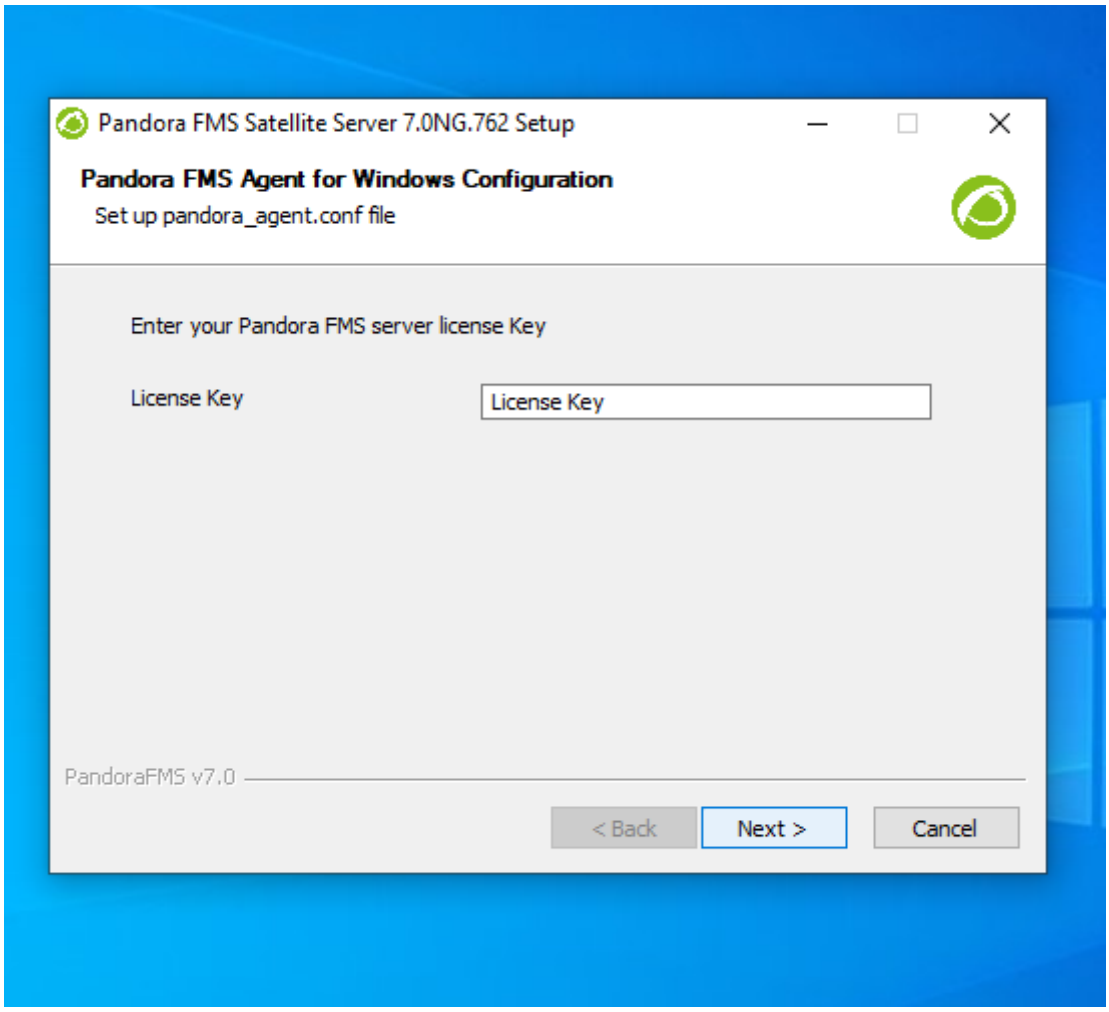


Une fois l'installation de WinCap terminée, vous verrez l'écran suivant :

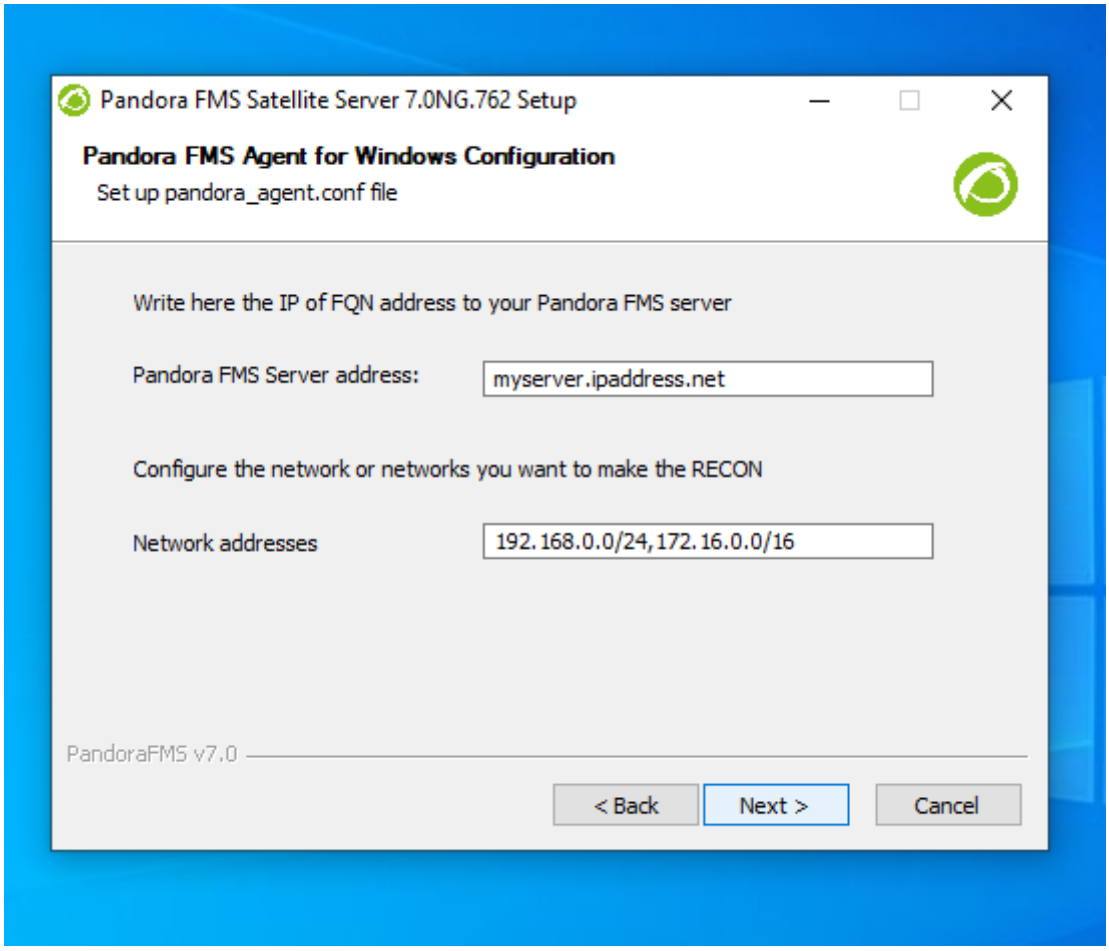


Il sera nécessaire d'introduire la clé de licence de Pandora FMS pour pouvoir continuer l'installation

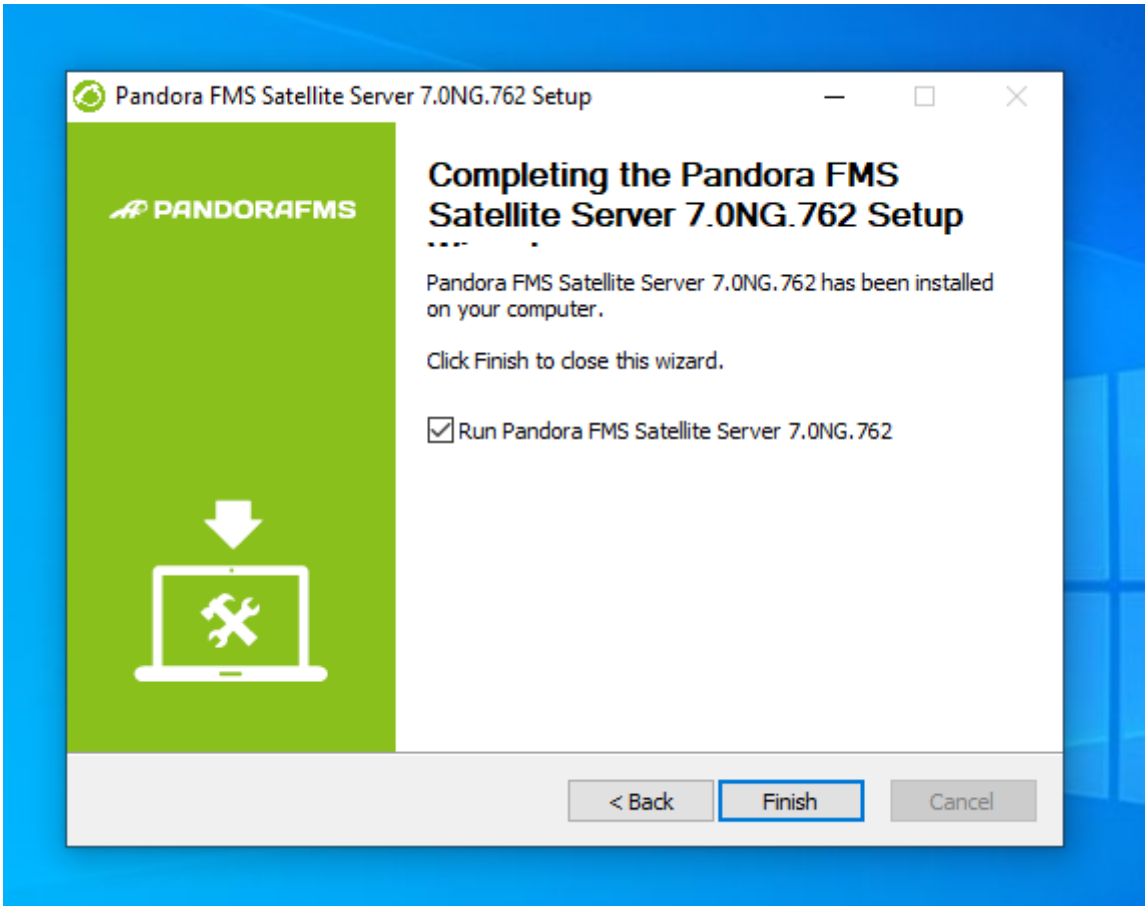
:



Dans la section suivante, configurez l'adresse du serveur FMS Pandora pour envoyer les données; vous pourriez définir les règles d'exploration du réseau pour le Satellite Server.



La machine devra être redémarrée pour que toutes les modifications soient prises en compte.



Une fois le processus terminé, vous pouvez démarrer et arrêter le service Satellite Server à partir du menu Start de MS Windows®.

Dépendant de l'année de votre version MS Windows vous devres installer quelque(s) des cettes librairies :

Microsoft Visual C++ Redistributable (derniers téléchargements avec support technique) :

- Visual Studio 2015, 2017, 2019, et 2022.
- Visual Studio 2013 (VC++ 12.0).
- Visual Studio 2012 (VC++ 11.0) Update 4.
- Visual Studio 2010 (VC++ 10.0) SP1 (sans support technique).
- Visual Studio 2008 (VC++ 9.0) SP1 (sans support technique).

Ils sont disponibles pour processeurs de 32 bits (X86), 64 bits (X64) et ARM64 dans le lien suivant :

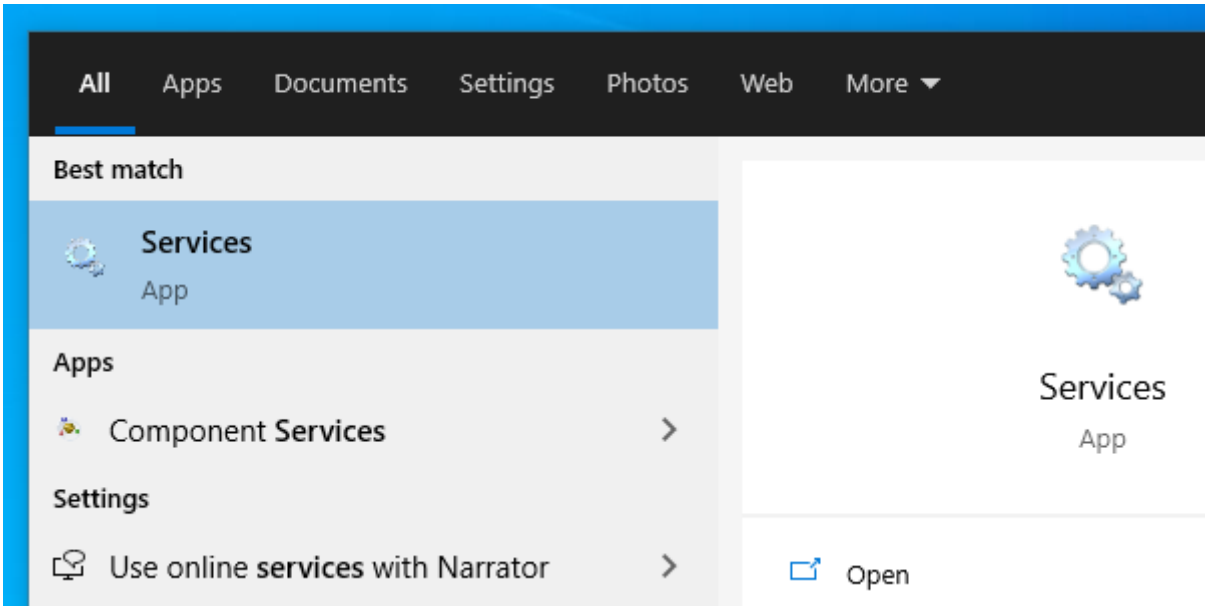
https://docs.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-170&viewFallbackFrom=msvc-170

Fonctionnement du module WMI dans certaines versions de Windows

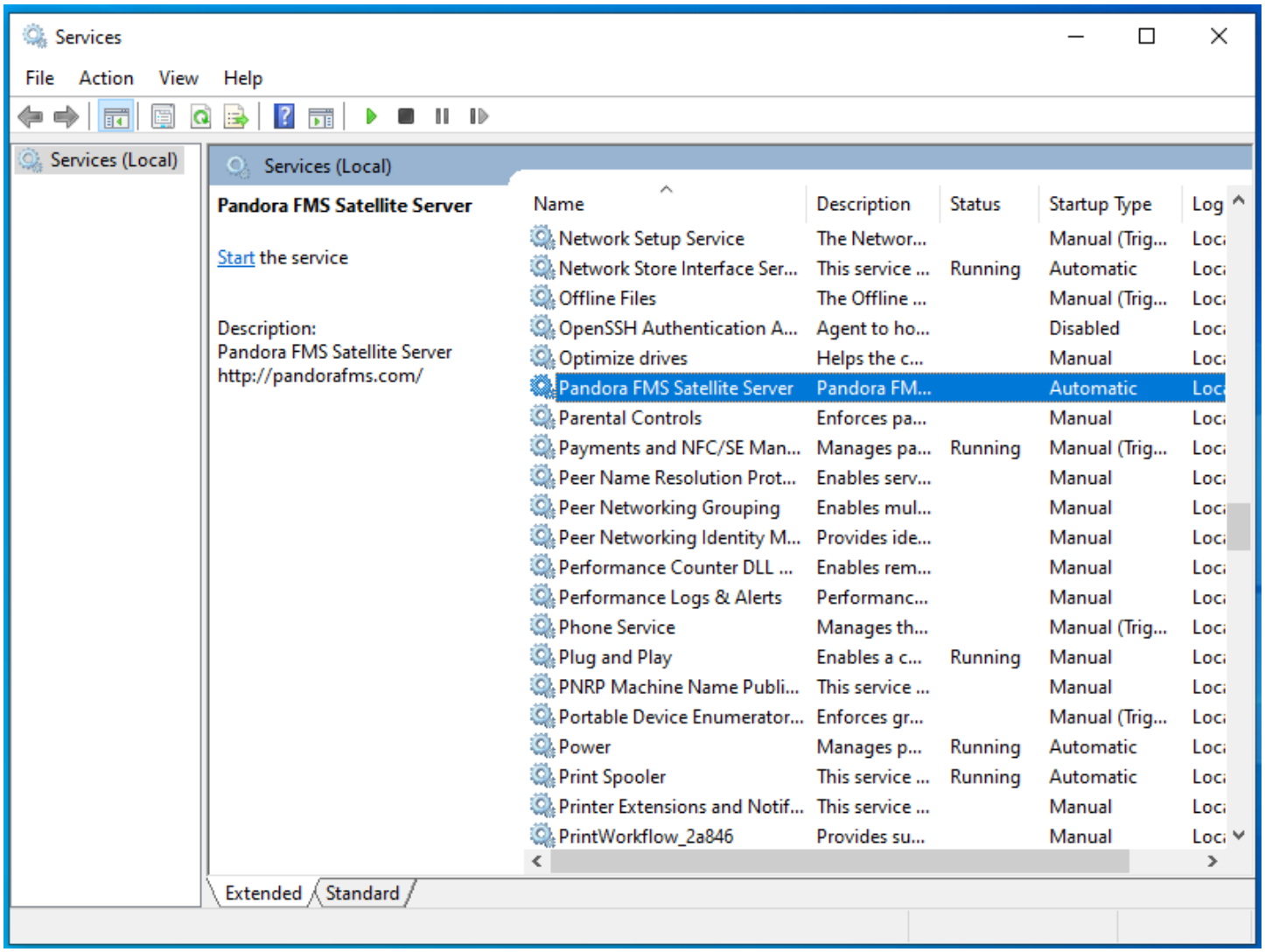
Pour des raisons de sécurité de Windows, certaines versions ont des utilisateurs limités avec qui effectuer des requêtes WMI à distance. Dans le cas où ces requêtes ne sont pas effectuées, la solution consiste à exécuter le service de serveur satellite en tant qu'utilisateur administrateur.

Le processus à suivre est le suivant :

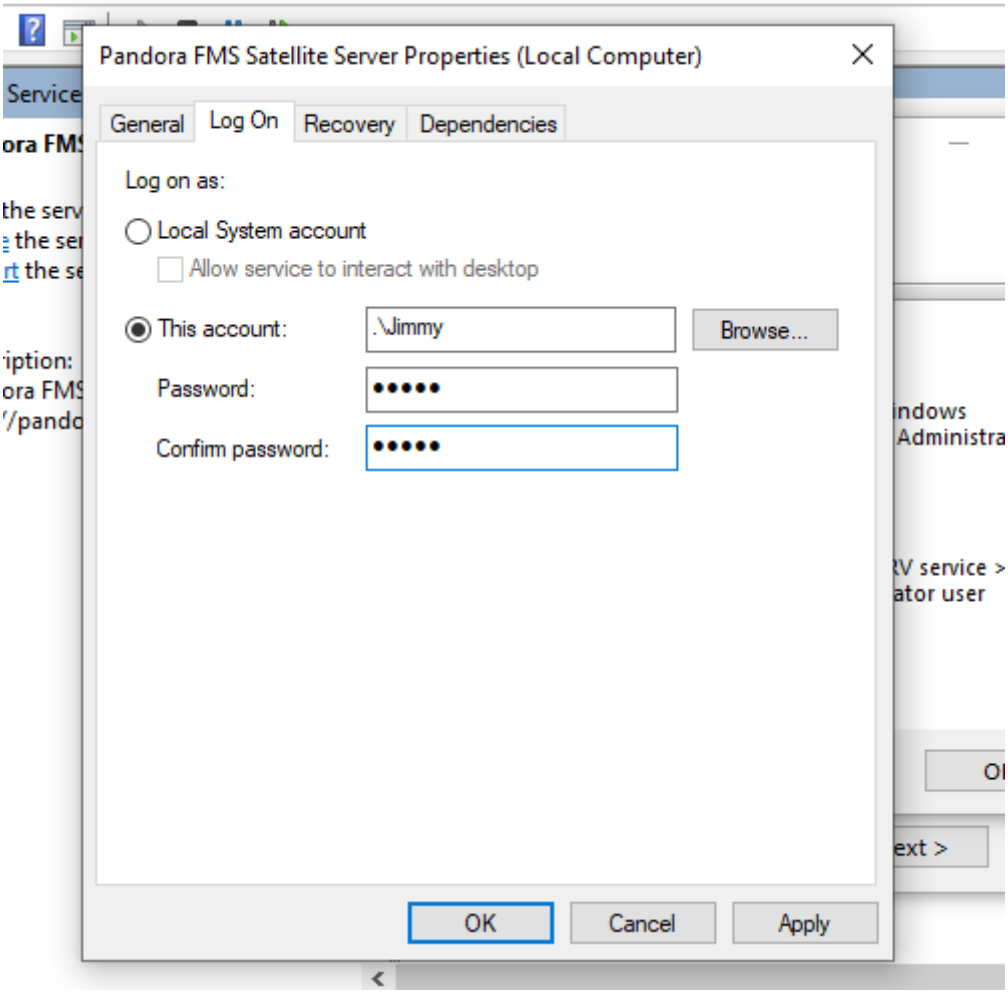
On ouvre les services :



Cliquez sur le service et allez dans Propriétés :



Dans la fenêtre de Login, sélectionez un compte avec des permissions d'administrateur et appliquez les modifications :



Le service doit être redémarré pour appliquer les modifications.

Configuration

Tous les paramètres nécessitant un délai d'attente sont spécifiés en secondes (par exemple 300 = 5 minutes).

Il est important de souligner que les intervalles de latence et de SNMP sont spécifiques au changement d'état. Dans le cas des contrôles booléens (état d'un port, état de la machine), le seuil qui définit le changement d'état est automatique ; dans le cas des valeurs numériques (latence, trafic réseau sur une interface, espace disque, CPU, etc. Par défaut, aucun seuil n'est défini ; cela doit être fait dans la définition du module.

agent_interval

```
agent_interval xxx
```

Par défaut, 300 secondes (5 minutes) ; créer des agents avec un intervalle de 5 minutes. Ce sera le temps après lequel il enverra les données au serveur, indépendamment du fait que les contrôles

effectués par le serveur satellite soient à un intervalle plus court. Si besoin et par défaut il crée des agents dans le serveur Pandora FMS correspondant selon le temps spécifié ici.

agent_threads

```
agent_threads xxx
```

Nombre de fils utilisés pour envoyer des fichiers de données XML.

xxxxxx_interval

```
xxxxxx_interval xxx
```

Il exécute toutes les vérifications (latence, SNMP, etc.) toutes les xxx secondes. Si les données collectées changent par rapport à la précédente, il les envoie à ce moment-là. Si c'est la même chose, il l'enverra quand l'intervalle de cet agent le commandera. Il est utile de faire des tests très intensifs, et de ne notifier qu'en cas de changement d'état.

xxxxx_retries

```
xxxxx_retries xxx
```

Nombre de tentatives xxx dans les contrôles (latence, SNMP, ping...)

xxxxx_timeout

```
xxxxx_timeout xxx
```

Délai d'attente en secondes pour les tests de type SNMP, de latence et de ping.

xxxxx_block

```
xxxxx_block xxx
```

Il force le serveur à exécuter les requêtes (latence, ping et snmp) en blocs de requêtes XXX. Plus le nombre est élevé (jusqu'à 500), plus vous aurez de puissance de traitement, mais au prix d'une latence accrue. Dans certains cas, il peut être pratique de diminuer ce nombre.

xxxxx_threads

```
xxxxx_threads n
```

Nombre de fils n affectés à chaque type de contrôle pour travailler simultanément. Elle dépend de la puissance (CPU et RAM) de la machine. Plus il y aura de threads, plus le système sera chargé, mais plus il aura de capacité de traitement. Lorsque le nombre de filets dépasse 20, selon le système, cela peut détériorer les performances.

log_file

```
log_file <path_file>
```

Il indique le fichier dans lequel le journal du serveur satellite est écrit, par défaut dans */var/log/satellite_server.log* .

recon_task

```
recon_task xxxxx[,yyyy]
```

Les adresses IP/réseaux utilisés pour l'auto-découverte, séparés par des virgules. Par exemple :

```
192.168.50.0/24,10.0.1.0/22,192.168.70.64/26
```

server_ip

```
server_ip <IP>
```

Adresse IP ou nom DNS du serveur FMS de Pandora auquel nous voulons envoyer les informations. Les informations sont envoyées par Tentacle, la communication avec le serveur doit donc être possible par le port Tentacle `41121/tcp`.

recon_mode

```
recon_mode <mode_discovery>
```

Mode d'auto-découverte (`<mode_discovery>`). Le système utilisera ces protocoles pour découvrir les systèmes :

- `recon_mode icmp` : Il va simplement vérifier si l'hôte est vivant (ping) et mesurer le temps de latence.
- `recon_mode snmp` : S'il est capable de communiquer par SNMP (v1 et v2 seulement), il cherchera toutes les interfaces réseau, et enlèvera le trafic de toutes, ainsi que leur état de fonctionnement, en plus du nom de l'appareil et de l'emplacement. Il va essayer de se connecter avec les différentes communautés fournies dans le fichier de configuration pour se connecter. Pour utiliser SNMP v3

dont la reconnaissance est nécessaire, cliquez sur le lien pour apprendre comment configurer les identifiants d'accès connus.
- `recon_mode wmi` : Similaire au cas précédent, dans ce cas montrant la charge CPU, la mémoire et les disques (tous disponibles).

recon_community

```
recon_community <aaa>,<bbb>,<ccc>...
```

Il spécifie une liste de communautés SNMP <xxx> à utiliser dans la découverte SNMP, séparées par des virgules. Il utilisera cette liste dans la découverte SNMP : pour chaque IP trouvée, il essaiera de voir si elle répond à l'une de ces communautés.

wmi_auth

```
wmi_auth Administrator%password[,user%pass]
```

Il spécifie une liste de paires d'identifiants d'utilisateur, chacune sous le format >nom d'utilisateur>%<mot de passe> et séparés par virgules.

Par exemple : *admin%1234,super%qwerty*. Il utilisera cette liste dans la navigation WMI. Pour chaque IP trouvé, il essaiera de voir s'il répond à l'une de ces combinaisons.

wmi_ntlmv2

```
wmi_ntlmv2 [0|1]
```

Il active (1) ou désactive (0) l'autentification avec le protocole NTLMv2 pour WMI.

agent_conf_dir

```
agent_conf_dir <path>
```

Chemin (<path>) du répertoire qui crée es stocke automatiquement les fichiers de configuration de chaque agent créé par le Serveur Satellite. Par défaut `/etc/pandora/conf`. Vous pouvez aussi créer les agents manuellement.

group

```
''group <group_name>''
```

Il définit le nom du groupe <group_mame> par défaut des agents créés par le Serveur Satellite. Par exemple, “ Servers ”.

daemon

```
daemon [1|0]
```

Si sa valeur est 1, le démon est exécuté en arrière-plan (valeur par défaut).

hostfile

```
hostfile <path_filename>
```

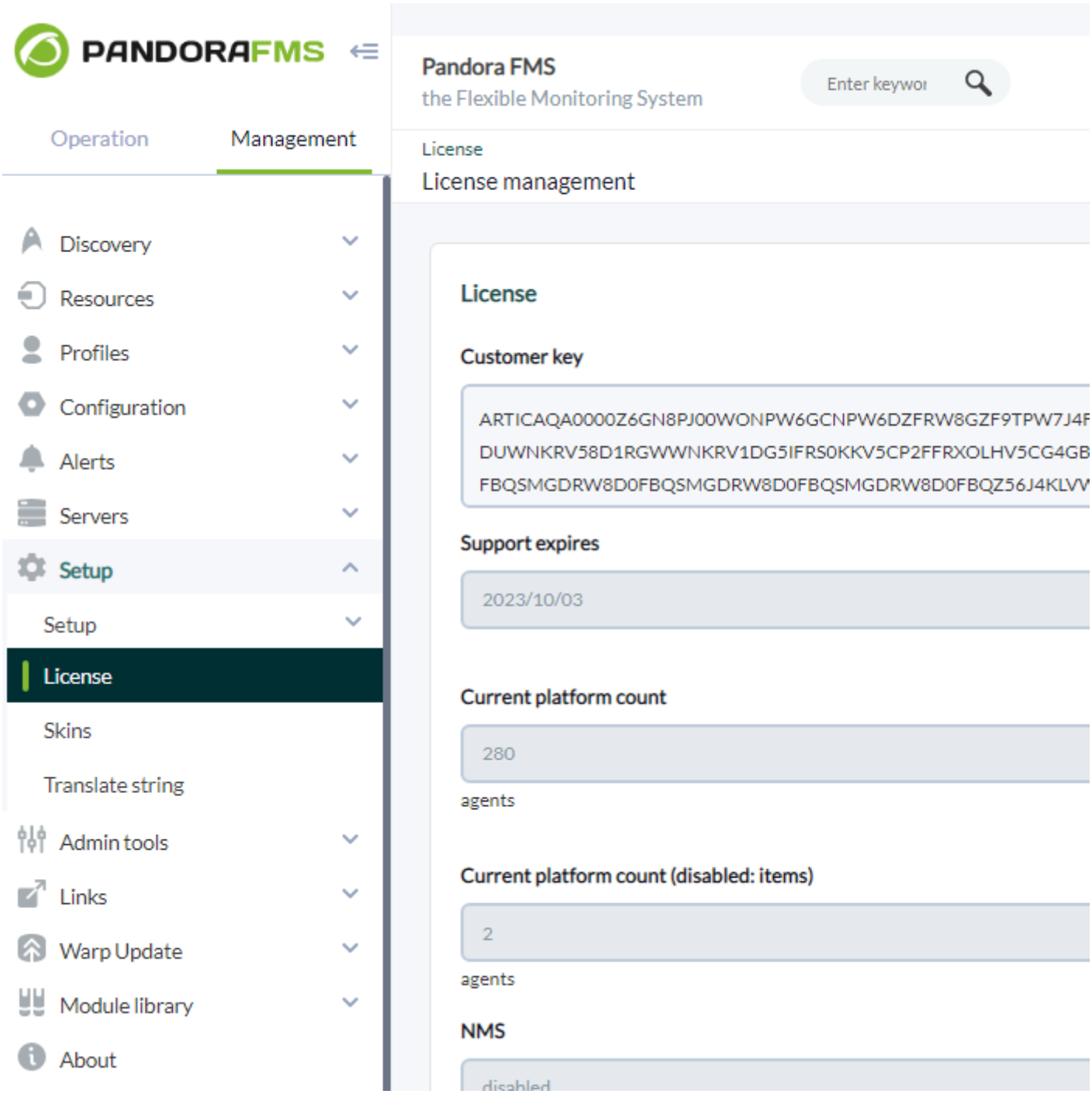
C'est une méthode alternative/complémentaire à l'analyse d'un réseau à la recherche d'hôtes. Dans ce fichier (<path_filename>), dans chaque ligne il y a une adresse. Alternativement, vous pouvez passer dans la même ligne le nom d'hôte suivi de l'IP, afin de créer l'agent avec ce nom et utiliser cette IP pour les modules (ex : `192.168.0.2 <hostname>`). Il est nécessaire que lorsque vous faites une requête avec fping aux adresses son résultat soit en ligne pour qu'elles soient valides.

pandora_license

Depuis la version 761 et ultérieures le serveur Satellite est
licencié automatiquement et ce jeton est obsolète.

```
pandora_license xxxxxxx
```

Il écrit et stocke la licence du serveur Pandora FMS, comme indiqué dans la section Setup → License de votre console Pandora FMS.



Vous pouvez utiliser la même licence dans autant de serveurs Satellite que vous le souhaitez, puisque le nombre total d'agents utilisant la licence est vérifié dans le serveur FMS Pandora, et non dans le Satellite.

remote_config

```
remote_config [1|0]
```

Il active par défaut la configuration à distance dans les agents détectés, nécessaire si vous voulez les gérer depuis la console après les avoir détectés. Il active également la configuration à distance du serveur satellite lui-même.

temporal_min_size

```
temporal_min_size xxx
```

Si l'espace libre (en mégaoctets) sur la partition où se trouve le répertoire temporaire est inférieur à cette valeur, aucun autre paquet de données n'est généré. Cela empêche le disque de se remplir si, pour une raison quelconque, la connexion au serveur est perdue pendant une période prolongée.

xml_buffer

```
xml_buffer [0|1]
```

Par défaut 0. Étant à 1, l'agent conservera les données XML qu'il n'a pas pu envoyer pour réessayer plus tard.

Sous Unix, si vous êtes dans un environnement sûr, vous devriez envisager de changer le répertoire temporaire, car `/tmp` a des droits d'écriture pour tous les utilisateurs.

snmp_version

```
snmp_version xx
```

Version SNMP à utiliser par défaut 1. Pour utiliser SNMP v3 consultez ce lien sur la façon de configurer les informations d'identification d'accès connues.

Certains modules pourraient cesser de fonctionner si cette valeur est modifiée.

braa

```
braa <path>
```

Chemin `<path>` vers le binaire braa (`/usr/bin/braa` par défaut).

fping

```
fping <path>
```

Chemin <path> d'accès au binaire fping (/usr/sbin/fping par défaut).

fsnmp

```
fsnmp <path>
```

Chemin <path> vers le binaire SNMP (/usr/bin/pandorafsnmp par défaut).

latency_packets

```
latency_packets xxx
```

Nombre de paquets xxx ICMP envoyés par demande de latence.

nmap

```
nmap <path>
```

Chemin <path> vers le binaire Nmap (/usr/bin/nmap par défaut).

nmap_timing_template

```
nmap_timing_template x
```

Une valeur xxx qui spécifie à quel point nmap doit être agressif, de 1 à 5. 1 signifie plus lent mais plus fiable, 5 signifie plus rapide mais moins fiable. La valeur par défaut est 2.

ping_packets

```
ping_packets xxx
```

Nombre de paquets ICMP envoyés par ping.

recon_enabled

```
recon_enabled [0|1]
```

Active (1) ou désactive (0) l'auto-découverte de l'équipement.

recon_timing_template

```
recon_timing_template xxx
```

Tout comme nmap_timing_template, mais appliqué aux scans réseau.

server_port

```
server_port xxxxx
```

Port du serveur Tentacle.

server_name

```
server_name xxxxx
```

Nom que vous voulez donner au serveur Satellite (par défaut il prend le nom d'hôte de la machine)

server_path

```
server_path <path>
```

Chemin <path> où les fichiers XML sont copiés si le transfer_mode est en local (par défaut `/var/spool/pandora/data_in`).

server_opts

Les paramètres du serveur qui sont transmis à Tentacle.

transfer_mode

```
transfer_mode [tentacle|local]
```

Mode de transfert de fichiers. Peut être seulement Tentacle ou local (Tentacle par défaut).

Serveur secondaire

```
secondary_mode [on_error|always]
```

Un type particulier de paramètre de configuration générale est la définition d'un serveur

secondaire. Cela vous permet de définir un serveur auquel les données sont envoyées, en plus du serveur défini en standard. Le mode serveur secondaire fonctionne de deux façons :

- on_error : envoie des données au serveur secondaire seulement s'il ne peut pas les envoyer au primaire.
- always : envoie toujours des données au serveur secondaire, qu'il puisse ou non contacter le serveur primaire.

Exemple de configuration :

```
secondary_server_ip     192.168.1.123
secondary_server_path   /var/spool/pandora/data_in
secondary_mode          on_error
secondary_transfer_mode tentacle
secondary_server_port   41121
```

snmp_verify

```
snmp_verify [0|1]
```

Active (1) ou désactive (0) la vérification des modules SNMPv1 qui font Braa tomber en panne en temps réel. Ces modules seront rejetés et cesseront d'être exécutés. Voyez snmp2_verify et nmp3_verify.

snmp2_verify

```
snmp2_verify [0|1]
```

Il active (1) ou désactive (0) la vérification des modules SNMPv2 qui font Braa échouer le braa en temps réel. Ces modules seront rejetés et cesseront d'être exécutés. voyez snmp_verify et snmp3_verify.

La vérification des modules SNMP version 2 peut être très
lente !

snmp3_verify

```
snmp3_verify [0|1]
```

Il active (1) ou désactive (0) la vérification des modules SNMPv3 qui font échouer le braa en temps réel. Ces modules seront rejetés et cesseront d'être exécutés. Voyez aussi snmp_verify et snmp2_verify.

Pour utiliser SNMPv3, voir ce lien sur la façon de configurer les informations d'identification d'accès connues.

snmp3_seclevel

Niveau de sécurité utilisé pour les messages SNMPv3 (`noauth`, `authnopriv` ou `authpriv`).

Pour utiliser SNMPv3, voir ce lien sur la façon de configurer les informations d'identification d'accès connues.

snmp3_secname

Nom de sécurité utilisé pour les messages SNMPv3.

Pour utiliser SNMPv3, voir ce lien sur la façon de configurer les informations d'identification d'accès connues.

snmp3_authproto

Protocole d'authentification (`md5` ou `sha`) pour les demandes SNMPv3 authentifiées.

Pour utiliser SNMPv3, voir ce lien sur la façon de configurer les informations d'identification d'accès connues.

snmp3_authpass

Mot de passe d'authentification pour la demande SNMPv3 authentifiée.

Pour utiliser SNMPv3, voir ce lien sur la façon de configurer les informations d'identification d'accès connues.

snmp3_privproto

Protocole de confidentialité (`des` ou `aes`) pour les requêtes SNMPv3 cryptées.

Pour utiliser SNMPv3, voir ce lien sur la façon de configurer les informations d'identification d'accès connues.

snmp3_privpass

Mot de passe de confidentialité pour les messages SNMPv3 cryptés.

Pour utiliser SNMPv3, voir ce lien sur la façon de configurer les informations d'identification d'accès connues.

startup_delay

```
starup_delay xxx
```

il attend xxx secondes avant d'envoyer des fichiers de données pour la première fois.

temporal

```
temporal <directory>
```

Répertoire temporaire où sont créés les fichiers XML, par défaut `/tmp`.

tentacle_client

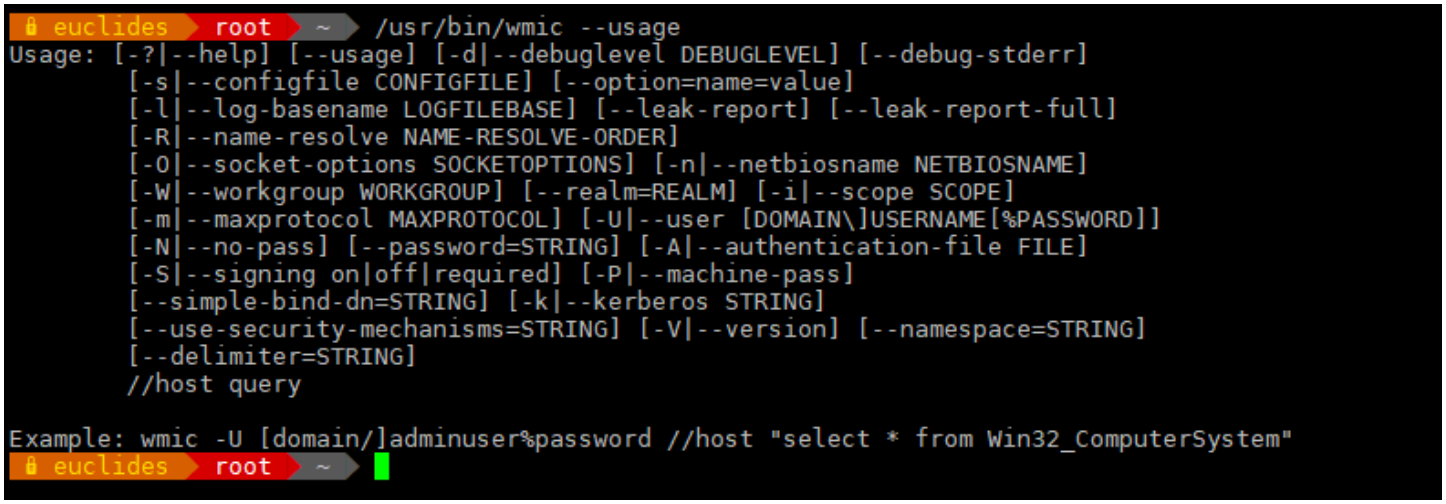
```
tentacle_client <path>
```

Chemin <path> d'accès au binaire du client Tentacle (`/usr/bin/tentacle_client` par défaut).

wmi_client

```
wmi_client <path>
```

Chemin <path> vers le binaire wmi_client (`/usr/bin/wmic` par défaut).



snmp_blacklist

```
snmp_blacklist <path>
```

Chemin <path> d'accès à la liste noire des modules SNMP (`/etc/pandora/satellite_server.blacklist` par défaut).

add_host

```
 add_host <adr_IP> [agent_name]
```

Il ajoute l'hôte donné `[agent_name]` à la liste des agents surveillés. Vous pouvez spécifier le nom de l'agent après l'adresse IP (<IP_addr>). Plusieurs hôtes peuvent être ajoutés, un par ligne. Par exemple :

```
add_host 192.168.0.1
add_host 192.168.0.2 localhost.localdomain
```

ignore_host

```
ignore_host <agent_name>
```

Il supprime l'hôte donné de la liste des agents surveillés, même s'il est trouvé dans une analyse du réseau par une tâche de reconnaissance. L'hôte doit être identifié par le nom de l'agent. Plusieurs hôtes peuvent être ignorés, un par ligne. Par exemple :

```
 ignore host 192.168.0.1
 ignore host localhost.localdomain
```

keepalive

```
keepalive xxx
```

Le serveur satellite rapporte son état et vérifie les changements dans la configuration à distance (des agents et de lui-même) toutes les secondes “ keepalive ”. Par défaut à 30 secondes.

credential_pass

```
credential_pass xxx
```

Mot de passe utilisé pour crypter les mots de passe des boîtes de justificatifs. Doit être le même que celui défini dans la console Pandora FMS. Par défaut, le nom d'hôte est utilisé.

timeout_bin

```
timeout_bin <path>
```

S'il est défini, le programme de timeout (généralement `/usr/bin/timeout`) sera utilisé lors de l'appel du client Tentacle.

timeout_seconds

```
timeout_seconds xxx
```

Délai d'attente en secondes pour le programme de timeout. Le paramètre timeout_bin doit être réglé.

proxy_traps_to

```
proxy_traps_to <dir_IP[:port]>
```

Il redirige les traps SNMP reçus par le serveur satellite vers l'adresse (et le port) spécifié. Par défaut, le port 162 est utilisé.

proxy_tentacle_from

```
proxy_tentacle_from <dir_IP[:port]>
```

Il redirige les données reçues par le serveur Tentacle à partir de l'adresse (et du port) spécifié. La valeur par défaut est le port 41121.

proxy_tentacle_to

```
Proxy_tentacle_to <dir_IP[:port]>
```

Redirige les requêtes des clients Tentacle reçues par le serveur satellite vers l'adresse (et le port) spécifié. Le port 41121 est utilisé par défaut.

Cette option peut entrer en conflit avec la configuration de l'agent distant.

Cela se produit si vous avez l'intention d'utiliser le Serveur Satellite comme proxy pour certains agents logiciels et de les surveiller à leur tour à distance à partir du Serveur Satellite lui-même (ICMP, SNMP, etc.) et la configuration à distance est activée dans les deux cas.

Dans cette situation, vous devez soit utiliser des agents différents pour les contrôles effectués (c'est-à-dire avec un `agent_name` différent), soit laisser la configuration distante activée uniquement dans l'un des deux (Satellite Server ou agents logiciels).

dynamic_inc

```
dynamic_inc [0|1]
```

Avec valeur 1 il déplace les modules dynamiques découverts de manière automatique (SNMP, WMI...) vers des fichiers séparés afin qu'ils n'interfèrent pas avec la configuration à distance des agents.

vlan_cache_enabled

```
vlan_cache_enabled [0|1]
```

Il active (1) ou désactive (0) le cache du VLAN des hôtes découverts.

verbosity

```
verbosity <0-10>
```

Niveau de verbosité du journal, où 10 est le niveau d'information le plus détaillé.

agents_blacklist_icmp

Version NG 713 ou supérieure.

```
agents_blacklist_icmp 10.0.0.0/24[,8.8.8.8/30]
```

Liste d'exclusion de vérifications CIPD. Ce champ peut être configuré avec une liste d'IPs utilisant la notation CIDR pour empêcher l'exécution de modules de type ICMP. Il est possible de spécifier plusieurs sous-réseaux en les séparant par des virgules.

agents_blacklist_icmp

Version NG 713 ou supérieure.

```
agents_blacklist_snmp 10.0.0.0/24[,8.8.8.8/30]
```

Liste noire des contrôles SNMP. Ce champ peut être configuré avec une liste d'IPs utilisant la notation CIDR pour éviter que des modules de type SNMP soient exécutés. Il est possible de spécifier plusieurs sous-réseaux en les séparant par des virgules.

agents_blacklist_wmi

Version NG 713 ou supérieure.

```
agents_blacklist_wmi 10.0.0.0/24[,8.8.8.8/30]
```

Liste de contrôle de la WMI. Ce champ peut être configuré avec une liste d'IPs utilisant la notation CIDR pour empêcher l'exécution de modules de type WMI. Il est possible de spécifier plusieurs sous-réseaux en les séparant par des virgules.

general_gis_exec

Version NG 713 ou supérieure.

```
general_gis_exec xxx
```

En activant cette option, un script sera utilisé pour fournir le positionnement SIG à tous les agents détectés par le serveur satellite. Le script doit avoir des permissions d'exécution et imprimer à l'écran les coordonnées avec le format<longitude>,<latitude>[,<altitude>]. Le troisième

paramètre, la altitude, est optionnel.

forced_add

Si la valeur est 1, les *hosts* ajoutés manuellement (via host_file ou add_host) seront toujours créés, même s'ils ne répondent pas au ping, avec un fichier de configuration sans modules.

Création d'agents dans le serveur satellite

Il y a trois façons de créer les agents dans le Satelllite Server : Recon Task, fichier satellite_hosts.txt ou de façon manuelle en créant le .conf des agents à surveiller.

Création d'agents par le Recon Task

La création d'agents par Recon Task est la plus utilisée par les utilisateurs de Pandora FMS. Pour le réaliser, nous devons avoir accès au fichier de configuration du Serveur Satellite et configurer les paramètres suivants :

- recon_community : Vous devez spécifier une liste de communautés SNMP à utiliser dans la découverte SNMP, séparées par des virgules (dans le cas d'une reconnaissance de type SNMP).
- recon_enabled : Il doit être réglé sur 1 pour activer la tâche de reconnaissance du serveur satellite.
- recon_interval : Intervalle de temps où le réseau que nous voulons est scanné, en secondes (par défaut 604800 secondes, 7 jours).
- recon_mode : Mode pour effectuer la tâche de reconnaissance (SNMP,ICMP,WMI), séparés par des virgules.
- recon_task : Liste des réseaux sur lesquels on veut faire la reconnaissance, séparés par des virgules.
- recon_timing_template : Une valeur qui spécifie à quel point nmap doit être agressif, de 1 à 5. 1 signifie plus lent mais plus fiable ; 5 signifie plus rapide mais moins fiable (par défaut 3).

Un exemple de la réalisation de Recon Task est :

```
recon_community public
recon_enabled 1
recon_interval 604800
recon_mode icmp,snmp,wmi
recon_task 192.168.0.0/24,192.168.1.0/24
recon_timing_template 3
```

Une fois les données configurées, exécutez le serveur satellite à l'aide de la commande :

```
/etc/init.d/satellite_serverd start
```

Les agents dont les fichiers de configuration ne contiennent aucun module

seront ignorés par le serveur Satellite.

Création d'agents par fichier

Tout d'abord, pour créer un agent en utilisant le fichier satellite_hosts.txt, allez dans le fichier de configuration du Satellite Server et il faut défaire la ligne :

```
host_file /etc/pandora/satellite_hosts.txt
```

Ensuite, créez le fichier satellite_hosts.txt avec l'IP des hôtes que vous voulez créer, en mettant l'IP et le nom de l'agent à créer :

```
192.168.10.5 Server5
192.168.10.6 Server6
192.168.10.7 Server7
```

Pour que les agents avec ces IP soient créés, il faut qu'ils
répondent à l'appel fping, sinon ils ne seront pas créés.

Une fois les données configurées, nous démarrons le serveur satellite au moyen de la commande :

```
/etc/init.d/satellite_serverd start
```

La lecture du fichier indiqué est faite tous les recon_interval secondes.

Création manuelle d'agents

Dans le répertoire `/etc/pandora/conf` les fichiers de configuration des nouveaux agents y sont stockés. Ouvrez une fenêtre terminale et allez vers le dossie :

```
cd /etc/pandora/conf
```

Une fois situé dans ce répertoire, créez un fichier `.conf` , par exemple “ fichier.conf ” et remplissez manuellement les champs suivants :

- agent_name : Le nom de l'agent.
- agent_alias : Le pseudo de l'agent.
- address : L'IP de l'élément à surveiller.
- group : Groupe auquel vous voulez assigner l'agent.
- gis_exec : Script de positionnement (optionnel). Si vous l'utilisez, écrasez l'emplacement fourni par le paramètre general_gis_exec du serveur satellite.
- La suite serait de créer les modules que vous voulez surveiller dans l'agent.

Un exemple serait :

```
agent_name Example

agent_alias Ceci est un exemple
adress 127.0.0.1
group Serveurs

 module_begin
module_name Ping
module_ping
module_end

 module_begin
module_name Latency
module_latency
module_end
```

Une fois les données configurées, démarrez le serveur satellite à l'aide de la commande

```
/etc/init.d/satellite_serverd start
```

Suppression de l'agent du serveur satellite

Il y a plusieurs cas de suppression d'agent de serveur satellite : suppression totale d'agent ou suppression partielle d'agent.

Au début, fiates une sauvegarde de tous les dossiers et
leurs fichiers avan ce procédure.

Pour l'élimination totale des agents, tenez sur compte de la méthode utilisée pour la création des agents.

- Manuel : Tout d'abord, il faut supprimer les fichiers `.conf` des agents créés dans le dossier `/etc/pandora/conf` et ensuite supprimer les agents dans la console.
- Satellite_hosts.txt file : Vous devrez créer le fichier `.txt`, ainsi que les fichiers .conf qui ont été créés dans le dossier `/etc/pandora/conf`, puis supprimer les agents dans la console.
- Recon_task : Vous devrez déconfigurer la `recon_task` dans le fichier conf du serveur satellite, puis supprimer les `.conf` qui ont été créés dans le dossier `/etc/pandora/conf` et ensuite supprimer les agents dans la console.

Pour l'élimination partielle, également tenez sur compte la méthode utilisée pour la création des agents.

- Manuel : Tout d'abord, supprimez les fichiers `.conf` des agents que vous voulez supprimer dans le

dossier `/etc/pandora/conf` et ensuite supprimez les agents dans la console.
- Satellite_hosts.txt file : Il faudra supprimer le fichier `.txt` les lignes des IP à supprimer, ainsi que les fichiers `.conf` qui ont été créés dans le dossier `/etc/pandora/conf` avec ces IP, puis supprimer les agents dans la console.
- Recon_task : Vous devez configurer la blacklist de la `recon_task` dans le fichier `.conf` du serveur satellite, puis supprimer les `.conf` qui ont été créés dans le dossier `/etc/pandora/conf` avec ces IPs et supprimer les agents dans la console.

Configurations personnalisées par agent

En plus des modules " automatiques ", il sera possible d'ajouter à la supervision tout contrôle TCP, SNMP, WMI ou SSH disponible, en utilisant une syntaxe similaire à celle utilisée pour les modules locaux dans les agents logiciels. Ensuite vous verrez quelques exemples de modules valides pour le Satellite Server, car ils sont autogénérés après avoir détecté le système.

Assurez-vous que l'OID commencent avec un point ou les
modules SNMP ne fonctionneront pas !

Etat de l'interface via SNMP. Le serveur satellite détecte automatiquement chaque interface :

```
module_begin
module_name if eth1 OperStatus
module_description IP address N/A. Description: The current operational state of
the interface. The testing(3) state indicates that no operational packets can be
passed.
module_type generic_data_string
module_snmp 192.168.70.225
module_oid .1.3.6.1.2.1.2.2.1.8.3
module_community artica06
module_end
```

Pour obliger le module à utiliser SNMP version 2c, ajoutez la ligne :

```
module_version 2c
```

Pour obliger le module à utiliser SNMP version 1, ajoutez la ligne :

```
module_version 1
```

Par exemple :

```
 module_begin
 module_name if eth1 OperStatus
 module_description IP address N/A. Description: The current operational state
of the interface. The testing(3) state indicates that no operational packets can
```

```
be passed.
 module_type generic_data_string
 module_snmp 192.168.70.225
 module_version 2c
 module_oid .1.3.6.1.2.1.2.2.1.8.3
 module_community artica06
 module_end
```

Connexion à une machine (via PING) :

```
 module_begin
 module_name ping
 module_type generic_data
 module_ping 192.168.70.225
 module_end
```

Vérification d'un port (via TCP) :

```
 module_begin
 module_name Port 80
 module_type generic_proc
 module_tcp
 module_port 80
 module_end
```

Requête SNMP générique. Dans ce cas, le Serveur Satellite tire automatiquement le trafic de chaque interface, avec son nom “ réel ” descriptif :

```
module_begin
module_name if eth0 OutOctets
module_description The total number of octets transmitted out of the interface,
including framing characters.
module_type generic_data_inc
module_snmp 192.168.70.225
module_oid .1.3.6.1.2.1.2.2.1.16.2
module_community public
module_end
```

Requête WMI pour l'utilisation du CPU (pourcentage) :

```
 module_begin
 module_name CPU
 module_type generic_data
 module_wmicpu 192.168.30.3
 module_wmiauth admin%none
 module_end
```

Requête WMI pour la mémoire libre (pourcentage) :

```
module_begin
module_name FreeMemory
module_type generic_data
module_wmimem 192.168.30.3
module_wmiauth admin%none
module_end
```

Consultation de WMI générique :

```
module_begin
module_name GenericWMI
module_type generic_data_string
module_wmi 192.168.30.3
module_wmiquery SELECT Name FROM Win32_ComputerSystem
module_wmiauth admin%none
module_end
```

Commande générique SSH :

```
module_begin
module_name GenericSSH
module_type generic_data
module_ssh 192.168.30.3
module_command ls /tmp | wc -l
module_end
```

Pour introduire un seuil, vous devez le faire à la fois dans la définition du texte du module (`module_min_warning`, `module_min_critical`) et dans la définition du seuil via l'interface Web. Par exemple :

```
module_begin
module_name Latency
module_type generic_data
module_latency 192.168.70.225
module_min_warning 80
module_min_critical 120
module_end
```

Vous pouvez créer manuellement des modules d'exécution. Les scripts ou les commandes exécutés par le Serveur Satellitaire doivent être préalablement affichés et accessibles par celui-ci. Dans ce sens, il fonctionne de la même manière qu'un `module_exec`. Tenez compte du fait que l'utilisation de `module_exec` pourrait faire que la performance du Satellite Server diminue.

```
module_begin
module_name Sample_Remote_Exec
module_type generic_data
module_exec /usr/share/test/test.sh 192.168.50.20
module_min_warning 90
```

```
module_min_critical 95
module_end
```

A partir de la version 7 de Pandora FMS, vous pouvez également ajouter des plugins. Comme ceux-ci, vous devez tenir compte du fait que les plugins seront exécutés dans la machine où le Satellite Server est en cours d'exécution. Par conséquent, vous devrez implémenter dans ces plugins une méthode pour vous connecter à la machine distante que vous voulez surveiller. L'avantage par rapport aux précédents est leur grande flexibilité. De cette façon, il est possible d'implémenter des conditions et d'autres mécanismes pour lesquels un `module_exec` ne suffit pas. La syntaxe est la même que celle des agents. Un exemple d'utilisation d'un plugin pourrait être le suivant :

```
module_plugin /usr/share/pandora/remote_advanced_checks.sh 192.168.0.1
```

SNMPv3

Pour configurer un module SNMPv3, définissez *module_version* à 3 et spécifiez :

- `module_seclevel` : Le niveau de sécurité (`noauth`, `authnopriv` ou `authpriv`),
- `module_secname` : Le nom de la sécurité.
- `modules_authproto` : Le protocole d'authentification (`md5` ou `sha`).
- `module_authpass` : La clé d'authentification.
- `module_privproto` : Le protocole de confidentialité (`aes` ou `des`).
- `module_privpass` : La clé de confidentialité selon les besoins.
- Par exemple :

```
module_begin
module_name snmp_noauth
module_type generic_data_string
module_snmp 127.0.0.1
module_version 3
module_oid .1.3.6.1.2.1.1.1.0
module_seclevel noauth
module_secname snmpuser
module_end
```

```
module_begin
module_name snmp_authnopriv
module_type generic_data_string
module_snmp 127.0.0.1
module_version 3
module_oid .1.3.6.1.2.1.1.2.0
module_seclevel authnopriv
module_secname snmpuser
module_authproto md5
module_authpass 12345678
module_end
```

```
module_begin
module_name snmp_authpriv
module_type generic_data_string
module_snmp 127.0.0.1
module_version 3
module_oid .1.3.6.1.2.1.1.2.0
module_seclevel authpriv
module_secname snmpuser
module_authproto sha
module_authpass 12345678
module_privproto aes
module_privpass 12345678
module_end
```

La configuration spécifique de SNMPv3 peut être partagée entre les modules en la retirant de la déclaration de module, dans le cas où elle est la même pour tous (il est également possible de la partager entre les agents en la déplaçant vers le fichier de configuration du satellite) :

```
agent_name snmp
address 127.0.0.1

seclevel authpriv
secname snmpuser
authproto md5
authpass 12345678
privproto des
privpass 12345678

module_begin
module_name snmp_authpriv_1
module_type generic_data_string
module_snmp
module_version 3
module_oid .1.3.6.1.2.1.1.1.0
module_end

module_begin
module_name snmp_authpriv_2
module_type generic_data_string
module_snmp
module_version 3
module_oid .1.3.6.1.2.1.1.2.0
module_end
```

Pour la création de modules SNMP (y compris SNMPv3) via la console Web de PFMS, consultez cette vidéo (en anglais). Pour la création de groupes de composants (y compris SNMPv3), voir "SNMP Wizard".

Fichier de configuration *par défaut* du serveur satellite pour SNMPv3 :

Vous devrez introduire vos propres valeurs et/ou informations d'identification, ainsi que modifier les protocoles ou les méthodes de cryptage nécessaires. Vous devrez redémarrer le serveur PFMS pour que les nouvelles valeurs de configuration soient lues et mises en mémoire.

```
# Security level used for SNMPv3 messages (noauth, authnopriv or authpriv).
#snmp3_seclevel authpriv

# Security name used for SNMPv3 messages.
#snmp3_secname

# Authentication protocol (md5 or sha) for authenticated SNMPv3 requests.
#snmp3_authproto sha

# Authentication password for authenticated SNMPv3 request.
#snmp3_authpass

# Privacy protocol (des or aes) for encrypted SNMPv3 requests.
#snmp3_privproto des

# Privacy password for encrypted SNMPv3 messages.
#snmp3_privpass
```

Boîtes d'identifiants

Sauf si l'authentification par clé est configurée, les modules SSH ont besoin d'un nom d'utilisateur (<user>) et d'un mot de passe (<pass>) pour fonctionner. Ceux-ci sont configurés dans le fichier de configuration principal, `satellite_server.conf`, à l'aide de boîtes d'identification (`credential_box`) ayant le format suivant :

réseau/masque,utilisateur,mop de pass

réseau/masque,utilisateur mot de passe crypté

Par exemple :

```
credential_box 192.168.1.1/32,user,pass1
credential_box 192.168.1.0/24,user,pass2
```

Les recherches dans les boîtes de justificatifs se font à partir de masques plus ou moins restrictifs.

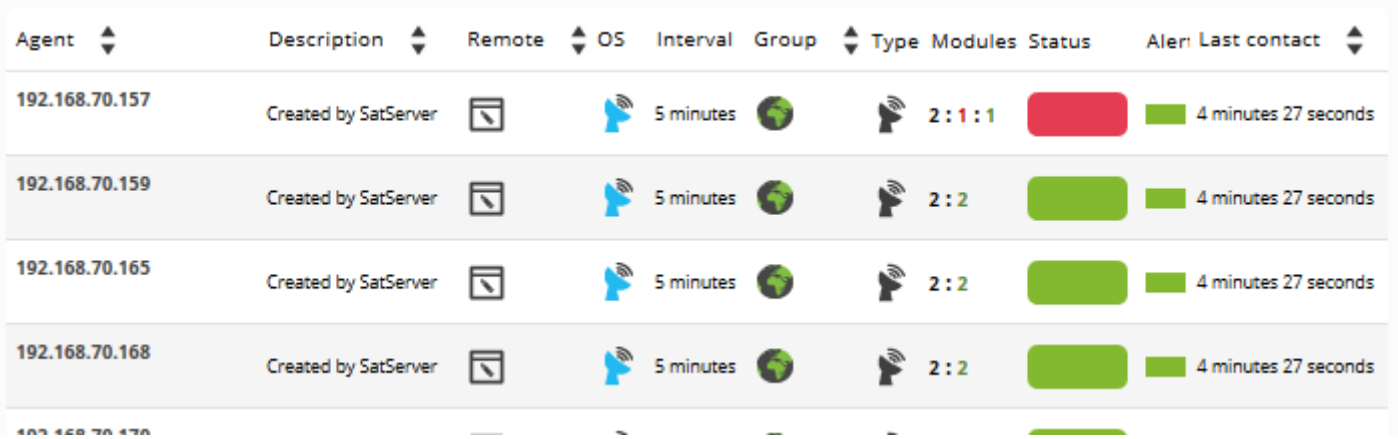
Les mots de passe peuvent être cryptés en utilisant Blowfish en mode ECB. Assurez-vous que credential_pass est défini ; sinon, le nom d'hôte sera utilisé comme mot de passe de chiffrement par défaut. La représentation hexadécimale du texte chiffré doit être entourée de doubles crochets :

```
```

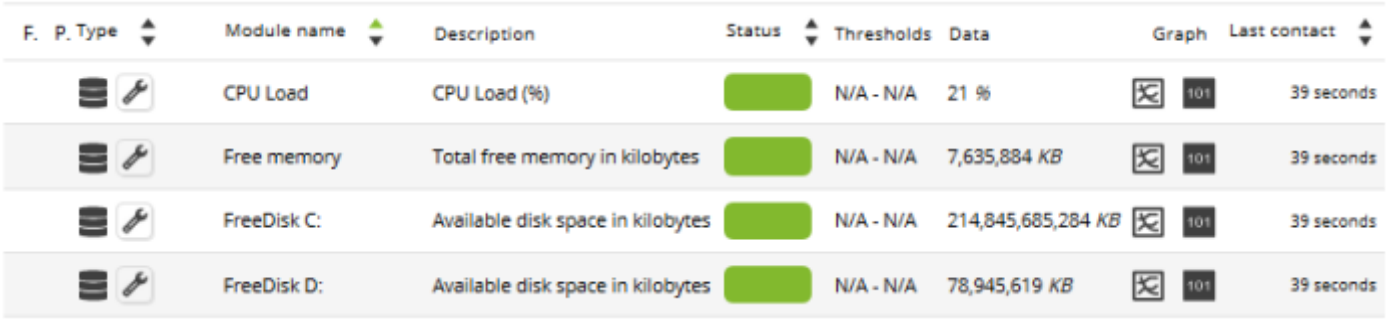
```
credential_box 192.168.1.0/24,user,[[80b51b60786b3de2|]]
```

Visualisation dans la console de tous les agents

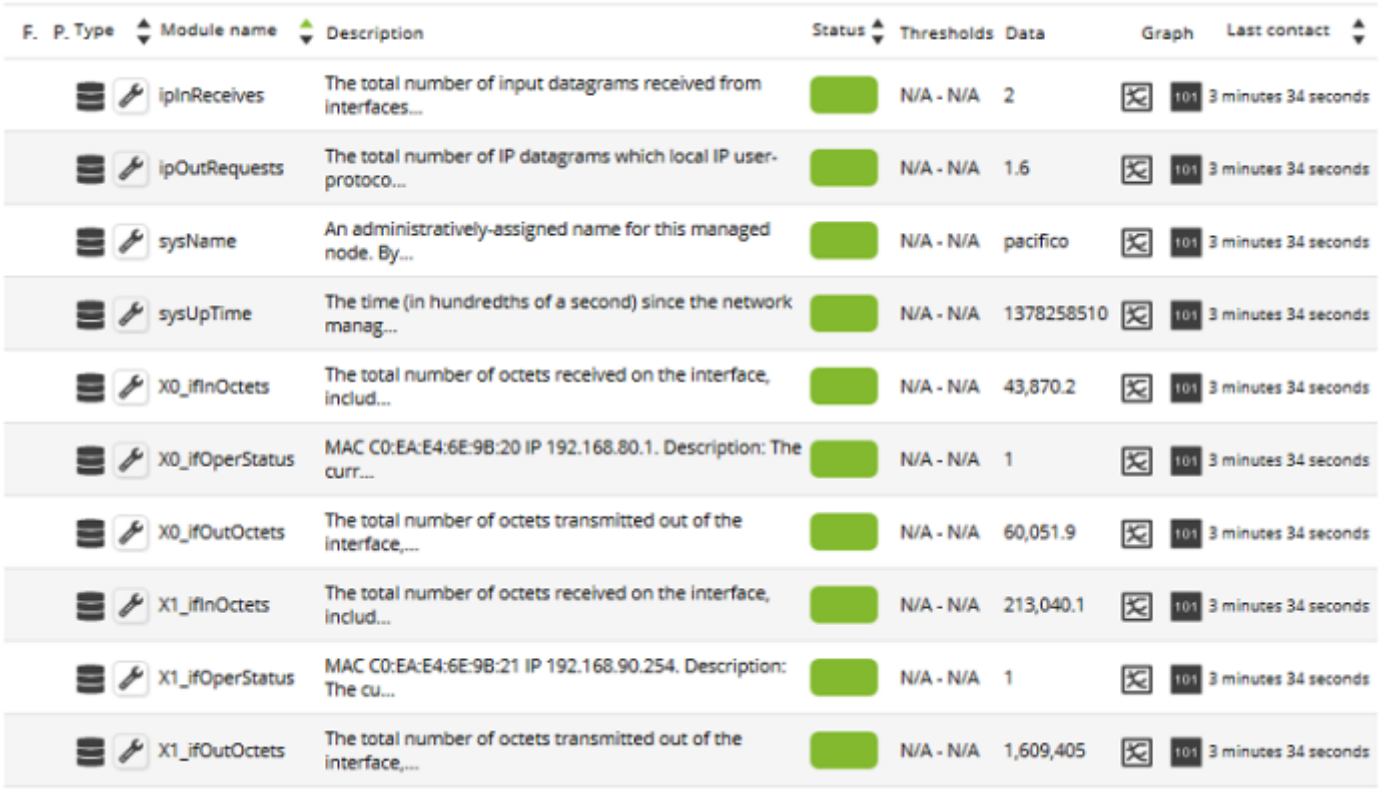
Si la configuration du Satellite Server était correcte, nous devrions obtenir une vue d'agent similaire à celle-ci :



Généralement, dans toutes les machines, des modules de type ICMP seront créés (Ping et Latence), mais dans certaines d'entre elles il est également possible de générer des modules de type SNMP et WMI. Dans ceux qui ont WMI activé, les modules suivants seront générés, s'ils sont disponibles :

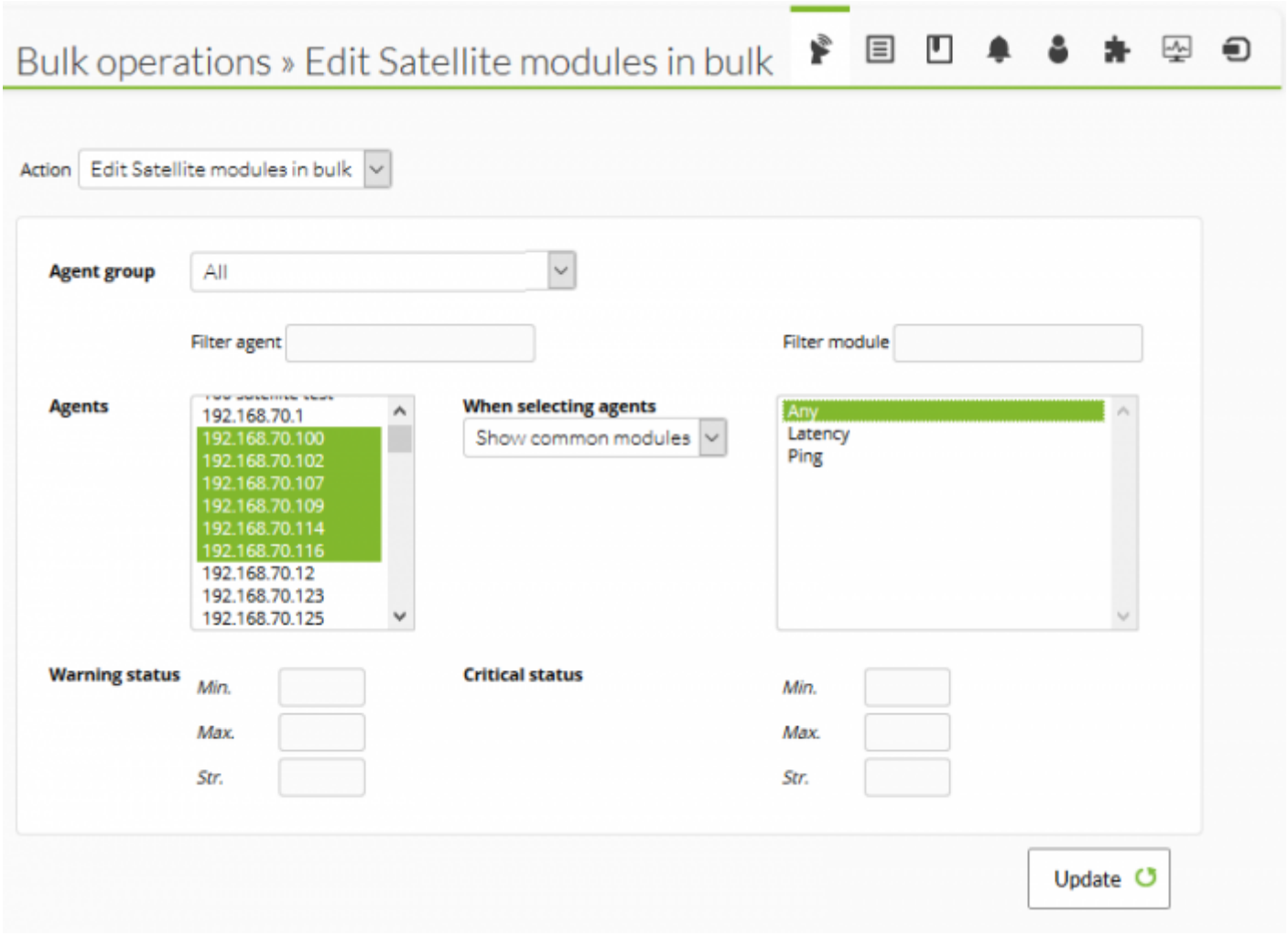


Dans les machines avec SNMP activé, les modules suivants seront générés, s'ils sont disponibles :

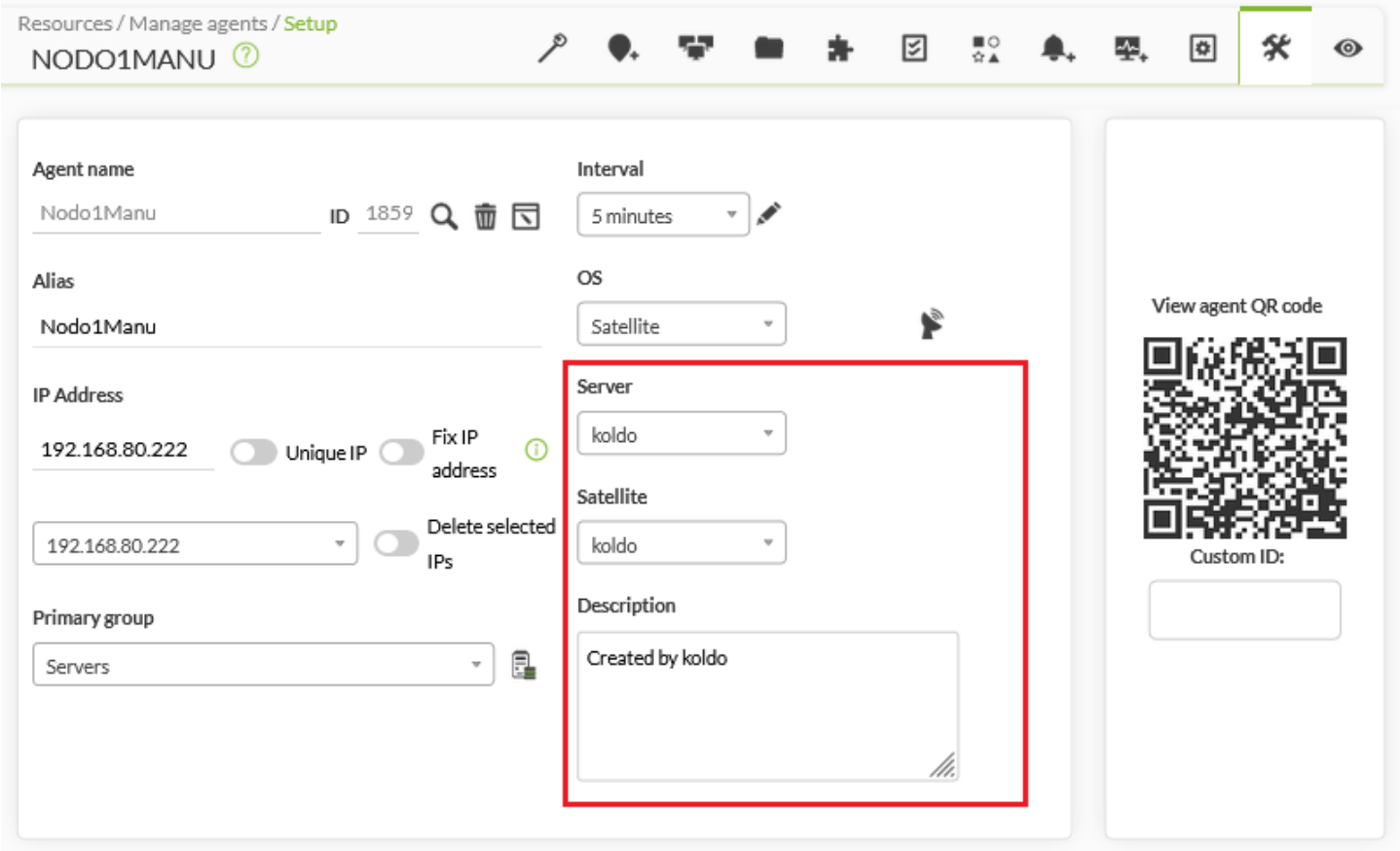


F.	P.	Type	Module name	Description	Status	Thresholds	Data	Graph	Last contact
			ipInReceives	The total number of input datagrams received from interfaces...		N/A - N/A	2		3 minutes 34 seconds
			ipOutRequests	The total number of IP datagrams which local IP user-protoco...		N/A - N/A	1.6		3 minutes 34 seconds
			sysName	An administratively-assigned name for this managed node. By...		N/A - N/A	pacifico		3 minutes 34 seconds
			sysUpTime	The time (in hundredths of a second) since the network manag...		N/A - N/A	1378258510		3 minutes 34 seconds
			X0_ifInOctets	The total number of octets received on the interface, includ...		N/A - N/A	43,870.2		3 minutes 34 seconds
			X0_ifOperStatus	MAC C0:EA:E4:6E:9B:20 IP 192.168.80.1. Description: The curr...		N/A - N/A	1		3 minutes 34 seconds
			X0_ifOutOctets	The total number of octets transmitted out of the interface,...		N/A - N/A	60,051.9		3 minutes 34 seconds
			X1_ifInOctets	The total number of octets received on the interface, includ...		N/A - N/A	213,040.1		3 minutes 34 seconds
			X1_ifOperStatus	MAC C0:EA:E4:6E:9B:21 IP 192.168.90.254. Description: The cu...		N/A - N/A	1		3 minutes 34 seconds
			X1_ifOutOctets	The total number of octets transmitted out of the interface,...		N/A - N/A	1,609,405		3 minutes 34 seconds

Dans la section des opérations massives de la console Pandora FMS il y a une section spéciale dédiée au Serveur Satellite, où vous pouvez faire plusieurs actions d'édition et de suppression d'agents et de modules de manière massive :



Version 763 ou ultérieure.



Liste noire SNMP

Lors de la surveillance de grands réseaux, les modules SNMP qui renvoient des données non valides peuvent affecter les performances du serveur satellite, et conduire d'autres modules à un état inconnu. Pour éviter cela, le Serveur Satellite peut lire une *liste noire* de modules SNMP qui seront jetés au démarrage avant l'exécution.

Pour créer une liste noire, éditez le fichier de configuration `/etc/pandora/satellite_server.conf` et assurez-vous que `snmp_blacklist` est décommenté et configuré avec le chemin du fichier où les modules en liste noire seront sauvegardés. Alors, exécutez :

```
satellite_server -v /etc/pandora/satellite_server.conf
```

Redémarrez le Satellite Server. La liste noire peut être régénérée autant de fois que nécessaire.

Le format de la liste noire est le suivant :

```
agent:OID
agent:OID
...
```

Par exemple :

```
192.168.0.1:1.3.6.1.4.1.9.9.27
192.168.0.2:1.3.6.1.4.1.9.9.27
```

Retour à l'index de documentation du Pandora FMS