



# サテライトサーバによる分散監視



<https://pandorafms.com/manual!/777/>

Permanent link:

[https://pandorafms.com/manual!/777/ja/documentation/pandorafms/complex\\_environments\\_and\\_optimization/05\\_satellite](https://pandorafms.com/manual!/777/ja/documentation/pandorafms/complex_environments_and_optimization/05_satellite)

2024/10/03 18:41

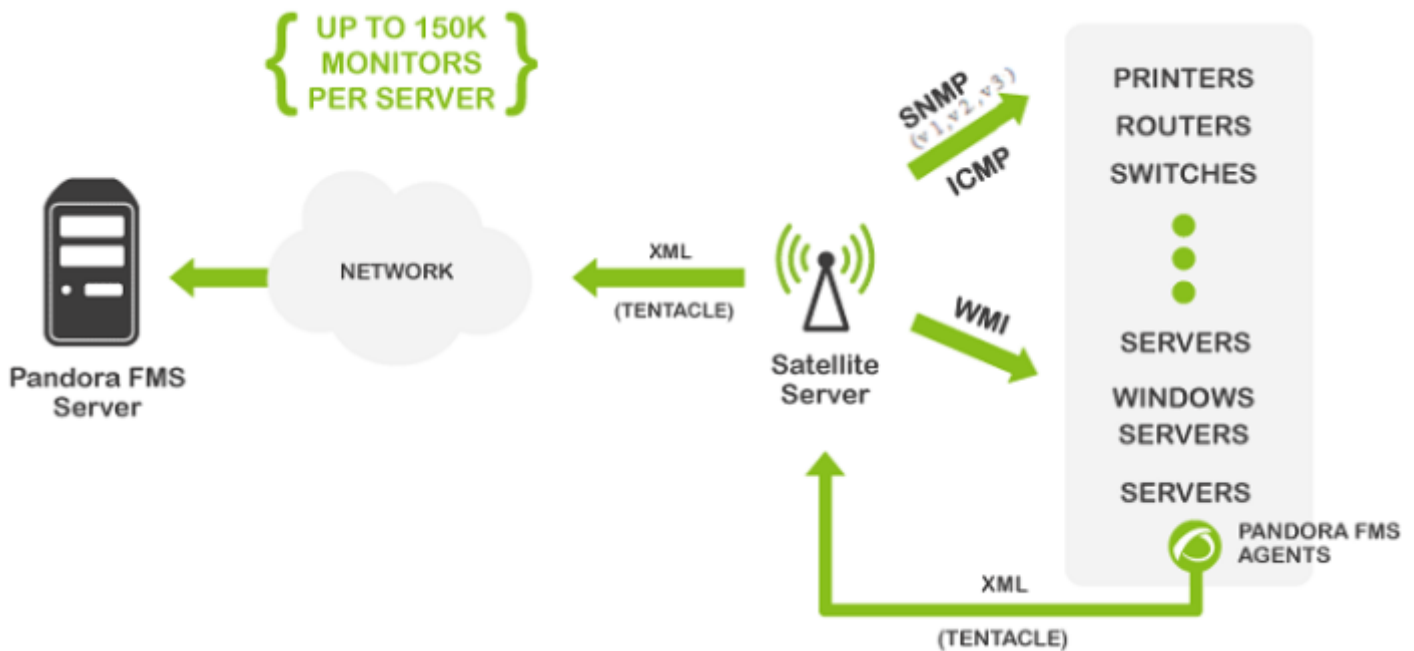


# サテライトサーバによる分散監視

[Pandora FMS ドキュメント一覧に戻る](#)

## 概要

**E** サテライトサーバは、ネットワークおよびリモートシステムの監視、検出に利用します。SNMP や ICMP を使ってネットワークの構成要素(ルータ、スイッチなど)や Windows(WMIを利用)や Linux サーバ(SNMPを利用)を検出することができます。これは通常のサーバではなく、拡張機能をもったブローカーエージェントであるとみなすことができます。Enterprise版の追加コンポーネントです。これは、特に Pandora ソフトウェアエージェントを使わずに、サーバから直接アクセスできないリモートネットワークを監視するのに便利です。



このサーバは、pandora データベースへの接続は必要ありません。エージェントのように tentacle プロトコルを使ってすべての情報を XML フォーマットで送ります。

サテライトサーバは、Windows および Linux で利用できます。インストール手順はそれぞれの場合で若干違う程度です。

このサーバはいくつかのユニークで特徴的な点があります。

- 非常に高い頻度(500チェック/秒)でネットワークテスト(ICMP遅延および SNMP v1 および v2)を実行できます。SNMP v3 では、**認証設定** およびデータの暗号化のため、そこまでの速度は出ません。
- 一定間隔(デフォルトでは300秒)でサーバに情報を送信します。しかし、遅延の計測、ICMP および SNMP 監視はより小さい間隔(例えば30秒)で実行できます。これにより、状態変化を検知したときに素早く Pandora サーバへ通知することができます。モジュールのタイプが generic\_proc でない(例えばネッ

トワークインタフェースや一般的なネットワークの接続性)場合は、この状態変化は事前に設定しておく必要があります。

- データベースへの接続は必要ありません。ブローカーエージェントやエクスポートサーバのように XML フォーマットでファイルをサーバに送信します。
- SNMP および WMI の自動検出機能があります(IPアドレスにより)エージェントを検出すると、動的に要素(ネットワークインタフェース、ストレージ)を検出し、自動的に監視を行います。
- Windows システムでは、ディスクドライブ、CPU、メモリを検出できます。
- SNMP に応答するシステムでは、インタフェースの状態とそれぞれの入出力トラフィック、システム名を検出することができます。
- 自動生成されたモジュールは、他のモジュール同様に、コンソールからエージェントの管理で編集することができます。また、他のエージェントと同様に一括操作メニューのサテライトメニューからモジュールを編集することもできます。
- エージェントは、サテライトサーバのエージェント設定ファイルディレクトリ内のエージェント設定ファイルにより直接作成することができます。

## バージョン 759 NG 以降

- バージョン 759NG 以降、サテライトサーバと Enterprise ネットワークサーバの両方のすべての高度な機能において IPv6 をサポートしています。以前は IPv4 でのみサポートされていた高性能なコードが IPv6 にも適用されるようになり、既存のポーリング機能が強化されました。

## キャパシティ

サテライトサーバのキャパシティをピンポイントで最大化するのは難しく、サーバ全体で何が動作しているかや実行する監視内容に依存します。最も良い場合で、1秒間に 500 の ICMP/SNMP チェックができますが、リモートデバイスの応答時間に大いに依存します(0.5ミリ秒で応答するデバイスと 2秒で応答するデバイスでは異なります)。理想的な条件下では、1台のサーバで 150,000項目の監視ができます。実際の環境では、閉じた LAN 環境でローエンドなハードウェア(Intel i5, 2GZ, 4GB RAM)による 1台のサテライトサーバで、約 50,000モジュールを監視できることを確認しています。

障害状態の監視項目が多いとパフォーマンスに影響します。障害状態の監視項目ごとにタイムアウトが1つしかないため、タイムアウト値が考慮されている必要があります。障害状態のモジュールが 1000 でタイムアウトが 4秒に設定されている場合、1つのスレッドですべてのチェックを実行するには 4000秒かかります。

## インストール

サテライトサーバはバイナリにて配布されています。追加のライブラリは必要ありません。Windows と Linux とともに、このサーバの機能は同じです。Windows システムではサービスとしてインストールされ、Linux システムではデーモンとしてインストールされます。設定ファイルと仕様はどちらも同

じです。



Linux 版のサテライトサーバは、このドキュメントで指定している外部パッケージへの依存がありません。

## オンラインインストールツール

**E** これは Pandora FMS の特別な機能です。 使用するには Enterprise ライセンスが必要です。 いずれの場合も、必須のインストールパラメータは Pandora FMS Enterprise サーバの IP アドレスまたは FQDN です。 ライセンスに関しては、[こちら](#) までお問合せください。

このツールは、Rocky Linux 8.x、AlmaLinux 8.x および RHEL8.x に対応しています。

オンラインインストールツール利用の前提条件:

- インターネットアクセスできること。
- curl がインストールされていること。(ほとんどのディストリビューションではデフォルトでインストールされています)
- 最低限の **ハードウェア要件** を満たすこと。
- root ユーザ権限を持っていること。
- 対応している OS であること。
- RHEL 8 を使用する場合は、事前にライセンスを有効化し、標準リポジトリをサブスクライブしていること。

オンラインインストールツールを使用するには、root 管理者ユーザでコマンドラインにアクセスし、次のコマンドを実行します。

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

オンラインインストールツールを使ったカスタムインストール:

- PANDORA\_SERVER\_IP: サテライトサーバが接続する Pandora FMS Enterprise サーバの IP アドレスまたは FQDN 必須パラメータ。

- TZ: サテライトサーバのタイムゾーン。オプションパラメータ。
- SATELLITE\_SERVER\_PACKAGE: サテライトサーバインストール tarball パッケージのカスタム URL オプションパラメータ。
- SATELLITE\_KEY: 自動的に有効化するためのサテライトサーバライセンス。オプションパラメータ。
- REMOTE\_CONFIG: リモート設定。オプションパラメータ。デフォルトでは有効化されています(値 1) □
- INSTALL\_AGENT: ソフトウェアエージェント(すべての [エージェントのオンラインインストーラ](#) の設定変数を利用できます)をインストールするオプションパラメータ。デフォルトでは有効化されています(値 1) □
- VMWARE\_DEPENDENCIES: VMware® プラグインをインストールするオプションです。デフォルトでは無効化されています(0) □
- ORACLE\_DEPENDENCIES: Oracle® プラグインをインストールするオプションです。デフォルトでは無効化されています(0) □
- MSSQL\_DEPENDENCIES: MS SQL Server® プラグインをインストールするオプションです。デフォルトでは無効化されています(0) □
- SKIP\_KERNEL\_OPTIMIZATIONS: kernel の最適化を行わないオプションです。デフォルトでは無効化されています(0) □

例:

```
env TZ='Europe/Madrid' \  
SATELLITE_KEY='SOPORTEDEV00RS0REB3M2T7ZHIS051IIQH52JISJ47VGHIRM... ' \  
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)"
```

## Linux システムへのサテライトサーバのインストール

推奨 GNU/Linux OS は RedHat Enterprise (RHEL) 8 / Rocky Linux 8 です。

インストールを行う前に、サテライトサーバの主な依存関係は次の通りです: Perl, Braa, Wmic (バージョン 761 以前) または PandoraWMIC (バージョン 762 以降), Fping, Nmap および libnsl

以前のバージョンの OS を利用している場合は、dnf の代わりに yum を利用してください。

次のコマンドで Perl をインストールします。

```
dnf install perl
```

インストーラに、依存する braa および wmic/PandoraWMIC は含まれています。fping および nmap

は先にEPELを設定し、個別にインストールする必要があります。以下のリンクを確認してください。

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

リポジトリの設定ができれば、以下を実行します。

```
dnf install fping nmap
```

最後に、依存関係で libnsl をインストールしていることを確認します。

```
dnf install libnsl
```

サテライトサーバを含むバイナリをダウンロードしたら、ダウンロードしたファイルのあるディレクトリへ行き、root 権限でファイルを展開します。

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

```
[root@localhost ~]# ls
pandorafms_satellite_server_7.0NG.726_180831_x86_64.tar.gz  README
[root@localhost ~]# tar -xvzf pandorafms_satellite_server_7.0NG.726_180831_x86_64.tar.gz
satellite_server/satellite_server
satellite_server/satellite_server.conf
satellite_server/satellite_serverd
satellite_server/satellite_server_installer
satellite_server/pandora_satellite_logrotate
satellite_server/README
satellite_server/bin/braa
satellite_server/bin/wmic
satellite_server/bin/tentacle_client
satellite_server/bin/pandorafsnmp
[root@localhost ~]#
```

ディレクトリ satellite\_server が作成されるので、そのディレクトリに入ります。

```
cd satellite_server/
```

サテライトサーバをインストールするには、インストールコマンドを実行します。

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

```
[root@localhost satellite_server]# ./satellite_server_installer --install
Pandora FMS Satellite Server installer for GENERIC. (c) 2014-2015 Artica ST.

>Installing the Pandora FMS Satellite Server binary to /usr/bin...
>Installing the tentacle_client binary to /usr/bin...
>Installing the braa binary to /usr/bin...
>Installing the pandorafsnmp binary to /usr/bin...
>Installing the wmic binary to /usr/bin...
>Copying configuration file to /etc/pandora...
>Creating agent configuration directory /etc/pandora/conf...
>Copying startup script to /etc/init.d...
>Linking startup script to /etc/rc.d/rc2.d
Creating logrotate.d entry for Pandora FMS log management

Edit the file /etc/pandora/satellite_server.conf and manually configure the Satellite Server.

[root@localhost satellite_server]#
```

完了したら、以下の `satellite_server.conf` を編集します。

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

761 より前のバージョンでは、ライセンスは `pandora_license` オプションに手動で入力する必要があります。テキストエディタ VIM などを用いて、`pandora_license` トークンを探してコメントを外し、**Pandora FMS Enterprise ライセンス**を入力します。

`server_ip` トークンを検索し、サテライトサーバが接続する Pandora FMS Enterprise サーバの IP アドレスまたはドメインを指定します。

ファイルを保存したのち、以下を実行してサテライトサーバを実行します。

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

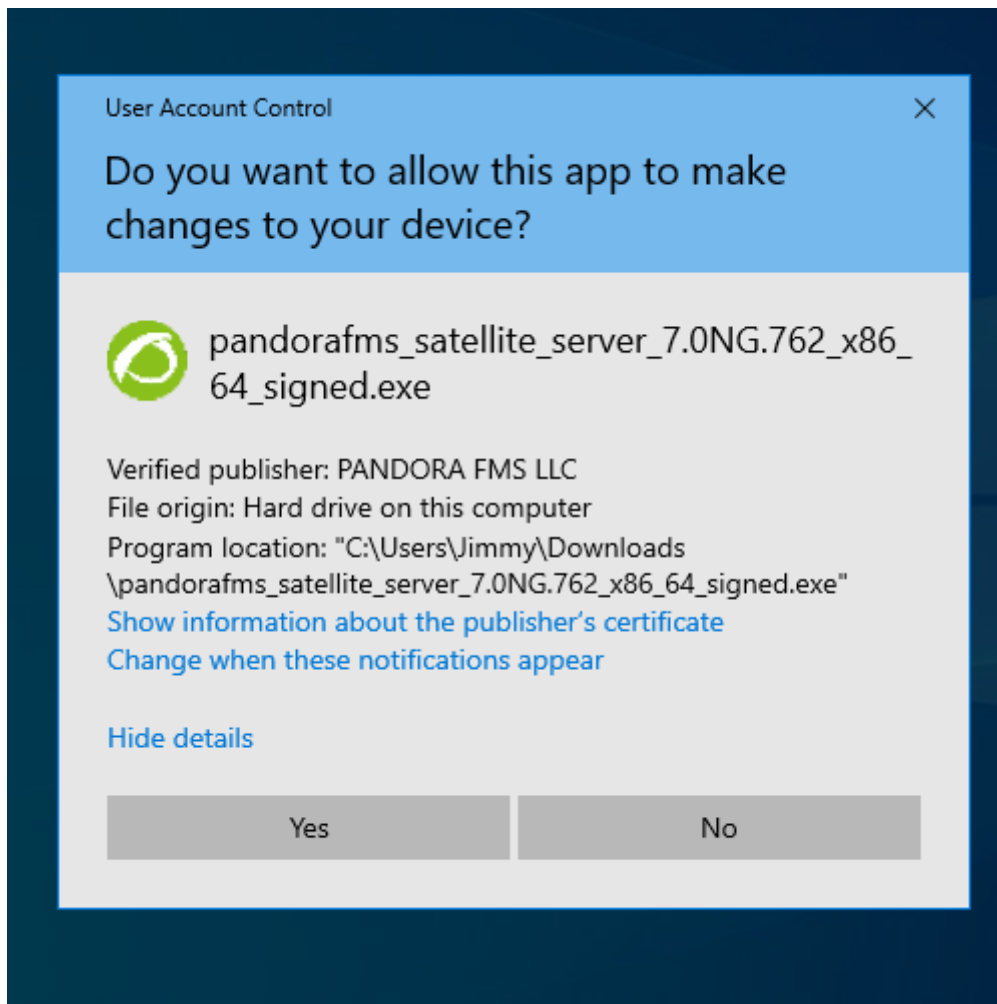
何らかのエラーが発生した場合は、次のログファイルを確認してください。

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

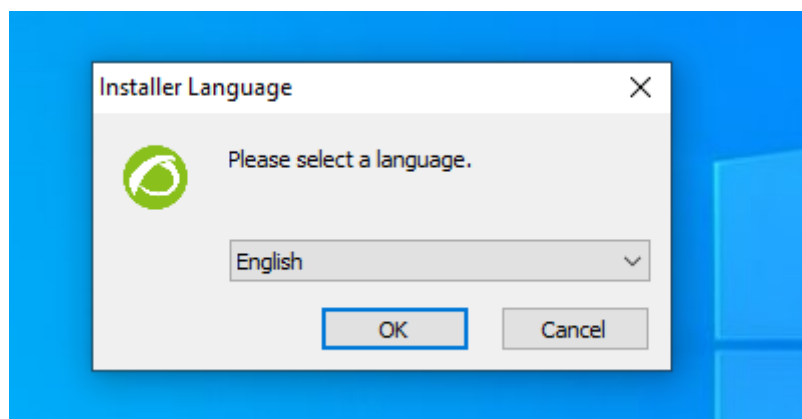
## Windows へのインストール

デジタル署名されたインストーラ(バージョン 762 およびそれ以上)を実行し、Yes をクリックします。

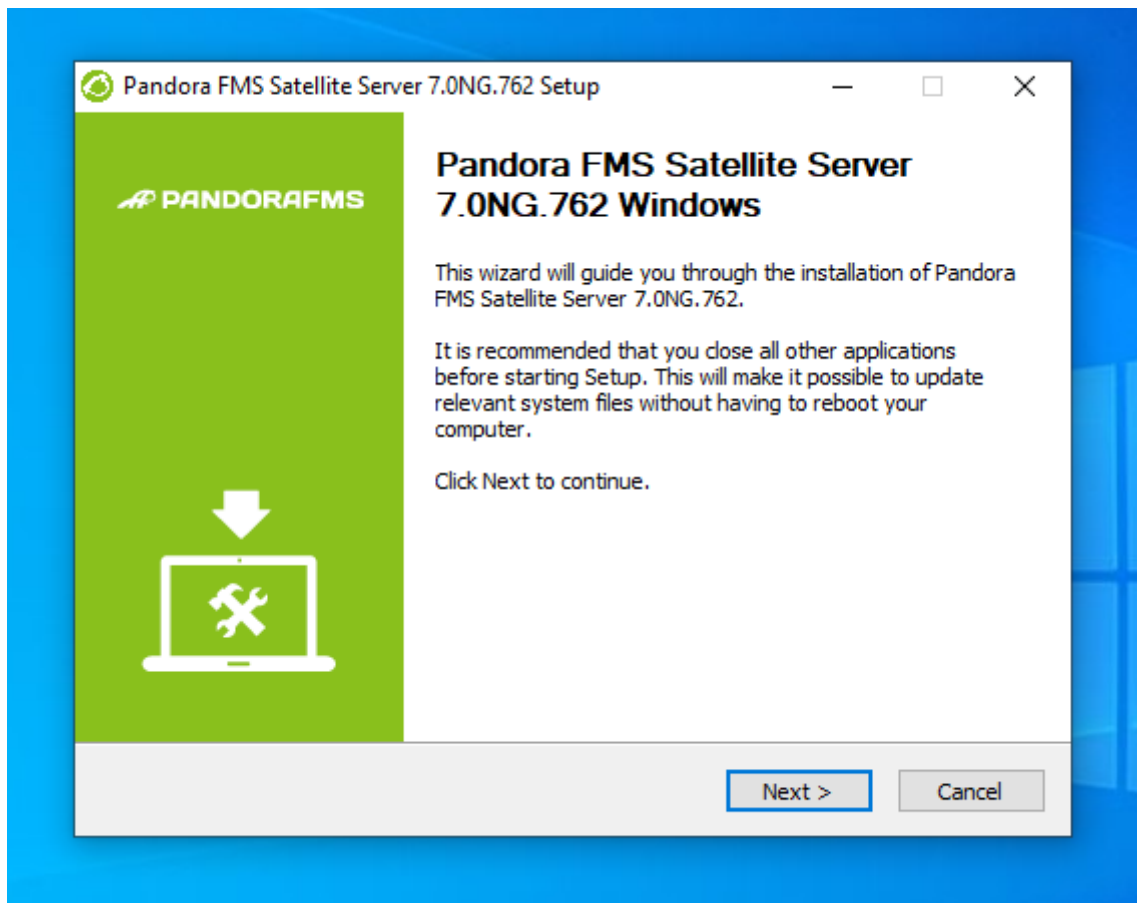




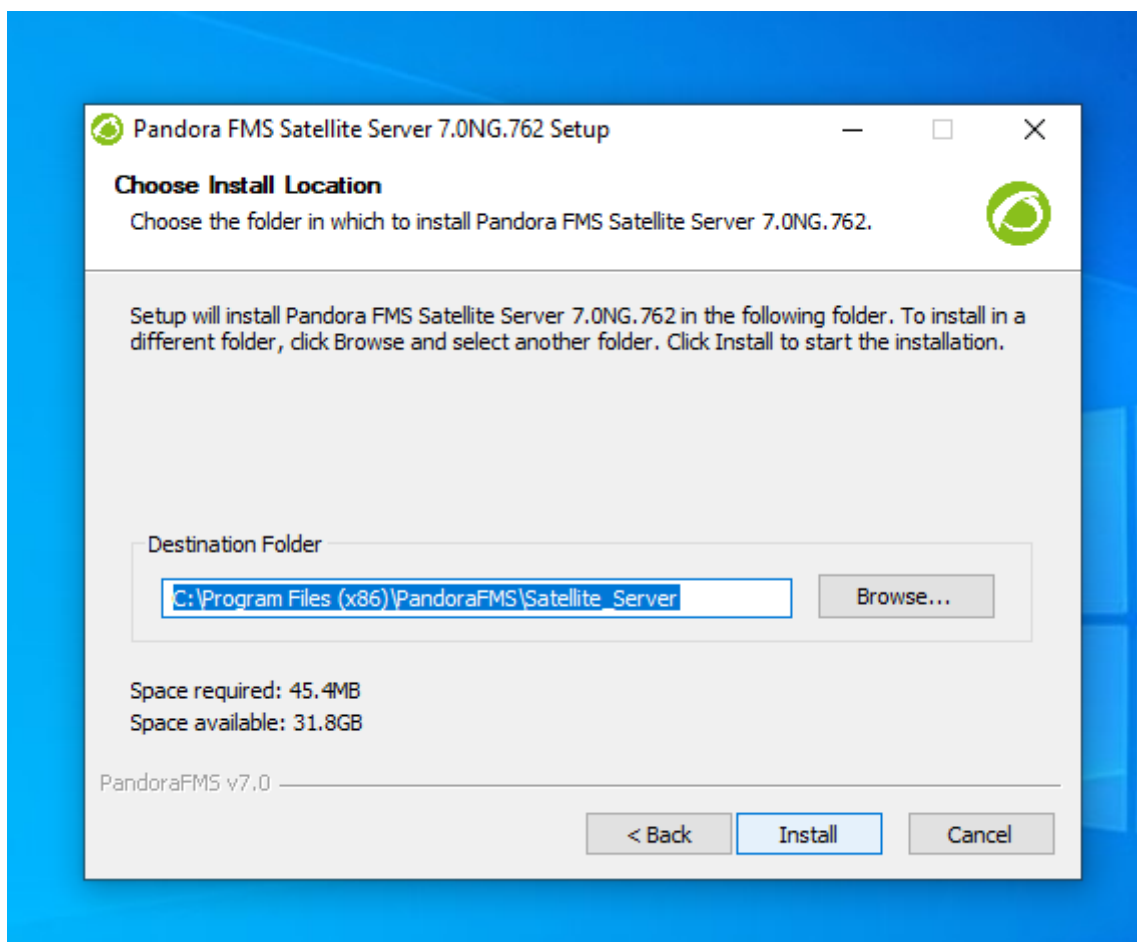
インストール言語を選択します。(日本語はありません)



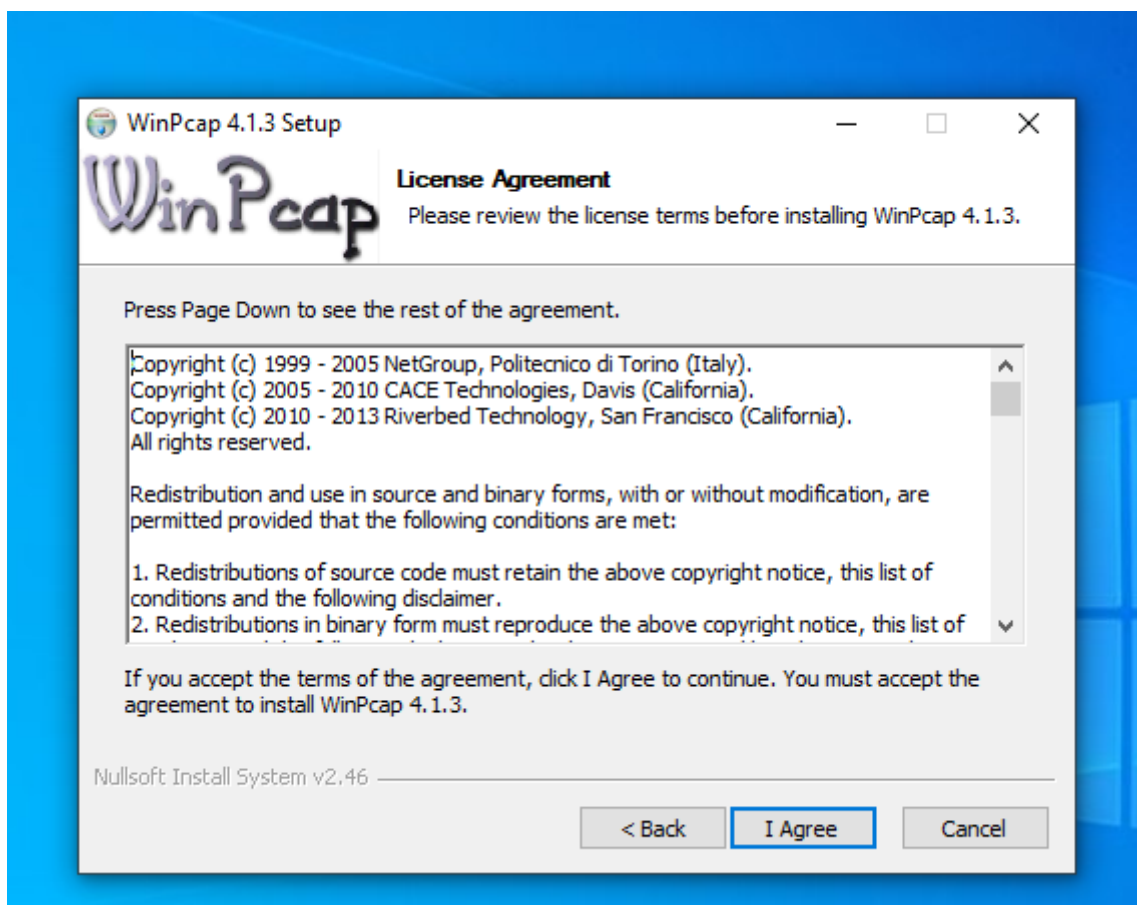
Next をクリックします。



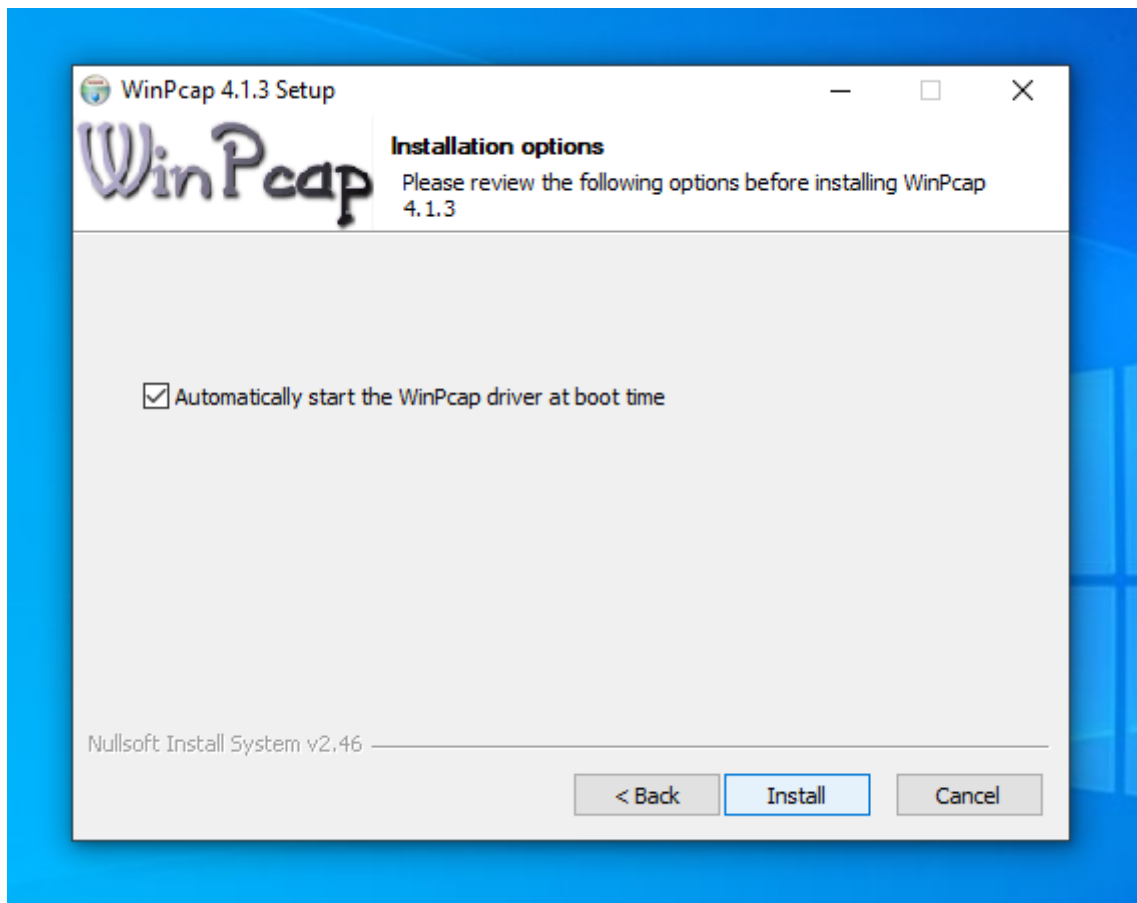
プログラムをインストールする場所を選択します。



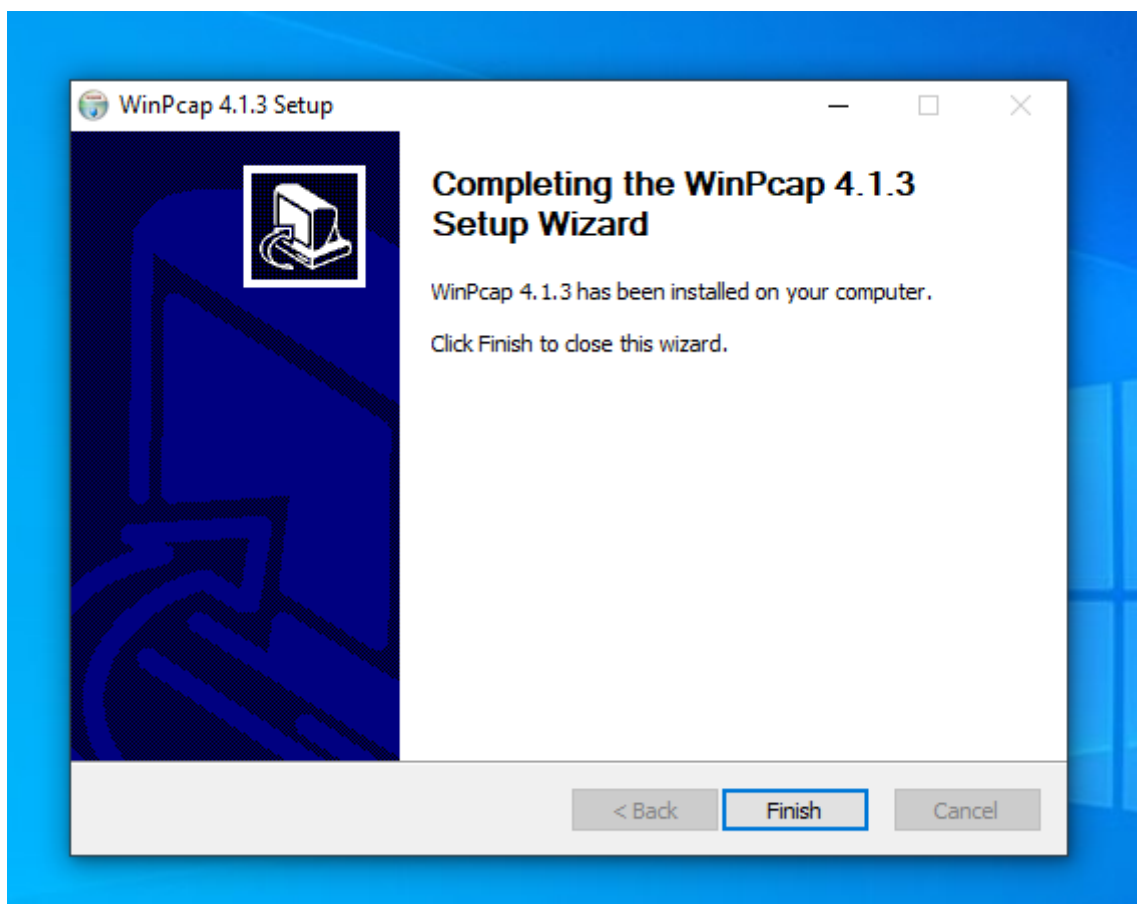
WinPCap のインストールが必要です。インストールウィンドウは、インストールプロセスの次のステップで表示されます。



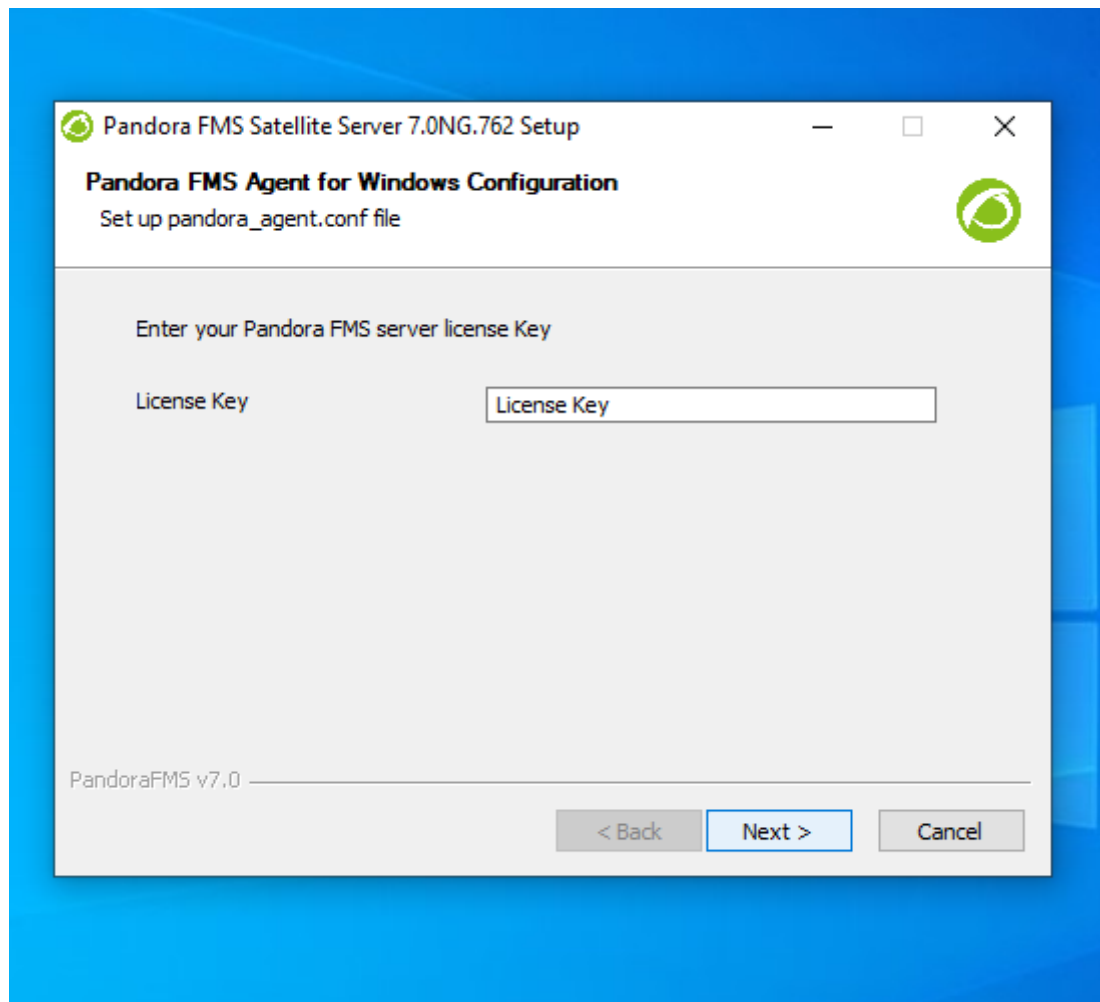
システムの起動時に起動するように WinPCap を設定します。



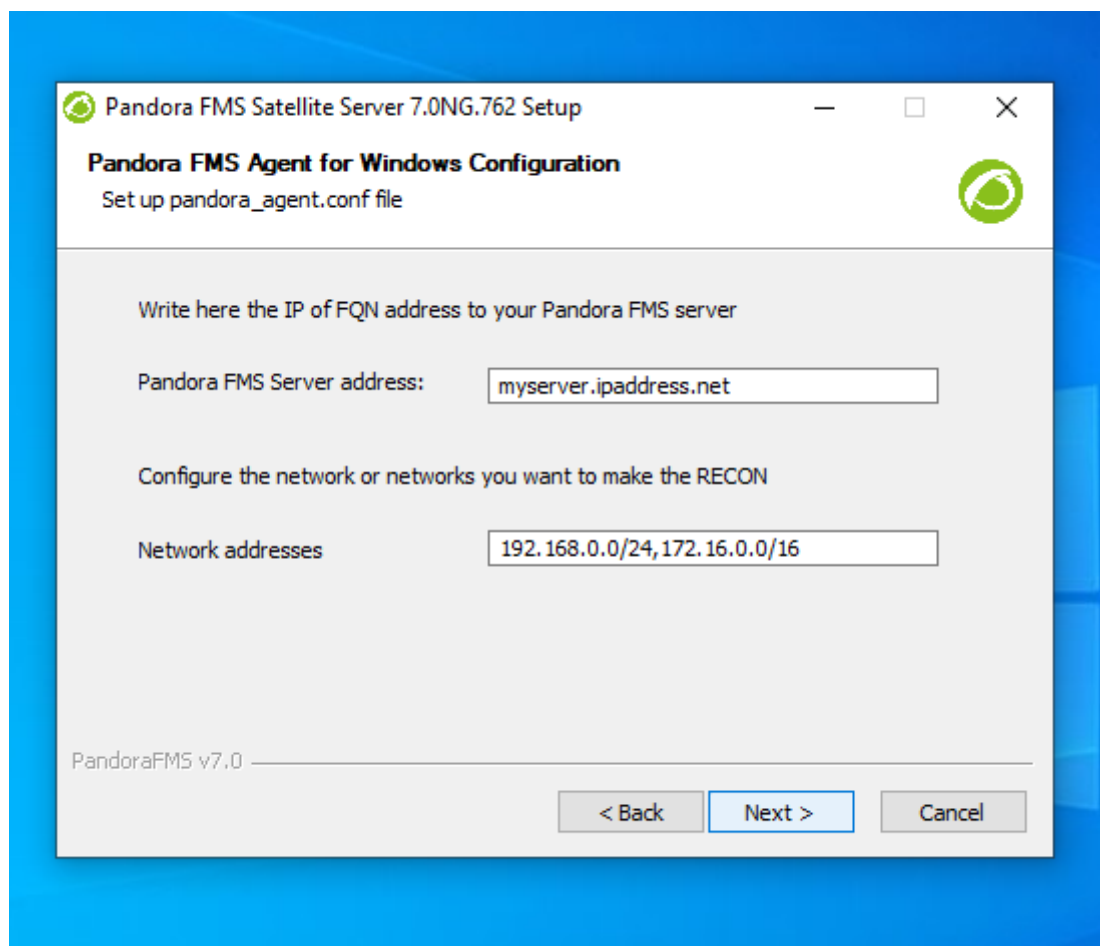
WinPCap のインストールが完了すると、次のウィンドウが表示されます。



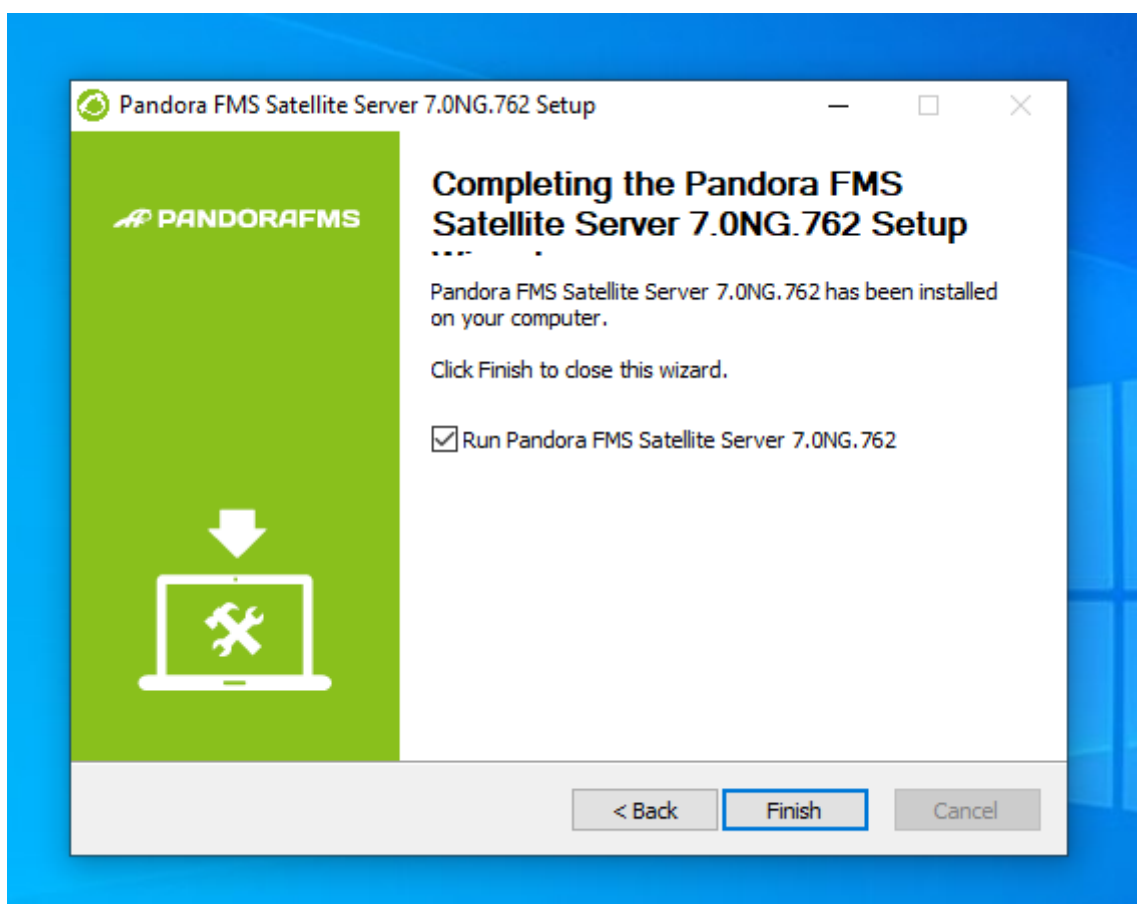
Pandora FMS ライセンス番号を入力して、インストールを続行します。



次に、データ送信先の Pandora FMS サーバアドレスを設定します。サテライトサーバーのネットワーク自動検出ルールを定義します。



すべての変更が適用されるように、マシンを再起動します。



処理が完了したら MS Windows® の [スタート] メニューから Pandora FMS サテライトサーバを起動および停止します。

MS Windows のバージョンに応じて、これらのライブラリのいくつかをインストールする必要があります:

Microsoft Visual C++ Redistributable (最新のサポート版):

- Visual Studio 2015, 2017, 2019, および 2022.
- Visual Studio 2013 (VC++ 12.0).
- Visual Studio 2012 (VC++ 11.0) Update 4.
- Visual Studio 2010 (VC++ 10.0) SP1 (サポートされなくなっています).
- Visual Studio 2008 (VC++ 9.0) SP1 (サポートされなくなっています).

32 ビット(X86)、64 ビット(X64)、および ARM64 プロセッサ用いずれも、以下のリンクから入手できます。

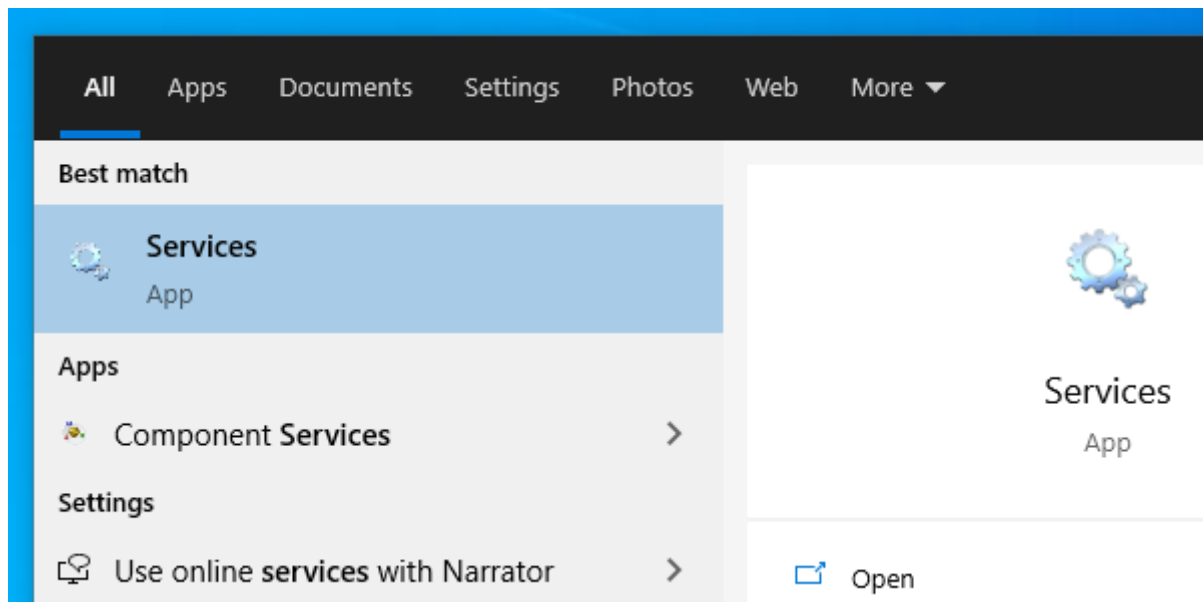
<https://docs.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-170&viewFallbackFrom=msvc-170>

### いくつかの Windows バージョンにおける WMI モジュールの実行

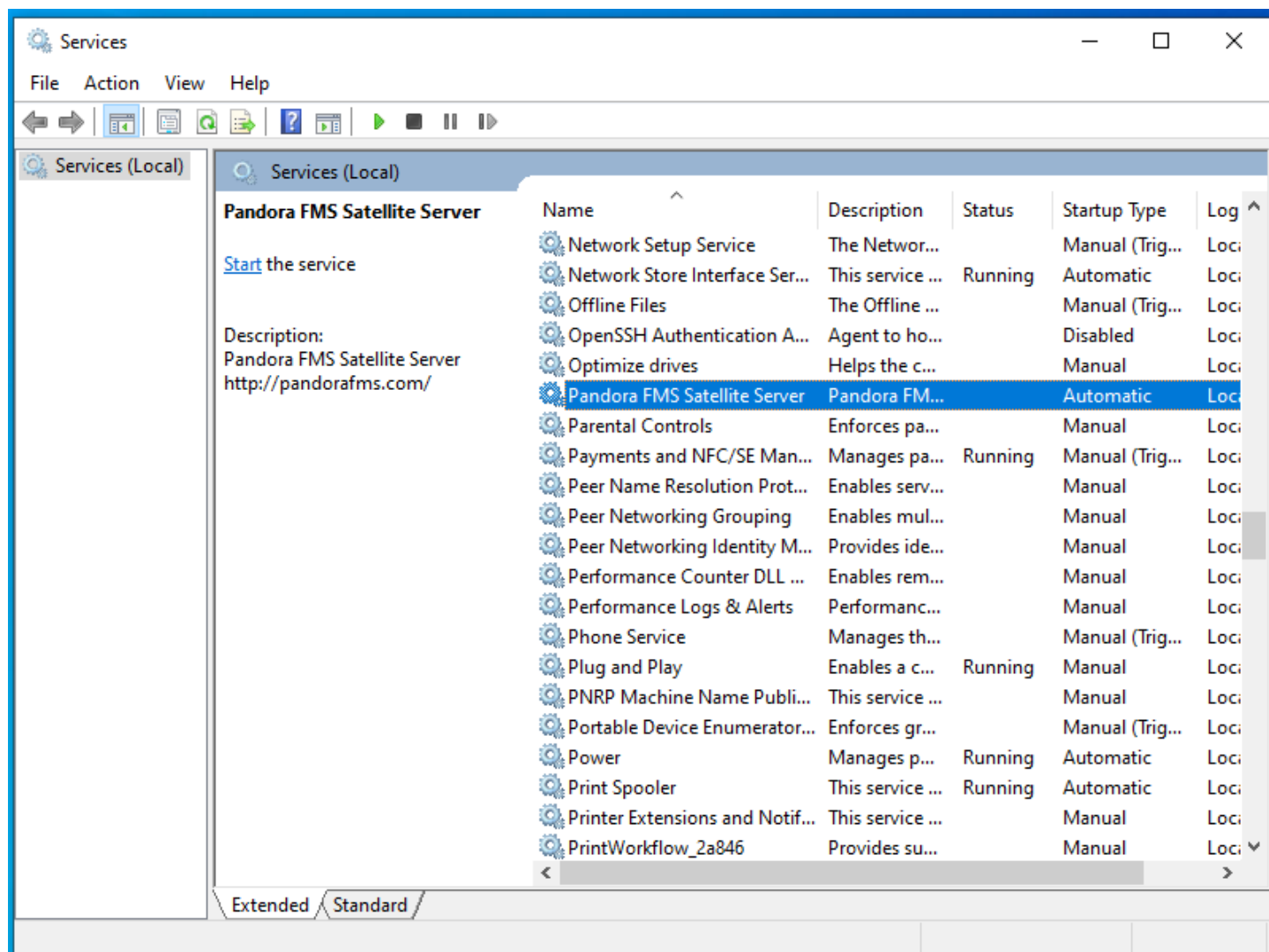
Windows のセキュリティ上の理由により、いくつかのバージョンではリモートから WMI クエリを実行できるユーザは限定されています。このような場合は、Administrator ユーザ権限でサテライトサーバのサービスを実行します。

手順は次の通りです。

サービスを開きます。

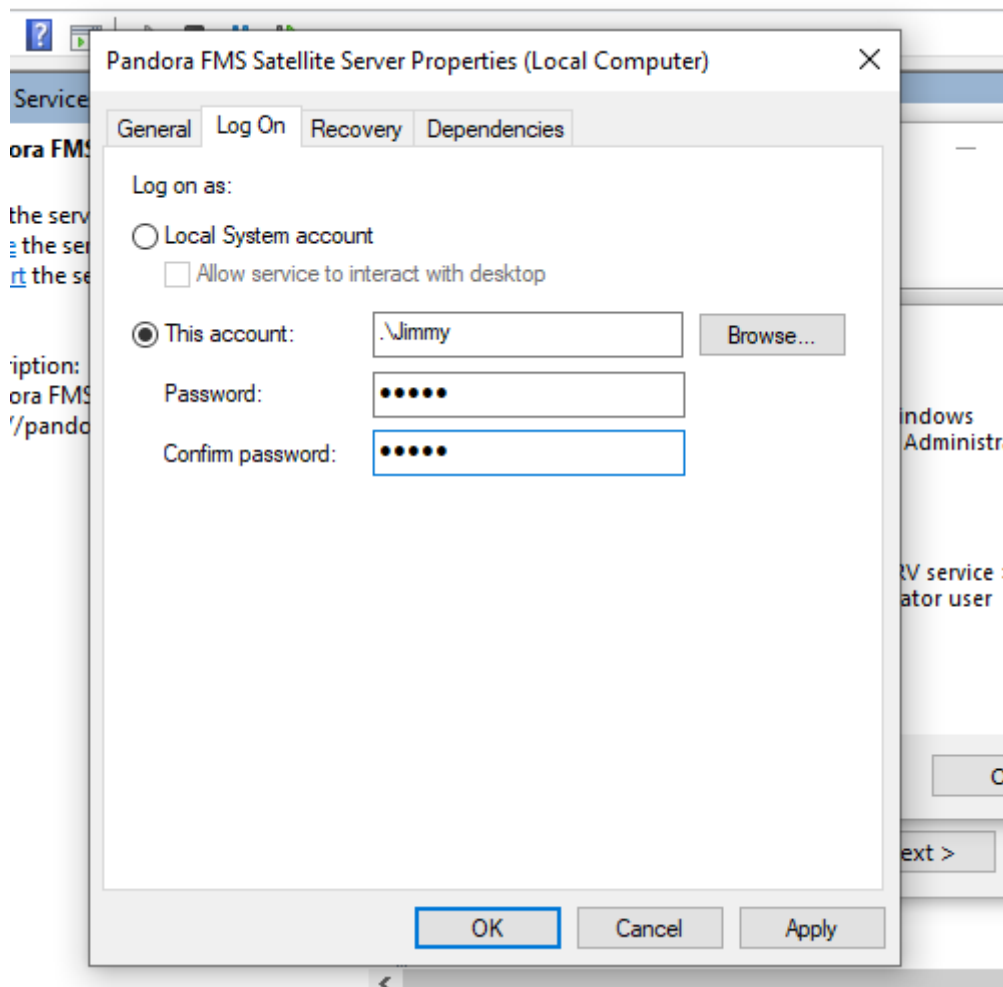


サービスを右クリックし、プロパティに入ります。



ログオンウィンドウで、Administrator 権限のあるアカウント選択し、変更を適用します。





そして、以降の対応を行い、サービスを再起動します。

## 設定

すべてのパラメータにおいて、タイムアウト値やいくつかの時間は秒単位で指定する必要があります。例えば、5分は 300 です。

遅延と snmp を使った監視では、状態変化が特別であるということを理解しておくことが重要です。二値(boolean)監視(ポートやマシンの生き死に)では、状態変化は自動で行われます。数値(遅延、各インタフェースのネットワークトラフィック、ディスクの空き、CPUなど)では、それぞれのモジュールで定義された閾値に基づいています。

### agent\_interval

```
agent_interval xxx
```

デフォルトでは 300 秒 (5分) です。ネットワークサーバにてより短い間隔でチェックが行われても、それには関係なく、この間隔で情報はサーバーに送信されます。必要に応じて、デフォルトで、指定された時間に従って、対応する Pandora FMS サーバにエージェントが作成されます。

収集したデータが以前のものと異なる場合は、すぐに送信します。同じであれば、エージェントの実行間隔で送信します。集中的なチェックを実行し、状態が変化した場合にのみ通知するのに便利です。

## agent\_threads

```
agent_threads xxx
```

エージェントの XML データファイルを送信するのに利用するスレッドの数です。

## log\_file

```
log_file <ファイルのパス>
```

サテライトサーバのログファイルです。デフォルトのパスは /var/log/satelite\_server.log です。

## recon\_task

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

自動検出のための IP アドレスおよびネットワークアドレスです。以下に例を示します。

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

## server\_ip

```
server_ip <IP>
```

情報を送信する先の Pandora FMS サーバの IP アドレスまたは名前です。tentacle プロトコルを使うため、そのポート番号(41121/tcp)でアクセスできる必要があります。

## recon\_mode

```
recon_mode <自動検出モード>
```

自動検出モードです。<自動検出モード> には、次のプロトコルを指定できます。

- icmp: ホストの応答と遅延時間のみを確認します。
- snmp: SNMP(v1 および v2 のみ)で通信できる場合、すべてのインターフェイスを検索し、すべてのインターフェイスからトラフィックを取得します。また、動作ステータス、デバイス名、場所も取得します。設定ファイルで定義されているさまざまなコミュニティを試して接続します。

- wmi: 上記と似ていますが、CPU 利用率、メモリおよびディスクドライブ(存在するものすべて)を表示します。

## recon\_community

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

SNMP の自動検出で利用する SNMP コミュニティのカンマ区切りの一覧です。SNMP 探索でこのリストを使用します。見つかった IP ごとに、これらのコミュニティのいずれかに応答するかどうかを確認します。

## wmi\_auth

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

User%Password の一覧です。例えば admin%1234,super%qwerty です。この一覧は WMI の自動検出で利用されます。見つかった IP ごとに、これらの組み合わせのいずれかに応答するかどうかを確認します。

## wmi\_ntlmv2

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

It enables (1) or disables (0) authentication with [protocol NTLMv2](#) for WMI.

WMI の [NTLMv2 プロトコル](#) による認証を有効化(1)または無効化(0)します。

## agent\_conf\_dir

```
agent_conf_dir <パス>
```

<パス> に指定したディレクトリに、サテライトサーバが検出したエージェントの設定ファイルが保存されます。デフォルトでは /etc/pandora/conf です。後述するように、手動で作成することもできます。

## group

```
group <グループ名>
```

サテライトサーバで作成されたエージェントのデフォルトグループの指定です。例えば "Servers" です。

## daemon

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

1 に設定すると、デーモンとしてバックグラウンドで起動します(デフォルト)。

## hostfile

ネットワークスキャンの別手法です。各行にアドレスを書いたファイルを用意します。ファイルには、各行にアドレスを記載します。IP アドレスに続けてホスト名を書くこともでき、エージェントはその名前で作成され、IP はモジュールでのアクセスに使用します(例：193.168.0.2 ホスト名)。正しく動作させるためには、各アドレスに fping を送信可能でなければなりません。

## pandora\_license\_key

バージョン NG 765 以上

```
# Encryption key for the Pandora FMS license.  
# pandora_license_key
```

ライセンスをサテライトサーバに安全に送信するには、このトークンで設定するのと同じ暗号化キーを [Web コンソール](#) または [メタコンソール](#) で設定する必要があります。

[server\\_ip](#) トークンについても確認してください。

## pandora\_license

バージョン 761 以降では、サテライトサーバのライセンスは自動的に設定され、このトークンは廃止となりました。

```
pandora_license xxxxxxxx
```

Pandora FMS コンソールの [セットアップ](#) [ライセンス](#) のメニューで表示される Pandora FMS サーバのライセンス番号を入力する必要があります。

The screenshot displays the Pandora FMS web interface. The top left features the Pandora FMS logo and a search bar. The main navigation menu is on the left, with 'Management' selected. The 'License' menu item is highlighted. The main content area shows the 'License management' page with the following details:

- License**
- Customer key**: ARTICAQA0000Z6GN8PJ00WONPW6GCNPW6DZFRW8GZF9TPW7J4F  
DUWUNKRV58D1RGWWNKRKRV1DG5IFRS0KKV5CP2FFRXOLHV5CG4GB  
FBQSMGDRW8D0FBQSMGDRW8D0FBQSMGDRW8D0FBQZ56J4KLVV
- Support expires**: 2023/10/03
- Current platform count**: 280 agents
- Current platform count (disabled: items)**: 2 agents
- NMS**: disabled

ライセンスを使用するエージェントの合計は、サテライトサーバではなく Pandora FMS サーバで確認されるため、必要な数のサテライトサーバで同じライセンスを使用できます。

## remote\_config

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

デフォルトでは、検出されたエージェントのリモート設定が有効になります。それらを検出した後、コンソールから管理したい場合には必須です。また、サテライトサーバのリモート設定も有効化します。より詳細については、[ソフトウェアエージェントを使ったモニタリング](#)を参照してください。

## temporal\_min\_size

```
temporal_min_size xxx
```

テンポラリディレクトリのパーティション空き容量がここで指定したサイズ (MB 単位) より小さくなったら、データパケットの生成を停止します。何らかの理由でサーバへの接続が長時間切れた場合に、ディスクがいっぱいになるのを避けます。

## xml\_buffer

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

デフォルトは 0 です。1 に設定すると、送信できなかった XML データファイルを保存します。

セキュアな環境の UNIX で XML バッファを有効にしたい場合は、テンポラリディレクトリの変更を検討してください。'/tmp' は誰もが書き込めます。

## snmp\_version

```
snmp_version xx
```

デフォルトで利用する SNMP バージョンです(1 および 2c のみ対応しています)。デフォルトは 1 です。SNMP v3を使用する場合は、アクセス権限情報を設定する方法について[こちら](#)を確認してください。

この設定を変更すると、いくつかのモジュールは動作しなくなります。

## braa

```
braa <パス>
```

braa バイナリのパスです。(デフォルトは /usr/bin/braa)

## fping

```
fping <path>
```

fping バイナリのパスです。(デフォルトは /usr/sbin/fping)

## fsnmp

```
fsnmp <パス>
```

fsnmp バイナリ(SNMPv3)のパス(<パス>)です。デフォルトは /usr/bin/pandorafsnmp です。

## latency\_packets

```
latency_packets xxx
```

遅延要求ごとに送信する ICMP パケット数です。

## nmap

```
nmap <パス>
```

nmap バイナリのパス(<パス>)です。デフォルトは /usr/bin/nmap です。

## nmap\_timing\_template

```
nmap_timing_template x
```

x には、1 から 5 で nmap がどの程度の精度で動作するかを指定します。1 は遅いですが信頼性が高く、5 は早いですが信頼性は低くなります。デフォルトは 2 です。

## ping\_packets

```
ping_packets xxx
```

ping リクエストごとに送信する ICMP パケット数です。

## recon\_enabled

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

動検出を有効化(1)または無効化(0)する設定です。

## recon\_timing\_template

```
recon_timing_template xxx
```

`nmap_timing_template` のように、ネットワーク検出に適用されます。

### **server\_port**

```
server_port xxxxx
```

tentacle サーバのポート番号です。

### **server\_name**

```
server_name xxxxx
```

サテライトサーバの名前です。(デフォルトはマシンのホスト名です)

### **server\_path**

```
server_path <path>
```

`transfer_mode` が local の場合の XML ファイルがコピーされるパスです。(デフォルトは /var/spool/pandora/data\_in)

### **server\_opts**

tentacle に渡されるサーバパラメータです。

### **transfer\_mode**

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

ファイル転送モードです。tentacle または local を指定できます。(デフォルトは tentacle)

### **snmp\_verify**

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

braa でのリアルタイムの実行を妨げる SNMPv1 モジュールの確認を有効化(1)または無効化(0)します。これらのモジュールは、実行が停止されます。



## snmp2\_verify

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

braa でのリアルタイムの実行を妨げる SNMPv2 モジュールの確認を有効化(1)または無効化(0)します。これらのモジュールは、実行が停止されます。

SNMP バージョン 2 モジュールの確認は非常に遅くなる可能性があります。

## snmp3\_verify

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

braa でのリアルタイムの実行を妨げる SNMPv3 モジュールの確認を有効化(1)または無効化(0)します。これらのモジュールは、実行が停止されます。

SNMP v3を使用する場合は、アクセス権限情報を設定する方法について[こちら](#)を確認してください。

## snmp3\_seclevel

SNMPv3 メッセージのセキュリティレベル(noauth, authnopriv または authpriv)です。

SNMP v3を使用する場合は、アクセス権限情報を設定する方法について[こちら](#)を確認してください。

## snmp3\_secname

SNMPv3 メッセージのセキュリティ名です。

SNMP v3を使用する場合は、アクセス権限情報を設定する方法について[こちら](#)を確認してください。

## snmp3\_authproto

SNMPv3 リクエストの認証プロトコル(md5 または sha)です。

SNMP v3を使用する場合は、アクセス権限情報を設定する方法について[こちら](#)を確認してください。

## snmp3\_authpass

SNMPv3 リクエストの認証パスワードです。

SNMP v3を使用する場合は、アクセス権限情報を設定する方法について[こちら](#)を確認してください。

### **snmp3\_privproto**

SNMPv3 リクエストの暗号化プロトコル(des または aes)です。

SNMP v3を使用する場合は、アクセス権限情報を設定する方法について[こちら](#)を確認してください。

### **snmp3\_privpass**

SNMPv3 メッセージの暗号化パスワードです。

SNMP v3を使用する場合は、アクセス権限情報を設定する方法について[こちら](#)を確認してください。

### **startup\_delay**

```
startup_delay xxx
```

初回の XML データファイル送信を startup\_delay に指定した秒間遅らせます。

### **temporal**

```
temporal <ディレクトリ>
```

XML ファイルが生成されるテンポラリディレクトリ。デフォルトは /tmp です。

### **tentacle\_client**

```
tentacle_client <パス>
```

tentacle クライアントのフルパスです。(デフォルトは /usr/bin/tentacle\_client)

### **wmi\_client**

```
wmi_client <パス>
```

WMI クライアントバイナリのフルパスです。デフォルトは /usr/bin/wmic です。

```
euclides root ~ /usr/bin/wmic --usage
Usage: [-?|--help] [--usage] [-d|--debuglevel DEBUGLEVEL] [--debug-stderr]
       [-s|--configfile CONFIGFILE] [--option=name=value]
       [-l|--log-basename LOGFILEBASE] [--leak-report] [--leak-report-full]
       [-R|--name-resolve NAME-RESOLVE-ORDER]
       [-O|--socket-options SOCKETOPTIONS] [-n|--netbiosname NETBIOSNAME]
       [-W|--workgroup WORKGROUP] [--realm=REALM] [-i|--scope SCOPE]
       [-m|--maxprotocol MAXPROTOCOL] [-U|--user [DOMAIN\]USERNAME[%PASSWORD]]
       [-N|--no-pass] [--password=STRING] [-A|--authentication-file FILE]
       [-S|--signing on|off|required] [-P|--machine-pass]
       [--simple-bind-dn=STRING] [-k|--kerberos STRING]
       [--use-security-mechanisms=STRING] [-V|--version] [--namespace=STRING]
       [--delimiter=STRING]
       //host query

Example: wmic -U [domain/]adminuser%password //host "select * from Win32_ComputerSystem"
euclides root ~
```

## snmp\_blacklist

```
snmp_blacklist <パス>
```

SNMP モジュールブラックリストファイルのパスです。(デフォルトは /etc/pandora/satellite\_server.blacklist)

## add\_host

```
add_host <IPアドレス> [エージェント名]
```

監視対象エージェントの一覧にホストを追加します。IP アドレスの後にエージェント名を指定できます。1行に一ホストで複数のホストを追加できます。例:

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

## ignore\_host

```
ignore_host <エージェント名>
```

自動検出タスクで見つかったものを含めて、監視対象エージェントの一覧から指定のホストを削除します。ホストはエージェント名で指定する必要があります。1行に一ホストで複数のホストを除外できます。例:

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

## delete\_host

```
delete_host <agent_name>
```

設定ファイルを削除することにより、監視対象のエージェントのリストから指定されたホストを完全に削除します。ホストはエージェント名で指定される必要があります。1行に1つずつ書くことにより複数のホストを削除できます。例:

```
delete_host 192.168.0.1
delete_host localhost.localdomain
```

## keepalive

```
keepalive xxx
```

ここで設定された秒数ごとに、サテライトサーバは Pandora Server に接続して状態を報告し、(ソフトウェアエージェントおよびサテライトサーバ自身の)リモート設定による更新が無いかを確認します。デフォルトは 30 秒です。

## credential\_pass

```
credential_pass xxx
```

credential box に記述したパスワードの暗号化に使ったパスワードを指定します。コンソールで定義されているパスワードに一致しなければなりません。デフォルトではホスト名が使われます。

## timeout\_bin

```
timeout_bin <パス>
```

定義すると `tentacle` クライアントを呼び出すときにタイムアウトコマンド(通常は `/usr/bin/timeout`)が使われます。

## timeout\_seconds

```
timeout_seconds xxx
```

タイムアウトコマンドのタイムアウト秒数です `timeout_bin` が設定されている必要があります。

## proxy\_traps\_to

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

サテライトサーバで受信した SNMP トラップを、指定のアドレス(およびポート)にプロキシします。

デフォルトでは 162番ポートが利用されます。

## proxy\_tentacle\_from

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

tentacle サーバが受け付けるアドレスとポート番号です。デフォルトではポート番号は 41121 が使われます。

## proxy\_tentacle\_to

```
proxy_tentacle_to <IP[:port]>
```

サテライトサーバで受信した Tentacle クライアントリクエストを指定のアドレス(およびポート)にプロキシします。デフォルトでは 41121番ポートが利用されます。

このオプションは、リモートエージェント設定と競合することがあります。

これは、サテライトサーバが一部のソフトウェアエージェントのプロキシとして使用され、サテライトサーバ自体(ICMP/SNMPなど)から遠隔で監視され、両方でリモート設定が有効になっている場合に発生します。

このような場合は、実行されたチェックに異なるエージェントを使用するか、異なるエージェント名を使用するか、リモート設定を2つのサテライト・サーバーまたはソフトウェア・エージェントのいずれか一方のみに有効にしておく必要があります。

## dynamic\_inc

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

1 に設定すると、リモートエージェント設定の動作を妨げないように、動的に自動検出したモジュール(SNMP/WMI...)を別のファイルに移動します。

## vlan\_cache\_enabled

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

自己検出ホストで VLAN キャッシュを有効化(1)または無効化(0)します。

## verbosity

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

ログレベル 0(最も少ない) から 10(最も多い)。

## agents\_blacklist\_icmp

バージョン NG 713 以上

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

ICMP チェックのブラックリストです。このフィールドは、ICMP タイプのモジュールが実行されないようにするために CIDR 表記を使用して IP のリストを設定することができます。カンマ区切りで複数のサブネットを指定できます。

## agents\_blacklist\_snmp

バージョン NG 713 以上

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

SNMP チェックのブラックリストです。このフィールドは、SNMP タイプのモジュールが実行されないようにするために CIDR 表記を使用して IP のリストを設定することができます。カンマ区切りで複数のサブネットを指定できます。

## agents\_blacklist\_wmi

バージョン NG 713 以上

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

WMI チェックのブラックリストです。このフィールドは、WMI タイプのモジュールが実行されないようにするために CIDR 表記を使用して IP のリストを設定することができます。カンマ区切りで複数のサブネットを指定できます。

## general\_gis\_exec

バージョン NG 713 以上

```
general_gis_exec xxx
```

サテライトサーバによって検出される全エージェントの位置設定スクリプト。スクリプトは実行可能で、<経度>,<緯度>,[<高度>] というフォーマットを出力する必要があります。3つ目のパラメータはオプションです。

## forced\_add

```
force_add [0|1]
```

1 に設定すると、([host\\_file](#) または [add\\_host](#) を介して)モジュール無しの設定で手動で追加されたホストは、ping に応答しない場合でも常に作成されます。

## agent\_block

```
agent_block XX
```

Tentacle クライアントの 1 回の呼び出しで送信される XML データ ファイルの数。デフォルトは 50 です。

## conf\_interval

```
conf_interval XXX
```

リモート設定チェック間隔。デフォルトは 300 秒です。

## exec\_interval

```
exec_interval XXX
```

監視実行間隔。デフォルトは 300 秒です。

## exec\_threads

```
exec_threads X
```

モジュール実行に利用するスレッド数。デフォルトは 5 です。

マシンの容量(CPU と RAM)に依存します。スレッド数が多いほど、マシンの負荷は高くなりますが、処理能力は高くなります。システムによっては、20 スレッドを超えるとパフォーマンスが低下する場合があります。

## latency\_block

```
latency_block XXX
```

単一の nmap (レイテンシ)呼び出しで処理されるホスト数です。デフォルトは 400 です。

これは、サーバで XXX 個の監視をブロック実行させます。数値が大きいほど(最大 500)より多くの性能が得られますが、遅延が増加します。場合によっては、その数値を下げることをお勧めします。

## latency\_interval

```
latency_interval XXX
```

レイテンシ監視間隔です。デフォルトは 180 秒です。

## latency\_retries

```
latency_retries X
```

レイテンシモジュールのリトライ数です。デフォルトは 2 です。

## latency\_threads

```
latency_threads X
```

レイテンシ監視に使うスレッド数です。デフォルトは 4 です。

## latency\_timeout

```
latency_timeout X
```

秒単位でのレイテンシ監視のタイムアウトです。デフォルトは 1 です。



## ping\_block

```
ping_block XXX
```

単一の nmap (ping) 呼び出しで処理するホストの数です。デフォルトは 400 です。

これは、サーバで XXX 個の監視をブロック実行させます。数値が大きいくほど(最大 500)より多くの性能が得られますが、遅延が増加します。場合によっては、その数値を下げることをお勧めします。

## ping\_interval

```
ping_interval XXX
```

ping 監視の実行間隔。デフォルトは 120 秒です。

## ping\_retries

```
ping_retries X
```

ping 監視モジュールのリトライ回数です。デフォルトは 2 です。

## ping\_threads

```
ping_threads X
```

ping 監視に利用するスレッド数です。デフォルトは 4 です。

## ping\_timeout

```
ping_timeout 1
```

ping 監視の秒単位のタイムアウト値です。デフォルトは 1 です。

## plugin\_interval

```
plugin_interval XXX
```

プラグインの監視間隔です。デフォルトは 300 です。

## plugin\_threads

```
plugin_threads X
```

プラグイン監視に利用するスレッド数です。デフォルトは 2 です。

## plugin\_timeout

```
plugin_timeout XX
```

プラグイン監視の秒単位のタイムアウト値です。デフォルトは 10 です。

## recon\_interval

```
recon_interval XXXXXX
```

秒単位のネットワークスキャン間隔です。デフォルトは 604800 です。

## snmp2\_block

```
snmp2_block XX
```

単一の braa (SNMPv2c) 呼び出しにおける処理ホスト数。デフォルトは 50 です。

## snmp2\_interval

```
snmp2_interval XXX
```

SNMP 監視 (SNMPv2c) の間隔です。デフォルトは 100 秒です。

## snmp2\_retries

```
snmp2_retries X
```

Number of retries for SNMP modules (SNMPv2c), 2 by default.

SNMP モジュール (SNMPv2c) のリトライ回数です。デフォルトは 2 です。

## snmp2\_threads

```
snmp2_threads X
```

---

SNMP 監視 (SNMPv2c) で使われるスレッド数です。デフォルトは 8 です。

### **snmp2\_timeout**

```
snmp2_timeout X
```

SNMP (SNMPv2c) 監視の秒単位のタイムアウト値です。デフォルトは 5 です。

### **snmp3\_block**

```
snmp3_block XX
```

単一の braa (SNMPv3) 呼び出しにおいて処理するホスト数です。デフォルトは 50 です。

### **snmp3\_interval**

```
snmp3_interval XXX
```

SNMP 監視 (SNMPv3) の間隔です。デフォルトは 180 です。

### **snmp3\_retries**

```
snmp3_retries X
```

SNMP モジュール (SNMPv3) のリトライ回数です。デフォルトは 2 です。

### **snmp3\_threads**

```
snmp3_threads X
```

SNMP 監視 (SNMPv3) で使われるスレッド数です。デフォルトは 4 です。

### **snmp3\_timeout**

```
snmp3_timeout X
```

SNMP (SNMPv3) 監視の秒単位のタイムアウト値です。デフォルトは 5 です。

## snmp\_block

```
snmp_block XX
```

単一の braa (SNMPv1) 呼び出しで処理するホスト数。デフォルトは 50 です。

## snmp\_interval

```
snmp_interval XXX
```

SNMP 監視 (SNMPv1) の実行間隔です。デフォルトは 100 です。

## snmp\_retries

```
snmp_retries X
```

SNMP モジュール (SNMPv1) のリトライ回数です。デフォルトは 2 です。

## ssh\_interval

```
ssh_interval XXX
```

SSH 監視の実行間隔です。デフォルトは 300 秒です。

## ssh\_threads

```
ssh_threads X
```

SSH モジュールで利用するスレッド数です。デフォルトは 5 です。

## ssh\_timeout

```
ssh_timeout X
```

SSH 監視の秒単位のタイムアウト値です。デフォルトは 2 です。

## tcp\_interval

```
tcp_interval XXX
```

TCP 監視の実行間隔です。デフォルトは 300 秒です。

### **tcp\_threads**

```
tcp_threads X
```

TCP 監視に利用するスレッド数です。デフォルトは 5 です。

### **tcp\_timeout**

```
tcp_timeout X
```

TCP 監視の秒単位のタイムアウト値です。デフォルトは 1 です。

### **snmp\_threads**

```
snmp_threads X
```

SNMP 監視 (SNMPv1) で利用するスレッド数です。デフォルトは 8 です。

### **snmp\_timeout**

```
snmp_timeout X
```

SNMP 監視 (SNMPv1) の秒単位のタイムアウト値です。デフォルトは 5 です。

### **wmi\_interval**

```
wmi_interval XXX
```

WMI 監視の実行間隔です。デフォルトは 300 秒です。

### **wmi\_threads**

```
wmi_threads X
```

WMI 監視で利用するスレッド数です。デフォルトは 5 です。

## ipam\_task

```
ipam_task <id IPAM TASK> , <CIDR>
```

IPAM をスキャンするネットワーク(スラッシュ表記)のカンマ区切りリスト[] Pandora FMS の IPAM タスクの ID を先頭につける必要があります(自動検出サーバフィールドは 未割り当て の状態で、サテライトサーバに割り当てする必要があります)。例: 1,192.168.0.0/24

## ipam\_interval

```
ipam_interval XXXXXX
```

IPAM スキャンの秒単位の実行間隔。

## セカンダリサーバ

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

特殊な設定パラメータとしては、セカンダリサーバの設定があります。これにより、データを送信する別のサーバを設定します。セカンダリサーバモードには、二種類の動作があります。

- on\_error: プライマリサーバにデータを送信出来なかった場合のみ、セカンダリサーバにデータを送信します。
- always: プライマリサーバにデータが送信できるかどうかに関わらず、常にセカンダリサーバにもデータを送信します。

設定例:

```
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
```

## リモート設定

### リモートファイル設定

メニュー サーバ(Servers) → サーバ管理(Manage servers) を介して、サテライトサーバが属する Pandora FMS Enterprise サーバで、サテライトサーバのリモート設定用の高度なエディターにアクセスできます[] Web ブラウザにページをロードしたら、リモート設定 アイコンをクリックします。

Pandora FMS  
the Flexible Monitoring System

Enter keywords 🔍

Servers / Manage Servers  
Pandora FMS servers

pandorafms	🟢	📶	📘	7.0NG.770 (P) 230505	0 of 0	-/0	4:0	3 seconds	
pandorafms	🟢	📶	📘	7.0NG.770 (P) 230505	0 of 0	-/0	2:0	3 seconds	
pandorafms	🟢	📶	📘	7.0NG.770 (P) 230505	0 of 0	-/0	1:0	3 seconds	
pandorafms	🟢	📶	📘	7.0NG.770 (P) 230505	0 of 0	-/0	1:0	3 seconds	
pandorafms	🟢	📶	📘	7.0NG.770 (P) 230505	0 of 0	-/0	1:0	3 seconds	
pandorafms	🟢	📶	📘	7.0NG.770 (P) 230505	0 of 0	-/0	1:0	3 seconds	
pandorafms	🟢	📶	📘	7.0NG.770 (P) 230505	0 of 0	-/0	1:0	3 seconds	
pandorafms	🟢	📶	📘	7.0NG.770 (P) 230505	N/A	N/A	1:1	3 seconds	
pandorafms	🟢	📶	📘	7.0NG.770 (P) 230505	0 of 0	-/0	4:0	3 seconds	
pandorafms	🟢	📶	📘	7.0NG.770 (P) 230505	0 of 0	-/0	4:0	3 seconds	
pandorafms	🟢	📶	📘	7.0NG.770 (P) 230505	0 of 0	-/0	0:0	14 seconds	🔧 🔄 📶 🗑️

Remote configuration

そして、高度なエディタ(Advanced editor) アイコンをクリックします。

Pandora FMS  
the Flexible Monitoring System

Enter keywords 🔍

Servers / Pandora FMS servers  
Remote configuration ⓘ

Advanced editor

Delete remote conf server files 🗑️ ⓘ

Configuration

```
# -----
# Pandora FMS Satellite Server configuration file
# (c) 2014-2023 Pandora FMS
# -----

# Directory where agent configuration files are stored.
agent_conf_dir /etc/pandora/conf

# Number of XML data files sent in a single call to the Tentacle client.
agent_block 50
```

設定(Configuration) に対応するテキストボックスで、前のセクションで説明した各トークンを編集および/または追加できます。編集が終了したら、ページの下部にある 更新(Update) ボタンをクリック

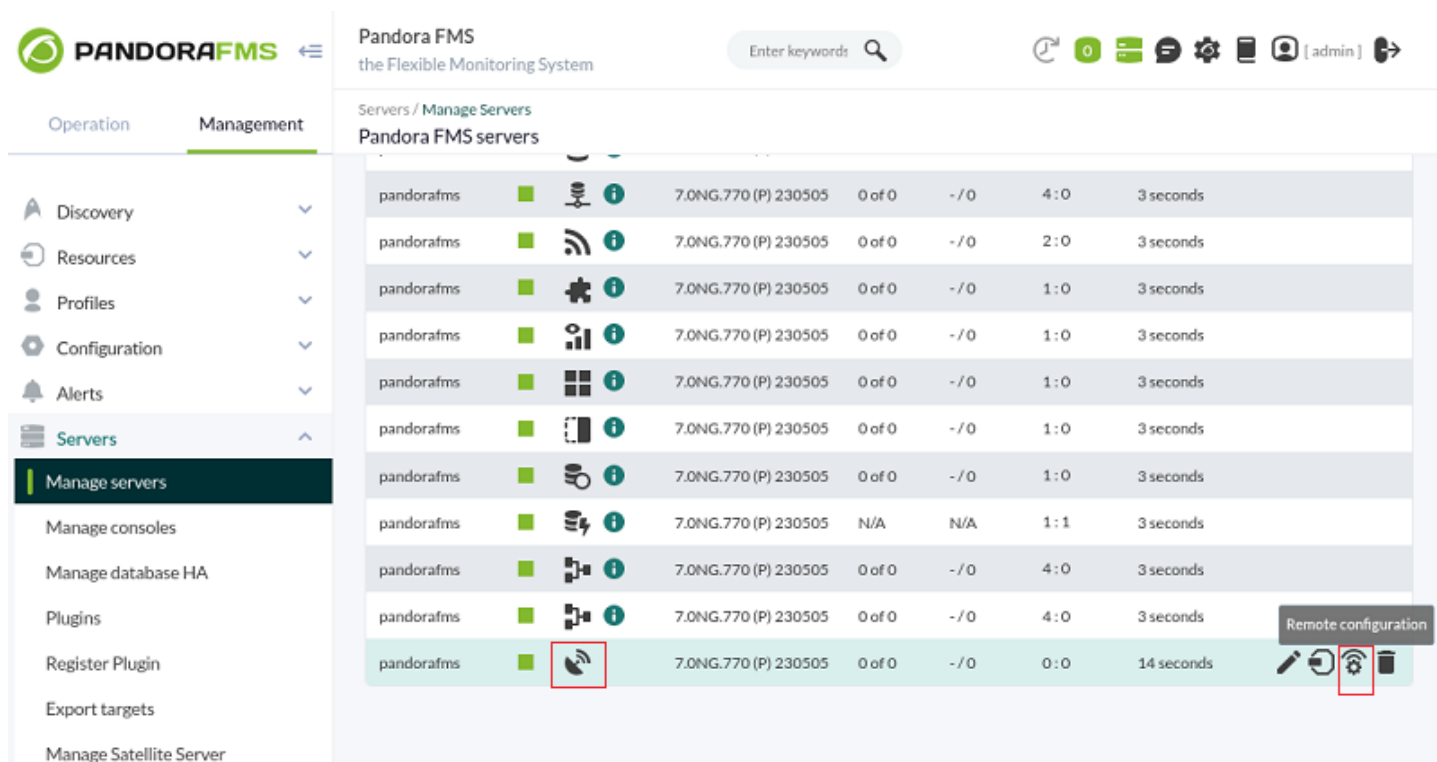
クして変更を保存します。

新しいトークンの同期と読み込みには時間がかかります。  
変更が反映されるまでしばらくお待ちください。

## リモート設定グラフィカルインタフェース

バージョン NG 764 以降

メニュー サーバ(Servers) → サーバ管理(Manage servers) を選択し、リモート設定(Remote configuration) アイコンをクリックすると、サテライトサーバが属する Pandora FMS Enterprise サーバのグラフィカルインタフェースにアクセスできます。



The screenshot shows the Pandora FMS web interface. The top navigation bar includes the Pandora FMS logo, a search bar, and user information. The left sidebar shows the 'Servers' menu with 'Manage servers' selected. The main content area displays a table of Pandora FMS servers. The table has columns for server name, status, IP address, version, and configuration. A 'Remote configuration' button is visible next to the last server entry.

Server Name	Status	IP Address	Version	Configuration
pandorafms	Online	7.0NG.770 (P) 230505	0 of 0	-/0
pandorafms	Online	7.0NG.770 (P) 230505	0 of 0	-/0
pandorafms	Online	7.0NG.770 (P) 230505	0 of 0	-/0
pandorafms	Online	7.0NG.770 (P) 230505	0 of 0	-/0
pandorafms	Online	7.0NG.770 (P) 230505	0 of 0	-/0
pandorafms	Online	7.0NG.770 (P) 230505	0 of 0	-/0
pandorafms	Online	7.0NG.770 (P) 230505	N/A	N/A
pandorafms	Online	7.0NG.770 (P) 230505	0 of 0	-/0
pandorafms	Online	7.0NG.770 (P) 230505	0 of 0	-/0
pandorafms	Online	7.0NG.770 (P) 230505	0 of 0	-/0
pandorafms	Online	7.0NG.770 (P) 230505	0 of 0	-/0
pandorafms	Online	7.0NG.770 (P) 230505	0 of 0	-/0

動的検索(Dynamic search) を使用して、キーテキスト(1文字以上)を入力し、特定のトークンを検索します。



Pandora FMS  
the Flexible Monitoring System

Enter keywords

Servers / Pandora FMS servers  
Remote configuration ?

Operation Management

Discovery Resources Profiles Configuration

Dynamic search

一部のトークンは、2つの値 (ON /OFF) のみを受け入れます。この値を変更するには、最初のボタンを使用します。2番目のボタンで対応するトークンを有効または無効にすると、サテライトサーバは最初のボタンで設定された値に関係なくそれに割り当てられたデフォルト値 (1 または 0) を使用します。

Pandora FMS  
the Flexible Monitoring System

Enter keywords

Servers / Pandora FMS servers  
Remote configuration ?

Dynamic search

General server configuration

VLAN cache enabled ?  
  
Default value: 1

Enable XML buffer  
  
Default value: 0

すべての変更を行ったら、更新(Update) ボタンをクリックして設定を保存します。

## サテライトサーバでのエージェント作成

サテライトサーバでのエージェント作成には、自動検出タスク []Satelite\_hosts.txt ファイル、エージェントの .conf ファイルの手動作成の3通りがあります。

### 自動検出タスクでのエージェント作成

自動検出タスクを通してのエージェント作成は、Pandora FMS ユーザに最も使われている方法です。

これをできるようにするには、サテライトサーバ設定ファイルで次のパラメータを設定する必要があります。

- recon\_community: カンマ区切りで SNMP 検出で利用する SNMP コミュニティを指定します (snmp タイプの自動検出を実行する場合)
- recon\_enabled: サテライトサーバの自動検出タスクを有効化するには、1 に設定する必要があります。
- recon\_interval: 秒単位のネットワークをスキャンする間隔です。(デフォルトは 604800 秒です。)
- recon\_mode: カンマ区切りで自動検出タスクのモード (snmp,icmp,wmi) を指定します。
- recon\_task: カンマ区切りで自動検出したいネットワークの一覧を指定します。
- recon\_timing\_template: 1 から 5 で nmap をどの程度アグレッシブに動作させるかを指定します。1 はゆっくりですが信頼性があります。5 は、早いですが信頼性は低いです。(デフォルトは 3 です)

自動検出タスクの例を以下に示します。

```
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
```

設定を行ったら、以下のコマンドでサテライトサーバを実行します。

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

設定ファイルにモジュールが無いエージェントは、サテライトサーバに無視されます。

## ファイルを利用したエージェント作成

最初に、satellite\_hosts.txt ファイルを通してエージェントを作成するためには、サテライトサーバの設定ファイルで以下の行のコメントを外す必要があります。

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

次に、作成したいエージェントの IP と名前を記載した satellite\_hosts.txt ファイルを作成します。

```
192.168.10.5 Server.5
192.168.10.6 Server.6
192.168.10.7 Server.7
```

指定の IP で作成されるためには、一覧に記載した IP に対して fping ができる必要があります。そうでないと作成されません。

設定が完了したら、次のコマンドでサテライトサーバを実行します。

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

指定のファイルの読み込みは `recon_interval` で指定した秒ごとに行われます。

## 手動エージェント作成

最初に、サテライトサーバの設定ファイルの `agent_conf_dir` パラメータを見ます。ここが、新たに作成するエージェントの設定ファイルを置く場所です (`/etc/pandora/conf`)

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

パスを確認したら、例に示すような作成したいエージェントの `.conf` ファイルを作成するだけです。次のフィールドは手動で設定します。

- `agent_name`: エージェント名を設定します。
- `agent_alias`: エージェントの別名を設定します。
- `address`: 監視したいエージェントの IP を設定します。
- `group`: エージェントを割りあてるグループを設定します。
- `gis_exec`: 位置を決定するスクリプトです。これは `general_gis_exec` の指定を上書きします。
- `modules`: 自動検出タスクを実行するときに作成したいモジュールを指定します。

例を以下に示します:

```
agent_name Example
agent_alias This is an example
address 127.0.0.1
group Servers

module_begin
module_name Ping
module_ping
module_end

module_begin
module_name Latency
module_latency
module_end
```

設定を行ったら、次のコマンドでサテライトサーバを起動します。

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

## サテライトサーバでのエージェント削除

サテライトサーバからのエージェント削除には、全エージェントの削除か一部のエージェントの削除があります。

実行の前に、フォルダおよびファイルをバックアップしてください。

全てのエージェントの場合は、エージェントをどのように作成したかを考慮する必要があります。

- 手動: 最初に、/etc/pandora/conf フォルダに作成したエージェントの .conf ファイルを削除し、次にコンソールからエージェントを削除します。
- Satellite\_hosts.txt ファイル: /etc/pandora/conf フォルダに作成したファイルだけでなく file.txt を削除する必要があります。その後、コンソールからエージェントを削除します。
- 自動検出タスク: サテライトサーバの設定ファイルから、自動検出タスクを無効化する必要があります。そして /etc/pandora/conf フォルダに作成された conf を削除し、コンソールからエージェントを削除します。

一部の削除でも、エージェントをどのように作成したかを考慮する必要があります。

- 手動: 最初に、/etc/pandora/conf フォルダに作成したエージェントの .conf ファイルを削除し、その後コンソールからエージェントを削除します。
- Satellite\_hosts.txt ファイル: /etc/pandora/conf フォルダに作成した対象エージェントの conf だけでなく file.txt 内の削除したい IP の行を削除します。その後、コンソールからエージェントを削除します。
- 自動検出タスク: サテライトサーバの設定ファイル内の recon\_task のブラックリストを設定する必要があります。そして /etc/pandora/conf 内の該当の conf ファイルを削除し、コンソールから対象のエージェントを削除します。

## エージェントによるカスタム設定

自動検出モジュールに加えて、ソフトウェアエージェントのモジュールに似たような書式で TCP, SNMP, WMI のすべての種類の監視を追加することができます。以下は、システムによって検出された後に自動生成される、サテライトサーバに有効なモジュールの例です。

### ICMP/TCP クエリ

マシンへの接続性(PING の利用):

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

ポートチェック(TCP の利用):

```
module_begin
```

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

## WMI クエリ

WMI CPU 使用量チェック(パーセント):

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

WMI 空きメモリチェック(パーセント):

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

一般的な WMI クエリ:

```
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
```

## SNMPv1 および SNMPv2 クエリ

OID はドットから始まっていることを確認してください。  
そうでないと SNMP モジュールが動作しません。

インタフェースのステータスです(SNMP) サテライトサーバはそれぞれのインタフェースを自動検出  
します。

```
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
```

モジュールで SNMP バージョン 2c を使うには、次の行を追加します。

```
module_version 2c
```

モジュールで SNMP バージョン 1 を使うには、次の行を追加します。

```
module_version 1
```

例:

```
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
```

一般的な SNMP 監視です。この場合、サーバはそれぞれのインタフェースのトラフィックを指定した名前で自動的に展開します。

```
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
```

## SNMPv3

SNMPv3 モジュールを設定するには、`module_version` を 3 に設定し、以下を設定します。

- module\_seclevel: セキュリティレベル ( noauth, authnopriv または authpriv ).
- module\_secname: セキュリティ名
- module\_authproto: 認証プロトコル ( md5 または sha ).
- module\_authpass: 認証パスワード
- module\_privproto: 暗号化プロトコル ( aes または des ).
- module\_privpass: 暗号化パスワード(必要な場合)

以下に例を示します。

```
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
```

SNMPv3 の特定の設定内容は、モジュール設定の外に置くことでモジュール間で共有することができます。(サテライトの設定ファイルに移動することにより、エージェント間でも共有することができます)

```
agent_name snmp
```

```
address 127.0.0.1

seclvl 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
```

Pandora FMS Web コンソールから SNMP モジュール(SNMPv3 を含む)作成は、[こちらの動画](#)を確認してください。コンポーネントグループ(SNMPv3 を含む)の作成は、[SNMP ウィザード](#)を確認してください。

SNMPv3 のデフォルトのサテライトサーバ設定:

独自の値や認証情報を入力する必要があります。また、必要なプロトコルや暗号化方式を変更する必要があります。新しい設定値を有効にするには、サテライトサーバを再起動する必要があります。

```
# 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
```

## SSH クエリ

MS Windows® にインストールされたサテライトサーバでの SSH クエリは、まだ実装中です。Pandora FMS 開発チームが取り組んでいます。

一般的な SSH コマンド。

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

閾値を設定するには、次のようなモジュール設定(module\_min\_warning, module\_min\_critical)を行い、コンソールでモジュール設定を行う必要があります。

```
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
```

手動で実行モジュールを作成することができます。サテライトサーバが実行するスクリプトやコマンドは、事前に作成してサーバに置いておく必要があります。エージェントの module\_exec と同じように動作します。module\_exec の利用は、サテライトサーバの実行速度を低下させる可能性があります。

```
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
```

## プラグインクエリ

Pandora FMS バージョン 7 からプラグインを追加できます。プラグインはサテライトサーバが稼

働しているマシンで実行されることに注意してください。したがって、これらのプラグインで、監視したいリモートコンピュータに接続するための実装をする必要があります。以前のバージョンよりも優れている点は、その大きな柔軟性です。このようにして、前提条件やその他 `module_exec` ではできないことを実装することができます。書式はエージェントと同じです。プラグインの使用例は次のとおりです。

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

## 認証ボックス

秘密鍵と公開鍵で構成された鍵を用いた認証が使われていない場合、SSHモジュールが機能するには、ユーザ名 (<user>) とパスワード (<pass>) が必要です。これらは、次の形式の認証情報ボックス (`credential_box`) を使用して、メインの設定ファイル `satellite_server.conf` で設定します。

```
network/mask,username,password
```

```
network/mask,username,[[encrypted password|]]
```

MS Windows® にインストールされたサテライトサーバでの SSH クエリは、まだ実装中です。Pandora FMS 開発チームが取り組んでいます。

例:

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

認証情報ボックスは、制限の厳しいマスクから制限の緩いマスクの順に検索されます。

パスワードは、Blowfish を用いて ECB モードで暗号化できます。 `credential_pass` が定義されていることを確認してください。定義されていない場合、ホスト名がデフォルトの暗号化パスワードとして使用されます。暗号文の 16 進数表現は、二重括弧で囲む必要があります。

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

## 認証設定

鍵認証が正しく設定されていない場合、sshモジュールはユーザ(<user>)とパスワード(<pass>)による認証を必要とします。その認証情報は、サテライトサーバの設定ファイル `satellite_server.conf` に `credential_box` を使って次のようなフォーマットで設定します:

```
network/mask,username,password
```

```
network/mask,username,[[encrypted password]]
```

例:

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

credential\_box で記述された認証情報は、ネットワークマスクの適用範囲が狭いものから順に検索します。

blowfishのECBモードで暗号化したパスワードも記述できます。暗号化されたパスワードは、文字列を二重の括弧(サテライトサーバによる分散監視)で囲みます。暗号化に利用するパスワードは credential\_pass に記述してください (デフォルトではホスト名が使われます):

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

## コンソールでの全エージェントの表示

サテライトサーバの設定が正しくできると、エージェント詳細で次のような画面を見ることができます。

Agent	Description	Remote	OS	Interval	Group	Type	Modules	Status	Alert	Last contact
192.168.70.157	Created by SatServer			5 minutes			2:1:1			4 minutes 27 seconds
192.168.70.159	Created by SatServer			5 minutes			2:2			4 minutes 27 seconds
192.168.70.165	Created by SatServer			5 minutes			2:2			4 minutes 27 seconds
192.168.70.168	Created by SatServer			5 minutes			2:2			4 minutes 27 seconds

全マシンで ICMP モジュール(PING および遅延)が作成されていますが、いくつかのマシンでは SNMP および WMI モジュールもあります。

WMI が有効なマシンでは、次のようなモジュールが生成されます。

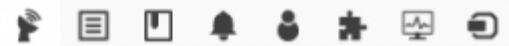
F.	P.	Type	Module name	Description	Status	Thresholds	Data	Graph	Last contact
			CPU Load	CPU Load (%)		N/A - N/A	21 %		101 39 seconds
			Free memory	Total free memory in kilobytes		N/A - N/A	7,635,884 KB		101 39 seconds
			FreeDisk C:	Available disk space in kilobytes		N/A - N/A	214,845,685,284 KB		101 39 seconds
			FreeDisk D:	Available disk space in kilobytes		N/A - N/A	78,945,619 KB		101 39 seconds

SNMP が有効なマシンでは、次のようなモジュールが生成されます。

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		101 3 minutes 34 seconds
			ipOutRequests	The total number of IP datagrams which local IP user-protoco...		N/A - N/A	1.6		101 3 minutes 34 seconds
			sysName	An administratively-assigned name for this managed node. By...		N/A - N/A	pacifico		101 3 minutes 34 seconds
			sysUpTime	The time (in hundredths of a second) since the network manag...		N/A - N/A	1378258510		101 3 minutes 34 seconds
			X0_ifInOctets	The total number of octets received on the interface, includ...		N/A - N/A	43,870.2		101 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		101 3 minutes 34 seconds
			X0_ifOutOctets	The total number of octets transmitted out of the interface,...		N/A - N/A	60,051.9		101 3 minutes 34 seconds
			X1_ifInOctets	The total number of octets received on the interface, includ...		N/A - N/A	213,040.1		101 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		101 3 minutes 34 seconds
			X1_ifOutOctets	The total number of octets transmitted out of the interface,...		N/A - N/A	1,609,405		101 3 minutes 34 seconds

Pandora コンソールの一括操作メニューでは、サテライトサーバ用のメニューがあります。エージェントおよびモジュールを一括して編集したり削除したりできます。

# Bulk operations » Edit Satellite modules in bulk



Action Edit Satellite modules in bulk

Agent group All

Filter agent

Filter module

Agents

- 192.168.70.1
- 192.168.70.100
- 192.168.70.102
- 192.168.70.107
- 192.168.70.109
- 192.168.70.114
- 192.168.70.116
- 192.168.70.12
- 192.168.70.123
- 192.168.70.125

When selecting agents

Show common modules

- Any
- Latency
- Ping

Warning status

Min.   
Max.   
Str.

Critical status

Min.   
Max.   
Str.

Update

バージョン NG 763 以降

Resources / Manage agents / Setup  
NODO1MANU ?

Agent name: Nodo1Manu ID 1859 Interval: 5 minutes


Alias: Nodo1Manu OS: Satellite

IP Address: 192.168.80.222 Unique IP: [off] Fix IP address: [off] Delete selected IPs: [off]

Primary group: Servers

Server: koldo  
Satellite: koldo  
Description: Created by koldo

View agent QR code



Custom ID:

## SNMP ブラックリスト

大きなネットワークで不正なデータを返す SNMP モジュールがある場合、サテライトサーバのパフォーマンスに影響したり、多くのモジュールが不明状態になります。サテライトサーバは、実行を行わない SNMP モジュールのブラックリストを読み込むことができます。

新たなブラックリストを作成するには、`/etc/pandora/satellite_server.conf` を編集し、`snmp_blacklist` のコメントが外され設定されているか確認します。そして、次のように実行します。

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

そして、サテライトサーバを再起動します。ブラックリストは、必要に応じて再生成できます。

ブラックリストファイルのフォーマットは次の通りです。

```
agent:0ID  
agent:0ID  
...
```

例:

```
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
```



---

[Pandora FMS ドキュメント一覧に戻る](#)