多くのホストに対する短時間での監視設定

多くのホストに対する短時間での監視設定

Pandora FMS ドキュメント一覧に戻る

概要

このガイドは、多くのマシン (5台、10台から、500台) を Pandora FMS の機能を利用して、すばやくモニタリング対象として設定しようとしている方のために用意しています。このドキュメントは次の 4つの部分から成ります。

- 自動検出サーバとテンプレートを使った、ネットワークデバイスのモニタリング
- 自動検出スクリプト SNMP を用いたSNMP ネットワークデバイスのモニタリング
- ポリシーを用いたエージェントモニタリング
- XML によるエージェントジェネレータを使った、カスタマイズスクリプトでのリモートモニタリング

自動検出サーバとテンプレートを使った、ネットワークデバイスモニタリング

状況

200台のサーバ、20台のスイッチ、10台のルータをモニタする必要があるとします。しかし、一つ一つ設定はできません。“ 全体 ” のモニタリングはとても簡単ですが、マシンにエージェントをインストールする時間はあまりありません。

解決策

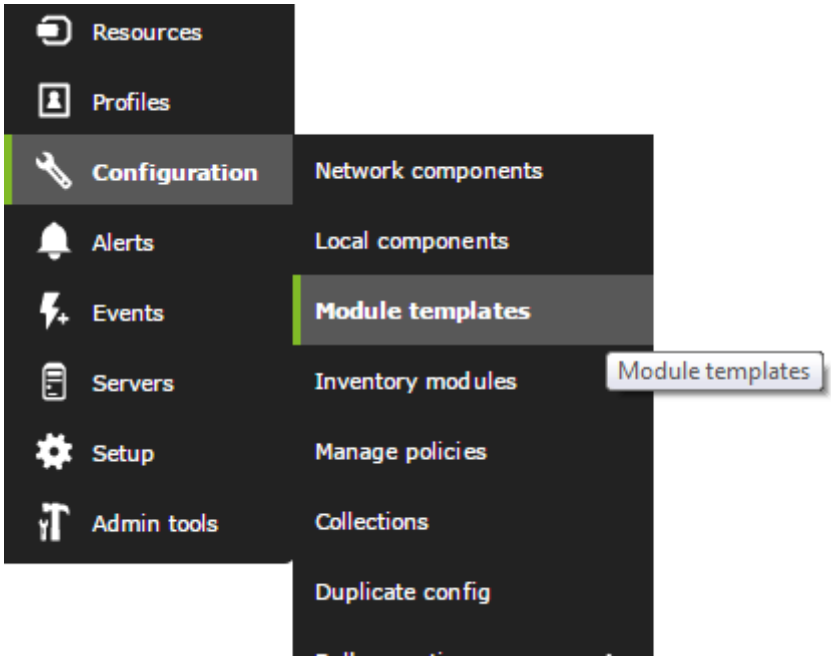
Pandora FMS はシステムを検出し、それがスイッチ、ルータ、サーバであるかによって異なるテンプレートを適用します。対象の種類を検出してテンプレートを適用することができます。

どれくらい時間がかかりますか

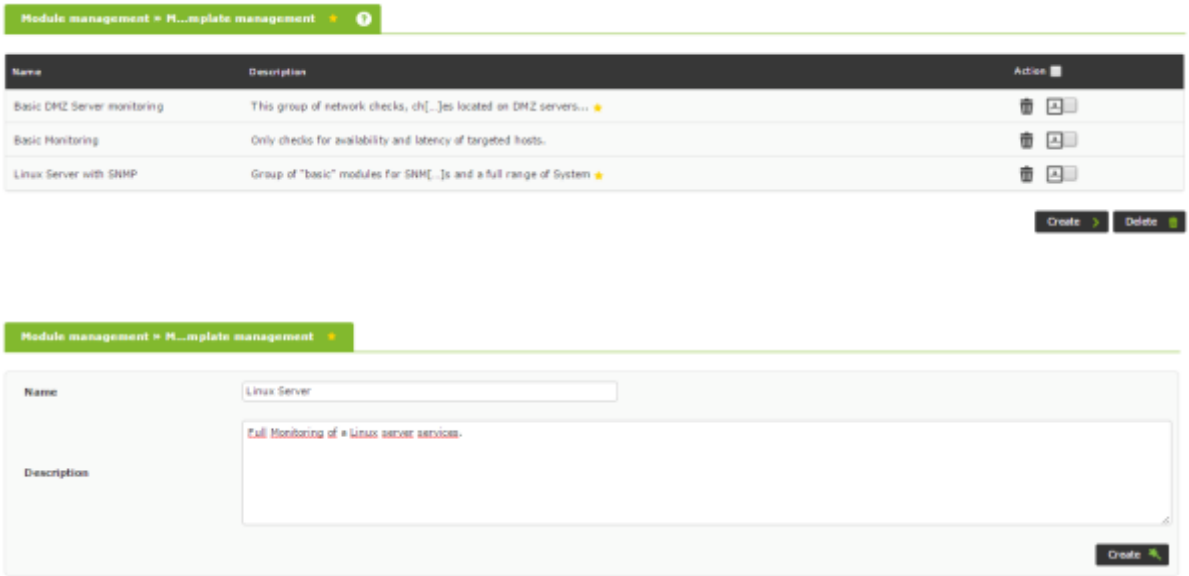
クラス C のネットワーク(255ホスト))で、バージョン 4.0 であれば 1分未満です。検出したマシンへのモニタリング設定の適用はすぐに実行されます。そのため、230台の対象に対してすべて設定するのに 10分未満です。

ステップ 1. モニタリングプロファイルの定義

最初に、Pandora FMS で “ モジュールテンプレート ” と呼ばれる機能で、モニタリングのテンプレートの定義を行います。そのためには、以下のメニューへ行きます。



すでに一般的な監視のいくつかのプロファイルが定義されています。それらの中から一つ(Linux Server)を編集します。これは、リモートから一般的な Linux サーバをモニタリングするのに便利です。



上の画面ショットを見るとわかるように、このプロファイルには“Check SSH Server” といった基本的な TCP のチェック、基本的な ICMP チェック(“Host Alive”)Linux MIB を使った SNMP モジュールなど、いくつかのモジュールがあります。

これらのチェック“ テンプレート ”は、インストールと同時に Pandora FMS の基本モジュールライブラリに定義されています。また、汎用的なモジュール定義を含んでいます。

このモジュールにはIP アドレスがありません。なぜなら、このモジュールがエージェントに適用されたときにエージェントから割り当てられるためです。しきい値、snmpコミュニティなど残りのフィールドは、このモジュールをエージェントに割り当てた時のデフォルトになります。

以上で、モニタリングテンプレート、テンプレートの一般的なモジュール、ネットワークコンポーネントがどんなものであるのか理解できたかと思います。次に、WMI の一般的なモニタリングおよび、基本的なモニタリングのテンプレートを見てみましょう。

最初に、Windows 用の 3つの WMI モジュールがあります。これらのモジュールは、オリジナルのコンポーネントや生成されたモジュールをカスタマイズする必要があります。WMI のリモートクエリを行えるようにするために、ユーザとパスワードが必要です。

2つ目は、基本的な ICMP の応答チェックです。次の画面ショットのように、HTTP、FTP、SMTP などのサービス動作確認といった他の基本的なチェックを追加することができます。

F.	P.	Type	Module name	Description	Status	Warn	Data	Graph	Last contact
			Connections opened	Network connections used in this machine		0/400 - 0/450	439 *conns*		7 minutes 25 seconds
			CPU Usage	% of CPU usage in this machine		0/60 - 0/90	10 %		7 minutes 25 seconds
			Disk_Free	Disk space available in MB.		20/10 - 10/0	35.0 *MB*		7 minutes 25 seconds
			Dropped Bits of nothing	Simulation of big number with absolute nosense, real like li...		N/A - N/A	317,615,070 *gamusins*		7 minutes 26 seconds
			Memory_free			N/A - 50/0	7,869.2 *MB*		7 minutes 26 seconds
			Network Traffic (Incoming)	Network throughtput for incoming data		N/A - 0/900K	764,725 *kbit/sec*		7 minutes 26 seconds
			Network Traffic (Outgoing)	Network throughtput for Outgoing data		N/A - 0/900K	385,559 *kbit/sec*		7 minutes 26 seconds
			Server Status A	Status of my super-important daemon / service / process		N/A - N/A	11		7 minutes 26 seconds
			Server Status B	Status of my super-important daemon / service / process		N/A - N/A	78		7 minutes 26 seconds
			Server Status C	Status of my super-important daemon / service / process		N/A - N/A	39		7 minutes 26 seconds
			System Log File	Messages from the system in logfile format		N/A - N/A	HWTbUZwsg8DL		7 minutes 26 seconds

ステップ 2. 自動検出サーバでのネットワークタスクの利用

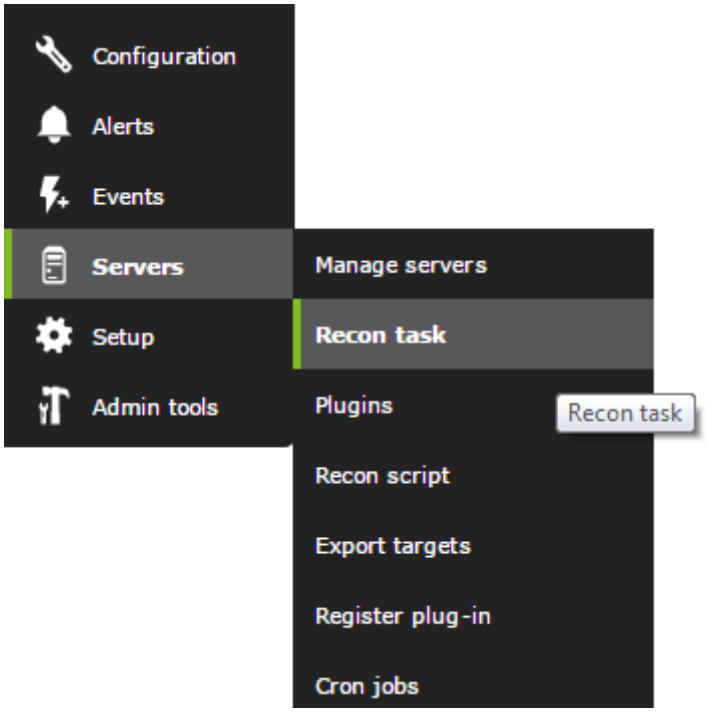
現在、Linux, Windows および network の 3つのモニタリングの基本プロファイルがあります。

例えば、network グループのすべてのマシンをモニタリングしたいと仮定します。

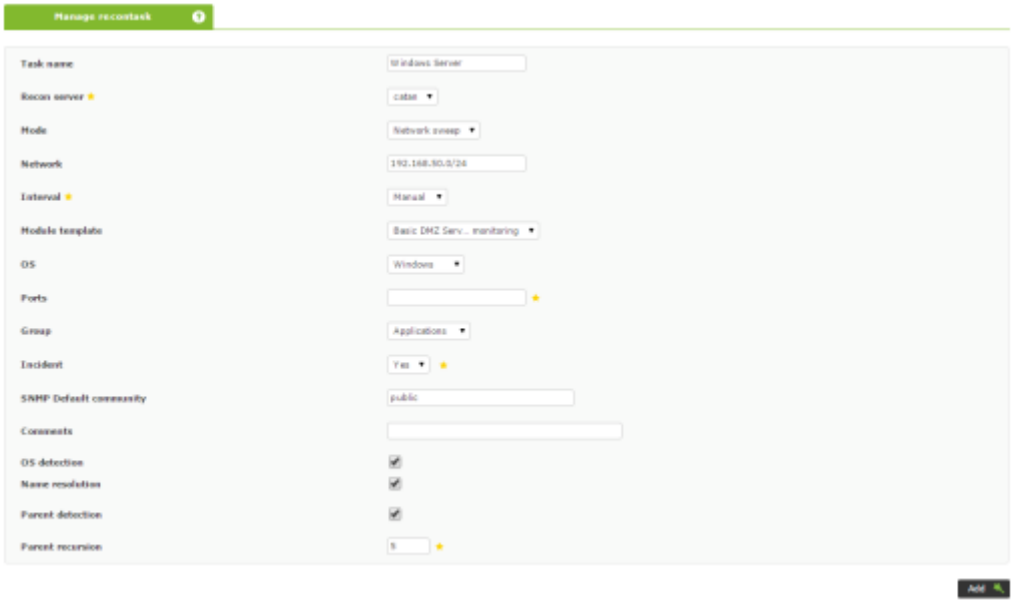
- 192.168.50.0/24 for servers.
- 192.168.50.0/24 はサーバ
- 192.168.50.0/24,192.168.1.0/24 for communications.
- 192.168.50.0/24 および 192.168.1.0/24 はネットワーク機器

また、ネットワーク上の全マシンを検出し、OS によってテンプレートを適用するかしないのかを決定したいとします。それ以外にも、スイッチの種類やモジュールの違いもあります。ここでは、基本的なポートがオープンしているかどうかにて、対象を“識別”します。例えば、23番ポート(telnet)が開いていれば、一般的なマシン(スイッチ、ルータ)と認識します。

新たな定義をするために、自動検出管理へ行きます。



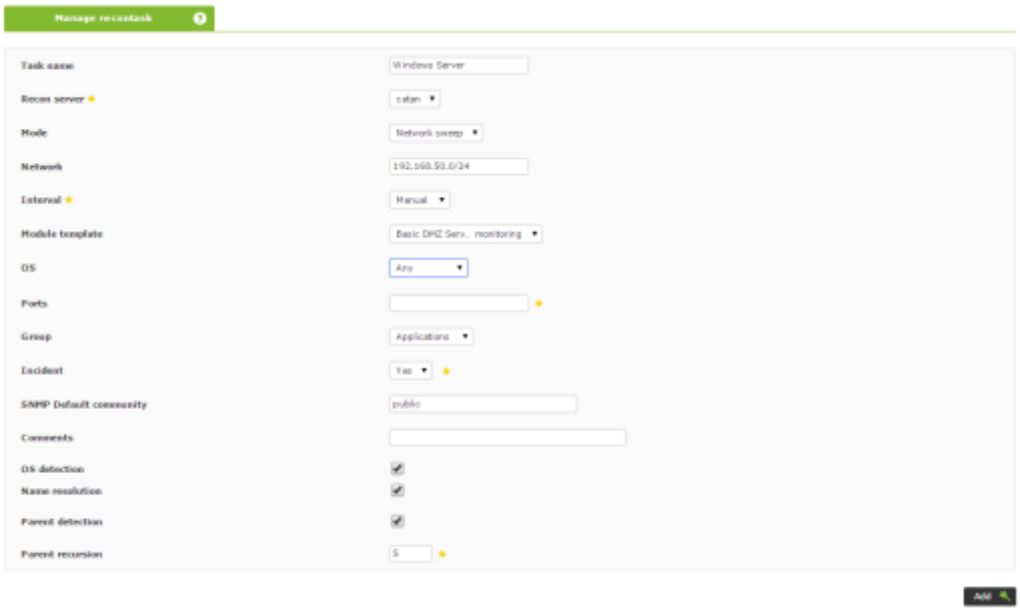
Windowsサーバを検出、登録し、Windowsマシンの標準モニタリング設定を適用する設定をします。



ここで“OS” フィールドで Windows を選択しているのがわかります。これは、このモニタリングプロファイルを Windows マシンに適用するという設定です。Windows でなければ無視します。

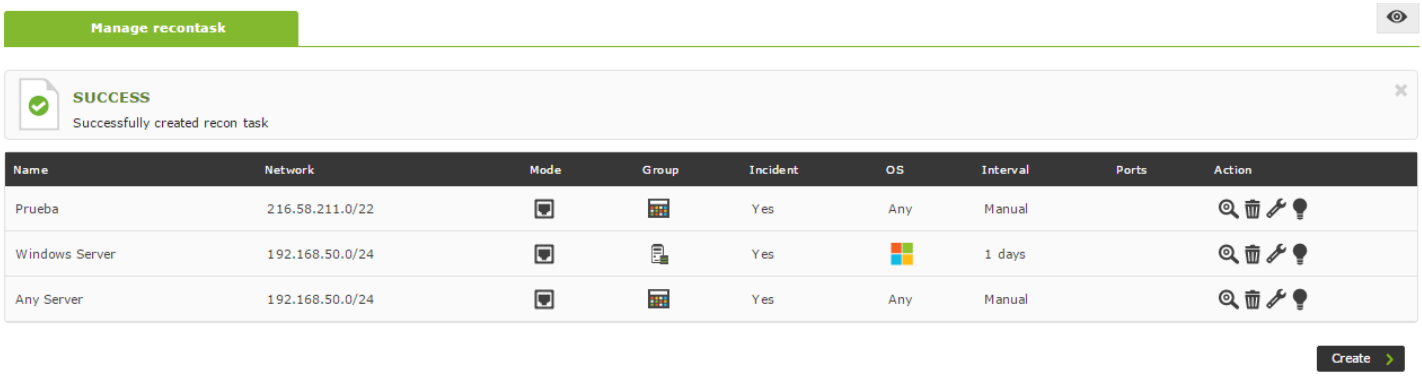
OS の種類の自動検出は、100% の確実性はありません。特定のポートを指定するなど、他の手法を選択することも可能です。

これにより、指定したポートが開いている全マシンがテンプレートの対象に入ります。以下の例では別タスクを作成しています。一般的なネットワークデバイスモニタリングを適用するために、OS の代りにポートのフィルタを利用しています。

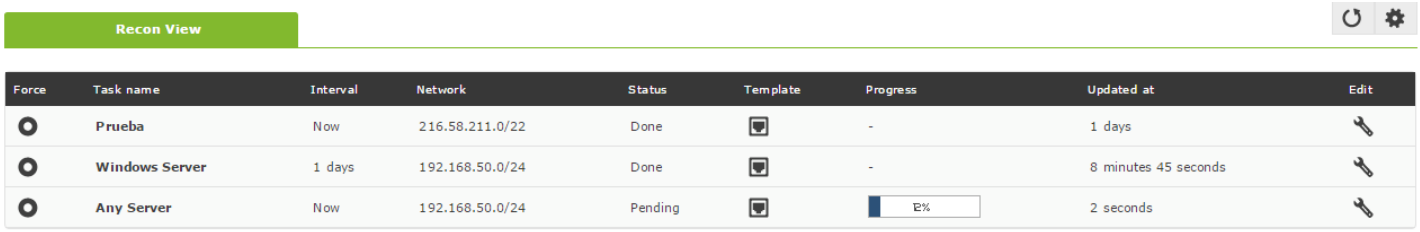


2つのネットワークを指定する方法は重要です。一行に書いて、192.168.50.0/24 と 192.168.1.0/24 をスペースで区切っています。

最後に、Linux も同じように設定します。3つの定義が完了すると次のようになります。



自動検出タスクを定義すると、自動的に検出が開始されます。必要であれば状態を参照したり強制実行ができます。それには、目のアイコンをクリックし、自動検出サーバの管理画面へ行きます。

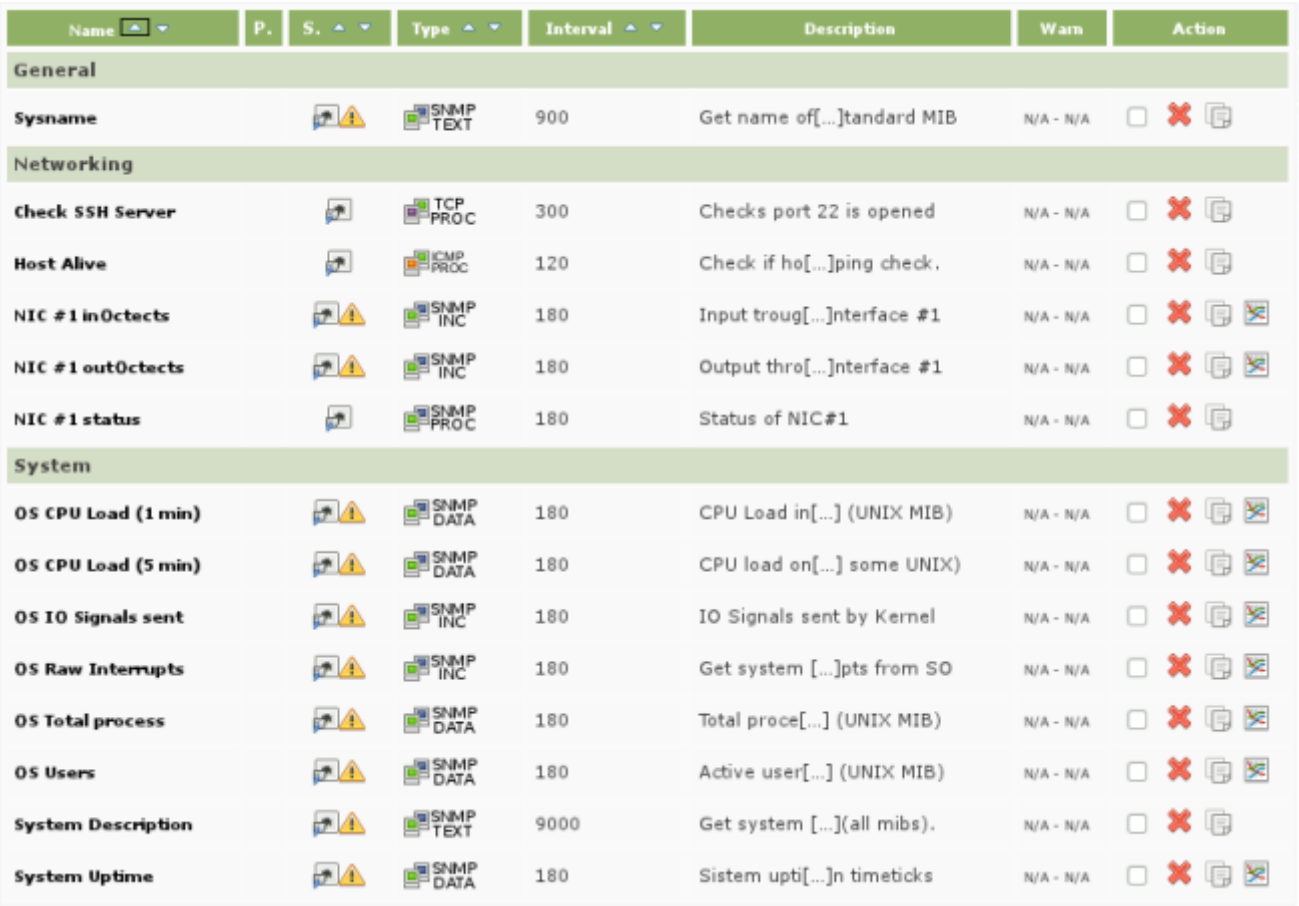


デフォルトでは、自動検出サーバは 1スレッドです。一度に一つのタスクのみ実行可能です。残りのタスクは、実行中のタスクが終了するのを待ちます。タスクの左にある緑のアイコンをクリックするとタスクの強制実行ができます。

自動検出サーバは、モニタリング対象になっていない新たなマシンを検出します。検出したら、そ

れらを自動的に登録(オプションが有効であれば名前解決も試みます)し、プロファイルに含まれる全モジュールを割り当てます。

一つのプロファイルに割り当てられている多くのモジュールが何であるかは十分理解しておいてください。特定のエージェントにとっては適した設定にはならなくなるかもしれません。このエージェントでは Linux システムを正しく検出していますが SNMP の応答はありません。そのため全ての SNMP モジュールの応答がありません。モジュールにデータを反映できる状態にありますが、未初期化状態となります。次回のデータベースメンテナンススクリプトが実行されたときに、未初期化モジュールは自動的に削除されます。



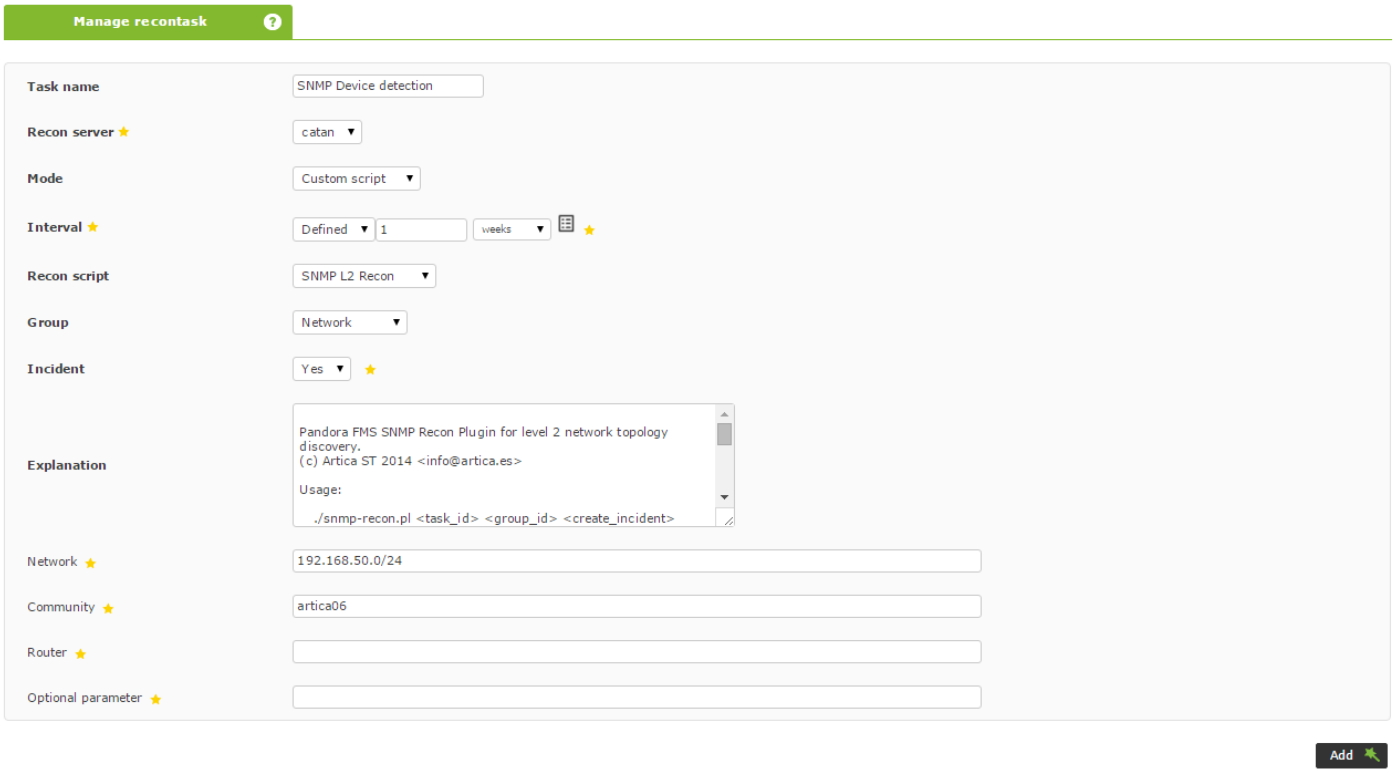
Name	P.	S.	Type	Interval	Description	Warn	Action
General							
Sysname			SNMP TEXT	900	Get name of[...]tandard MIB	N/A - N/A	
Networking							
Check SSH Server			TCP PROC	300	Checks port 22 is opened	N/A - N/A	
Host Alive			ICMP PROC	120	Check if ho[...]ping check.	N/A - N/A	
NIC #1 inOctects			SNMP INC	180	Input troug[...]nterface #1	N/A - N/A	
NIC #1 outOctects			SNMP INC	180	Output thro[...]nterface #1	N/A - N/A	
NIC #1 status			SNMP PROC	180	Status of NIC#1	N/A - N/A	
System							
OS CPU Load (1 min)			SNMP DATA	180	CPU Load in[...] (UNIX MIB)	N/A - N/A	
OS CPU Load (5 min)			SNMP DATA	180	CPU load on[...] some UNIX)	N/A - N/A	
OS IO Signals sent			SNMP INC	180	IO Signals sent by Kernel	N/A - N/A	
OS Raw Interrupts			SNMP INC	180	Get system [...]pts from SO	N/A - N/A	
OS Total process			SNMP DATA	180	Total proce[...] (UNIX MIB)	N/A - N/A	
OS Users			SNMP DATA	180	Active user[...] (UNIX MIB)	N/A - N/A	
System Description			SNMP TEXT	9000	Get system [...](all mibs).	N/A - N/A	
System Uptime			SNMP DATA	180	Sistem upti[...]n timeticks	N/A - N/A	

SNMP 自動検出スクリプトを用いた SNMP ネットワークデバイススクリプト

上記の場合とほとんど同じです。ここでは SNMP デバイスには多くのインタフェースがあり、それぞれのインタフェースの状態、入力トラフィック、エラー率などを自動的にモニタリングする必要があると考えます。

そのためには、自動検出スクリプトを利用します。これは、スクリプトで複雑なアクションを実行可能にする仕組みです Pandora FMS には、このような SNMP デバイスを検出するためのスクリプトが付属しています。

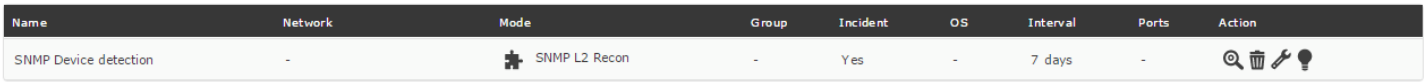
利用するには、次のようにネットワークタスクを作成します。



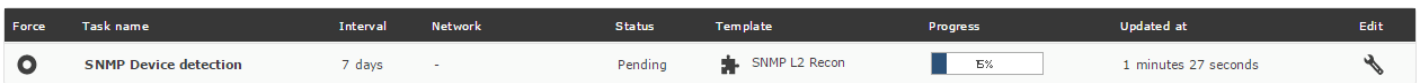
最初のフィールドには、対象のネットワークを入力します。 2つ目のフィールドには、デバイスを検出するのに利用する SNMP コミュニティを入力します。 3つ目のフィールドには、いくつかのオプションパラメータを入力します。ここでは、ダウン状態のインタフェースを登録するように -n を指定しています。デフォルトでは、アクティブなインタフェースのみ登録されます。

このスクリプトは、未登録のインタフェースかつ、実行時にアクティブなインタフェースを登録します。そのため、新たなインタフェースがアップした場合、それを検出します。ネットワークタスクは 1日に 1度実行したり、1時間ごとに実行したりできます。

自動検出スクリプトを登録すると次のように表示されます。



そして、自動検出スクリプトの実行は次のように表示されます。



モニタリングテンプレートおよび一括操作によるエージェントモニタリング

作成中

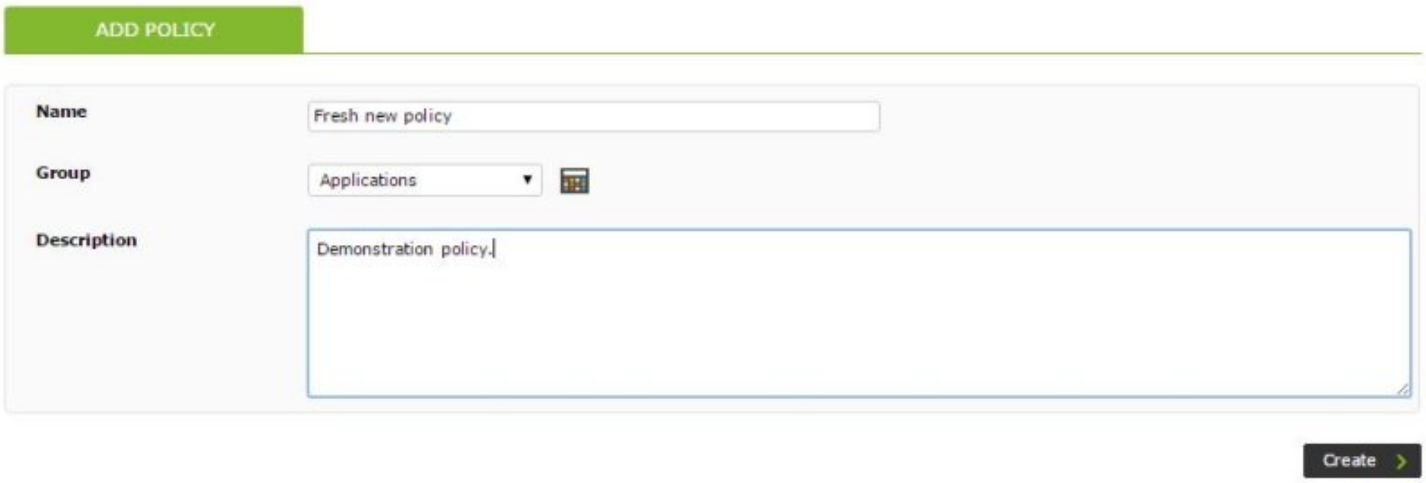
ポリシーによるエージェントモニタリング

ソフトウエアエージェントがインストールされた多くの監視の対象の管理には、ポリシーを使うことができます。

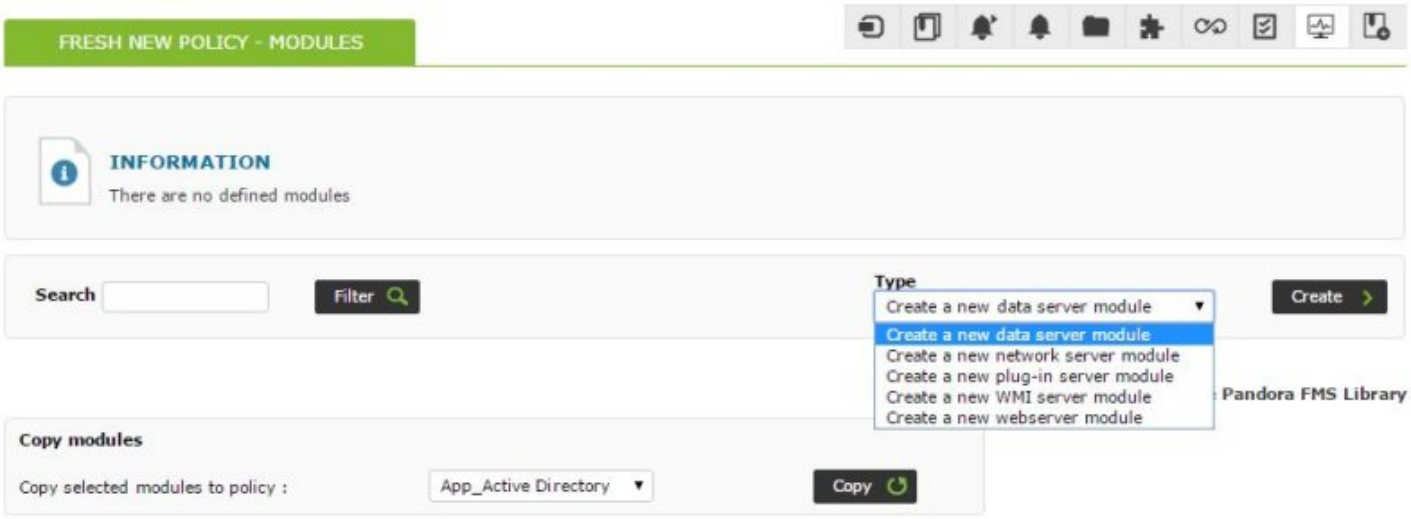
最初に、ソフトウエアエージェントは *remote_config* が有効化された状態でインストールされている必要があります。そうでないとモジュールの作成ができません。

```
remote_config 1
```

次に、*ポリシーの追加(Add policy)* へ行き、新たなポリシーを作成します。名前、グループ、説明といったいくつかのパラメータを入力します。

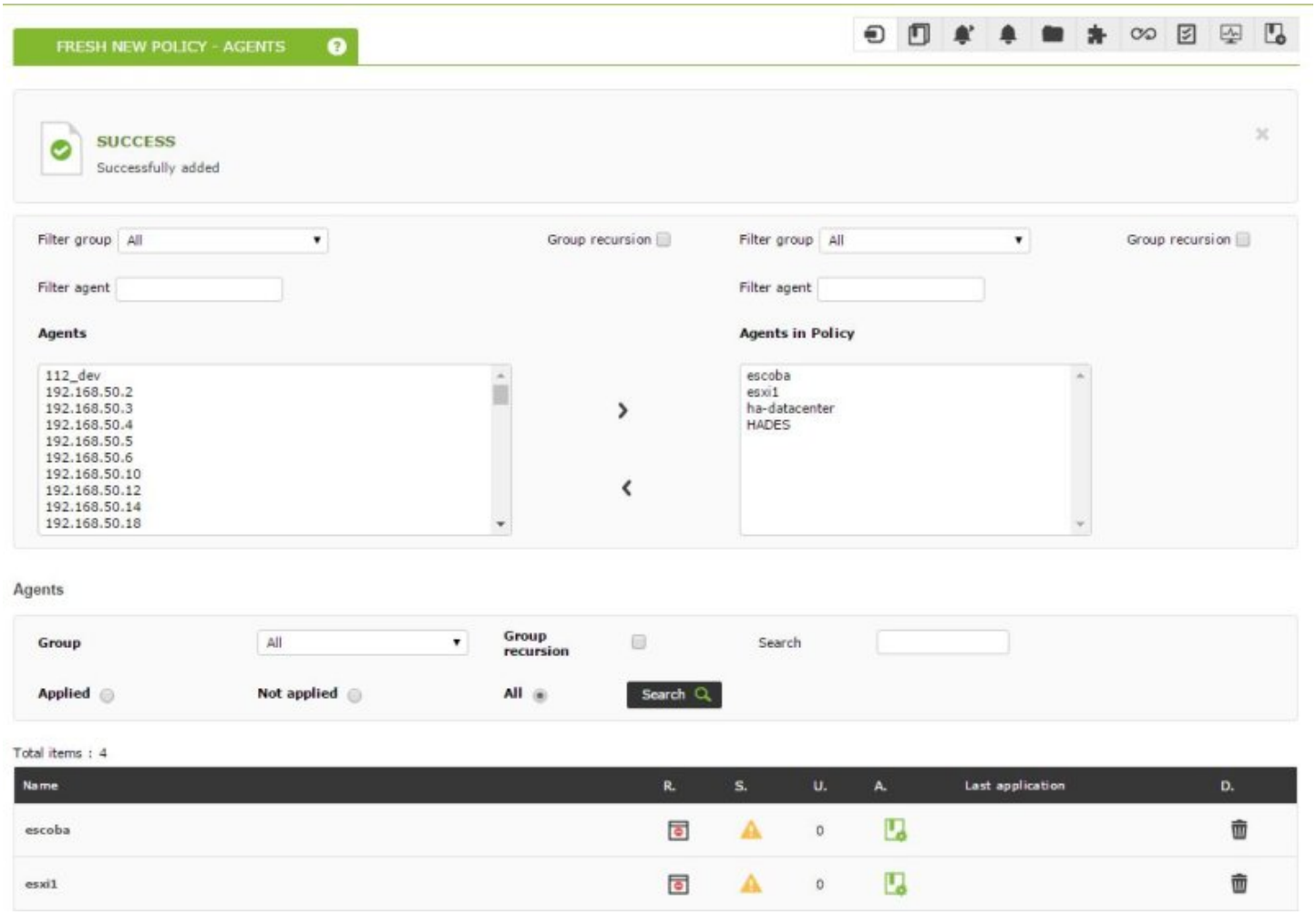


ここからは、ポリシーに新たなモジュールを作成する画面へ行き、新たなローカルモジュール(*dataserver module*)を作成します。

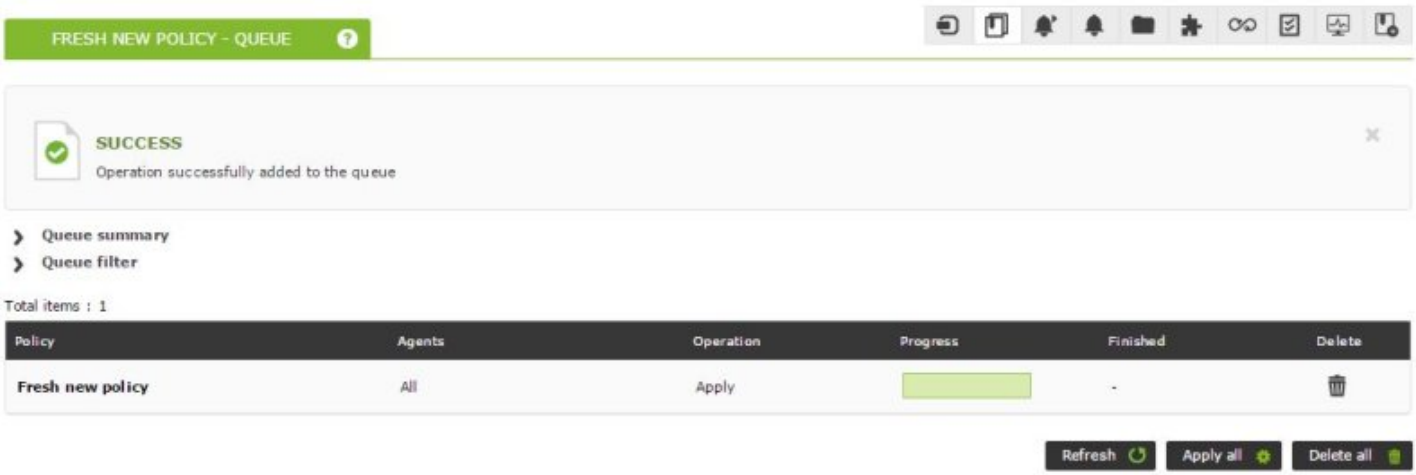


ローカル(*dataserver module*)またはリモートの必要なモジュールを作成したら、ポリシーへの必要なエージェント追加を開始できます。それには、ポリシーに対応するタブへ行き、“ ポリシー内エー

ジェント(Agents in policy)" セクションへエージェントを移動します。



エージェントが追加されたら、キュ(*Queue*) で変更を適用します。全変更を適用し進捗バーが完了するのを待ちます。



完了すると、ポリシー内に作成されたすべてのモジュールが選択したエージェントに展開されます。

ポリシーでは、エージェントのグループにモジュールを追加するだけでなく、アラート、コレクション、プラグインなどの他の要素も含めることができます。さらに、ポリシーはモジュールのしきい値変更など任意の変更も可能で、適用することでポリシーに含まれるエージェントすべてに対して

自動的に伝播させることができます。

カスタムスクリプトを用いたエージェントモニタリング

ここで説明するのは、大規模なシステムをモニタリングする高度な方法で、完全にアドホックな手法です。そのためには、システムの情報を取得するためのツールが必要です。例えば以下の通りです。

- リモートシステムに関する情報を取得するためのスクリプトがある。
- 再利用可能なデータを生成する他のモニタリングシステムが稼働中。
- XXXマシンのグループといった監視をしており、単一データではなく複数のデータを同時に返す。一つ一つデータを返せれば、リモートサーバのプラグインとして利用可能。

考え方は簡単です。希望のエージェント名およびモジュールデータを埋め込んだ、エージェントの XML ヘッダを生成する外部スクリプトを利用します。この外部スクリプトは、Pandora の XML フォーマットでデータを生成します(非常に簡単です)。メインのスクリプトは、XML を作成し、それを XML データファイルを処理するための標準パス(/var/spool/pandora/data_in)に置きます。スクリプトを CRON から動作するようにします Pandora FMS がデータを受け取るための XML フォーマットについての詳細は、補足資料の技術情報を参照してください。

リモートエージェントスクリプト

/usr/share/pandora_server/util/pandora_remote_agent.sh というスクリプトがあり、2つのパラメータがあります。

```
 -a <エージェント名>
 -f <実行するスクリプトファイル>
```

このとき、*/tmp/sample_remote.sh* というスクリプトがあり、次のような内容だとします。

```
#!/bin/bash

PING=`ping 192.168.50.1 -c 1 | grep " 0% packet loss" | wc -l`

echo "<module>"
echo "<name>Status</name>"
echo "<type>generic_proc</type>"
echo "<data>$PING</data>"
echo "</module>"

ALIVE=`snmpget -Ot -v 1 -c artica06 192.168.70.100 DISMAN-EVENT-
MIB::sysUpTimeInstance | awk '{ print $3>=8640000 }'`

echo "<module>"
echo "<name>Alive_More_than_24Hr</name>"
echo "<type>generic_proc</type>"
```

```
echo "<data>$ALIVE</data>"
echo "</module>"

# Another script with returns XML
EXT_FILE=/tmp/myscript.sh

if [ -e "$EXT_FILE" ]
then
    $EXT_FILE
fi
```

これは、次のようにリモートエージェントスクリプトを実行することにより、エージェント名 “agent_test” にて XML を生成します。

```
/usr/share/pandora_server/util/pandora_remote_agent.sh -a agent_test -f
/tmp/sample_remote.sh
```

同じスクリプトを XX マシンに対しても実行したいとします。ユーザ□IP□パスワードなどを、同じスクリプトに渡します。

```
/usr/share/pandora_server/util/pandora_remote_agent.sh -a agent_test -f
"/tmp/sample_remote.sh 192.168.50.1"
```

コマンドラインパラメータで、/tmp/sample_remote.sh のパラメータを決めます。

スクリプトの cron 設定

10台のマシンを以下のようにモニタリングすることを考えます。

```
/usr/share/pandora_server/util/pandora_remote_agent.sh -a agent_test1 -f
"/tmp/sample_remote.sh 192.168.50.1"
/usr/share/pandora_server/util/pandora_remote_agent.sh -a agent_test2 -f
"/tmp/sample_remote.sh 192.168.50.2"
/usr/share/pandora_server/util/pandora_remote_agent.sh -a agent_test3 -f
"/tmp/sample_remote.sh 192.168.50.3"
/usr/share/pandora_server/util/pandora_remote_agent.sh -a agent_test4 -f
"/tmp/sample_remote.sh 192.168.50.4"
/usr/share/pandora_server/util/pandora_remote_agent.sh -a agent_test5 -f
"/tmp/sample_remote.sh 192.168.50.5"
/usr/share/pandora_server/util/pandora_remote_agent.sh -a agent_test6 -f
"/tmp/sample_remote.sh 192.168.50.6"
/usr/share/pandora_server/util/pandora_remote_agent.sh -a agent_test7 -f
"/tmp/sample_remote.sh 192.168.50.7"
/usr/share/pandora_server/util/pandora_remote_agent.sh -a agent_test8 -f
"/tmp/sample_remote.sh 192.168.50.8"
/usr/share/pandora_server/util/pandora_remote_agent.sh -a agent_test9 -f
"/tmp/sample_remote.sh 192.168.50.9"
/usr/share/pandora_server/util/pandora_remote_agent.sh -a agent_test10 -f
"/tmp/sample_remote.sh 192.168.50.10"
```

これら全ての行を、例えば “/tmp/my_remote_mon.sh” というスクリプトに書き、実行権限を与え、以下の行を root の crontab に書きます。

1. */5 * * * * root /tmp/my_remote_mon.sh

これにより、スクリプトは 5分間隔で実行されます。スクリプトにマシンを追加することもできます。

監視に関するより詳細、利点や正しい監視方法をを知りたい方は、我々の blog system monitoring blog も参照してください。

Pandora FMS ドキュメント一覧に戻る