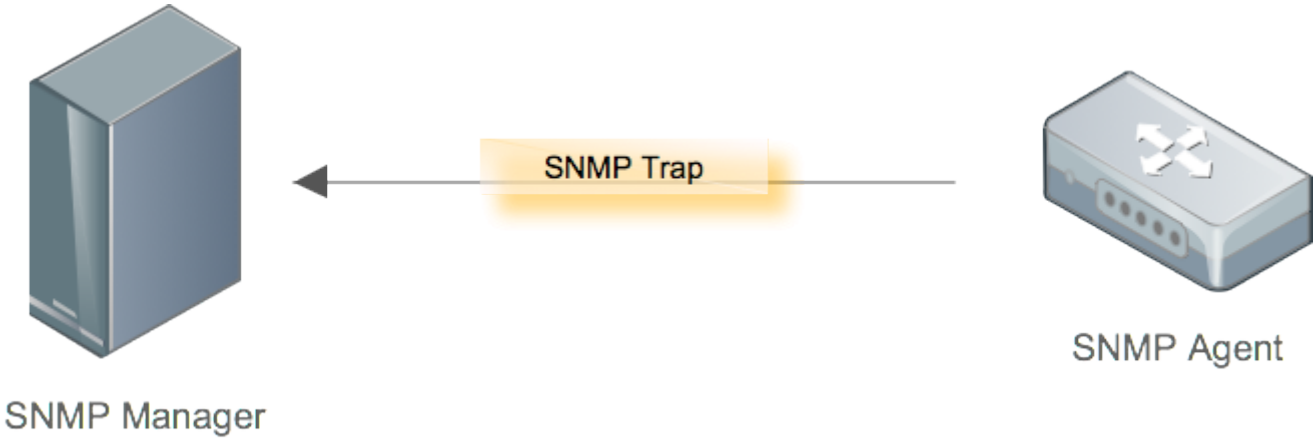
SNMP トラップ監視

SNMP トラップ監視

Pandora FMS ドキュメント一覧に戻る

概要

スイッチ、ルータ、サーバ、プリンタ、AP など、SNMP をサポートするネットワークデバイスは、インターフェイスの障害、CPU またはネットワークの負荷が高すぎたり、UPS の状態が変化したり、ディスクの障害が発生したときなどに、アラーム(SNMP トラップ)を送信することができます。各デバイスには、可能なイベントの独自のコレクションがあり、これは MIB と呼ばれます。この場合、デバイスのポーリングに使用する MIB とは異なります。



トラップは、デバイスにて何かが発生した場合に非同期で送信されます。Pandora FMS には、モニタリング対象から送られてくるトラップを表示するトラップ受信コンソールがあり、また、トラップに対してアラートを設定することができます。SNMP トラップは、Pandora FMS の起動時に起動される、OS の SNMP サーバデーモンで受け取ります。このサーバは通常ログファイルを以下に保存します。

```
/var/log/pandora /pandora_snmptrap.log
```

トラップは、常に生データで受信されます。つまり、数値 OID で受信します。ただし、OS に MIB ファイルがインストールされている場合は、文字に変換することができます。エンタープライズ版の Pandora FMS の SNMP コンソールでは、トラップをより認識しやすいように、OID を数値や文字で表したり、何らかの説明をつけたりする (“ インターフェースダウン ” など) ためのルールを設定できます。このようなルールを自動的に定義するために、Pandora FMS はトラップのベンダ MIB を読み込むことができます。ビデオチュートリアル "Loading MIBs in Pandora FMS" もご覧ください。

最初に、SNMP コンソールを有効化するためには /etc/pandora/pandora_server.conf 内の以下のパラメータを編集する必要があります。

```
snmpconsole 1
```

トラップを文字列変換したい場合(変数のバインディングまたは Enterprise 文字列のいずれか)は、次のオプションも有効化します(Enterpriase 版のみ)。

```
translate_variable_bindings 1
translate_enterprise_strings 1
```

また/etc/snmp/snmptrapd.conf ファイルを必要なパラメータで設定する必要があります。 例:

```
authCommunity log public
disableAuthorization yes
```

この設定では、コミュニティ “public”認証無しでトラップを受け付けます。

SNMPv3

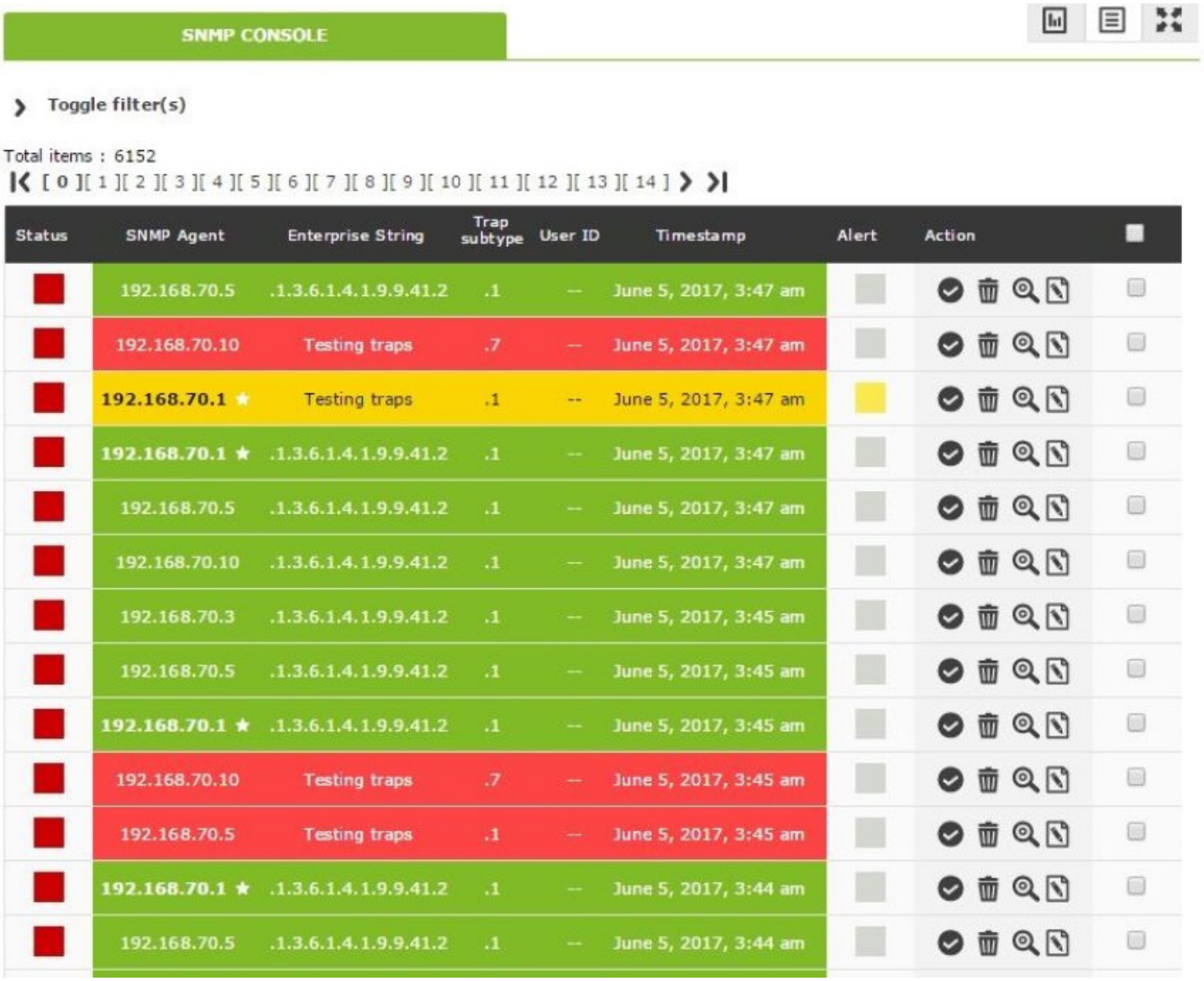
SNMPv3 トラップは、送信ユーザが createUser ディレクティブを用いて */etc/snmp/snmptrapd.conf* に追加されていないと受信を拒否します。例を以下に示します。

```
disableAuthorization yes
createUser -e 0x0102030405 snmpv3user SHA mypassword AES
```

engineID は、*-e* オプションと共に指定する必要があります。
そうしないとSNMPv3 INFORM のみを受信します。

トラップ受信コンソールへのアクセス

トラップコンソールへアクセスするには、モニタリング(Monitoring) → SNMP → SNMP コンソール(SNMP Console) をクリックします。受信したトラップ一覧が表示されます。全トラップ情報を表示することができるアイコン(目)があります。ここでSNMP トラップの詳細を見ることができます。



それぞれのトラップで、次のカラムが表示されます。

状態(Status)

承諾されたトラップは緑の四角で、そうでないものが赤の四角です。

SNMPエージェント(SNMP Agent)

トラップを送信したエージェントです。

OID

トラップの OID です。トラップは、このフィールドでは 1つのデータのみ送信できます。

値(Value)

トラップの値です。トラップは、このフィールドでは 1つのデータのみ送信できます。

カスタム OID (Custom OID), カスタム値 (Custom Value)

トラップで送信されるカスタムフィールドです。これは、トラップを送るデバイスに依存した特別なロジックで複雑なデータになることがあります。トラップはこのフィールドを使って複数のデータを送ることができます。

タイムスタンプ(Time Stamp)

トラップを受信した時間です。

アラート(Alert)

アラート通知をしたトラップは黄色、そうでないものはグレーです。

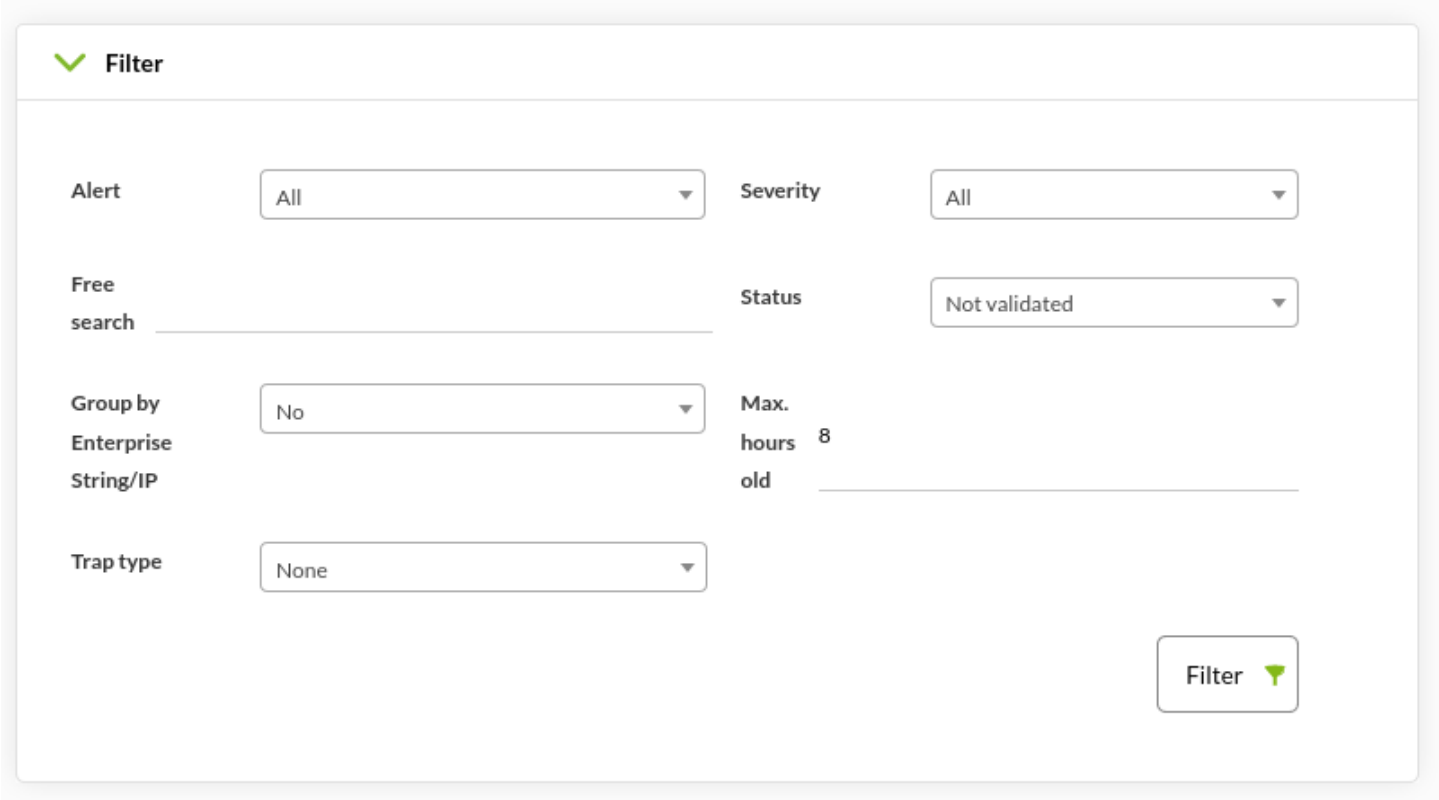
アクション(Action)

トラップを削除したり承諾するフィールドです。

トラップのタイプによって異なる色で表示されます。

- 青: メンテナンストラップ
- 紫: 情報トラップ
- 緑: 正常トラップ
- 黄: 警告トラップ
- 赤: 障害トラップ

トラップコンソールの上部に、'フィルタ設定(Toggle Filter)' というオプションが表示されています。これをクリックすると、トラップフィルタリングフィールドのオプションが表示されたり隠れたりします。



トラップの承諾

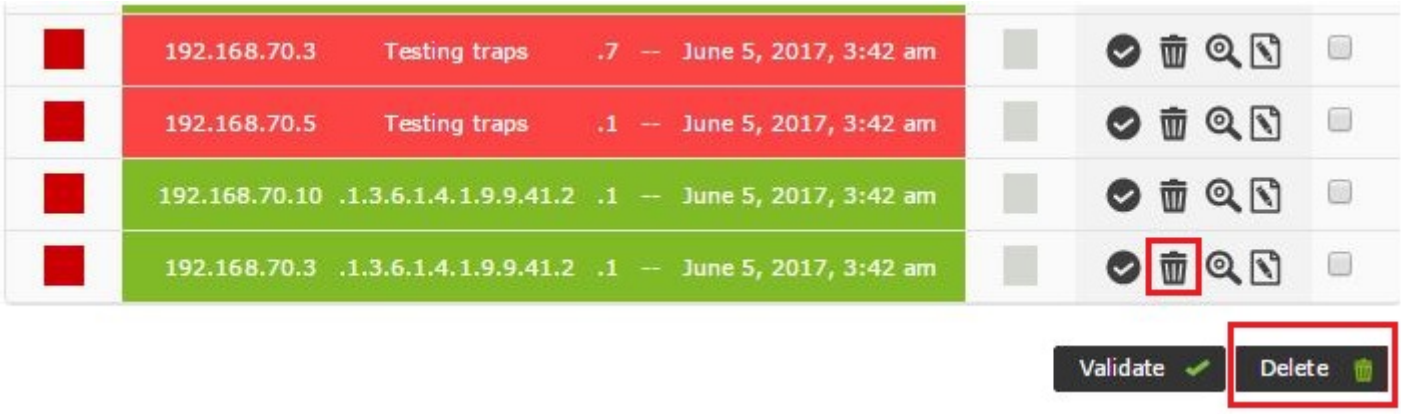
効果的にトラップを管理するために、トラップを承諾する操作が可能です。これにより、管理者は、すでに確認したトラップであるのか、そうでないのかを識別することができます。

トラップを承諾するためには、トラップの右にある緑の丸いボタンをクリックします。複数のトラップをマークして“ 承諾(validate)” ボタンをクリックすることにより、複数のトラップを承諾することもできますPandora FMS のイベントと似ています。

複数のトラップをマークして 承諾(validate) ボタンをクリックすることにより、複数のトラップを承諾することもできます。

トラップの削除

個別に、または複数の選択と「削除」アクションによって、編集後のトラップを削除することができます。

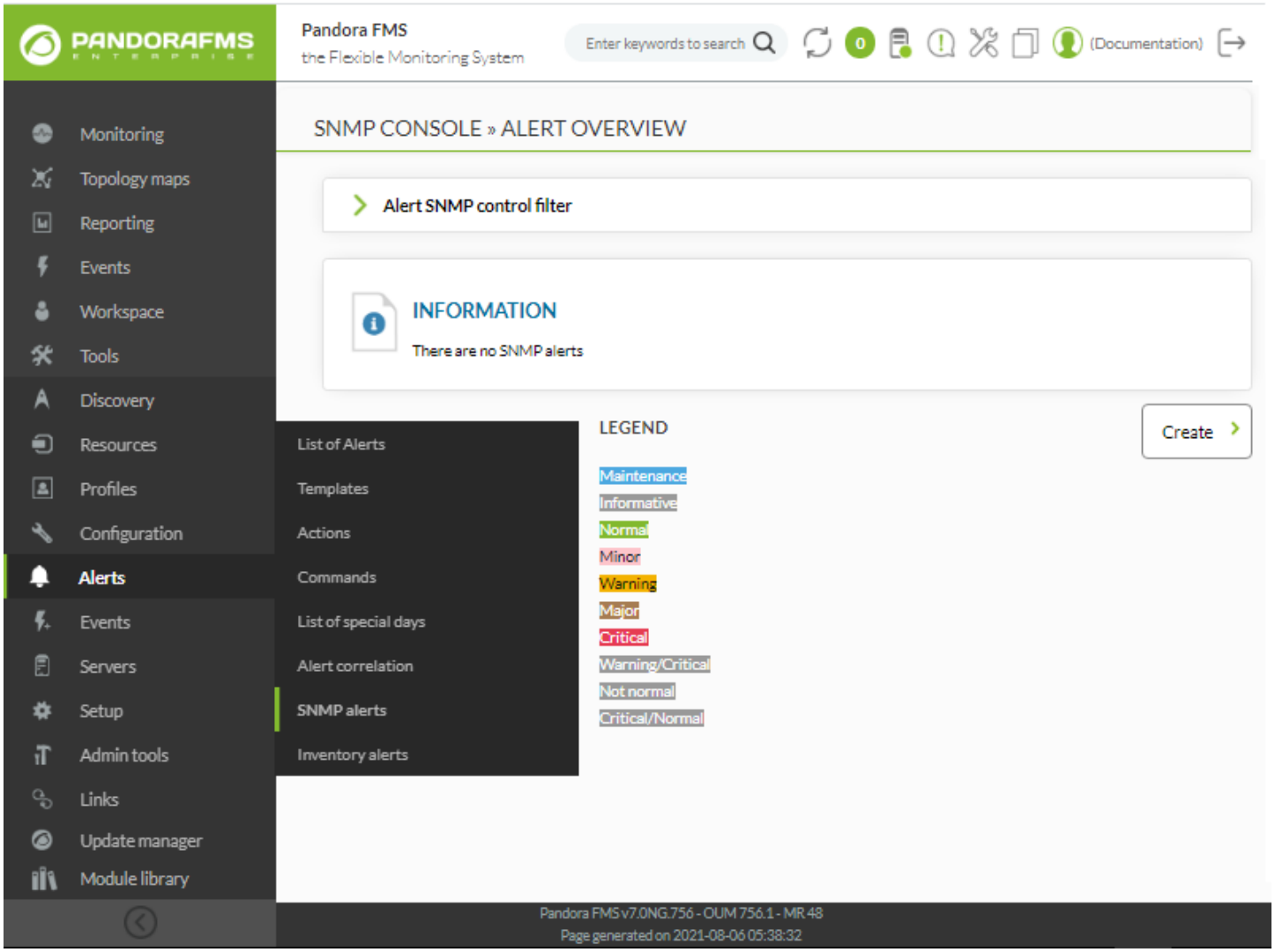


トラップが蓄積されないようにするため、10日以上経過したトラップを自動的に削除するデフォルト設定オプションがあります。

SNMP トラップアラート

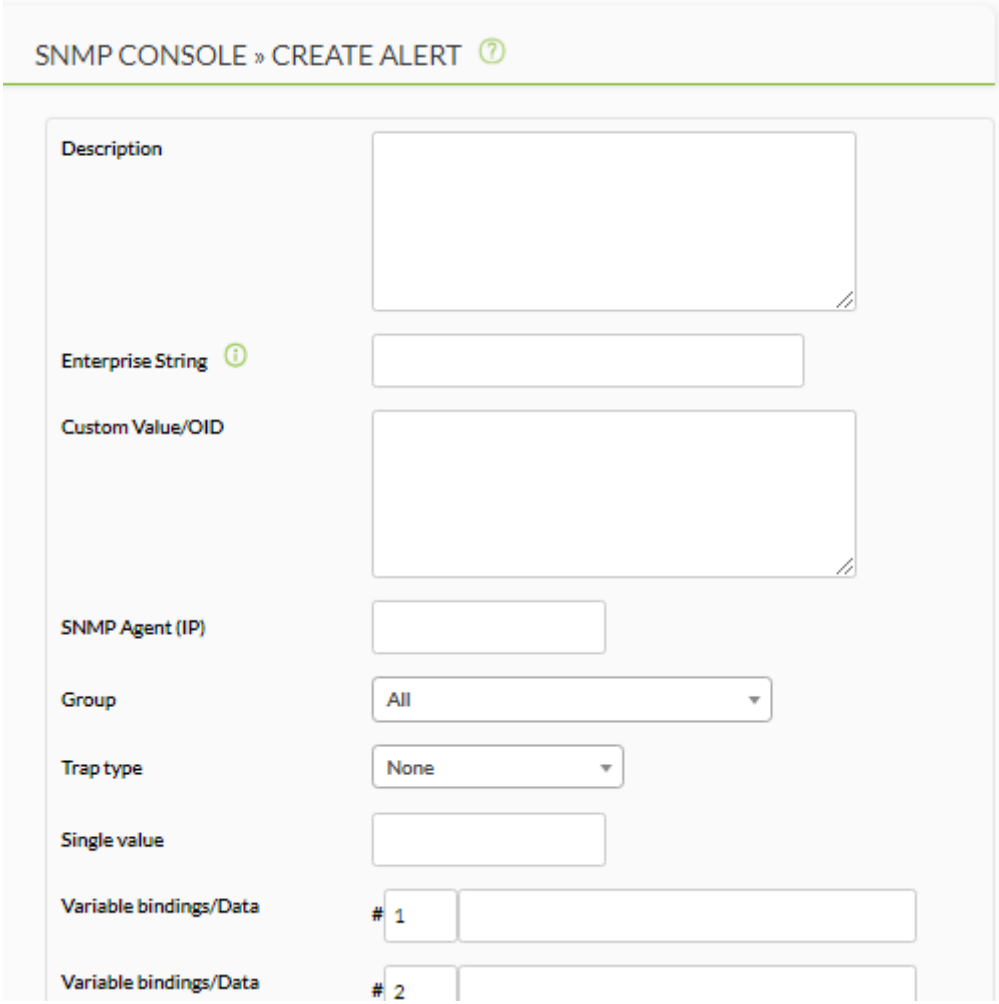
概要

Pandora FMS には、受信する SNMP トラップのアラートシステムもあります。 それらは主にフィルタリングルールに基づいており、アラートを発報するように設定したルールに従って、すべてのフィールドで条件に一致するものを検索します。 この章を読む前に、Pandora FMS のアラートに関しても確認してください。



アラートの追加

SNMP トラップアラートには、コンソールで受信した SNMP トラップがアラート条件にマッチするかを検索するために使用されるいくつかのフィールドがあります。 オプションで、必要に応じて、より一般的なルールやより具体的なルールを作成するフィールドを使用することができます。



説明(Description)

アラートの説明です。

Enterprise 文字列(Enterprise String)

トラップのメイン OID です。 文字列を検索します。 たとえばOID の一部を検索するなら ば、1.21.34.2.3 という表現が利用でき、それを含むすべての OID をフィルタリングすることができます。同様に、*1.21.34.2.3.* も可能です。ただし、* 文字を使用する必要はありません。

カスタム値/OID(Custom Value/OID)

トラップのその他フィールドである“Value” フィールドおよび“Custom OID”“Custom Value” フィールドを検索します。たとえば“Testing TRAP 225” という文字列を送信するトラップがある場合“Testing.*TRAP.*” という正規表現で “Testing TRAP” を検索できます。

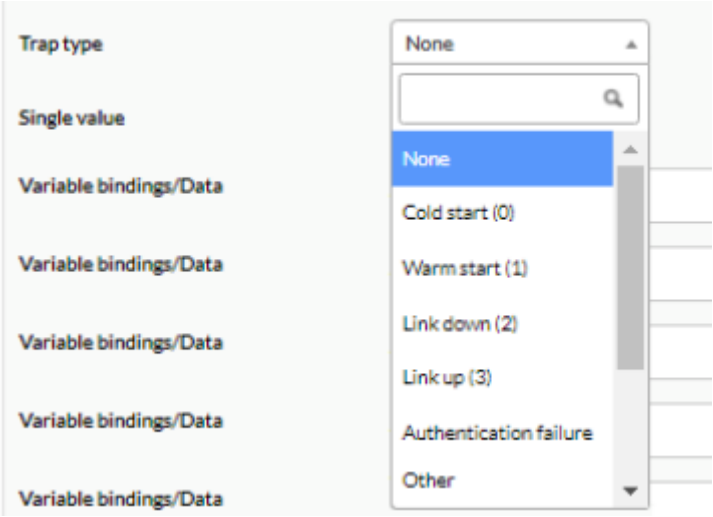
SNMP エージェント(SNMP Agent)

ラップを送信するエージェントの IP です。同様に、正規表現や文字列検索が使えます。

トラップタイプ(Trap type)

Cold Start, Warm start, Link down, Link up, Authentication failuer, other などのトラップタイプによ

るフィルタです。何も指定しなければ、トラップは通常 “Other” タイプとなり、任意のタイプが検索されます。

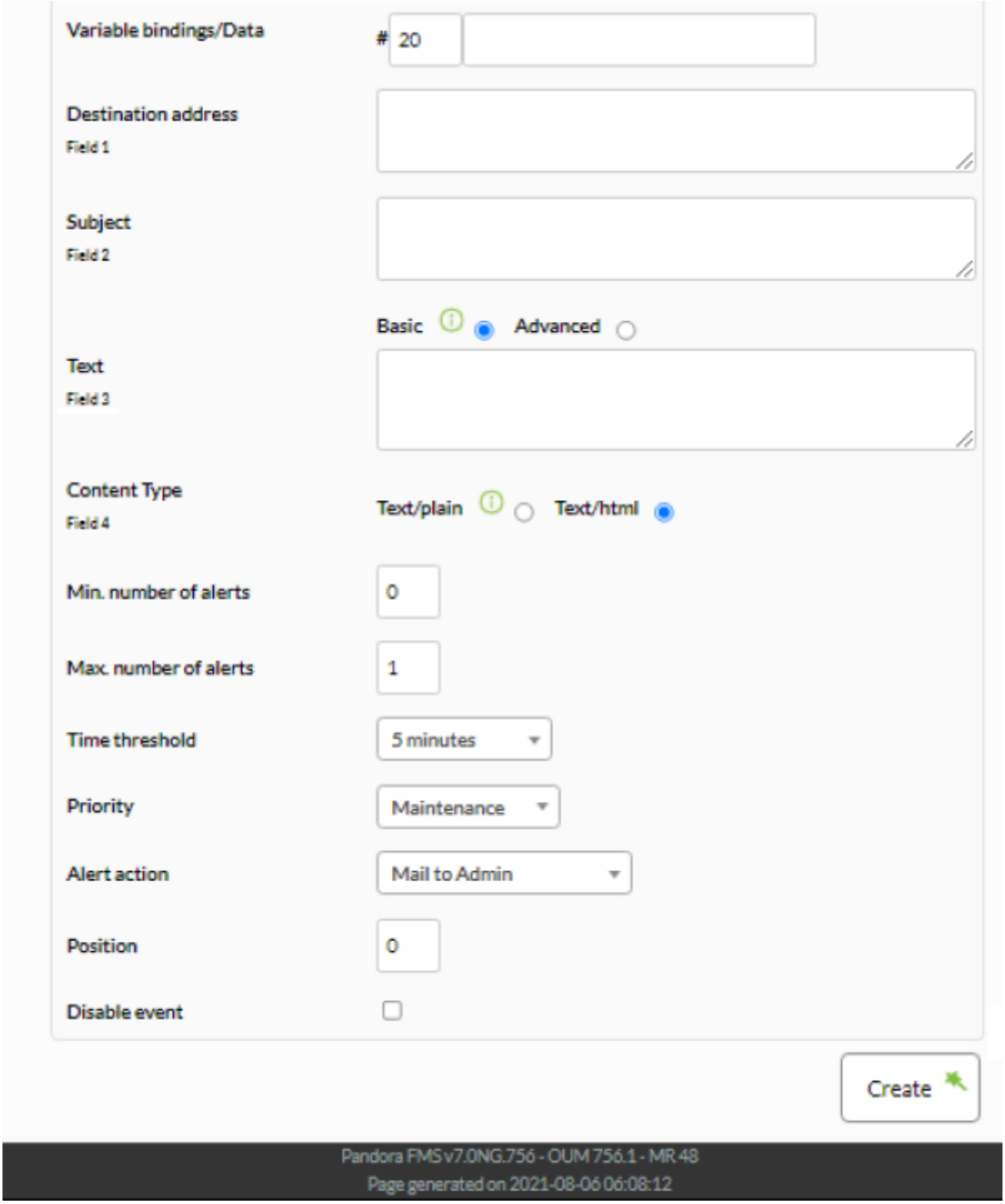


単一値(Single value)

トラップの値によるフィルタです。この例では、.666 で MAIN OID の値のみ参照し、カスタムデータの追加 OID は参照しないことに注意してください。

バインド変数/データ #1-20(Variable bindings/Data #1-20)

マッチする正規表現で、1 から 20 まであります。マッチすると、アラートが発報されます。設定した値は、_snmp_fx_ マクロ(_snmp_f1_, _snmp_f2_,...)で利用できます。マッチング対象として利用できるのは最初の 20変数のみですが、マクロにはいくつでも(_snmp_f11_, _snmp_f12_, ...)指定できます。



フィールド1(Field 1)

アラートのコマンドパラメータに指定するフィールド1です。このフィールドは、イベントの生成を選択した場合に使用されるか、メールアクションを選択した場合の宛先に使われます(アクションのデフォルトのメールの宛先を上書きする場合)。 アクション/アラートテンプレートでのカスタムフィールドの動作を完全に理解するにはPandora FMS のアラートについて説明している章を参照してください。

フィールド2(Field 2)

アラートのコマンドパラメータに指定するフィールド2です。例えば、電子メールを送信する場合は件名になります。 空白のままにするとアクションで定義した内容が使用されます。

フィールド3(Field 3)

アラートのコマンドパラメータに指定するフィールド3です。例えば、電子メールを送信する場合は本文になります。 空白のままにするとアクションで定義した内容が使用されます。

最小アラート数(Min. Number of Alerts)

アラートを発生させるトラップの最小数を指定します。

最大アラート数(Max. Number of Alerts)

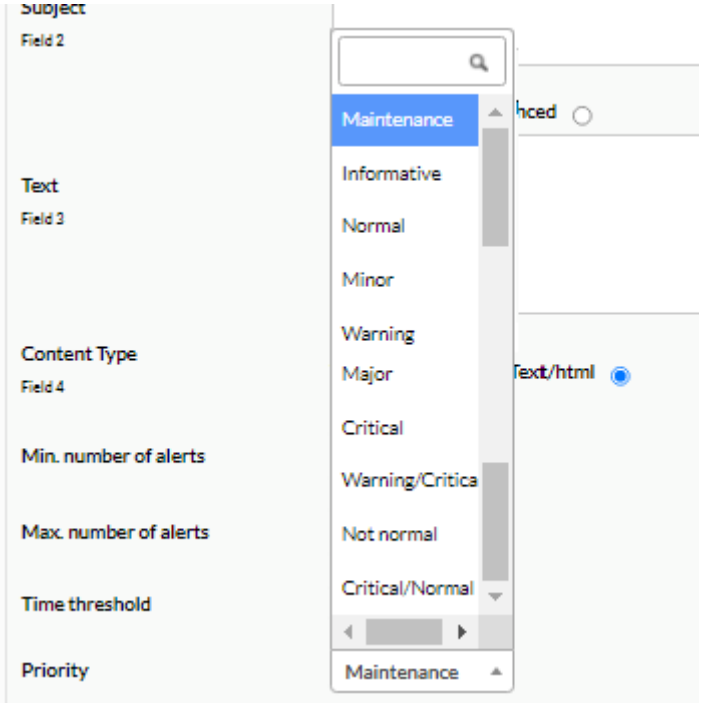
指定された間隔(または時間しきい値)でアクションが実行される最大回数を指定します。

再通知間隔(Time Threshold)

アラートカウンタをリセットする時間を指定します。このカウントは、最小アラート数で利用されます。

優先度(Priority)

アラートの優先度の選択です。



アラートの優先順位は、トラップの優先順位や Pandora FMS イベントとも何の関係もありません。

アラートアクション(Alert Action)

アラート実行時のアクションを選択します。イベントを選択すると、通常のアラート作成イベント

は生成されません。

位置(Position)

低位のアラートが最初に評価されます。複数のアラートが単一のトラップにマッチした場合は、マッチした同じ位置のすべてのアラートが発報されますが、低位のアラートがマッチしても発報されません。

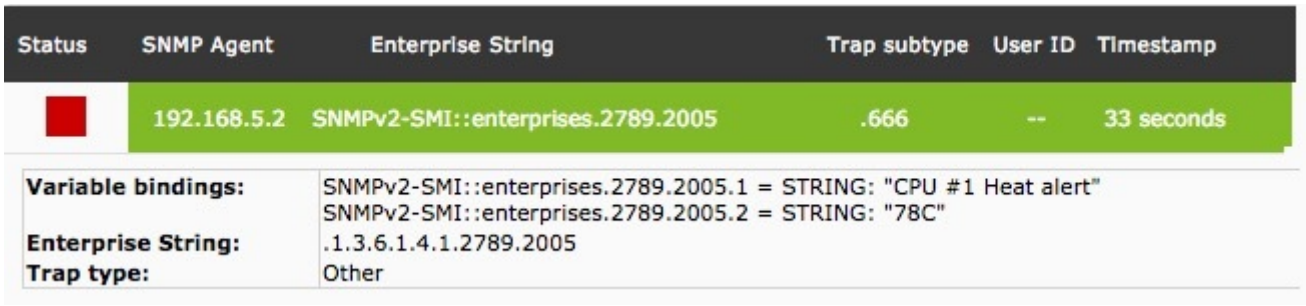
アラートフィールドマクロ

アラート *フィールド* で以下のマクロを利用できます。

- _data_: トラップ全体
- _agent_: エージェント名
- _address_: IP アドレス
- _timestamp_: トラップ日時
- _snmp_oid_: トラップ OID
- _snmp_value_: トラップ OID の値

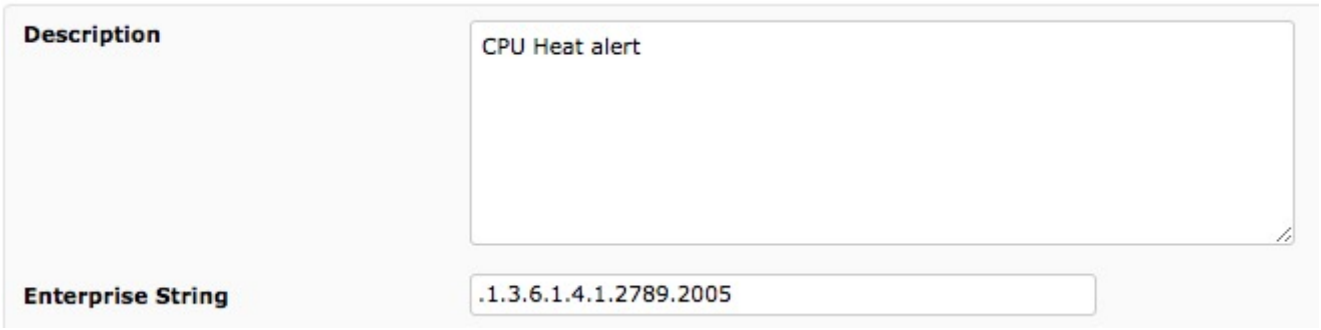
トラップアラートの例

次のようなトラップを受信したと仮定します。

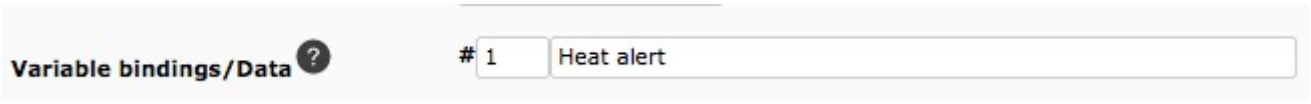


この場合、CPUオーバーヒートメッセージを含む可能性のあるトラップを識別するメインOID (1.3.6.1.4.1.2789.2005)がありますが（それ以外のものはわかりませんが）、1 と 2 の 2つの変数でその時の CPU のヒート状態と温度を表しています。 CPU のオーバーヒートトラップだけを識別したいので、トラップの最初の変数のヒートアラート文字列にマッチさせます(検索には最大20個まで設定できます)。

トラップの最初の部分を定義するのは簡単です。最初の最も重要なプレフィルタを作成するために、メイン OID のみを使用します。

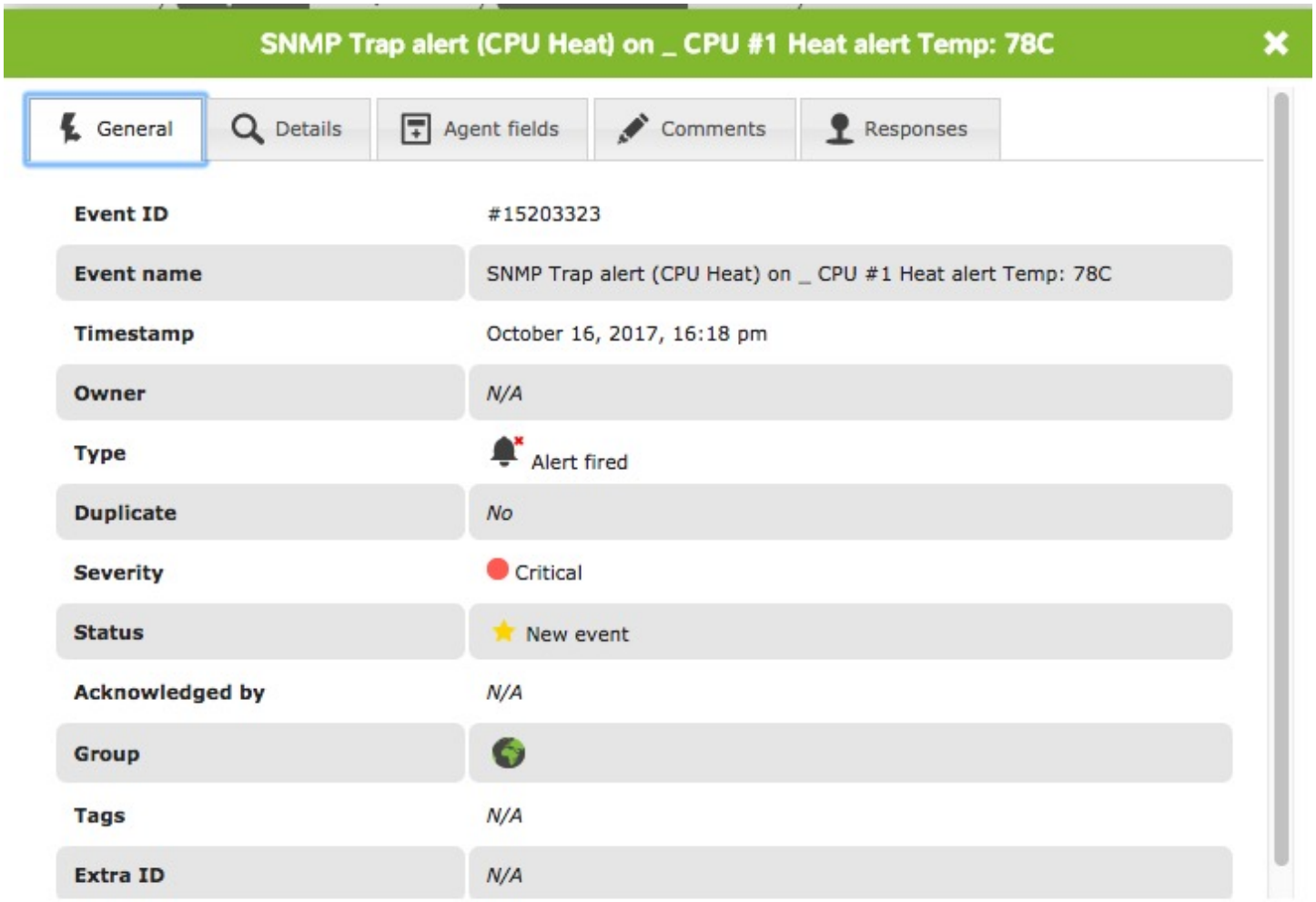


トラップ定義の 2番目は、必須部分を含みます。 トラップの最初の変数で “Heat alert” という文字列を探しますが、トラップをメインの OID で受信すると変数にはテキスト文字列が含まれていないため、アラートは発報されません。



最後に、“Pandora Event” タイプのアラートを選択することで、受け取ったトラップので値を含む変数 1 と 2 を使用してメッセージをマッチさせます。

アラートがオフになると、生成されるイベントは次のようになります。



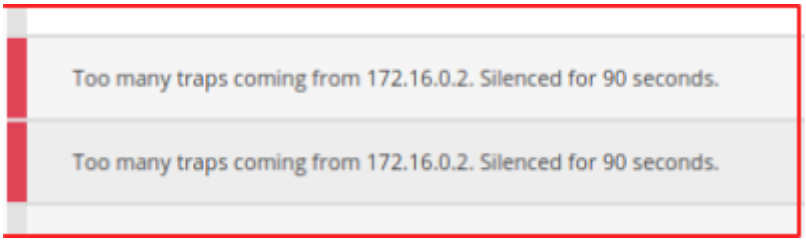
大量のトラップがある環境での動作

トラップストーム保護

同一の発信元から来るトラップストームからシステムを守るために利用する 2つのサーバパラメータがあります。これは、*pandora_server.conf* にて行う次の設定です。

- `snmp_storm_protection`: 同一の発信元 IP から指定した間隔(以下参照)内で処理する SNMP トラップの最大数です。
- `snmp_storm_timeout`: SNMP トラップストームから守る秒単位の間隔です。ここで指定した時間の間は、同一発信元(IP)からは *snmp_storm_protection* で指定した数のトラップのみを処理します。
- `snmp_storm_silence_period`: 特定のソースに対してストーム保護が起動されるたびに 0 より大きい場合は、現在の時間と静観時間が加算されます。 この時間が経過するまで、特定のソースからの新しいトラップは登録されません。

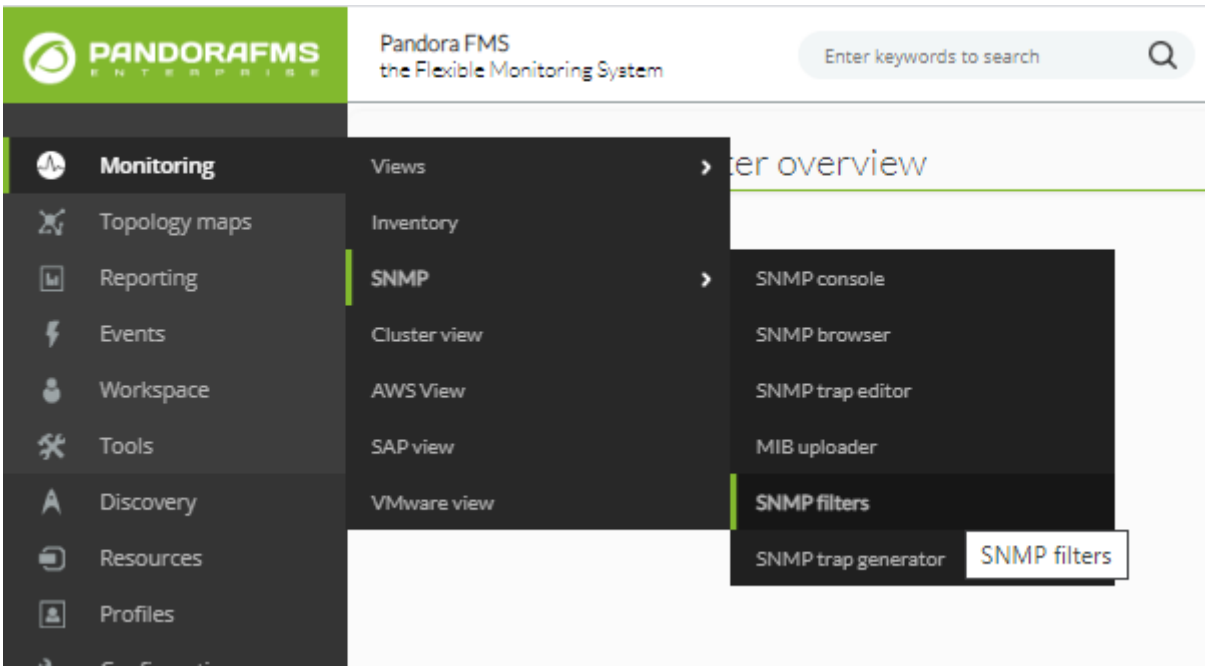
この保護が行われると、コンソールのイベントに反映されます。



トラップストーム保護は、トラップフィルタリングと合わせて、1日に何百、何千ものトラップを受け取っている場合に、不要なトラップを排除し一部のトラップのみを扱うことができます。

サーバにおけるトラップフィルタリング

あるシステムでは多数のトラップが受信されますが、監視に必要なのはわずかな割合でしかありません Pandora FMS では、アプリケーションの不要な読み込みを避けるために、サーバが受け取るトラップをフィルタリングすることが可能です。モニタリング(Monitoring) > SNMP> SNMP フィルタ(SNMP Filters) から、フィルタを定義することができます。



+ を使用して、説明と必要な数のフィルタを追加します。

SNMP ログ(デフォルトは */var/log/pandora/pandora_snmptrap.log*) のトラップエントリーに対して正規表現が適用されています。それは、次のような固定フォーマットです。

```
%4y-%02.2m-%l[**]%02.2h:%02.2j:%02.2k[**]%a[**]%N[**]%w[**]%W[**]%q[**]%v\n
```

それぞれ次の意味です。

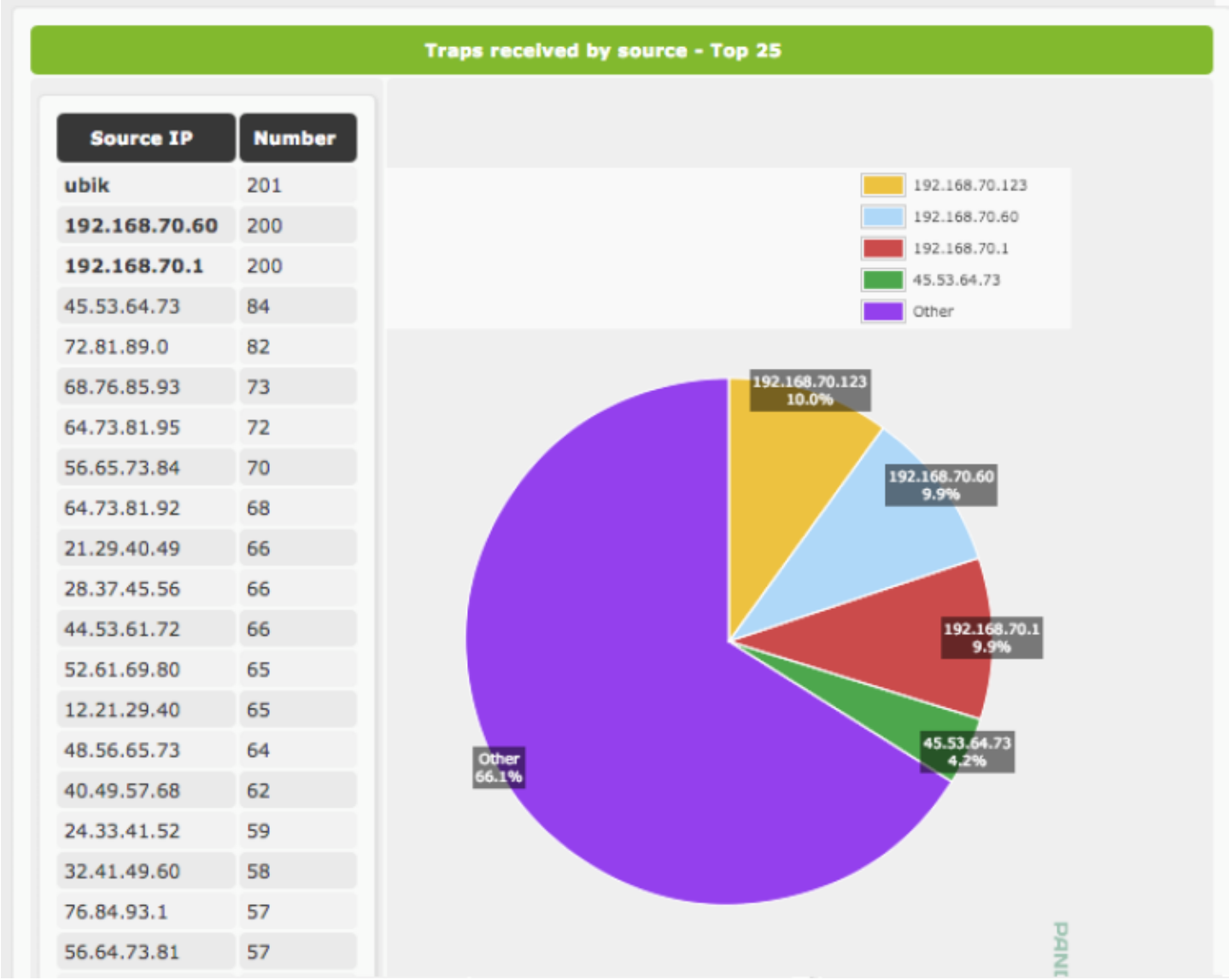
- %y: 現在の年。
- %m: 現在の月。(数値)
- %l: 現在の月における日付。
- %h: 現在の時間。
- %j: 現在の分。
- %k: 現在の秒。
- %a: 発信元アドレス。(トラップバージョン 1 のみ)
- %N: OID
- %w: トラップタイプ。(数値)
- %W: トラップの説明。
- %q: トラップのサブタイプ。(数値)
- %v: タブで区切られた値のリスト。(カスタム OID)

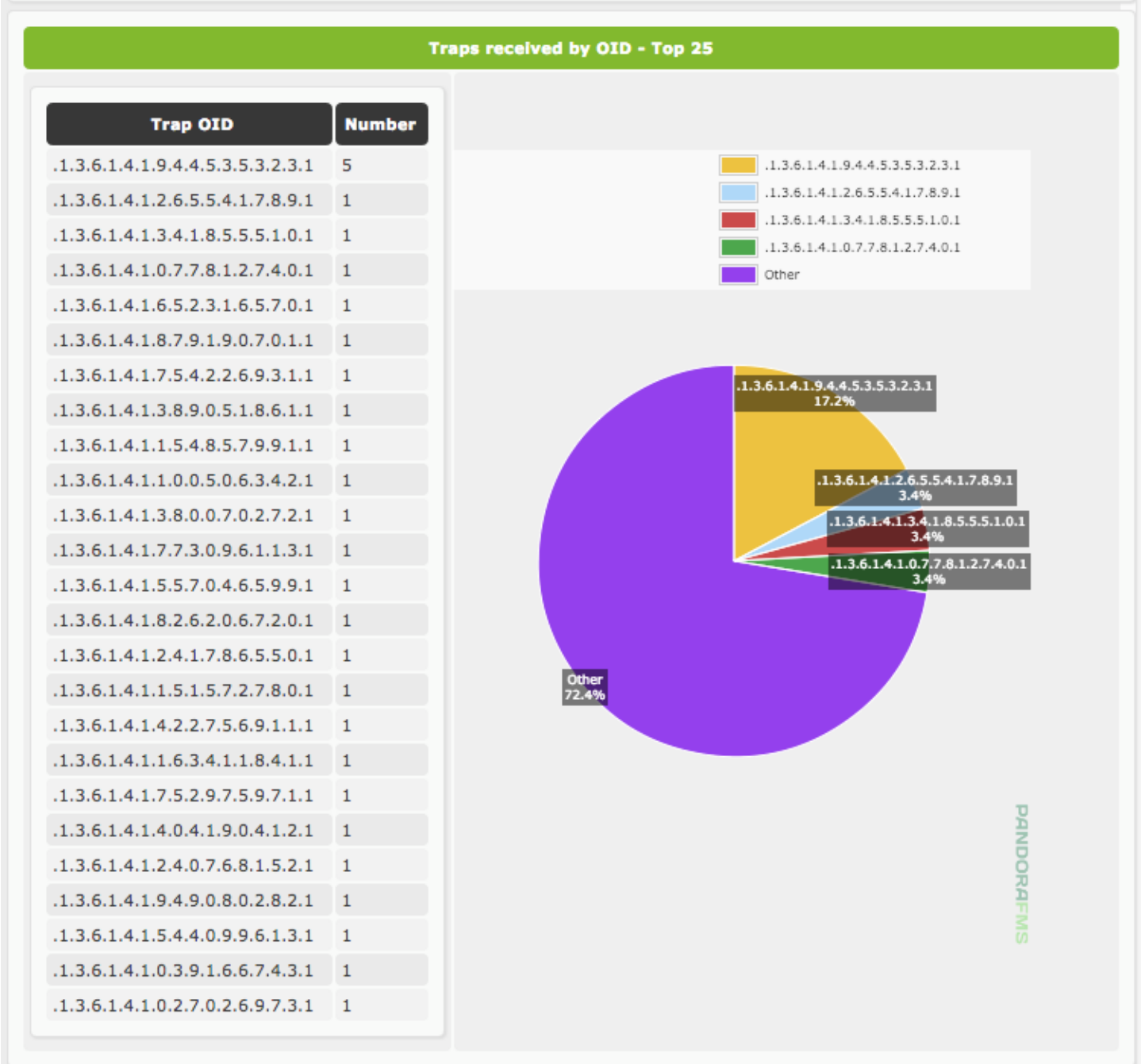
例えば、192.168.50.20 からのすべてのトラップをフィルタするには、次のフィルタ設定をします。

複数のフィルタを同時に作成できるため、検索ではすべてのフィルタリング条件を満たすトラップが対象となります。

SNMP トラップ統計

このビューでは、トラップの統計情報を発信元(IP)とOIDの両方で見ることができます。これにより、より多くのトラップを生成する IP とより多く繰り返す OID を特定し、フィルタの効果的な管理が可能になります。 過去 30日間のトラップ統計が表示されます。





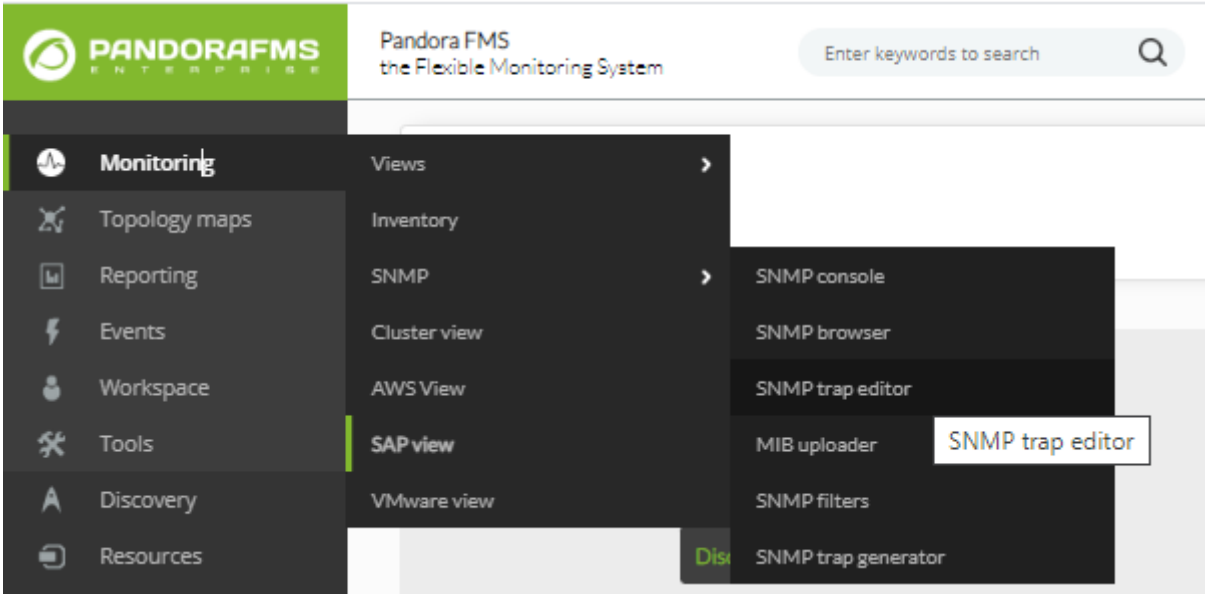
トラップのカスタマイズ

E この機能は Enterprise 版のみです。

モニタ対象デバイスから送られるトラップをオペレータがわかりやすくするために Pandora FMS にベンダ MIB をロードしたり、トラップを編集することができます。

トラップのリネーム/カスタマイズ

コンソールのトラップの部分に“ トラップの編集 ”, “ カスタマイズ ” があります。トラップを編集するには、操作(Operation) > SNMPコンソール(SNMP Console) > SNMP トラップエディタ(SNMP trap editor) へ行きます。



カスタマイズしたいトラップの編集アイコンをクリックします。

次のようなウインドウが表示されます。

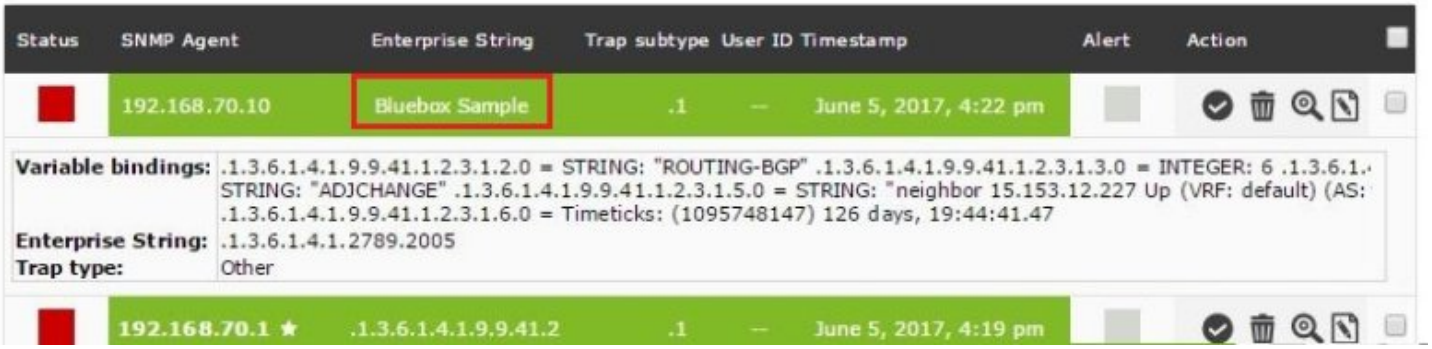
これにより、OID ".1.3.6.1.4.1.2789.2005" を見たとき、"Bluebox sample" と表示されます。区別するのがより簡単になります。 その内容自体(元の OID を含む)は変更されません。

カスタムOID には、変数バインディング文字列に部分マッチさせる Perl 互換の正規表現を記述します。一般的にはカスタムOID を使ったトラップの変換は必要ありません。

カスタムOIDに記述可能な文字数には制限があるため、この制限を超える可能性のある変数バインディング文字列全体ではなく、一部の変数にマッチするような正規表現を使用して下さい。

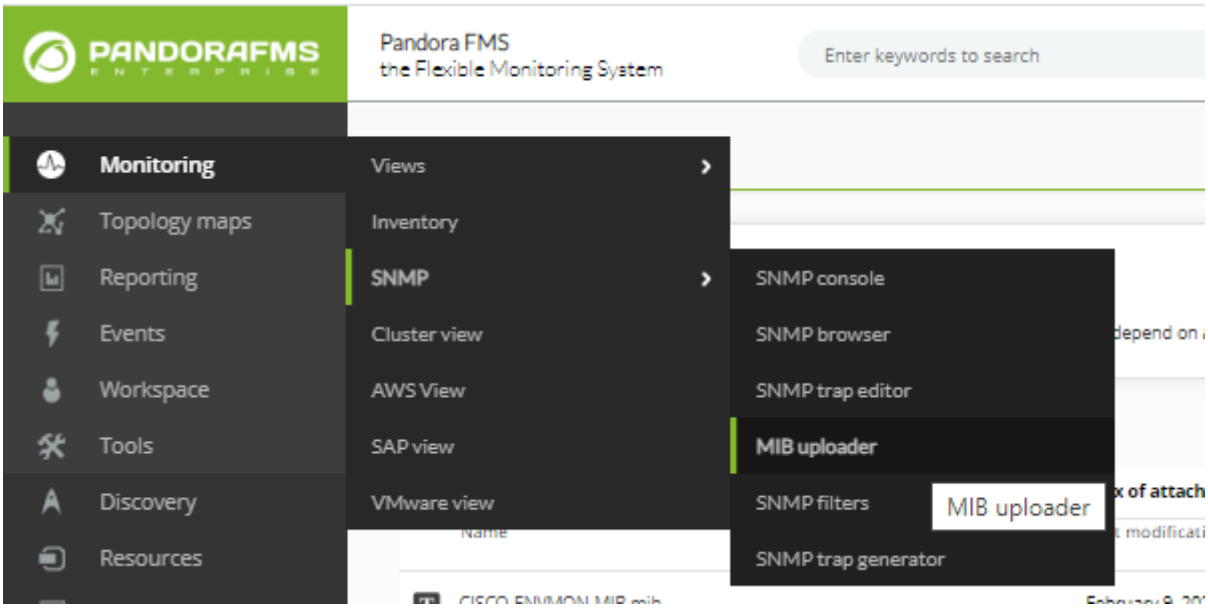
古いトラップは変更されないことに注意してください。この機能は、有効にした時点以降新たに受信したトラップに対してのみ有効です。

最終的に、ユーザ定義トラップは次のようになります。

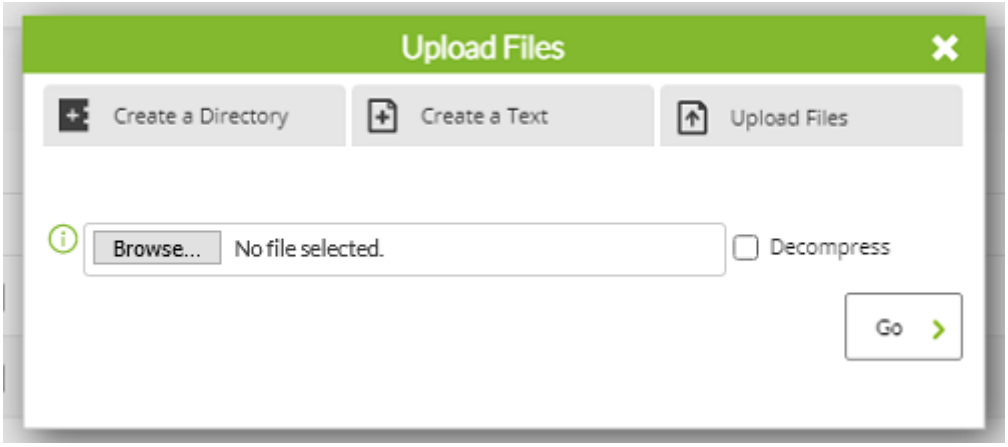


ベンダ MIB のロード

このオプションによりトラップの MIB をアップロードすることができPandora の内部変換データベースに展開されます。トラップを受信すると自動的に文字列に変換します。モニタリング(Monitoring) > SNMP > MIBアップローダ(MiB uploader) へ行きます。



ベンダ MIB をアップロードするには、ファイルアップロード(Upload file(s)) をクリックしてファイルを選択し、Go をクリックします。



アップロードが完了すると、システムはそれをトラップライブラリに取り込みます。

複雑な SNMP トラップへのアラート関連付け

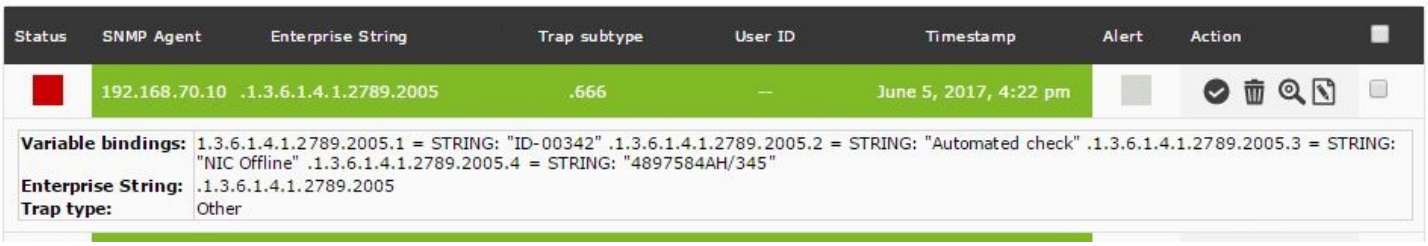
前述のアラートは、トラップが適切に定義されている場合にのみ使用されます。復旧通知のための関連データはありません。

しかしながら、いくつかの場合、次のような構造のトラップが見られます。

 OID: .1.3.6.1.4.1.2789.2005 Value: 666 Custom data: 1.3.6.1.4.1.2789.2005.1 = STRING: “ID-00342” .1.3.6.1.4.1.2789.2005.2 = STRING: “Automated check” .1.3.6.1.4.1.2789.2005.3 = STRING: “NIC Offline” .1.3.6.1.4.1.2789.2005.4 = STRING: “4897584AH/345”

これは “ 複雑 ” なトラップで、OID および値の他に、他の OID や値を元にした複雑なデータを含んでいます。トラップは複雑なパートを含むことができ、OID/値のペア (カウンタ、数値、文字、データなど) を元にした完全にランダムな構造です。

このトラップは、トラップコンソールでは次のように表示されます。



拡張情報 (Custom data) を見てみると、いくつかの有用なデータが含まれています。インスタンスでは、OID が 2005.1 で終わっている最初のフィールドは、識別子のように見えます、OID が 2005.3 で終わっている 3つ目のフィールドは、エラーメッセージのように見えます。フィールド 2 と 4 は、我々にとって不明なコードであり、使い道が無さそうです。

トラップのテキストデータの特定の部分を使って、トラップからイベントを生成できるということを考えてみましょう。次のような情報を含むイベントを生成したいと仮定します。

```
Chassis Alert: <error message> in device <identifier>
```

取得したデータにマッチさせてアラートを生成し、またデータを使ってアラート内のメッセージを生成します。 Pandora FMS では、セレクタを用いて、高度な正規表現にて実現することができます。正規表現に関する詳細はこちらを参照してください。
http://en.wikipedia.org/wiki/Regular_expression#Expressive_power_and_compactness.

() を使ったセレクタでは、検索の表現を使って情報のコピーができます。

識別子を取得するための正規表現は次のようになります。

```
.*.1.3.6.1.4.1.2789.2005.1 \= STRING\: \"([0-9\-\_A-Za-z]+)\"
```

エラーメッセージを取得するための正規表現はつぎのようになります。

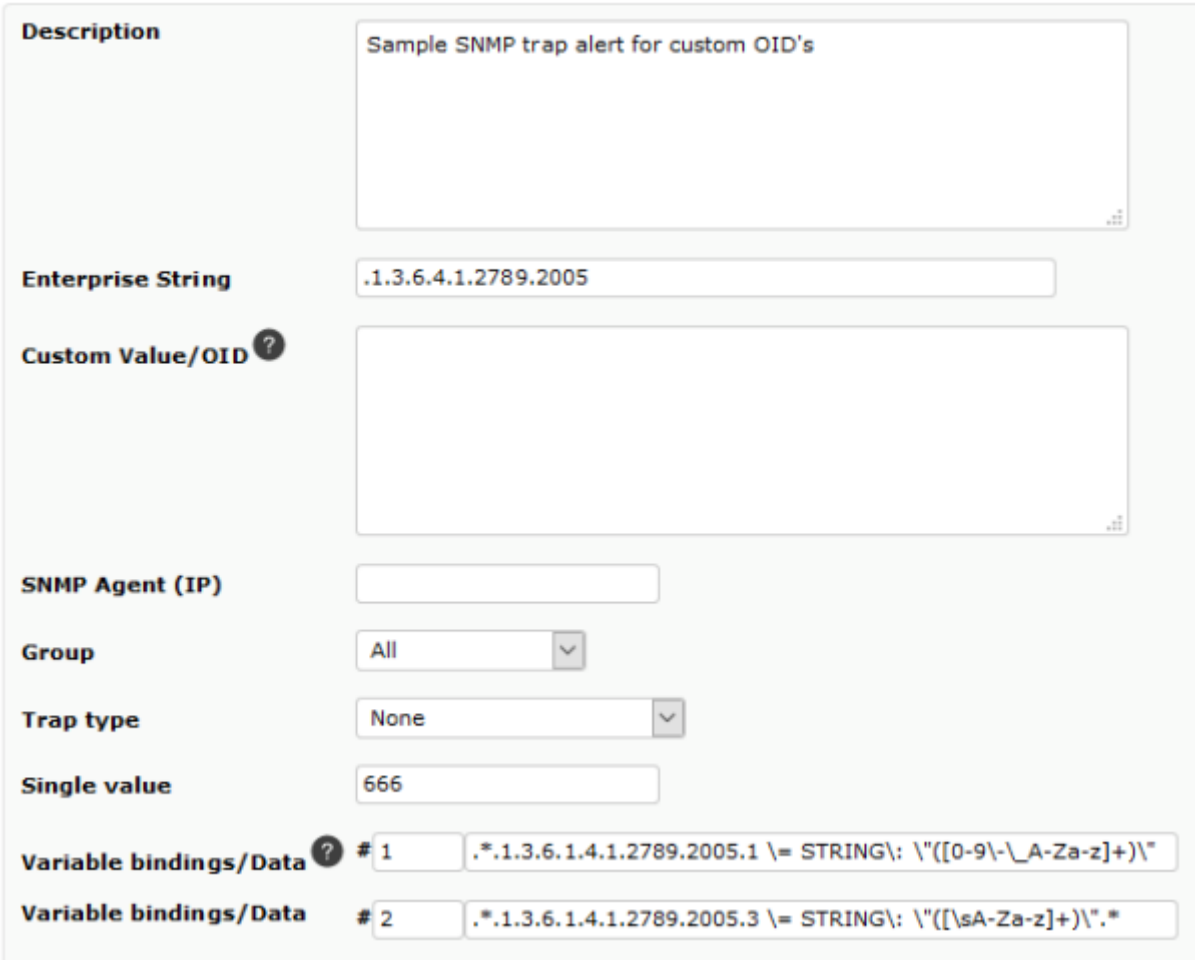
```
.*.1.3.6.1.4.1.2789.2005.3 \= STRING\: \"([\sA-Za-z]+)\".*
```

データフィールドを取得したら、それをアラートで利用する必要があります。この目的のためには、特別なマクロ _snmp_f1_、_snmp_f2_ および _snmp_f3_ を利用します。これらのマクロは、SNMP トラップアラート以外では無視されます。

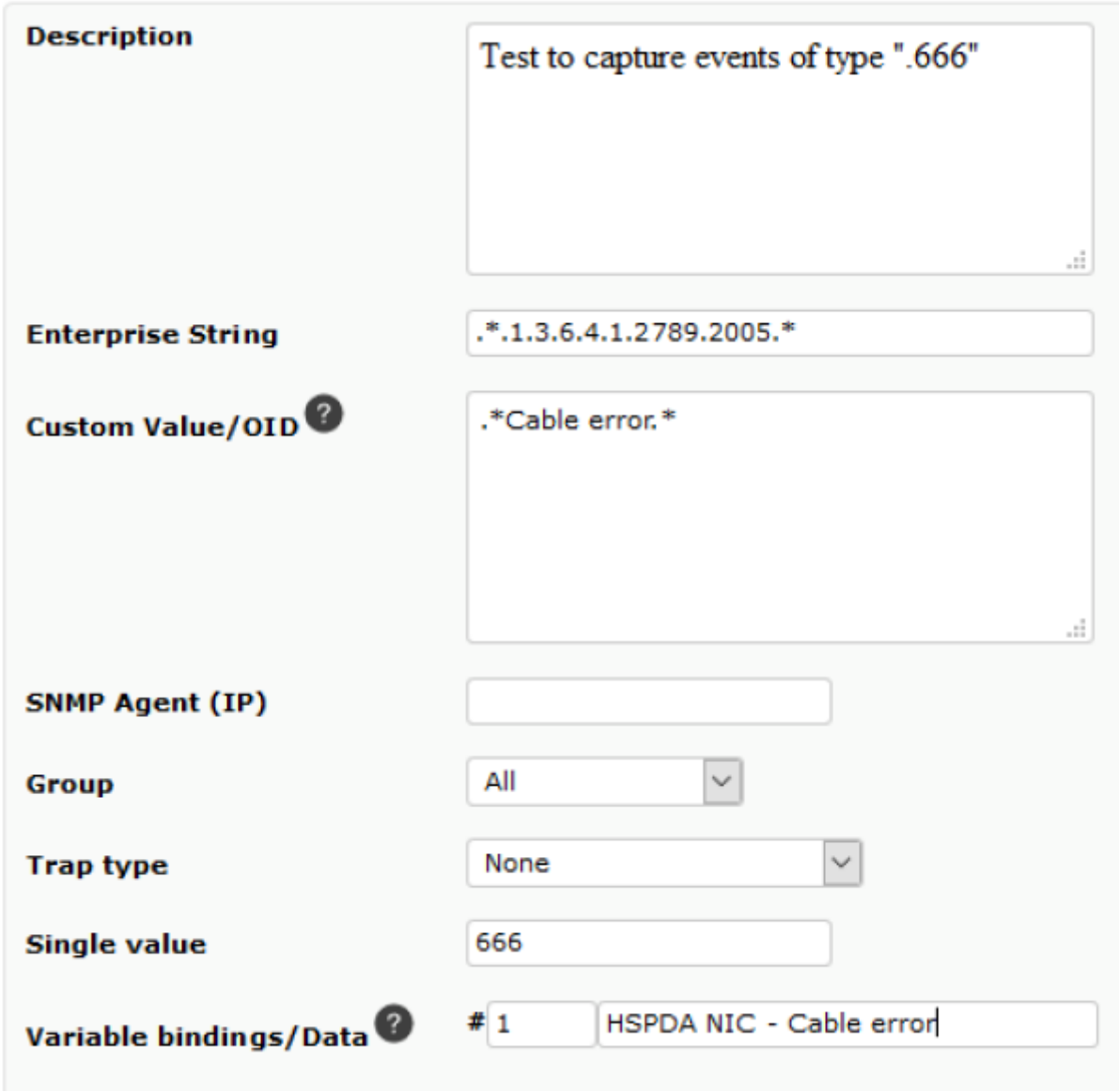
メッセージを生成するために、次のような文字列を使います。

Chassis Alert: _snmp_f2_ in device _snmp_f1_

以下の画像は、作成したアラートを表示しています。



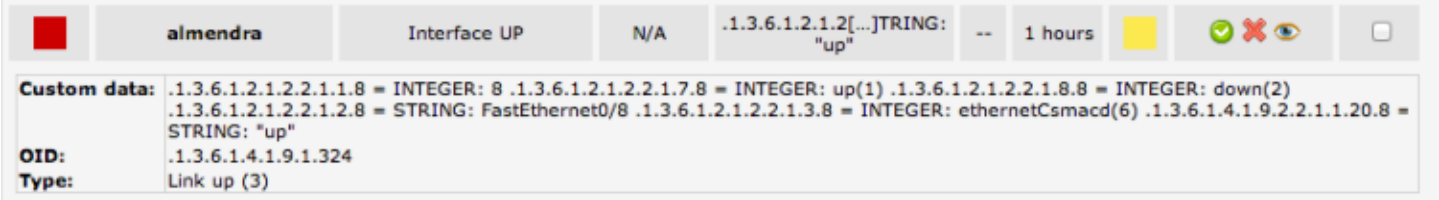
この種類のアラートをうまく作成するためには、正規表現の知識が必要です。不正な場所にスペースや記号、その他文字が入っていたりすると、正しく動作しません。SNMP アラートでは、正規表現を利用している意味を常に意識してください。 正規表現を使って簡単にアラートを生成するには、次のようにします。



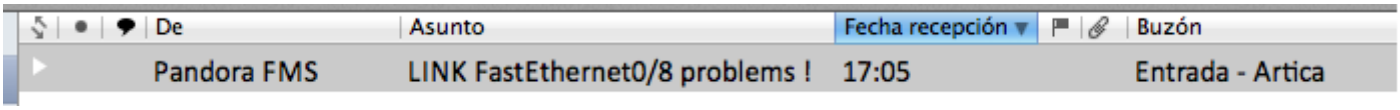
追加の例

別の例です。特定のトラップで受信するインタフェース名に関する情報を送信するメールアラートを利用します。トラップ受信時に、デバイス名、IP およびインタフェース名をメール送信します。

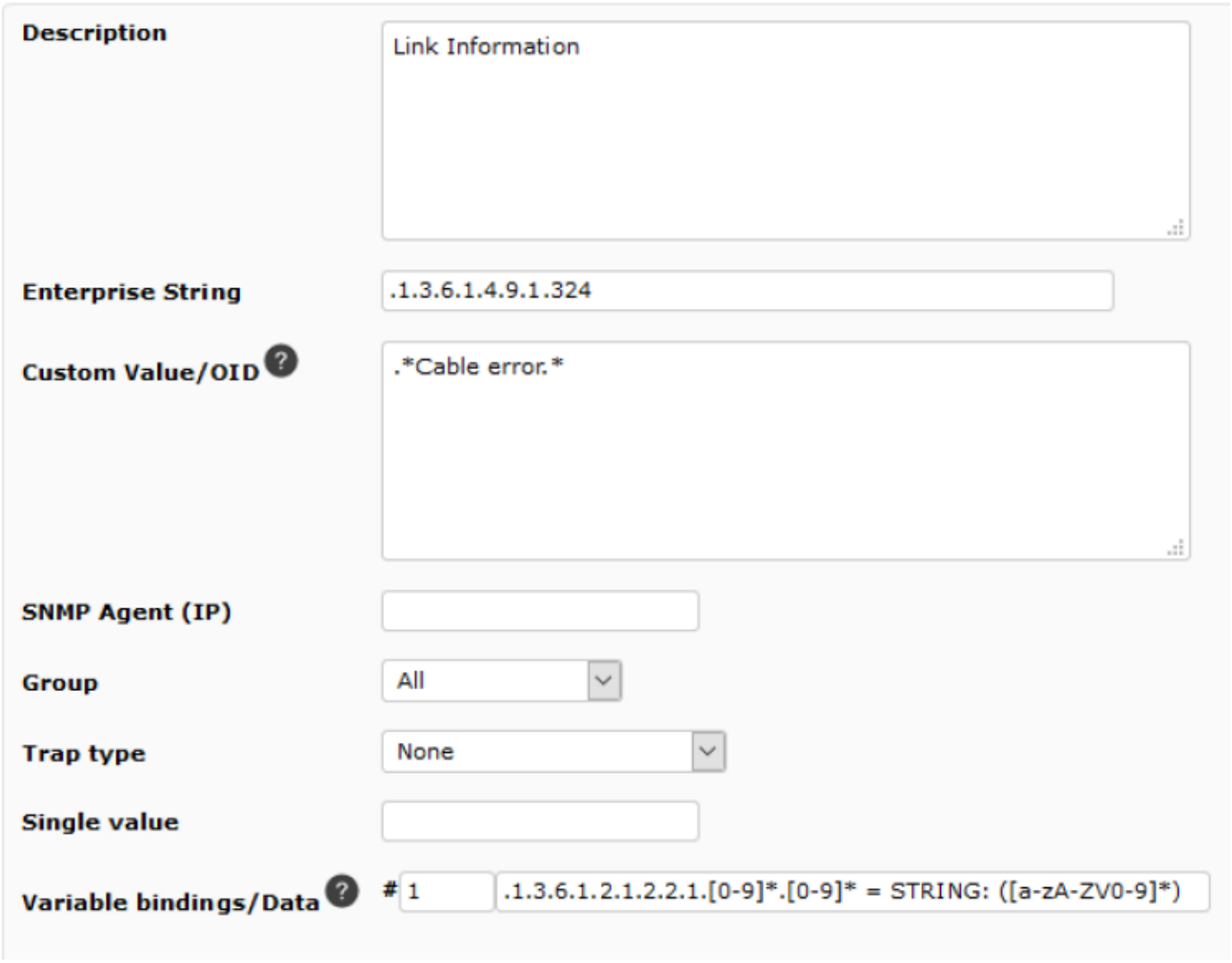
以下は、スイッチから受信したトラップです。



これは、受信したメールです。



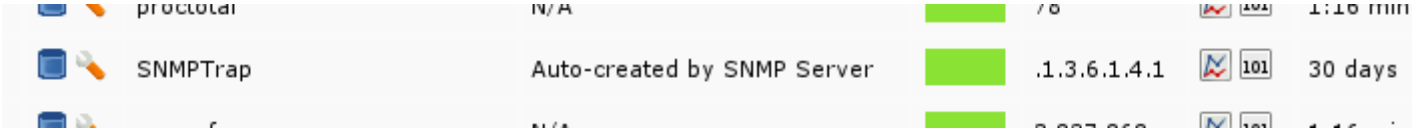
以下がトラップの設定です。



Pandora アラートとトラップの関係 / SNMP エージェントのトラップ転送

トラップに定義されたアラートは、Pandora のアラートエンジンとは完全に独立しています。そのために、“ 温度が 29度に達したらアラート上げるといったことや、冗長化電源ダウンのトラップを受信したときにアラートを上げる ” といったことがアラート管理からは設定できません。この種のアラートは、Pandora FMS のモジュールとは独立しており、トラップコンソールも、これらの要素に関連していません。

SNMP コンソールからフォワードされたトラップを含む、特殊な SNMP トラップモジュール:



これを解決するために、“エージェントへの SNMP トラップ転送(Agent SNMP Trap Forwarding)”という手法が存在します。このオプションは、トラップの発信元 IP アドレスが特定のエージェントのアドレスであった場合、トラップをそのエージェントの “SNMPTrap” という名前のモジュールへ文字列として転送します。この場合は常に、トラップはエージェントのモジュールにてテキストで受信します。モジュールは、最初のトラップを受信したときに定義されます。

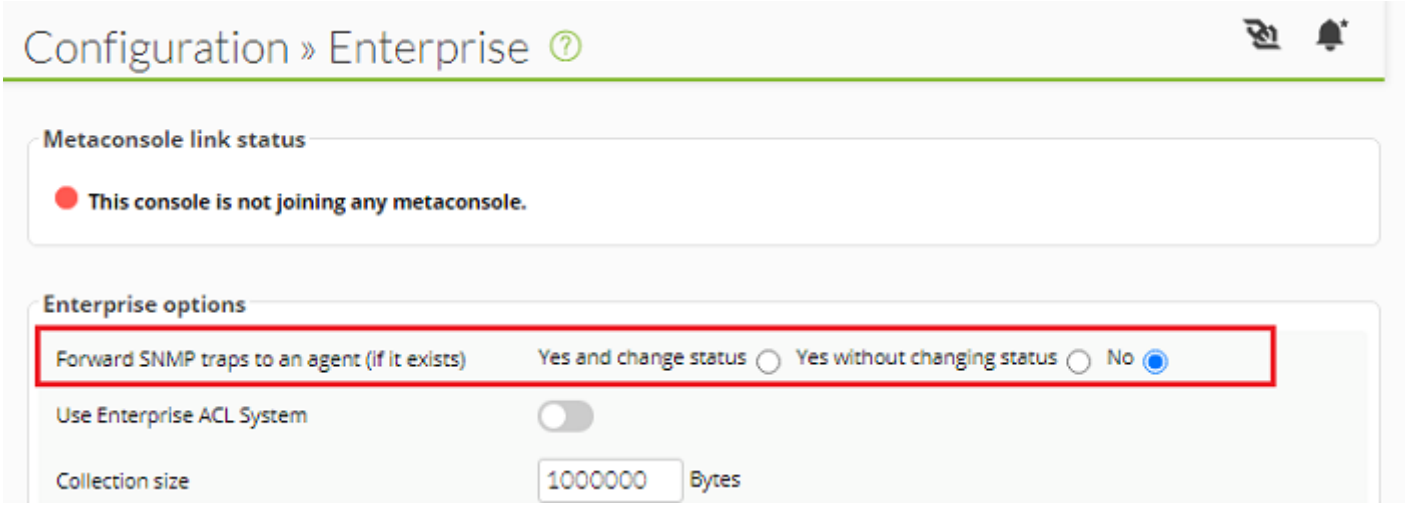
テキストのアラートは、通常のモジュールにおける標準的な方法で定義できます。これにより、別の発信元からのトラップを、それぞれ別のモジュールで扱うための SNMP モニタリングのカスタマイズが可能です。それにより、アラートの設定もそれぞれ可能になります。

SNMP トラップデータサンプル:

Delete	Timestamp	Data
✖	June 16, 2010, 8:42 pm	.1.3.6.1.4.1.9.1.324 0 .1.3.6.1.2.1.2.2.1.1.19 19 .1.3.6.1.2.1.2.2.1.7.19 = INTEGER: up(1) .1.3.6.1.2.1.2.2.1.8.19 = INTEGER: down(2) .1.3.6.1.2.1.2.2.1.2.19 = STRING: FastEthernet0 19 .1.3.6.1.2.1.2.2.1.3.19 = INTEGER: ethernetCsmacd(6) .1.3.6.1.4.1.9.2.2.1.1.20.19 = STRING: Lost Carrier
✖	June 16, 2010, 8:42 pm	.1.3.6.1.4.1.9.1.324 0 .1.3.6.1.2.1.2.2.1.1.19 19 .1.3.6.1.2.1.2.2.1.7.19 = INTEGER: up(1) .1.3.6.1.2.1.2.2.1.8.19 = INTEGER: down(2) .1.3.6.1.2.1.2.2.1.2.19 = STRING: FastEthernet0 19 .1.3.6.1.2.1.2.2.1.3.19 = INTEGER: ethernetCsmacd(6) .1.3.6.1.4.1.9.2.2.1.1.20.19 = STRING: up
✖	June 16, 2010, 8:42 pm	.1.3.6.1.4.1.9.1.324 0 .1.3.6.1.2.1.2.2.1.1.19 19 .1.3.6.1.2.1.2.2.1.7.19 = INTEGER: up(1) .1.3.6.1.2.1.2.2.1.8.19 = INTEGER: down(2) .1.3.6.1.2.1.2.2.1.2.19 = STRING: FastEthernet0 19 .1.3.6.1.2.1.2.2.1.3.19 = INTEGER: ethernetCsmacd(6) .1.3.6.1.4.1.9.2.2.1.1.20.19 = STRING: Lost Carrier
✖	June 16, 2010, 2:54 pm	.1.3.6.1.4.1.9.1.324 0 .1.3.6.1.2.1.2.2.1.1.22 22 .1.3.6.1.2.1.2.2.1.7.22 = INTEGER: up(1) .1.3.6.1.2.1.2.2.1.8.22 = INTEGER: down(2) .1.3.6.1.2.1.2.2.1.2.22 = STRING: FastEthernet0 22 .1.3.6.1.2.1.2.2.1.3.22 = INTEGER: ethernetCsmacd(6) .1.3.6.1.4.1.9.2.2.1.1.20.22 = STRING: up
✖	June 16, 2010, 2:54 pm	.1.3.6.1.4.1.9.1.324 0 .1.3.6.1.2.1.2.2.1.1.22 22 .1.3.6.1.2.1.2.2.1.7.22 = INTEGER: up(1) .1.3.6.1.2.1.2.2.1.8.22 = INTEGER: down(2) .1.3.6.1.2.1.2.2.1.2.22 = STRING: FastEthernet0 22 .1.3.6.1.2.1.2.2.1.3.22 = INTEGER: ethernetCsmacd(6) .1.3.6.1.4.1.9.2.2.1.1.20.22 = STRING: Lost Carrier

これは、エンタープライズ版の機能です。次のように、メインの設定画面で有効にできます。

エージェントへのトラップフォワーディングを有効にするオプション設定:



設定を変更したら、有効にするためには Pandora FMS サーバを再起動する必要があります。

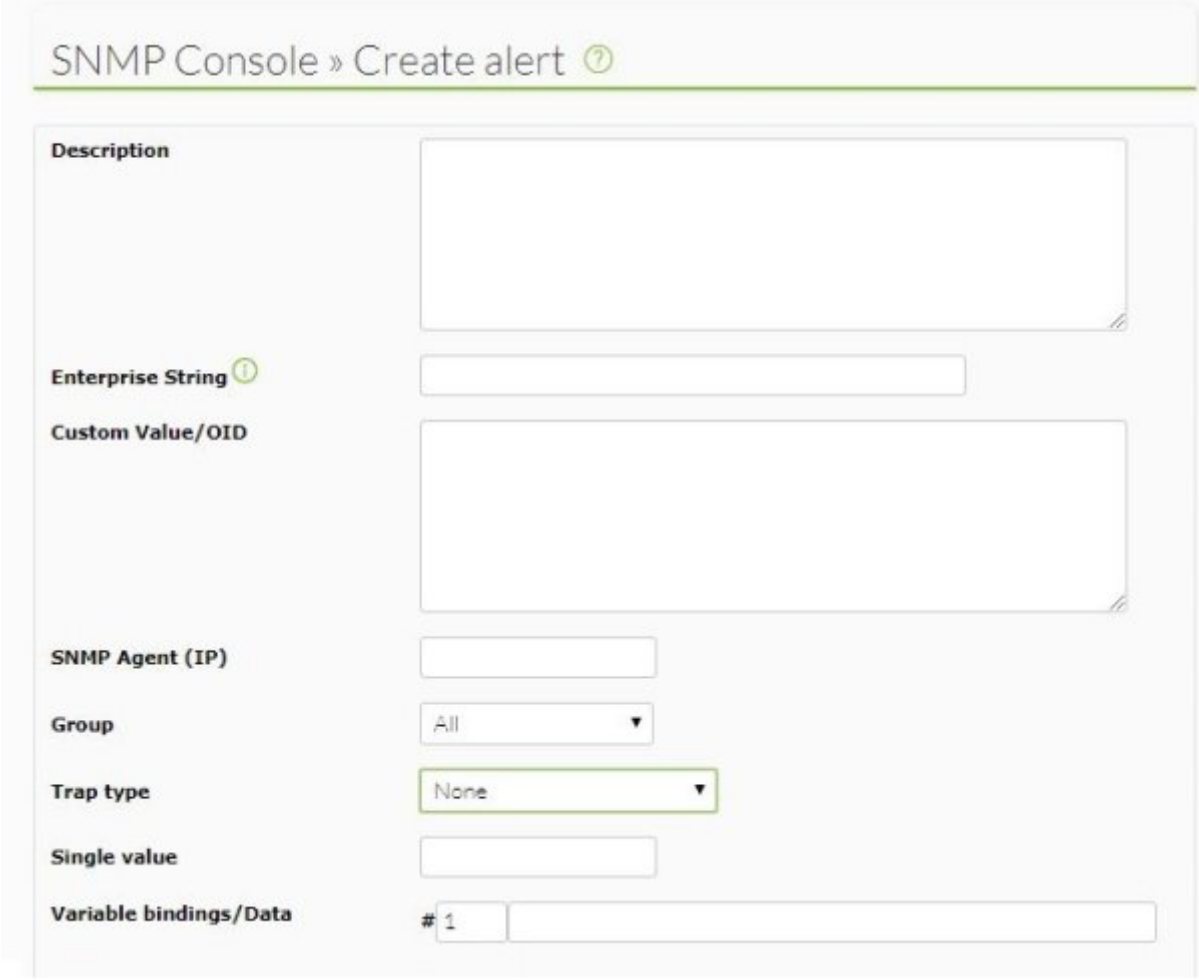
ほかには、エージェントのモジュールに、トラップを関連付ける手段もあります。例えば、トラップ受信で何らかのログファイルに“1”を書くようにし、それを読むモジュールを定義します。この方法により、想定したトラップを受信したときにモジュールの状態が変化するようになり、トラップの受信と関連付けることができます。

外部 SNMP トラップハンドラ

SNMP コンソールは、トラップを取得する目的のためだけに作られています。トラップは個々のアイテムとしてのみ処理されます。 1つのトラップは多くの情報を含むことができます。 場合によっては、実行できる唯一の監視がトラップに基づいている場合があります。 そのために、取得した 1つのトラップの情報をプラグインとして動作する外部スクリプトで後処理することができます。

トラップの詳細データを処理するために、アラートの結果としてトラップのすべてのデータをスクリプトへ渡すことができます。以下に例を示します。これは Pandora の SNMP コンソールログで見ることができるトラップビューです。

```
 2010-08-26 12:01:46 pandora 10.201.246.2 .1.3.6.1.4.1.1722
.1.3.6.1.4.1.1722.2.10.1.1.1 233 .1.3.6.1.4.1.1722.2.10.1.1.3 = STRING:
 AIX_Software_Failure .1.3.6.1.4.1.1722.2.10.1.1.2 = STRING: 08 25 2010
08:23:43:697685 .1.3.6.1.4.1.1722.2.10.1.1.8 = STRING: 1: A
 software error PERM with label CORE_DUMP, identifier C69F5C9B occurred at Wed
Aug 2 5 10:22:28 DFT 2010 on dvs02 for resource
 SYSPROC. Cause is SOFTWARE PROGRAM ABNORMALLY TERMINATED.
.1.3.6.1.4.1.1722.2.10.1.1.6 = STRING: 8
 .1.3.6.1.4.1.1722.2.10.1.1.11 = STRING: An application may not work properly
.1.3.6.1.4.1.1722.2.10.1.1.10 = STRING: An application
 may not work properly .1.3.6.1.4.1.1722.2.10.1.1.12 = INTEGER: 4
.1.3.6.1.6.3.1.1.4.3.0 = OID: .1.3.6.1.4.1.1722
```



スクリーンショットで、どのようにアラートを作成するかがわかると思います。トラップの内容 (_data_) に応じてスクリプトを実行します。また SNMP アラートが作成されます。この場合、特定の OID (.1.3.6.1.4.1.1722.2.10.1.1.1) にマップされます。マップする OID の範囲を広くすることも可能で、例えば、(.1.3.6.1.4.1.1722) を指定すれば、この OID すべてのトラップ (AIX の特定 MIB の一部を想定) でスクリプトが呼び出されるようになります。

このデータを処理するスクリプトが実行されます。 また、エージェントからのデータであるかのようにXMLファイルを作成し、'/var/spool/pandora/data_in' に移動することによってPandora FMSに直接データを書き込み、トラップを分析できます。 この場合の基本的なスクリプトは、複雑な情報を生成するスクリプトです。 すでにこのトラップに関する十分な情報を処理する処理は、以下から構成されます。

- オリジナルIPアドレス
- メインイベント (コールドスタート)
- 2つ目のイベント(説明): AIX_Software_Failure, 1: A software error PERM with label CORE_DUMP, identifier C69F5C9B occurred at Wed Aug 2 5 10:22:28 DFT 2010 on dvs02 for resource SYSPROC. Cause is SOFTWARE PROGRAM ABNORMALLY TERMINATED, An application may not work properly, An application may not work properly.

これらのデータをパースするスクリプトを作る場合、例えば “miscript.pl” というスクリプトであれば/var/spool/pandora/data_in にランダムな番号をつけたファイル名で XML ファイルを書くようにします。 例えばsnmp_gateway.31415.data です。

生成した XML ファイルは次のようになります。

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<agent_data description='' group='' os_name='aix' os_version='' interval='300'
version='3.1(Build 100608)' timestamp='2010/08/26 12:20:26'
agent_name='10.201.246.2'>
  <module>
    <name><![CDATA[Critical_Event]]></name>
    <description><![CDATA[]]></description>
    <type>async_proc</type>
    <data><![CDATA[1]]></data>
  </module>
<module>
    <name><![CDATA[events]]></name>
    <description><![CDATA[]]></description>
    <type>generic_string</type>
    <datalist>
      <data><value><![CDATA[AIX_Software_Failure]]></value></data>
      <data><value><![CDATA[A software error PERM with label CORE_DUMP,
identifier C69F5C9B occurred at Wed Aug 2 5 10:22:28 DFT 2010 on dvs02 for
resource SYSPROC.]]></value></data>
      <data><value><![CDATA[Cause is SOFTWARE PROGRAM ABNORMALLY TERMINATED, An
application may not work properly, An application may not work
properly.]]></value></data>
    </datalist>
  </module>
</agent_data>
```

アプリケーションは何でも実現できカスタマイズ可能です。とても強力な構造になっています。多くのシステムでは、情報はテキストだけではなく数値でも取得できます。グラフを書きたい場合等では、数値情報のモジュールを利用します。なお、全てのデータは常に非同期であることに注意してください。

実例: トラップを用いた ESX のモニタリング

最も問題となりうるモニタリングの一つにVMware ESX のようにバージョンによって情報の収集方法が異なるベンダが提供のシステムがあります。この章では、外部 SNMP トラップハンドラを使ってESX システムをモニタする方法を説明します。

ESX のトラップは次のようになっています。

```
 .1.3.6.1.4.1.6876.4.3.301 = STRING: "host"  .1.3.6.1.4.1.6876.4.3.302 = STRING:
"c7000-06-01.tsm.inet"      .1.3.6.1.4.1.6876.4.3.303 = ""
 .1.3.6.1.4.1.6876.4.3.304 = STRING: "Green" .1.3.6.1.4.1.6876.4.3.305 = STRING:
"Yellow"    .1.3.6.1.4.1.6876.4.3.306 = STRING: "Host
 cpu usage - Metric Usage = 1%"
```

```
 .1.3.6.1.4.1.6876.4.3.301 = STRING: "host"  .1.3.6.1.4.1.6876.4.3.302 = STRING:
"dl360-00.tsm.inet" .1.3.6.1.4.1.6876.4.3.303 = ""
 .1.3.6.1.4.1.6876.4.3.304 = STRING: "Yellow".1.3.6.1.4.1.6876.4.3.305 = STRING:
"Green"     .1.3.6.1.4.1.6876.4.3.306 = STRING: "Host
 memory usage - Metric Usage = 84%"
```

```
 .1.3.6.1.4.1.6876.4.3.301 = STRING: "host"  .1.3.6.1.4.1.6876.4.3.302 = ""
.1.3.6.1.4.1.6876.4.3.303 = ""  .1.3.6.1.4.1.6876.4.3.304 =
 STRING: "Red"       .1.3.6.1.4.1.6876.4.3.305 = STRING: "Green"
.1.3.6.1.4.1.6876.4.3.306 = STRING: "Datastore usage on disk - Metric
 Storage space actually used = 55%"
```

見ての通り、トラップに CPUディスク、メモリ の情報がまとめられています。トラップの内容を分析し、XML ファイルを作成する小さなスクリプトを書き、トラップハンドラにすることを考えます。トラップハンドラの書き方は共通です。手順は 4つのステップとなります。

1. ハンドラスクリプトを作成します。以下に示すスクリプトを参考にしてください。
2. アラートコマンドを作成します。
3. 上記のコマンドを使って、アラートアクションを作成します。必要に応じて適用先のエージェントを指定します。(複数の ESX がある場合は、それぞれ別のエージェントに適用したいような場合もあるかもしれません。)
4. エンタープライズ OID (この手法でモニタする全トラップのもの) および発信元 IP アドレスにマップする SNMP トラップアラートを作成します。

では、トラップハンドラを作成する、最初のステップを見てみましょう。

トラップハンドラ: esx_trap_manager.pl

```
#!/usr/bin/perl
# (c) Sancho Lerena 2010 <slerena@artica.es>
# Specific Pandora FMS trap collector for ESX

use POSIX qw(setsid strftime);
```

```
sub show_help {
    print "\nSpecific Pandora FMS trap collector for ESX\n";
    print "(c) Sancho Lerena 2010 <slerena@artica.es>\n";
    print "Usage:\n\n";
    print "   esx_trap_manager.pl <destination_agent_name> <TRAP DATA>\n\n";
    exit;
}

sub writexml {
    my ($hostname, $xmlmessage ) = @_;
    my $file = "/var/spool/pandora/data_in/$hostname.".rand(1000).".data";

    open (FILE, ">> $file") or die "[FATAL] Cannot write to XML '$file'";
    print FILE $xmlmessage;
    close (FILE);
}

if ($#ARGV == -1){
    show_help();
}

$chunk = "";

# First parameter is always destination host for virtual server
$target_host = $ARGV[0];

foreach $argnum (1 .. $#ARGV) {
    if ($chunk ne ""){
        $chunk .= " ";
    }
    $chunk .= $ARGV[$argnum];
}

my $hostname = "";
my $now = strftime ("%Y-%m-%d %H:%M:%S", localtime());
my $xmldata = "<agent_data agent_name='$target_host' timestamp='$now'
version='1.0' os='Other' os_version='ESX_Collectordime '
interval='9999999999'>";

if ($chunk =~ m/.1.3.6.1.4.1.6876.4.3.302 \= STRING\: ([A-Za-z0-9\-\.]*)\s\.1/){
    $hostname = "_".$1;
}

if ($chunk =~ m/Host cpu usage \- Metric Usage \= ([0-9]*)\z/){
    $value = $1;
    $module_name = "CPU_OCUPADA$hostname";
}

if ($chunk =~ m/Host memory usage \- Metric Usage = ([0-9\.]*)\z/){
    $value = $1;
    $module_name = "MEMORIA_OCUPADA$hostname";
}
```

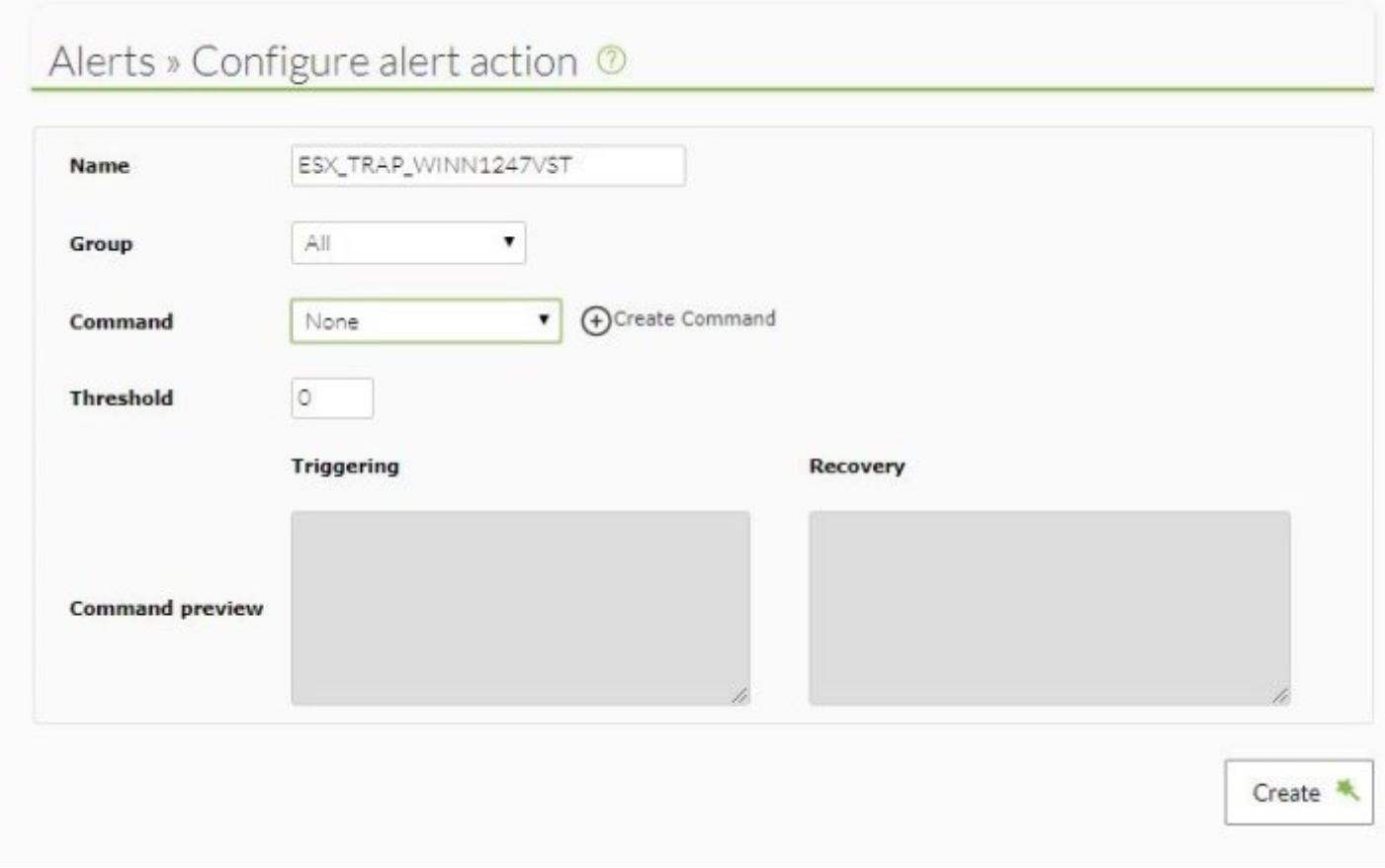
```
if ($chunk =~ m/Datastore usage on disk \- Metric Storage space actually used \=
([0-9\.]*)\z/){
    $value = $1;
    $module_name = "DISCO_OCUPADO$hostname";
}

$xmldata .=
"<module><name>$module_name</name><type>async_data</type><data>$value</data></mo
dule>\n";

$xmldata .= "</agent_data>\n";
writexml ($target_host, $xmldata);
```

ステップ 2: アラートコマンドの作成

この例では、コマンドスクリプトを /tmp に置いています。(実際には、より安全な場所に置いてください。) そして、それに実行権限を与えます (chmod 755)

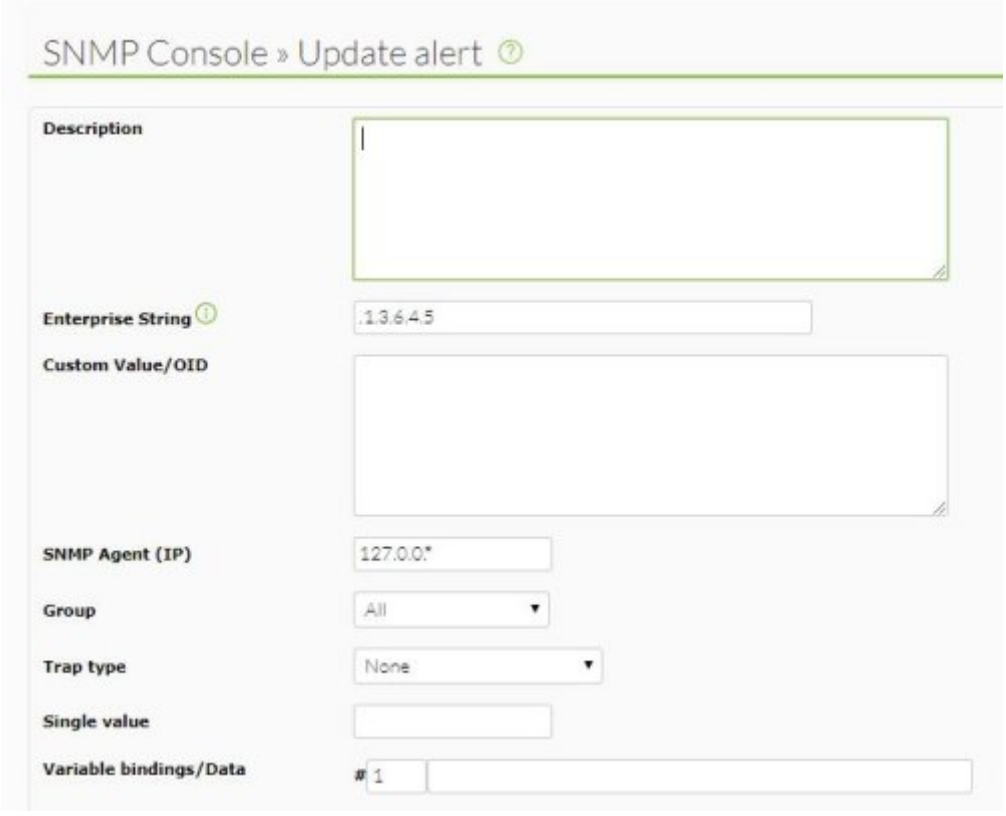
ステップ 3: アラートアクションの作成

特定のエージェントのトラップの情報を送信するアクションを作成します。この場合、情報は WINN1247VSR というエージェントに送られます。上記のコマンドは、エージェント名をパラメータとして受け取り、(ESX バーチャルセンターの) 全ての情報を処理し、トラップからのデータを分割します。特に制限はなく、トラップで送られる全ての情報を処理します。



ステップ 4: SNMP アラートの作成

作成したアクションを使って、トラップのアラートを設定します。

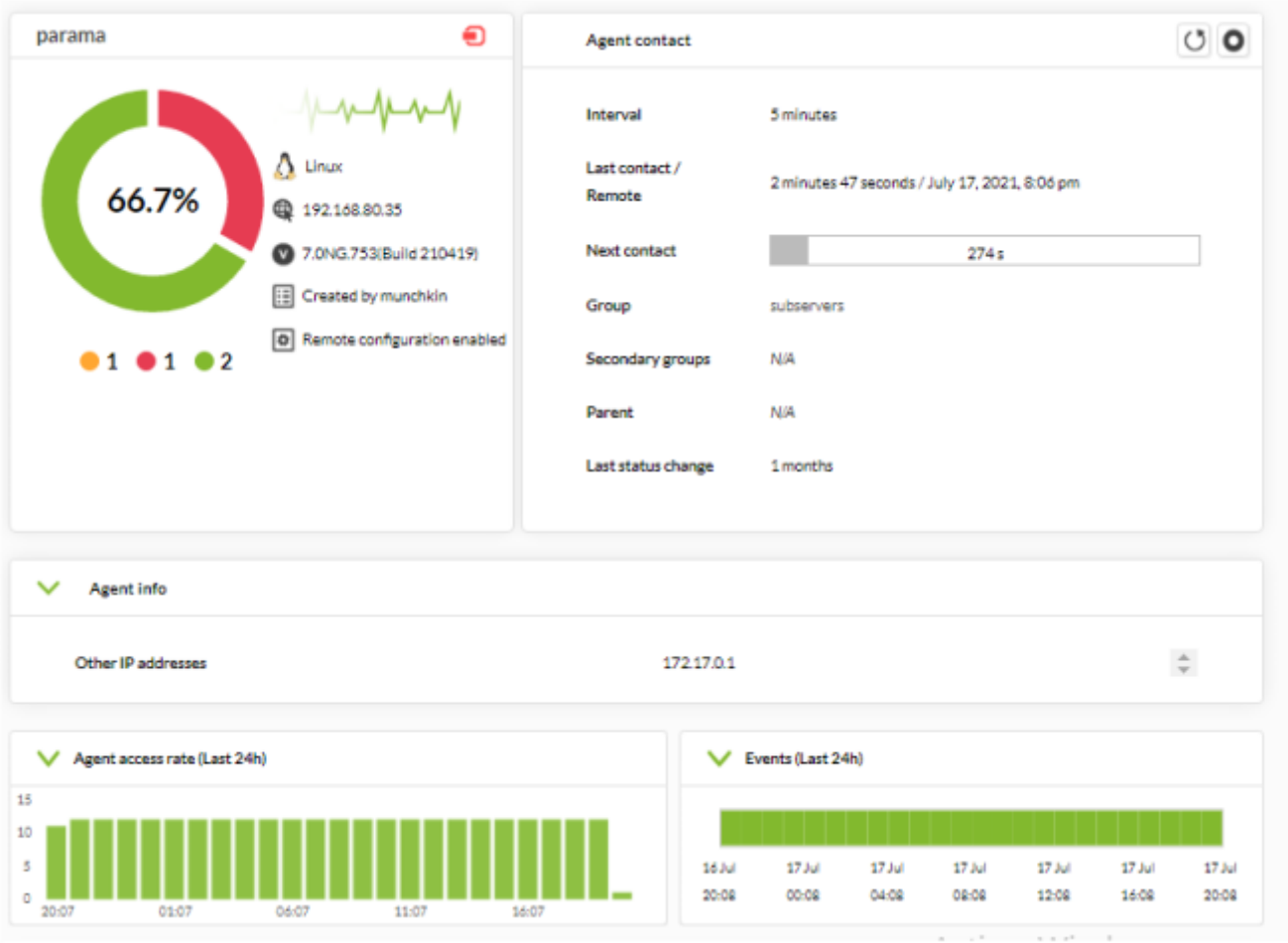


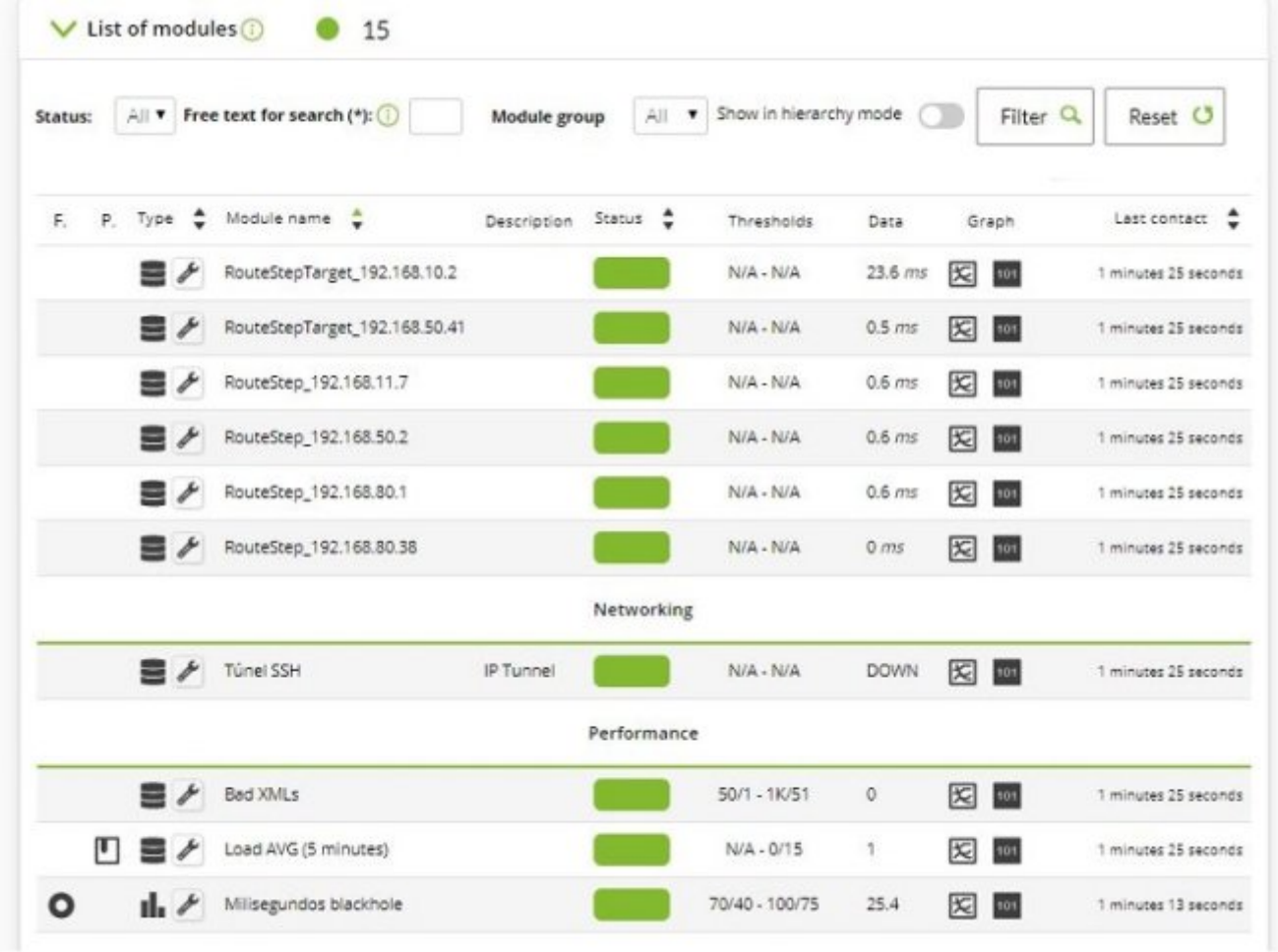
ESX の全てのトラップを処理するために、ESX のトラップにマップする OID である

.1.3.6.1.4.1.6876.4.3.301 を指定します。IP アドレスフィルタリングにより、それぞれの VirtualCenter のソース IP アドレスでフィルタリングすることもできます。

データの表示

以下に情報の参照例を示します。これにより、通常のモジュールとして管理できます。





SNMP trap 転送

Pandora FMS では、Pandora サーバの設定ファイルでsnmp_forward_trapトークンを有効にすることにより、SNMP trap を外部のホストへ転送することができます。

SNMP v1 を使った trap 転送設定例

```
snmp_forward_trap 1
snmp_forward_ip 192.168.1.145
snmp_forward_version 1
snmp_forward_community public
snmp_forward_secName
snmp_forward_engineid
snmp_forward_authProtocol
snmp_forward_authPassword
snmp_forward_privProtocol
snmp_forward_privPassword
snmp_forward_secLevel
```

SNMP v2c を使った trap 転送設定例

```
snmp_forward_trap 1
```

```
snmp_forward_ip 192.168.1.145
snmp_forward_version 2c
snmp_forward_community public
snmp_forward_secName
snmp_forward_engineid
snmp_forward_authProtocol
snmp_forward_authPassword
snmp_forward_privProtocol
snmp_forward_privPassword
snmp_forward_secLevel
```

SNMP v3 を使った trap 転送設定例

この例は、SNMP v3 trap の知識が必要になるため特に難しいです。リモートの SNMP エージェントがsnmp_forward_ipで定義されており、次の設定が /etc/snmp/snmptrapd.conf ファイルに書かれていることを想定します。

```
createUser -e 0x0102030405 myuser MD5 mypassword DES myotherpassword
```

Pandora サーバの設定ファイルは次のようになります。

```
snmp_forward_trap 1
snmp_forward_ip 192.168.1.145
snmp_forward_version 3
snmp_forward_secName myuser
snmp_forward_engineid 0x0102030405
snmp_forward_authProtocol MD5
snmp_forward_authPassword mypassword
snmp_forward_privProtocol DES
snmp_forward_privPassword myotherpassword
snmp_forward_secLevel authNoPriv
```

より詳細は、 NET-SNMP's v3 Trapsを参照してください。

snmptrapd デーモンの個別管理

何らかの理由により snmptrapd デーモンを Pandora FMS から独立して管理したい場合(Pandora FMS デーモンとは独立して停止・起動をしたい場合)は、いくつか考慮すべきことがあります。

1. Pandora FMS サーバにおいて snmpconsole パラメータを有効化する必要があります

2. Pandora FMS サーバで設定されるログは、snmptrapd を独立して管理する場合でも同じでなければいけません。

3. snmptrap の呼び出しは特定のフォーマットである必要があり、標準的なシステムからの呼び出し

は利用できません。呼び出しは次のようにする必要があります(パラメータ -A はとても重要です)。

```
/usr/sbin/snmptrapd -A -t -On -n -a -Lf /var/log/pandora/pandora_snmptrap.log -p
/var/run/pandora_snmptrapd.pid --format1=SNMPv1[**]%4y-%02.2m-
%l[**]%02.2h:%02.2j:%02.2k[**]%a[**]%N[**]%w[**]%W[**]%q[**]%v\n --
format2=SNMPv2[**]%4y-%02.2m-%l[**]%02.2h:%02.2j:%02.2k[**]%b[**]%v\n
```

4. snmptrapd トークンが、Pandora FMS 設定ファイル内に設定されている必要があります。

```
snmp_trapd manual
```

5. この機能を有効化したら、次の手順を実施する必要があります。

- /etc/pandora/pandora_server.conf の設定を変更
- Pandora FMS サーバを停止
- snmptrapd プロセスが動作していないことを確認 (もし動いていたら、停止するまで待つか kill します)
- snmptrapd を手動で起動 (上記のフォーマットにて)
- Pandora FMS サーバを起動

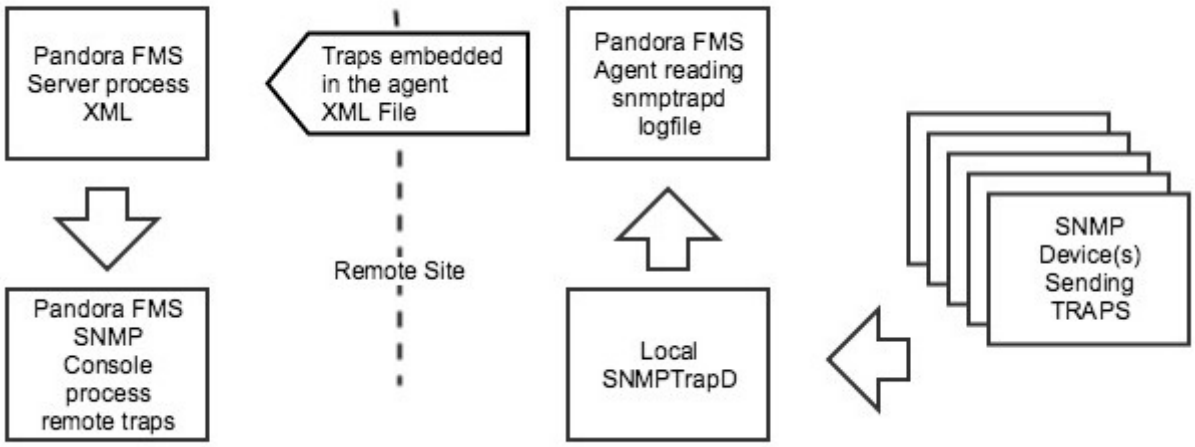
トラップログファイルの管理

pandora_snmptrap.log.index および *pandora_snmptrap.log* が変更されていなければ snmptrapd プロセスは、pandora サーバプロセスの停止および起動に依存せず、停止および起動することができます。これらのファイルに変更が加わっている場合は、pandora サーバの再起動が必要です。トラップのログファイルを外部でローテートする必要がある場合は、前述の 2つのファイルを削除したあとに pandora サーバを再起動する必要があります。

SNMP トラップバッファリング

SNMPトラップが信頼できない接続を介して外部マネージャに送信されると、情報が失われます Pandora FMS では、トラップをローカルの *snmptrapd* からの転送ではなく、信頼できる方法で Pandora FMS サーバに転送することができます。

アーキテクチャ



- SNMP エージェントは、ローカルの *snmptrapd* にトラップを送信します。
- ローカルの Pandora FMS エージェントが *snmptrapd* のログファイルからトラップを読み取り、XML データファイルを用いて指定の Pandora FMS サーバへ送信します。それは XML バッファに保存され必要に応じてリトライされます。
- データサーバは、XML データファイルからトラップを読み込み、プレーンテキストファイルに展開します。
- SNMP コンソールは、プレーンテキストファイルからトラップを処理します。

SNMPコンソールが *snmptrapd* のログファイルから直接トラップを処理する方が効率的です。この設定は、直接の接続や信頼性に不安がある場合にのみ利用します。

前提条件

- ローカルの *snmptrapd* がトラップを受信すること。
- ローカルの Pandora FMS エージェントがあること。
- Pandora FMS がインストールされていること。

設定

snmptrapd

/etc/snmp/snmptrapd.conf を編集し、Pandora FMS と互換性があるフォーマットでログをファイルに記録する設定になっているか確認します。(必要に応じてログファイル名を変更することができます)

```
[snmp] logOption f /var/log/snmptrapd.log
format1 SNMPv1[**]%4y-%02.2m-
%l[**]%02.2h:%02.2j:%02.2k[**]%a[**]%N[**]%w[**]%W[**]%q[**]%v\n
format2 SNMPv2[**]%4y-%02.2m-%l[**]%02.2h:%02.2j:%02.2k[**]%b[**]%v\n
```

Pandora FMS エージェント

snmptrapd のログファイルからデータを読む Pandora FMS エージェントに付属の *grep_snmptrapd* プラグインを利用します。

ローカルのエージェント設定ファイル */etc/pandora/pandora_agent.conf* を編集し、必要に応じて *snmptrapd* のログファイルのパスを指定する次の行を追加します。

```
module_plugin grep_snmptrapd /var/log/snmptrapd.log
```

Pandora FMS サーバ

SNMP コンソールが、データサーバにて書かれた外部ログファイルからトラップを処理するよに設定する必要があります。

サーバ設定ファイル */etc/pandora/pandora_server.conf* 編集し、次の設定をします。

- SNMP コンソールが有効であるか確認します。

```
snmpconsole 1
```

- データサーバが有効であるか確認します。

```
dataserver 1
```

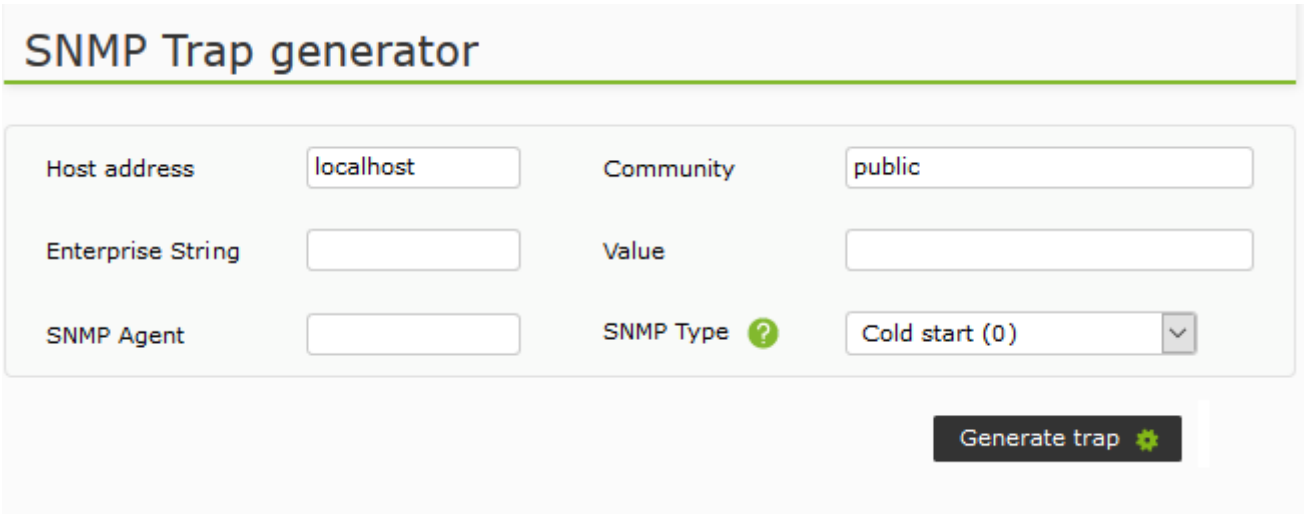
- 外部 SNMP ログファイルを設定します。存在しない場合は、SNMP コンソールが作成します。

```
snmp_extlog /var/log/pandora/pandora_snmptrap.ext.log
```

snmp_extlog は Pandora FMS サーバで書き込み可能な任意のファイルです。(同様に */etc/pandora/pandora_agent.conf* で定義されている)*snmp_logfile* とは異なります。

トラップジェネレータ

このツールは、SNMP コンソールから参照できるトラップを生成します。



トラップジェネレータを正しく設定するには、次のフィールドを入力する必要があります。

ホストアドレス(Host Address)

トラップ送信先 IP アドレス。

コミュニティ(Community)

トラップジェネレータでアクセスするときの SNMP コミュニティ。

エンタープライズ文字列(Enterprise String)

トラップの OID です。例: 1.3.6.1.2.1.2.2.1.8

値(Value)

トラップで送信するデータであり、値です。

SNMP エージェント(SNMP Agent)

トラップをシミュレートするエージェントです。

SNMP タイプ (SNMP Type)

以下から SNMP タイプを選択します。

- Cold Start: エージェントが開始または再開されたことを意味します。
- Warm Start: エージェント設定が変更されたことを意味します。
- Link down: 通信インタフェースが利用できない状態になった(無効化)ことを意味します。
- Linu up: 通信インタフェースが利用できる状態になったことを意味します。
- Authentication failure: エージェントが(コミュニティによって)認証できない NMS を受信したことを意味します。
- EGP neighbor loss: ルータが EGP プロトコルを使用しているシステムで、近くのホストが利用できない状態になったことを示します。
- Enterprise: ベンダトラップを含む、すべての新規トラップです。

Pandora FMS ドキュメント一覧に戻る