3 SNMP のトラップ監視

概要

SNMPのトラップの受信は、SNMP対応のデバイスに対してクエリーを発行することの反対の機能です。

情報はSNMP対応のデバイスから送信されZabbixによって収集または「トラップ」されます。

通常のトラップは、状況の変化が起こったときに送信され、クエリーに使用されるエージェントサイドのポート番号161と対照的に、ポート番号162でエージェントがサーバに接続します。トラップの使用によって、クエリーの間に発生しクエリーデータでは見落とされるような短期間の障害を感知することができます。

ZabbixでのSNMPトラップの受信は、**snmptrapd**と、Zabbixに対してトラップを受け渡すために組み込まれたメカニズム - PerlスクリプトまたはSNMPTTのどちらかで動作するように設計されています。

トラップの受信の仕事の流れは以下の通りです：

1. **snmptrapd** がトラップを受信
2. snmptrapd がトラップを、SNMPTTに渡すか Perl のトラップレシーバを呼び出し
3. SNMPTT または Perl のトラップレシーバが構文を解析し、書式を整えてファイルにトラップを書き出し
4. Zabbix SNMP トラッパーが、トラップファイルを読み込み、構文を解析
5. 各トラップについてZabbix が、受信したIPまたはDNSアドレスに相当するホスト上の SNMP インターフェースを検索
6. 検出したSMNP インターフェースについてsnmptrap[regex] アイテムのすべての正規表現にトラップが比較され：一致した場合、そのトラップをすべての一致したアイテムの値として設定、一致しないけれどsnmptrap.fallbackアイテムが存在する場合は、そのトラップをそのアイテムの値として設定
7. 相当するSNMPインターフェースのどれとも一致しなかった場合、デフォルトでZabbixが一致しなかったトラップのログを残す。（これは、[監視]→[一般]→[その他]の「Log unmatched SNMP traps」で設定されます）

3.1 SNMP トラップの設定

フロントエンドの以下のフィールドの設定は、このアイテムタイプに固有のものです：

- ホストがSNMPインターフェースを持っている必要があります。

[設定]→[ホスト]で、ホストインターフェースのフィールドにIPアドレスまたはDNSアドレスを指定してSNMPインターフェースを設定します。受信するトラップそれぞれのアドレスは、対応するホストを探すために、すべてのSNMPインターフェースのIPアドレスおよびDNSアドレスと比較されます。

- アイテムの設定

キーのフィールドで、SNMPトラップキーの1つを使用します：

キー		
説明	**戻り値**	**コメント**
snmptrap[regex]		

キー		
説明	戻り値	コメント
対応するアドレスから**正規表現**に一致するすべてのSNMPトラップを取得します。	SNMPトラップ	このアイテムはSNMPインターフェースにのみ設定可能です。 **このアイテムは、バージョン2.0.0からサポートされました。** 注意：Zabbix2.0.5から、このアイテムキーのパラメータでユーザーマクロとグローバル正規表現がサポートされました。
snmptrap.fallback		
そのインターフェースに対してどのsnmptrap[]アイテムでも取得されなかったアドレスからすべてのSNMPトラップを取得します。	SNMPトラップ	このアイテムはSNMPインターフェースにのみ設定可能です。 **このアイテムは、バージョン2.0.0からサポートされました。**

パースされるタイムスタンプの**データ型**は「ログ」に設定します。「数値」のように他の書式も設定可能ですが、カスタムのトラップハンドラを必要とする場合があることに注意してください。

SMMPトラップ監視が動作するためには、まず最初に正しくセットアップされている必要があります。

3.2 SNMPトラップ監視のセットアップ

Zabbix サーバ/プロキシの設定

トラップを読み込むためにはZabbix サーバまたはZabbix プロキシがSNMPトラッパのプロセスを開始するように設定されており、SNMPTTまたはPerlのトラップレシーバによって書き込まれるトラップファイルを指している必要があります。 その設定を行うためには、設定ファイル（zabbix_server.conff または zabbix_proxy.conf）を編集します。

1. StartSNMPTrapper=1
2. SNMPTrapperFile=[TRAP FILE]

SNMPTTの設定

最初に、snmptrapdがSNMPTTを使用するように設定します。

最高のパフォーマンスのためにはSNMPTTはトラップを受け渡すsnmptthandler-embeddedを使用するデーモンとして設定されていることSMNPTTのホームページの設定の概要を参照してください：
http://snmptt.sourceforge.net/docs/snmptt.shtml

SNMPTTがトラップを受信するよう設定されたら、トラップのログを記録する設定をします：

1. Zabbixによって読み込まれるトラップファイルにトラップのログを保存：
   log_enable = 1
   log_file = [TRAP FILE]
2. 日時の書式を設定：
   date_time_format = %H:%M:%S %Y/%m/%d = [DATE TIME FORMAT]

これで□Zabbixが認識するように、トラップをフォーマットします□□snmptt.confを編集）：

1. 各FORMATのステートメントは、□ZBXTRAP [address]□で始まるようにします□[address]の値が、Zabbix 上のSNMP インターフェスのIPアドレスとDNSアドレスアドレスと比較されます。例：EVENT coldStart .1.3.6.1.6.3.1.1.5.1 “Status Events” Normal \\FORMAT ZBXTRAP $aA Device reinitialized (coldStart)
2. さらに詳細は、下記のSNMPトラップのフォーマットを参照してください。

知らないトラップを使用しないようにしてください - Zabbixは、それらを認識できません。知らないトラップは、snmptt.conf 内で general イベントと定義することで、扱うことができるようになります□ EVENT general .* “General event” Normal

Perlトラップレシーバーの設定

要件□Perl□–enable-embedded-perl でコンパイルされたNet-SNMP□Net-SNMP 5.4からは、デフォルトで実行されています）

Perl トラップレシーバ□ misc/snmptrap/zabbix_trap_receiver.pl を探してください）は□snmptrapd からトラップをZabbix サーバに直接受け渡すために使用されます。これを設定するには：

- snmptrapd 設定ファイル□snmptrapd.conf□にperlのスクリプトを追加します。例：
  perl do “[FULL PATH TO PERL RECEIVER SCRIPT]”;
- レシーバーを設定します。例：
  $SNMPTrapperFile = '[TRAP FILE]'
  $DateTimeFormat = '[DATE TIME FORMAT]';

SNMP トラップのフォーマット

すべてのカスタマイズされたperlのトラップレシーバとSNMPTTトラップ設定は、次の方法でトラップをフォーマットする必要があります：

```
[timestamp] [the trap, part 1] ZBXTRAP [address] [the trap, part 2]
```

それぞれの意味は以下の通りです。

- [timestamp] - ログアイテムに使用されているタイムスタンプ
- ZBXTRAP - この行で新しいトラップを開始することを示すヘッダ
- [address] - このトラップに対してホストを発見するのに使用されるIP アドレス

処理中のメッセージから□ZBXTRAP□と「[address]□は、削除されることに注意してください。トラップが他の方法でフォーマットされている場合は、Zabbixがそのトラップを期待通りにはパースしない場合があります。

ファイル内のトラップの行の例：

```
11:30:15 2011/07/27 .1.3.6.1.6.3.1.1.5.3 Normal "Status Events" localhost - ZBXTRAP 192.168.1.1 Link down on interface 2.  Admin state: 1.  Operational state: 2
```

これは、次のような□IP=192.168.1.1のSNMPインターフェースのトラップになります：

```
11:30:15 2011/07/27 .1.3.6.1.6.3.1.1.5.3 Normal "Status Events" localhost -
Link down on interface 2.  Admin state: 1.  Operational state: 2
```

3.3 システム要件

ログローテーション

Zabbixは、いかなるログローテションシステムも提供していません – したがって、ユーザーがそれを処理します。ログローテーションは、最初に古いファイルの名前を変更し、トラップが1つも残らないように後からファイルを削除します:

1. Zabbixが最新のロケーションのトラップファイルを開いて、ステップ3へ進みます。
2. Zabbixが、i ノード番号を既定のトラップファイルのiノード番号と比較して、その時点でファイルがローテートされていないかどうかチェックします。開かれたファイルが1つもない場合Zabbixは、最新のロケーションをリセットしてステップ1に戻ります。
3. Zabbixが、その時点で開かれているファイルからデータを読み込み、新しいロケーションを設定します。
4. 新しいデータがパースされます。これがローテートファイルだった場合、ファイルは閉じられ、ステップ2に戻ります。
5. 新しいデータが1つもない場合Zabbixは1秒間スリープ状態になり、ステップ2に戻ります。

ファイルシステム

トラップファイルの実装のために、ファイルシステムがファイルを識別するためにiノードをサポートしている必要があります。（情報はstat() のコールで統合されます）

3.4 セットアップの例

この例ではZabbix サーバへのトラップの受け渡しにsnmptrapd と SNMPTTを使用します。次のようにセットアップします:

1. **zabbix_server.conf** - ZabbixがSNMP トラッパを開始するように設定し、トラップファイルを以下のように設定:
   StartSNMPTrapper=1
   SNMPTrapperFile=/tmp/my_zabbix_traps.tmp
2. snmptrapd.conf - トラップハンドラとして SNMPTT を追加:
   traphandle default snmptt
3. **snmptt.ini** – 出力ファイルと日時の書式を設定:
   log_file = /tmp/my_zabbix_traps.tmp
   date_time_format = %H:%M:%S %Y/%m/%d
4. **snmptt.conf** – デフォルトのトラップフォーマットを定義:
   EVENT general .* “General event” Normal
   FORMAT ZBXTRAP $aA $ar
5. SNMP アイテムTESTを作成します:
   ホストの SNMP インターフェースの IP: 127.0.0.1
   キー: snmptrap[“General”]

ログの時間の形式: hh:mm:ss yyyy/MM/dd

この結果は、以下のようになります:

1. トラップの送信にコマンドが使用されます。
   snmptrap -v 1 -c public 127.0.0.1 '.1.3.6.1.6.3.1.1.5.3' '0.0.0.0' 6 33 '55' .1.3.6.1.6.3.1.1.5.3 s “teststring000”
2. 受信されるトラップ:
   15:48:18 2011/07/26 .1.3.6.1.6.3.1.1.5.3.0.33 Normal “General event” localhost - ZBXTRAP 127.0.0.1 127.0.0.1
3. アイテム[TEST]の値:
   15:48:18 2011/07/26 .1.3.6.1.6.3.1.1.5.3.0.33 Normal “General event” localhost - 127.0.0.1

このシンプルな例では、**トラップハンドル**としてSNMPTTを使用しています。よりよいパフォーマンスのためには[snmptrapd から SNMPTTあるいは直接Zabbixにトラップを受け渡すのに、埋め込みのPerlを使用するようにしてください。

本ページは2013/04/30時点の原文を基にしておりますので、内容は必ずしも最新のものとは限りません。最新の情報は右上の「Translations of this page」から英語版を参照してください。