SNMP コマンドを使用した RMON アラームとイベントの設定方法

目次

概要
前提条件
要件
使用するコンポーネント
表記法
背景説明
ステップバイステップ手順
イベントの作成
アラームの作成
例
確認
トラブルシューティング
関連情報

概要

このドキュメントでは、SNMP コマンドを使用してリモート モニタリング（RMON）アラームとイベント設定を設定する例を紹介します。

前提条件

要件

このドキュメントに関する固有の要件はありません。

使用するコンポーネント

この資料の以降の手順を実行するには、使用しているデバイスで RMON-MIB がサポートされていることが必要です。「IOS MIB ツール（登録ユーザ専用）」で確認できます。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されたものです。 このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。 ネットワークが稼働中の場合は、コマンドが及ぼす潜在的な影響を十分に理解しておく必要があります。

表記法

ドキュメント表記の詳細は、『シスコ テクニカル ティップスの表記法』を参照してください。

背景説明

remote monitoring (RMON; リモート モニタリング) アラームとイベントは、デバイス上の特定の MIB オブジェクトを監視し、それらのいずれかが定義された範囲を超過しそうになった場合にシステム管理者へ警告を発することを目的としています。

アラームモニタは、MIB の特定のオブジェクトを監視し、ある状態 (下降しきい値または上昇しきい値) に達したときにイベントを発生させます。

イベントとは、アラームがこのイベントを発生させた際に生成されるトラップまたはログのことです。 上昇しきい値および下降しきい値の例は次のとおりです。

```
n=value monitored by the alarm. The falling threshold is "5" and the rising threshold is "15"
```

これら 2 つの値のいずれかに到達したときに、アラームがイベントを発生させます。

値	トラップ	コメント
n1 = 16	rising	上昇値に到達： 15
n2 = 4	falling	下降値に到達： 5
n3 = 6	none	5 ~ 15
n4 = 6	rising	上昇値に到達： 15
n5 = 13	none	良好条件
n6 = 20	none	15 を超えたが、最後のイベント以降 5 未満になっていない
n7 = 4	falling	下降値に到達： 5
n8 = 20	none	5 未満だが、最後のイベント以降 15 を超えていない
n9 = 16	rising	上昇値に到達： 15

RMON アラームとイベントは、ルータ上でコマンド ライン インターフェイス (CLI) を使用して設定できます (「コマンドライン インターフェイスからの RMON アラームおよびイベントの設定」を参照) 。また、ルータやスイッチ上では Simple Network Management Protocol (SNMP) コマンドを使用して設定できます。 修正するパラメータは、RMON-MIB の一部です。

ステップバイステップ手順

イベントの作成

イベントを作成するには、次のコマンドを使用します。

```
# snmpset -c <read_write_community> <device_name> .1.3.6.1.2.1.16.9.1.1.x.y <variable type>
<value>
```

最初に、イベント ID (変数 *y*) を選択します。

イベントを作成するには、次の手順に従ってください。 それぞれのステップでは、ステップの説明、修正する MIB オブジェクトの名前、オブジェクト ID (OID)、および汎用コマンドの <variable *type*>、および <value> について記述しています。

1. ID に「y」を使用した最後の古いイベントを削除します (初めにそのイベントが今後は不要であることを確認してください。 今後も必要な場合は、他の ID を使用します)。`* eventStatus / .1.3.6.1.2.1.16.9.1.1.7.y`
   `* variable type=integer`
   `* value=4`
   注: 必要に応じて、同じコマンドを使用してイベントを消去します。
2. イベントの作成モードを入力します。`eventStatus / .1.3.6.1.2.1.16.9.1.1.7.y`
   `* variable type=integer`
   `* value v=2`
3. イベントに関する説明を指定します。`* eventDescription / .1.3.6.1.2.1.16.9.1.1.2.y`
   `* variable type=string (for Net-snmp) or octetsting (for Openview)`
   `* value = a description of the event`
4. 必要とするイベントのタイプを指定します。`* eventType / .1.3.6.1.2.1.16.9.1.1.3.y`
   `* variable type=integer`
   `* value =`
   `"1" => none`
   `"2" => log`
   `"3" => snmp-trap`
   `"4" => log-and-trap`
5. トラップ用のコミュニティ ストリングを指定します。 `* eventCommunity / .1.3.6.1.2.1.16.9.1.1.4.y`
   `* variable type=string (for Net-snmp) or octetsting (for Openview)`
   `* value="<trap_community_string>"`
6. イベントの所有者を指定します。`* eventOwner / .1.3.6.1.2.1.16.9.1.1.6.y`
   `* variable type=string (for Net-snmp) or octetsting (for Openview)`
   `* value="<event_owner>"`
7. イベントをアクティブにします。`* eventStatus / .1.3.6.1.2.1.16.9.1.1.7.y`
   `* variable type=integer`
   `* value=1`

アラームの作成

アラームを作成するには、次のコマンドを使用します。

```
# snmpset -c .1.3.6.1.2.1.16.3.1.1.x.y <read_write_community> <device_name> <variable type>
<value>
```

1. 「ID=y」を使用した最後の古いアラームを削除します (はじめにそのアラームが今後は不要であることを確認してください。 今後も必要な場合は、他の ID を使用します)。`* alarmStatus / .1.3.6.1.2.1.16.3.1.1.12.y`
   `* variable type=integer`
   `* value=4`
2. アラームの作成モードを入力します。`* alarmStatus / .1.3.6.1.2.1.16.3.1.1.12.y`

```
* variable type=integer
* value=2
```

3. データをサンプリングし、上昇しきい値および下降しきい値と比較するインターバルを秒で設定します。`* alarmInterval / .1.3.6.1.2.1.16.3.1.1.2.y`
```
* variable type=integer
* value=<n_seconds>
```
4. 監視対象とする OID を指定します。`* alarmVariable / .1.3.6.1.2.1.16.3.1.1.3.y`
```
* variable type=objid (for Net-snmp) or objectidentifier (for Openview)
* value=<oid_to_check>
```
5. 必要とするサンプルのタイプを指定します。`* alarmSampleType / .1.3.6.1.2.1.16.3.1.1.4.y`
```
* variable type=integer
* value=<rising_threshold> "1" => absoluteValue "2" => deltaValue
```
6. 発生させるアラームを指定します。`* alarmStartupAlarm / .1.3.6.1.2.1.16.3.1.1.6.y`
```
* variable type=integer
* value=
"1" => risingAlarm
"2" => fallingAlarm
"3" => risingOrFallingAlarm
```
7. 上昇しきい値を定義します。`* alarmRisingThreshold / .1.3.6.1.2.1.16.3.1.1.7.y`
```
* variable type=integer
* value=<rising_threshold>
```
8. 下降しきい値を定義します。`* alarmFallingThreshold / .1.3.6.1.2.1.16.3.1.1.8.y`
```
* variable type=integer
* value=<falling_threshold>
```
9. 上昇しきい値を超えた場合に発生させるイベント ID を指定します。`* alarmRisingEventIndex / .1.3.6.1.2.1.16.3.1.1.9.y`
```
* variable type=integer
* value=<event_ID>
```
10. 下降しきい値を超えた場合のイベント ID を指定します。`* alarmFallingEventIndex / .1.3.6.1.2.1.16.3.1.1.9.y`
```
* variable type=integer
* value=<event_ID>
```
11. アラームの所有者を指定します。`* alarmOwner / .1.3.6.1.2.1.16.3.1.1.11.y`
```
* variable type=string (for Net-snmp) or octetsting (for Openview)
* value=<owner>
```
12. アラームをアクティブにします。`* alarmStatus / .1.3.6.1.2.1.16.3.1.1.12.y`
```
* variable type=integer
* value=1
```

例

この例では、直前の 2 分間にインターフェイス 12 へ送られたバイト数が 140000000 を上回ったか、10 を下回ったときにトラップを送るために、「**safari**」が使用されています。

safari は Cisco IOS 2500 Software (C2500-JS-L), Version 12.1(9), RELEASE SOFTWARE (fc1) です。

この例は、WS-C6506 Software、バージョン NmpSW： 6.1(1b) でも正しく実行されます。

注: Catalyst には設定をチェックするための CLI コマンドがありませんが、サーバの **snmpwalk コマンド**を使用すればチェックを行えます。

ルータおよびスイッチでは、この設定によってリロードを切り抜けることができます。

```
safari# show rmon events Event table is empty # snmpset -c private safari
.1.3.6.1.2.1.16.9.1.1.7.123 integer 4 16.9.1.1.7.123 = 4 # snmpset -c private safari
```

```
.1.3.6.1.2.1.16.9.1.1.7.123 integer 2 16.9.1.1.7.123 = 2 safari#show rmon events Event 123 is
under creation, owned by Description is Event firing causes nothing, last fired 00:00:00 #
snmpset -c private safari .1.3.6.1.2.1.16.9.1.1.2.123 string "test_event" 16.9.1.1.2.123 =
"test_event" # snmpset -c private safari .1.3.6.1.2.1.16.9.1.1.3.123 integer 4 16.9.1.1.3.123 =
4 # snmpset -c private safari .1.3.6.1.2.1.16.9.1.1.4.123 string "public" 16.9.1.1.4.123 =
"public" # snmpset -c private safari .1.3.6.1.2.1.16.9.1.1.6.123 string "event_owner"
16.9.1.1.6.123 = "event_owner" # snmpset -c private safari .1.3.6.1.2.1.16.9.1.1.7.123 integer 1
16.9.1.1.7.123 = 1 safari# show rmon events Event 123 is active, owned by event_owner
Description is test_event Event firing causes log and trap to community public, last fired
00:00:00 safari# show rmon alarm Alarm table is empty # snmpset -c private safari
.1.3.6.1.2.1.16.3.1.1.12.321 integer 2 16.3.1.1.12.321 = 2 safari# show rmon alarm Alarm 321 is
under creation, owned by Monitors ccitt.0 every 10 second(s) Taking absolute samples, last value
was 0 Rising threshold is 0, assigned to event 0 Falling threshold is 0, assigned to event 0 On
startup enable rising or falling alarm # snmpset -c private safari .1.3.6.1.2.1.16.3.1.1.2.321
integer 120 16.3.1.1.2.321 = 120 # snmpset -c private safari .1.3.6.1.2.1.16.3.1.1.3.321 objid
.1.3.6.1.2.1.2.2.1.10.12 16.3.1.1.3.321 = OID: interfaces.ifTable.ifEntry.ifInOctets.12 #
snmpset -c private safari .1.3.6.1.2.1.16.3.1.1.4.321 integer 2 16.3.1.1.4.321 = 2 # snmpset -c
private safari .1.3.6.1.2.1.16.3.1.1.6.321 integer 3 16.3.1.1.6.321 = 3 # snmpset -c private
safari .1.3.6.1.2.1.16.3.1.1.7.321 integer 140000000 16.3.1.1.7.321 = 140000000 # snmpset -c
private safari .1.3.6.1.2.1.16.3.1.1.8.321 integer 10 16.3.1.1.8.321 = 10 # snmpset -c private
safari .1.3.6.1.2.1.16.3.1.1.9.321 integer 123 16.3.1.1.9.321 = 123 # snmpset -c private safari
.1.3.6.1.2.1.16.3.1.1.10.321 integer 123 16.3.1.1.10.321 = 123 # snmpset -c private safari
.1.3.6.1.2.1.16.3.1.1.11.321 string "alarm_owner" 16.3.1.1.11.321 = "alarm_owner" # snmpset -c
private safari .1.3.6.1.2.1.16.3.1.1.12.321 integer 1 16.3.1.1.12.321 = 1 safari# show rmon
alarm Alarm 321 is active, owned by alarm_owner Monitors ifEntry.10.1 every 120 second(s) Taking
delta samples, last value was 130244 Rising threshold is 140000000, assigned to event 123
Falling threshold is 10, assigned to event 123 On startup enable rising or falling alarm
```

確認

現在、この設定に使用できる確認手順はありません。

トラブルシューティング

現在のところ、この設定に関する特定のトラブルシューティング情報はありません。

関連情報

- コマンド ライン インターフェイスからの RMON アラームおよびイベント設定の設定
- イベント MIB のサポート
- RFC 1757
- テクニカルサポート - Cisco Systems