

Comment configurer les paramètres d'alarme et d'événement RMON à l'aide de commandes SNMP

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Conventions](#)

[Informations générales](#)

[Procédure pas à pas](#)

[Création D'Un Événement](#)

[Création D'Une Alarme](#)

[Exemple](#)

[Vérification](#)

[Dépannage](#)

[Informations connexes](#)

Introduction

Ce document fournit un exemple de configuration pour les paramètres d'alarme et d'événement de surveillance à distance (RMON) à l'aide des commandes SNMP.

Conditions préalables

Conditions requises

Aucune spécification déterminée n'est requise pour ce document.

Components Used

Pour suivre les procédures décrites dans ce document, votre périphérique doit prendre en charge la base RMON-MIB. Vous pouvez le vérifier dans [Cisco IOS MIB Tools](#) (clients [enregistrés](#) uniquement).

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

Pour plus d'informations sur les conventions des documents, référez-vous aux [Conventions utilisées pour les conseils techniques de Cisco](#).

Informations générales

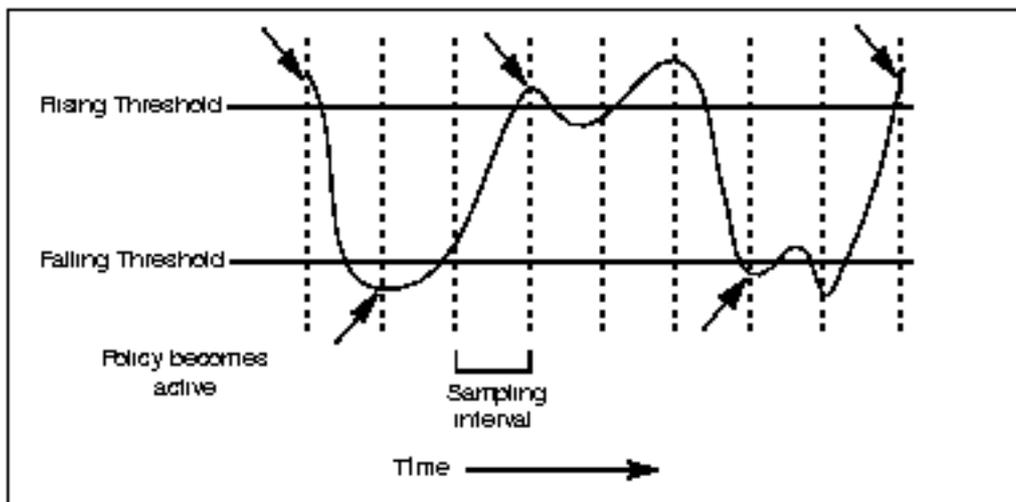
L'objectif de l'alarme et de l'événement de surveillance à distance (RMON) est de surveiller un objet MIB donné sur le périphérique et d'avertir l'administrateur système si l'une de ces valeurs sort de la plage définie.

L'alarme surveille un objet spécifique de la MIB et déclenche un événement lorsque la condition (seuil décroissant ou croissant) est atteinte.

L'événement est le déroutement ou le journal généré lorsque l'alarme le déclenche. Voici un exemple de seuil croissant ou décroissant :

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

L'alarme déclenche un événement lorsque l'une des deux valeurs est atteinte.



Valeur	Alerte	commentaire
n1 = 16	croissant	a atteint la valeur en hausse : 15
n2 = 4	tomber	a atteint la valeur en baisse : 5
n3 = 6	none	entre 5 et 15
n4 = 6	croissant	a atteint la valeur en hausse : 15
n5 = 13	none	bonne condition
n6 = 20	none	au-dessus de 15 mais n'est pas passé en dessous de 5 depuis le dernier

		événement
n7 = 4	tomber	a atteint la valeur en baisse : 5
n8 = 20	none	moins de 5 ans mais n'a pas dépassé 15 ans depuis le dernier événement
n9 = 16	croissan t	a atteint la valeur en hausse : 15

Vous pouvez configurer l'alarme et l'événement RMON sur les routeurs à l'aide de l'interface de ligne de commande (CLI) (reportez-vous à [Configuration des paramètres d'alarme et d'événement RMON à partir de l'interface de ligne de commande](#)), ainsi que sur les routeurs et les commutateurs à l'aide des commandes SNMP (Simple Network Management Protocol). Les paramètres à modifier font partie de la [RMON-MIB](#).

Procédure pas à pas

Création D'Un Événement

Utilisez cette commande pour créer un événement :

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

Choisissez d'abord l'ID d'événement (variable y).

Suivez cette procédure pour créer un événement. Pour chaque étape, il y a une description de l'étape, le nom de l'objet MIB à modifier, l'ID d'objet (OID), le *<type de variable>* et la *<valeur>* de la commande générique.

1. Effacez un ancien événement éventuel qui aurait utilisé ID=« y » (vérifiez d'abord que vous n'en avez plus besoin. Sinon, utilisez un autre ID).

```
* eventStatus / .1.3.6.1.2.1.16.9.1.1.7.y
* variable type=integer
* value=4
```

Remarque : utilisez la même commande pour effacer l'événement si nécessaire.

2. Passez en mode de création d'événement :

```
eventStatus / .1.3.6.1.2.1.16.9.1.1.7.y
* variable type=integer
* value v=2
```

3. Spécifier la description de l'événement :

```
* 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. Spécifiez le type d'événement souhaité :

```
* 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. Spécifiez la chaîne de communauté pour le déROUTement :

```
* eventCommunity / .1.3.6.1.2.1.16.9.1.1.4.y
* variable type=string (for Net-snmp) or octetstring (for Openview)
* value="<trap_community_string>"
```

6. Spécifiez le propriétaire de l'événement :

```
* eventOwner / .1.3.6.1.2.1.16.9.1.1.6.y
* variable type=string (for Net-snmp) or octetstring (for Openview)
* value="<event_owner>"
```

7. Activer l'événement :

```
* eventStatus / .1.3.6.1.2.1.16.9.1.1.7.y
* variable type=integer
* value=1
```

Création D'Une Alarme

Utilisez cette commande pour créer une alarme :

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

1. Effacer une ancienne alarme éventuelle qui aurait utilisé ID=y (première vérification si vous n'en avez plus besoin). Sinon, utilisez un autre ID) :

```
* alarmStatus / .1.3.6.1.2.1.16.3.1.1.12.y
* variable type=integer
* value=4
```

2. Passez en mode de création d'alarme :

```
* alarmStatus / .1.3.6.1.2.1.16.3.1.1.12.y
* variable type=integer
* value=2
```

3. Définissez l'intervalle (en secondes) pendant lequel les données sont échantillonnées et comparées aux seuils ascendants et descendants :

```
* alarmInterval / .1.3.6.1.2.1.16.3.1.1.2.y
* variable type=integer
* value=<n_seconds>
```

4. Spécifiez l'OID à surveiller :

```
* 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. Définissez le type d'échantillon souhaité :

```
* alarmSampleType / .1.3.6.1.2.1.16.3.1.1.4.y
* variable type=integer
* value=<rising_threshold> "1" => absoluteValue "2" => deltaValue
```

6. Spécifier le déclencheur d'une alarme :

```
* alarmStartupAlarm / .1.3.6.1.2.1.16.3.1.1.6.y
* variable type=integer
* value=
"1" => risingAlarm
```

```
"2" => fallingAlarm
"3" => risingOrFallingAlarm
```

7. Définissez le seuil ascendant :

```
* alarmRisingThreshold / .1.3.6.1.2.1.16.3.1.1.7.y
* variable type=integer
* value=<rising_threshold>
```

8. Définissez le seuil de chute :

```
* alarmFallingThreshold / .1.3.6.1.2.1.16.3.1.1.8.y
* variable type=integer
* value=<falling_threshold>
```

9. Spécifiez l'ID d'événement à déclencher lorsque le seuil ascendant est franchi :

```
* alarmRisingEventIndex / .1.3.6.1.2.1.16.3.1.1.9.y
* variable type=integer
* value=<event_ID>
```

10. Spécifiez l'ID d'événement lorsque le seuil de chute est franchi :

```
* alarmFallingEventIndex / .1.3.6.1.2.1.16.3.1.1.9.y
* variable type=integer
* value=<event_ID>
```

11. Spécifiez le propriétaire de l'alarme :

```
* 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. Activer l'alarme :

```
* alarmStatus / .1.3.6.1.2.1.16.3.1.1.12.y
* variable type=integer
* value=1
```

Exemple

Dans cet exemple, **safari** est utilisé pour envoyer un déroutement lorsque le nombre d'octets entrant dans l'interface 12 au cours des deux dernières minutes est supérieur à 14000000 ou inférieur à 10.

Safari est le logiciel Cisco IOS 2500 (C2500-JS-L), version 12.1(9), VERSION LOGICIELLE (fc1).

Cet exemple a également été essayé avec succès sur le logiciel WS-C6506, Version NmpSW : 6.1(1b).

Remarque : dans Catalyst, il n'y a aucune commande CLI pour vérifier la configuration, mais cela peut être fait avec la commande **snmpwalk** sur le serveur.

Sur le routeur et le commutateur, cette configuration survit à un rechargement.

```
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

Vérification

Aucune procédure de vérification n'est disponible pour cette configuration.

Dépannage

Il n'existe actuellement aucune information de dépannage spécifique pour cette configuration.

Informations connexes

- [Configuration des paramètres d'alarme et d'événement RMON à partir de l'interface de ligne de commande](#)
- [Support MIB d'événements](#)
- [RFC 1757](#)
- [Support technique - Cisco Systems](#)