

# Instellen van RMON - alarmlampjes en gebeurtenissen instellen met SNMP-opdrachten

## Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Conventies](#)

[Achtergrondinformatie](#)

[Stap voor stap Procedure](#)

[Een gebeurtenis maken](#)

[Een alarm maken](#)

[Voorbeeld](#)

[Verifiëren](#)

[Problemen oplossen](#)

[Gerelateerde informatie](#)

## Inleiding

Dit document biedt een voorbeeldconfiguratie voor het RMON - alarm en de instellingen voor gebeurtenissen op afstand met behulp van SNMP-opdrachten.

## Voorwaarden

### Vereisten

Er zijn geen specifieke vereisten van toepassing op dit document.

### Gebruikte componenten

Om de procedures in dit document te volgen, moet uw apparaat de RMON-MIB ondersteunen. U kunt dit controleren op [Cisco IOS MIB Gereedschappen](#) (alleen [geregistreerde](#) klanten).

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u de potentiële impact van elke opdracht begrijpen.

### Conventies

Raadpleeg voor meer informatie over documentconventies de [technische Tips](#) van [Cisco](#).

## Achtergrondinformatie

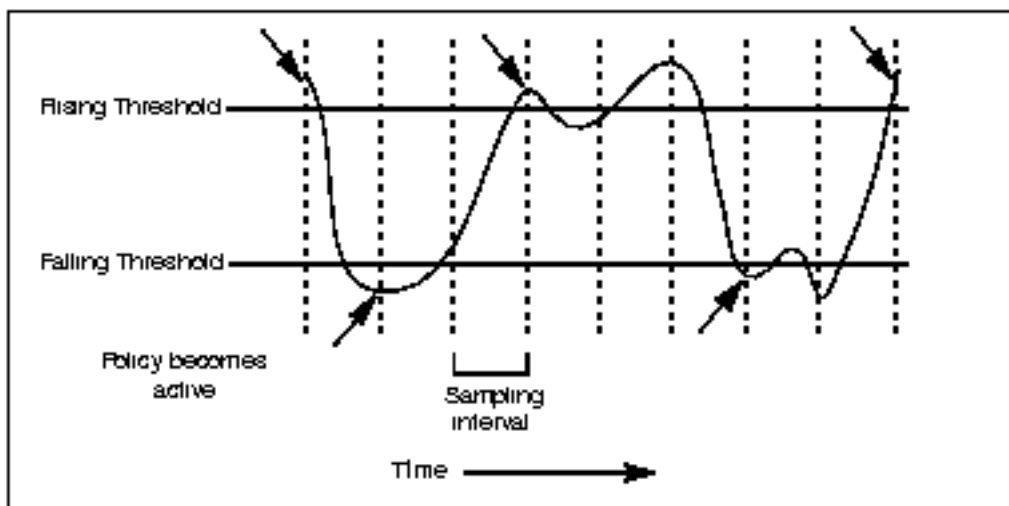
Het doel van afstandscontrole (RMON) alarm en gebeurtenis is om een bepaald MIB object op het apparaat te controleren, en de systeembeheerder te waarschuwen als een van die waarden het gedefinieerde bereik uitgaat.

Het alarm controleert een specifiek object in de MIB en veroorzaakt een gebeurtenis wanneer de toestand (dalende of stijgende drempel) wordt bereikt.

De gebeurtenis is de val of het logbestand dat wordt gegenereerd wanneer het alarm afgaat. Een voorbeeld van een stijgende en dalende drempel is:

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

Het alarm zet een gebeurtenis in werking wanneer een van de twee waarden wordt bereikt.



Waarde	Trap	Opmerking
n1=16	stijgend	heeft de stijgende waarde bereikt : 15
n2=4	dalen	heeft de dalende waarde bereikt : 5
n3=6	none	tussen 5 en 15
n4=6	stijgend	heeft de stijgende waarde bereikt : 15
n5=13	none	goede conditie
n6=20	none	meer dan 15 , maar niet minder dan 5 sinds de laatste gebeurtenis
n7=4	dalen	heeft de dalende waarde bereikt : 5
n8=20	none	onder 5 , maar niet boven de 15 sinds de laatste gebeurtenis .
n9=16	stijgend	heeft de stijgende waarde bereikt : 15

U kunt RMON - alarm en -gebeurtenis op routers configureren met behulp van de opdrachtregel

interface (CLI) (raadpleeg de [instellingen RMON-alarmen en gebeurtenissen configureren vanuit de opdrachtregel](#)) en op routers en switches met de opdrachten Simple Network Management Protocol (SNMP). De aan te passen parameters maken deel uit van de [RMON-MIB](#).

## Stap voor stap Procedure

### Een gebeurtenis maken

Gebruik deze opdracht om een gebeurtenis te maken:

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

Kies eerst de gebeurtenis ID (variabele y).

Volg deze procedure om een gebeurtenis te maken. Voor elke stap is er een beschrijving van de stap, de naam van het MIB object dat moet worden gewijzigd, het object-ID (OID), het <variabele type> en de <waarde> van de generieke opdracht.

1. Wis een mogelijk oud evenement dat ID="y" zou hebben gebruikt (controleer eerst of u het niet meer nodig hebt.) Gebruik anders een andere ID).

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

**Opmerking:** gebruik dezelfde opdracht om de gebeurtenis indien nodig te wissen.

2. Voer de generatiemodus van de gebeurtenis in:

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

3. Specificeer de beschrijving van de gebeurtenis:

```
* 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. Specificeer het gewenste type gebeurtenis:

```
* 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. Specificeer de string van de gemeenschap voor de val:

```
* 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. Specificeer de eigenaar van de gebeurtenis:

```
* 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. Activeert de gebeurtenis:

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

## Een alarm maken

Gebruik deze opdracht om een alarm te maken:

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

### 1. Wis een eventueel oud alarm dat ID=y zou hebben gebruikt (eerste controle als u het niet meer nodig hebt. Gebruik anders een andere ID):

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

### 2. Geef de modus voor het maken van alarm op:

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

### 3. Stel het interval (in seconden) vast waarover de gegevens worden bemonsterd en vergelijk het met de stijgende en dalende drempels:

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

### 4. Specificeer de OID die u wilt controleren:

```
* 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. Bepaal het type monster dat u wilt:

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

### 6. Specificeer wat een alarm in werking stelt:

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

### 7. Bepaal de stijgende drempel:

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

### 8. Bepaal de dalingsdrempel:

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

9. Specificeer de gebeurtenis-ID die u wilt activeren wanneer de stijgende drempel is overschreden:

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

10. Geef de gebeurtenis-ID op wanneer de dalende drempel wordt overschreden:

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

11. Specificeer de eigenaar van het alarm:

```
* 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. Activeert het alarm:

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

## Voorbeeld

In dit voorbeeld wordt **safari** gebruikt om een val te verzenden wanneer het aantal bytes dat tijdens de laatste twee minuten naar interface 12 gaat meer dan 140000000 of minder dan 10 bedraagt.

Safari is Cisco IOS 2500-software (C2500-JS-L), versie 12.1(9), RELEASE-SOFTWARE (FC1).

Dit voorbeeld werd ook met succes beproefd op WS-C6506 Software, versie NmpSW: 6.1(1 ter).

**Opmerking:** In Catalyst is er geen CLI-opdracht om de configuratie te controleren, maar het kan wel met de tussenloop opdracht op de server worden uitgevoerd.

Op de router en de switch, overleeft deze configuratie een herlading.

```
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
```

## [Verifiëren](#)

Er is momenteel geen verificatieprocedure beschikbaar voor deze configuratie.

## [Problemen oplossen](#)

Er is momenteel geen specifieke troubleshooting-informatie beschikbaar voor deze configuratie.

## [Gerelateerde informatie](#)

- [Instellingen voor RMON - alarmen en gebeurtenissen configureren vanuit de interface van de opdrachtregel](#)
- [Ondersteuning van Event MIB](#)
- [RFC 1757](#)
- [Technische ondersteuning - Cisco-systemen](#)