

# Como configurar o alarme RMON e definições de evento utilizando comandos SNMP

## Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Conventions](#)

[Informações de Apoio](#)

[Procedimento Passo a Passo](#)

[Criando um evento](#)

[Criando um alarme](#)

[Exemplo](#)

[Verificar](#)

[Troubleshoot](#)

[Informações Relacionadas](#)

## [Introduction](#)

Este documento fornece uma configuração de exemplo para configurações de alarme e evento de monitoração remota (RMON - Remote Monitoring) usando comandos SNMP.

## [Prerequisites](#)

## [Requirements](#)

Não existem requisitos específicos para este documento.

## [Componentes Utilizados](#)

Para seguir os procedimentos descritos neste documento, seu dispositivo precisa suportar o RMON-MIB. Você pode verificar isso em [Cisco IOS MIB Tools](#) (somente clientes [registrados](#)).

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](#)

Para obter mais informações sobre convenções de documento, consulte as [Convenções de dicas técnicas Cisco](#).

## Informações de Apoio

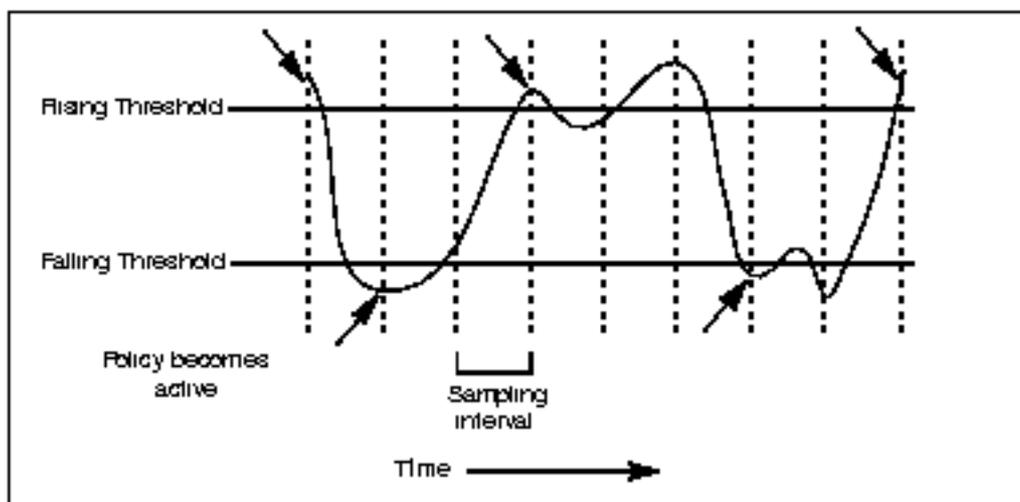
A finalidade do alarme e do evento do monitoramento remoto (RMON) é monitorar um determinado objeto do MIB no dispositivo e avisar ao administrador do sistema se um dos valores estiver fora da faixa definida.

O alarme monitora um objeto específico na MIB e dispara um evento quando a condição (limite de queda ou aumento) é atingida.

O evento é a armadilha ou registro gerado quando o alarme a dispara. Um exemplo de um limiar que se eleva e cai é:

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

O alarme dispara um evento quando um dos dois valores for alcançado.



Valor	Armadilha	Comentário
n1=16	ascensão	alcançou o valor crescente: 15
n2=4	queda	atingiu o valor decrescente : 5
n3=6	nenhum	entre 5 e 15
n4=6	ascensão	alcançou o valor crescente: 15
n5=13	nenhum	boas condições
n6=20	nenhum	acima de 15, mas não menos de 5 desde o último evento
n7=4	queda	atingiu o valor decrescente : 5
n8=20	nenhum	menos de 5 anos, mas não ultrapassou 15 anos desde o último evento
n9=16	ascensão	alcançou o valor crescente: 15

Você pode configurar o alarme e o evento RMON em roteadores usando a interface de linha de comando (CLI) (consulte [Configuração de Alarme e Configurações de Evento RMON a partir da](#)

[Interface de Linha de Comando](#)) e em roteadores e switches usando os comandos SNMP (Simple Network Management Protocol). Os parâmetros a serem modificados fazem parte do [RMON-MIB](#).

## Procedimento Passo a Passo

### Criando um evento

Use este comando para criar um evento:

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

Primeiro escolha a ID do evento (variável y).

Siga este procedimento para criar um evento. Para cada etapa, há uma descrição da etapa, o nome do objeto MIB a ser modificado, o ID do objeto (OID), o <tipo de variável> e o <value> do comando genérico.

1. Apague um possível evento antigo que tenha utilizado ID="y" (verificar primeiro se você não precisa mais dele). Caso contrário, utilize outro ID).

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

**Observação:** use o mesmo comando para apagar o evento quando necessário.

2. Digite o modo de criação do evento:

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

3. Especifique a descrição do evento:

```
* 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. Especifique o tipo de evento que deseja:

```
* 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. Especifique a string de comunidade para o desvio:

```
* 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. Especifique o proprietário do evento:

```
* 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. Ativar o evento:

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

## Criando um alarme

Utilize este comando para criar um alarme:

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

### 1. Apague um eventual alarme antigo que teria usado ID=y (primeiro verifique se não precisa mais dele. Caso contrário, utilize outro ID):

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

### 2. Entre no modo de criação de alarme:

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

### 3. Defina o intervalo (em segundos) após o qual os dados serão medidos e comparados com os limiares de elevação e de queda.

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

### 4. Especifique o OID que deseja monitorar.

```
* 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. Defina o tipo de amostra desejado:

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

### 6. Especifique o que dispara um 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. Defina a elevação de limiar:

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

### 8. Defina o limite de redução:

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

### 9. Especifique a ID de evento que você deseja que dispare quando a elevação de limiar for

cruzada:

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

10. Especifique o ID do evento quando o threshold de queda for ultrapassado:

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

11. Especifique o proprietário do 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. Ative o alarme:

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

## Exemplo

Neste exemplo, o **safari** é usado para enviar uma interceptação quando o número de bytes que entram na interface 12 durante os últimos dois minutos é superior a 14000000 ou inferior a 10.

Safari é o software Cisco IOS 2500 (C2500-JS-L), versão 12.1(9), RELEASE SOFTWARE (fc1).

Este exemplo também foi tentado com êxito no software WS-C6506, Versão NmpSW: 6.1(1b).

**Observação:** no Catalyst, não há nenhum comando CLI para verificar a configuração, mas isso pode ser feito com o comando **snmpwalk** no servidor.

On the router and the Switch, this configuration survives a reload.

```
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
```

## [Verificar](#)

No momento, não há procedimento de verificação disponível para esta configuração.

## [Troubleshoot](#)

Atualmente, não existem informações disponíveis específicas sobre Troubleshooting para esta configuração.

## [Informações Relacionadas](#)

- [Configurando as configurações de alarme e evento RMON a partir da interface de linha de comando](#)
- [Suporte a MIB de evento](#)
- [RFC 1757](#)
- [Suporte Técnico - Cisco Systems](#)