

Cómo configurar la Alarma RMON y los Parámetros de evento utilizando los Comandos SNMP.

Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Convenciones](#)

[Antecedentes](#)

[Procedimiento Paso a Paso](#)

[Creación de un acontecimiento](#)

[Creación de una alarma](#)

[Ejemplo:](#)

[Verificación](#)

[Troubleshoot](#)

[Información Relacionada](#)

[Introducción](#)

Este documento aporta una configuración de muestra para la alarma de Supervisión remota (RMON) y las configuraciones de eventos mediante los comandos SNMP.

[Prerequisites](#)

[Requirements](#)

No hay requisitos específicos para este documento.

[Componentes Utilizados](#)

Para seguir los procedimientos en este documento, es necesario que su dispositivo soporte RMON-MIB. Puede verificar esto en [Cisco IOS MIB Tools](#) ([sólo](#) 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.

[Convenciones](#)

Puede configurar la alarma y el evento RMON en los routers mediante la interfaz de línea de comandos (CLI) (consulte [Configuración de Alarma RMON y Configuración de Eventos desde la interfaz de línea de comandos](#)) y en los routers y switches mediante comandos SNMP (Simple Network Management Protocol). Los parámetros a modificar son parte de [RMON-MIB](#).

Procedimiento Paso a Paso

Creación de un acontecimiento

Utilice este comando para crear un evento:

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

Primero elija el ID de evento (variable *y*).

Siga este procedimiento para crear un evento. Para cada paso, hay una descripción del paso, el nombre del objeto MIB que se va a modificar, el ID de objeto (OID), el *<tipo de variable>* y el *<valor>* del comando genérico.

1. Elimine un eventual evento antiguo que habría utilizado ID="y" (primero asegúrese de revisar que ya no lo necesita). De lo contrario, utilice otra ID).

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

Nota: Utilice el mismo comando para borrar el evento cuando sea necesario.

2. Ingrese el modo de creación del evento:

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

3. Especifique la descripción del 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 el tipo de evento que desea:

```
* 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 la cadena comunitaria para la trampa:

```
* 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 el dueño del 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. Active el evento:

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

Creación de una alarma

Utilice este comando para crear una alarma:

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

1. Borre una alarma antigua que hubiera usado ID=y (primero verifique si ya no la necesita. De lo contrario, utilice otra ID):

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

2. Ingresar al modo de creación de alarma:

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

3. Establezca el intervalo (en segundos) según el cual se prueban los datos y se comparan con los umbrales ascendente y descendente:

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

4. Especifique el OID que desea el teléfono que desea supervisar.

```
* 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 el tipo de ejemplo que desea:

```
* 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 la causa por la que una alarma se dispara:

```
* 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 el umbral ascendente:

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

8. Defina el umbral descendente:

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

9. Especifique la identificación del evento que quiere activar cuando el umbral ascendente sea cruzado.

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

10. Especifique el ID del evento cuando se pase el umbral descendente:

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

11. Especifique el dueño de la alarma:

```
* 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. Active la alarma:

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

Ejemplo:

En este ejemplo, **safari** se utiliza para enviar una trampa cuando el número de bytes que ingresan a la interfaz 12 durante los últimos dos minutos es superior a 140000000 o inferior a 10.

Safari es el software Cisco IOS 2500 (C2500-JS-L), la versión 12.1(9) y el SOFTWARE RELEASE (fc1).

Este ejemplo también se probó con éxito en el software WS-C6506, versión NmpSW: 6.1.1 ter.

Nota: En el Catalyst, no hay ningún comando CLI para verificar la configuración, pero se puede hacer con el **comando snmpwalk** en el servidor.

En el router y el switch, esta configuración no se modifica al hacer una recarga.

```
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

```

Verificación

Actualmente, no hay un procedimiento de verificación disponible para esta configuración.

Troubleshoot

Actualmente, no hay información específica de troubleshooting disponible para esta configuración.

Información Relacionada

- [Configuración de Alarma RMON y Configuración de Eventos desde la Interfaz de Línea de Comandos](#)
- [Soporte de evento MIB](#)
- [RFC 1757](#)
- [Soporte Técnico - Cisco Systems](#)