

Come configurare le impostazioni degli allarmi e degli eventi RMON con i comandi SNMP

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Convenzioni](#)

[Premesse](#)

[Procedura dettagliata](#)

[Creazione Di Un Evento](#)

[Creazione Di Un Allarme](#)

[Esempio](#)

[Verifica](#)

[Risoluzione dei problemi](#)

[Informazioni correlate](#)

[Introduzione](#)

In questo documento viene fornita una configurazione di esempio per le impostazioni degli allarmi e degli eventi di monitoraggio remoto (RMON) con comandi SNMP.

[Prerequisiti](#)

[Requisiti](#)

Nessun requisito specifico previsto per questo documento.

[Componenti usati](#)

Per seguire le procedure descritte in questo documento, il dispositivo deve supportare RMON-MIB. Per verificare questa condizione, consultare [Cisco IOS MIB Tools](#) (solo utenti [registrati](#)).

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

[Convenzioni](#)

Per ulteriori informazioni sulle convenzioni usate, consultare il documento [Cisco sulle convenzioni nei suggerimenti tecnici](#).

Premesse

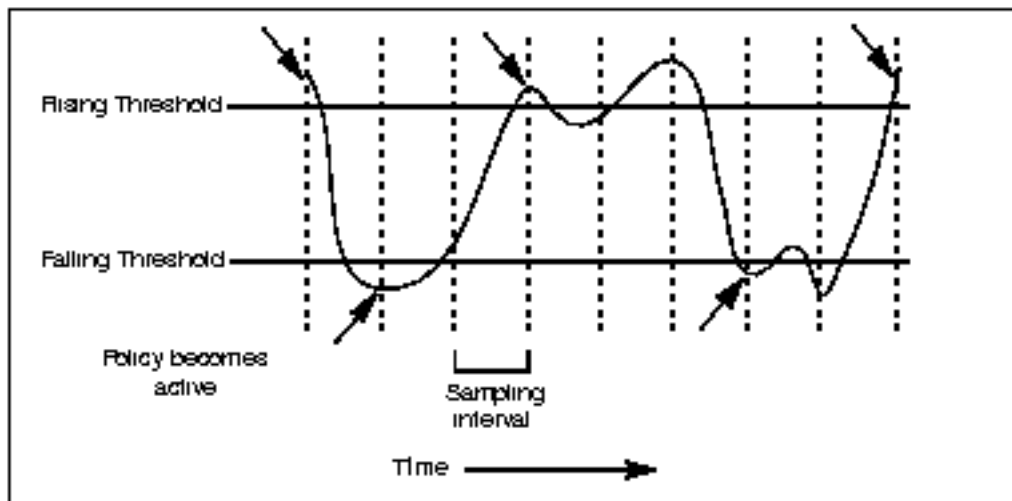
Lo scopo dell'allarme e dell'evento RMON (monitoraggio da remoto) è monitorare un determinato oggetto MIB sul dispositivo e avvisare l'amministratore di sistema se uno di questi valori esce dall'intervallo definito.

L'allarme monitora un oggetto specifico nel MIB e attiva un evento quando viene raggiunta la condizione (soglia di diminuzione o di aumento).

L'evento è la trap o il registro generato quando l'allarme lo attiva. Un esempio di soglia crescente e decrescente è:

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

L'allarme attiva un evento quando viene raggiunto uno dei due valori.



Valore	Trap	Commento
n1=16	aumento	ha raggiunto il valore crescente: 15
n2=4	caduta	ha raggiunto il valore decrescente: 5
n3=6	nessuna	tra 5 e 15
n4=6	aumento	ha raggiunto il valore crescente: 15
n5=13	nessuna	buono stato
n6=20	nessuna	più di 15 anni ma non meno di 5 dall'ultimo evento
n7=4	caduta	ha raggiunto il valore decrescente: 5
n8=20	nessuna	meno di 5 anni, ma non è andato oltre 15 dall'ultimo evento
n9=16	aumento	ha raggiunto il valore crescente: 15

È possibile configurare gli eventi e gli allarmi RMON sui router utilizzando l'interfaccia della riga di

comando (consultare il documento sulla [configurazione delle impostazioni degli allarmi e degli eventi RMON dall'interfaccia della riga di comando](#)) e sui router e sugli switch utilizzando i comandi SNMP (Simple Network Management Protocol). I parametri da modificare fanno parte di [RMON-MIB](#).

Procedura dettagliata

Creazione Di Un Evento

Utilizzare questo comando per creare un evento:

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

Scegliere innanzitutto l'ID evento (variabile y).

Seguire questa procedura per creare un evento. Per ogni passo è disponibile una descrizione del passo, il nome dell'oggetto MIB da modificare, l'ID dell'oggetto (OID), il <tipo di variabile> e il <valore> del comando generico.

1. Cancellare un evento precedente che avrebbe utilizzato ID="y" (assicurarsi di non averne più bisogno. In caso contrario, utilizzare un altro ID).

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

Nota: utilizzare lo stesso comando per cancellare l'evento quando necessario.

2. Accedere alla modalità di creazione degli eventi:

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

3. Specificare la descrizione dell'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. Specificare il tipo di evento desiderato:

```
* 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. Specificare la stringa della community per la trap:

```
* 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. Specificare il proprietario dell'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. Attivare l'evento:

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

Creazione Di Un Allarme

Utilizzare questo comando per creare un avviso:

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

1. Cancellare un eventuale vecchio allarme che avrebbe utilizzato ID=y (controllare prima se non è più necessario. Altrimenti, utilizzare un altro ID):

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

2. Accedere alla modalità di creazione dell'allarme:

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

3. Impostare l'intervallo (in secondi) di campionamento dei dati e confrontarlo con le soglie di aumento e diminuzione:

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

4. Specificare l'OID che si desidera monitorare:

```
* 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. Definire il tipo di campione desiderato:

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

6. Specificare cosa attiva un allarme:

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

7. Definire la soglia di aumento:

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

8. Definire la soglia discendente:

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

9. Specificare l'ID evento da attivare quando viene superata la soglia:

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

10. Specificare l'ID evento quando viene superata la soglia di riduzione:

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

11. Specificare il proprietario dell'avviso:

```
* 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. Attivare l'avviso:

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

Esempio

Nell'esempio, il comando **safari** viene usato per inviare una trap quando il numero di byte che vanno nell'interfaccia 12 negli ultimi due minuti è superiore a 14000000 o inferiore a 10.

Safari è il software Cisco IOS 2500 (C2500-JS-L), versione 12.1(9), SOFTWARE RELEASE (fc1).

L'esempio è stato provato con successo anche con il software WS-C6506, versione NmpSW: 6.1, paragrafo 1 ter.

Nota: in Catalyst, non è disponibile alcun comando CLI per controllare la configurazione, ma è possibile farlo con il comando **snmpwalk** sul server.

Sul router e sullo switch, questa configurazione rimane valida fino al caricamento successivo.

```
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
```

[Verifica](#)

Attualmente non è disponibile una procedura di verifica per questa configurazione.

[Risoluzione dei problemi](#)

Al momento non sono disponibili informazioni specifiche per la risoluzione dei problemi di questa configurazione.

[Informazioni correlate](#)

- [Configurazione delle impostazioni degli allarmi e degli eventi RMON dall'interfaccia della riga di comando](#)
- [Supporto MIB evento](#)
- [RFC 1757](#)
- [Supporto tecnico – Cisco Systems](#)