

Configuring RMON Alarm and Event Settings from the Command Line Interface (CLI)

Document ID: 17428

Introduction

Prerequisites

Requirements

Components Used

Conventions

Background Information

Syntax To Set Up An Event

Syntax To Set Up An Alarm

Examples

Related Information

Introduction

This document describes how to set up Remote Monitoring (RMON) Alarms and Events on a router from the command line interface (CLI).

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

This document is not restricted to specific software and hardware versions.

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

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Background Information

RMON is a method similar to Simple Network Management Protocol (SNMP) to track statistics on network device interfaces or ports.

The RMON feature typically is useful in a LAN switch environment, but is available on access routers (for example, the 2x00 Series) in Cisco IOS® Software Release 11.1 or later. Sometimes, you need to set up RMON on remote routers only when you can not get access to the LAN equipment (such as hubs) to view the traffic. RMON does not require you to actively poll for SNMP variables on a regular basis. The devices store the needed information, and then it is dumped periodically to a RMON network management station.

Note: By default all switches support mini-rmon, so that alarms, events, stats and history are directly received from the switches. In order to receive all other detailed information from switches, you require Network Analysis Module (NAM).

Syntax To Set Up An Event

Cisco IOS software allows you to set up RMON alarms and events from the CLI. This section and the next one provide the syntax of the required commands, with the same names that are used for the **eventTable** and the **alarmTable**.

1.3.6.1.2.1.16.9.1

eventTable OBJECT-TYPE

SYNTAX SEQUENCE OF EventEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A list of events to be generated."

::= { event 1 }

.1.3.6.1.2.1.16.3.1

alarmTable OBJECT-TYPE

SYNTAX SEQUENCE OF AlarmEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A list of alarm entries."

::= { alarm 1 }

Syntax

rmon event *eventIndex* [**log**] [**trap** *eventCommunity*] [**description** *eventDescription*] [**owner** *eventOwner*]

Syntax Description

1. **event** Configure an RMON event.
2. *eventIndex* Event number (1 5535)
3. **log** (Optional) Generate an RMON log when the event fires.
4. **trap** *eventCommunity* (Optional) Generate an SNMP trap when the event fires, for the specified SNMP community string.
5. **description** *eventDescription* (Optional) Specify a WORD or a description of the event.
6. **owner** *eventOwner* (Optional) Specify an owner for the event.

- If you do not specify either the **log** or the **trap** option, the **alarmTable** object **eventType** (1.3.6.1.2.1.16.9.1.1.3) is set to none.
- If you only specify **log**, **eventType** is set to log.
- If you only specify **trap**, the **eventType** is set to snmp-trap.
- If you specify both **log** and **trap**, **eventType** is set to log-and-trap.

Syntax To Set Up An Alarm

rmon alarm *alarmIndex* *alarmVariable* *alarmInterval* {**absolute** | **delta**} **rising-threshold** *alarmRisingThreshold* [*alarmRisingEventIndex*] **falling-threshold** *alarmFallingThreshold* [*alarmFallingEventIndex*] [**owner** *alarmOwner*]

Syntax Description

1. **alarm** Configure an RMON alarm.
2. *alarmIndex* Alarm number (1 5535)
3. *alarmVariable* MIB object to monitor (WORD)
4. *alarmInterval* Sample interval (1œ294967295)
5. **absolute** Test each sample directly.
6. **delta** Test delta between samples.
7. **rising–threshold** Configure the rising threshold.
8. *alarmRisingThreshold* Rising threshold value (–2147483648§147483647)
9. *alarmRisingEventIndex* (optional) Event to fire when the rising threshold is crossed (1 5535)
10. **falling–threshold** Configure the falling threshold.
11. *alarmFallingThreshold* Falling threshold value (–2147483648§147483647)
12. *alarmFallingEventIndex* (Optional) Event to fire when the falling threshold is crossed (1 5535)
13. **owner** *alarmOwner* (Optional) Specify an owner for the alarm (WORD).

The *alarmVariable* is specified one of these ways:

- As the entire dotted decimal Abstract Syntax Notation One (ASN.1) object identifier (OID) for the object (such as **.1.3.6.1.2.1.2.2.1.10.1**)
- With the table entry name followed by the table object number and the instance

For example, to specify ifInOctets for the first instance, use **ifEntry.10.1** for the *alarmVariable*.

Examples

In the examples in this section, `public` is the Read–Only (RO) SNMP community string and `171.68.118.100` is the host that receives the trap.

In order to set up an event to send a trap when triggered, issue these commands:

```
!-- Enter these commands on one line each.
```

```
rmon event 3 log trap public
  description "Event to create log entry and SNMP notification"
  owner "jdoe 171.68 118.100 2643"
```

```
rmon alarm 2 ifEntry.10.12 30 delta
  rising-threshold 2400000 3 falling-threshold 1800000 3
  owner "jdoe 71.68 118.100 2643"
```

In this example, a Cisco 2500 is configured to send a trap and to log an event, when the alarm threshold that monitors its own ifInOctets (ifEntry.10.1) exceeds an absolute value of 90000:

```
snmp-server host 171.68.118.100 public
```

```
SNMP-server community public RO
```

```
rmon event 1 log trap public description "High ifInOctets" owner jdoe
```

```
!-- Enter this command on one line:
```

```
rmon alarm 10 ifEntry.10.1 60 absolute
  rising-threshold 90000 1 falling-threshold 85000 owner jdoe
```

The monitoring occurs every 60 seconds, and the falling–threshold is 85000. In this case, the NetView management station received this trap:

```
router.rtp.cisco.com:
A RMON Rising Alarm:
Bytes received exceeded
  threshold 90000;

VALUE=483123 (sample TYPE=1; alarm index=10)
```

Issue these commands to view logged alarms and events:

- **show rmon events** Displays the contents of the RMON event table of the router. This command has no arguments or keywords.

```
Router#show rmon events
```

```
Event 12 is active, owned by manager 1
Description is interface-errors
Event firing causes log and trap to community public, last fired 00:00:00
```

- ◆ **Event 12 is active, owned by manager1** Unique index into the **eventTable**, which shows the event status as active and shows the owner of this row, as defined in the **eventTable** of RMON.
 - ◆ **Description is interface-errors** Type of event; in this case, an interface error.
 - ◆ **Event firing causes log and trap** Type of notification that the router will make about this event. Equivalent to **eventType** in RMON.
 - ◆ **community public** If an SNMP trap is to be sent, it is sent to the SNMP community that is specified by this octet string. Equivalent to **eventCommunity** in RMON.
 - ◆ **last fired** The last time that the event was generated.
- **show rmon alarms** Displays the contents of the RMON alarm table of the router. This command has no arguments or keywords.

```
Router#show rmon alarms
```

```
Alarm 2 is active, owned by manager1
Monitors ifEntry.1.1 every 30 seconds
Taking delta samples, last value was 0
Rising threshold is 15, assigned to event 12
Falling threshold is 0, assigned to event 0
On startup enable rising or falling alarm
```

- ◆ **Alarm2 is active, owned by manager1** Unique index into the **alarmTable**, which shows the alarm status as active and shows the owner of this row, as defined in the **alarmTable** of RMON.
- ◆ **Monitors ifEntry.1.1** OID of the particular variable to be sampled. Equivalent to **alarmVariable** in RMON.
- ◆ **every 30 seconds** Interval in seconds over which the data is sampled and compared with the rising and falling thresholds. Equivalent to **alarmInterval** in RMON.
- ◆ **Taking delta samples** Method to sample the selected variable and calculate the value to be compared against the thresholds. Equivalent to **alarmSampleType** in RMON.
- ◆ **Last value was 0** Value of the statistic during the last sampling period. Equivalent to **alarmValue** in RMON.
- ◆ **Rising threshold is 15** Threshold for the sampled statistics. Equivalent to **alarmRisingThreshold** in RMON.
- ◆ **assigned to event 0** Index of the EventEntry that is used when a rising threshold is crossed. Equivalent to **alarmRisingEventIndex** in RMON.

- ◆ Falling threshold is Threshold for the sampled statistic. Equivalent to *alarmFallingThreshold* in RMON.
 - ◆ Assigned to event Index of the EventEntry that that is used when a falling threshold is crossed. Equivalent to *alarmFallingEventIndex* in RMON.
 - ◆ On startup enable rising or falling alarm Alarm that may be sent when this entry is first set to valid. Equivalent to *alarmStartupAlarm* in RMON.
-

Related Information

- [Translate OID with the SNMP Object Navigator](#)
 - [Technical Support & Documentation – Cisco Systems](#)
-

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2008 – 2009 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Oct 26, 2005

Document ID: 17428
