Network-Based Application Recognition Protocol Discovery Management Information Base

The existing Network-Based Application Recognition (NBAR) feature is used to identify protocols so traffic can be classified appropriately for Quality of Service purposes. NBAR also contains a Protocol Discovery feature that displays various statistics of any NBAR-supported protocol traffic traversing an interface for the user.

The NBAR Protocol Discovery Management Information Base (MIB) expands the capabilities of NBAR Protocol Discovery by providing the following new Protocol Discovery functionalities through SNMP:

- Enable or Disable Protocol Discovery per interface
- Display Protocol Discovery statistics
- Configure and view multiple top-n tables that list protocols by bandwidth usage
- Configure thresholds based on traffic of particular NBAR-supported protocols or applications that report breaches and send notifications when these thresholds are crossed

Feature Specifications for Network-Based Application Recognition Protocol Discovery MIB

Feature History	
Release	**Modification**
Release 12.2(15)T	This feature was introduced.
Supported Platforms	
Cisco 1700, Cisco 2600, Cisco 3600, Cisco 3700, Cisco 7100, Cisco 7200, Cisco 7300, Cisco 7400, Cisco 7500 with VIP, Catalyst 6500 Family Switch with a FlexWAN card.	

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

This feature module describes the Network-Based Application Recognition Protocol Discovery MIB and contains the following sections:

- Prerequisites for NBAR Protocol Discovery MIBs, page 2
- Restrictions for NBAR Protocol Discovery MIBs, page 2
- Tables Supported by NBAR Protocol Discovery MIBs, page 2
- How to Use the NBAR Protocol Discovery MIB, page 8
- Configuration Examples For NBAR Protocol Discovery MIBs, page 14
- Additional References, page 55
- Command Reference, page 56

Prerequisites for NBAR Protocol Discovery MIBs

This feature is a MIB and therefore requires a method to read and configure MIBs in order to be used.

Restrictions for NBAR Protocol Discovery MIBs

- On some platforms, the NBAR Protocol Discovery MIB is not compatible with all possible Cisco IOS images on that platform. Access Cisco Feature Navigator at http://www.cisco.com/go/fn for information regarding Protocol Discovery MIB support in your Cisco IOS release.
- When multiple thresholds are active, a negative impact on the router could occur. The number of thresholds that are configurable on a particular platform vary based on platform and threshold types, but users should ensure that unneeded thresholds are deactivated and that thresholds are configured in such a way that unwanted breaches do not occur.

Tables Supported by NBAR Protocol Discovery MIBs

Using the NBAR Protocol Discovery MIB, the following tables can be created:

- cnpdSupportedProtocolsTable, page 3
- cnpdStatusTable, page 3
- cnpdAllStatsTable, page 4
- cnpdTopNConfigTable, page 5
- cnpdTopNStatsTable, page 6
- cnpdThresholdConfigTable, page 7
- cnpdThresholdHistoryTable, page 8

cnpdSupportedProtocolsTable

The cnpdSupportedProtocolsTable lists all the protocols and applications that NBAR is capable of recognizing on the router. It is important to note that a user can add support for additional NBAR protocols and applications by downloading Packet Description Language Modules (PDLMs) or upgrading to an IOS that has added support for additional NBAR protocols or applications; therefore, this table will not be identical on all routers.

The cnpdSupportedProtocolsTable is composed of the following elements:

- cnpdSupportedProtocolsIndex—The cnpdSupportedProtocolsIndex represents the object identifier (OID) of an NBAR-supported protocol or application. This OID is a number and is used to select or identify a particular protocol throughout this MIB.
- cnpdSupportedProtocolsName—The cnpdSupportedProtocolsName is the name of the protocol associated with a specific OID in the cnpdSupportedProtocolsTable line of this table. The last cnpdSupportedProtocolsName is always “unknown” and is used to classify protocols and applications that are not recognized using NBAR Protocol Discovery.

cnpdStatusTable

The cnpdStatusTable provides the following functionality:

- The ability to enable or disable Protocol Discovery on an interface
- The ability to view if Protocol Discovery is enabled or disabled on an interface
- The ability to view when Protocol Discovery was last enabled on an interface.

The cnpdStatusTable contains the following elements:

- IfIndex—The IfIndex is a number that represents a specific interface.
- cnpdStatusPdEnable—The cnpdStatusPdEnable object is used to determine if NBAR is enabled or disabled on the interface. If cnpdStatusPdEnable is set to true(1), then Protocol Discovery is enabled on that interface. If cnpdStatusPdEnable is set to false(2), then Protocol Discovery is not enabled on that interface.

  The cnpdStatusPDEnable object can be configured using the SNMP **setany** command. For an example of the cnpdStatusPDEnable object modification, see the “Enabling and Disabling NBAR Protocol Discovery on an Interface” section on page 9 of this document.
- cnpdStatusLastUpdateTime—The cnpdStatusLastUpdateTime displays the last time that Protocol Discovery was enabled on an interface. The number is given in timeticks which are converted to an actual time by the network management system.

cnpdAllStatsTable

The cnpdAllStatsTable is a table that contains the statistics collected on all NBAR-supported protocols on a per-interface basis.

The cnpdAllStatsTable contains the following elements:

- IfIndex—The IfIndex is a number that represents a specific interface.
- ProtocolsIndex—This number represents the protocol being monitored. To see which protocols are mapped to which numbers, see the cnpdSupportedProtocolsTable table.
- cnpdAllStatsInPkts—The cnpdAllStatsInPkts represents the number of packets that have been received by a specific interface. This value is given as a 32-bit variable; if the 32-bit variable is unable to handle the value (because SNMPv2 can support a 64-bit counter), the value used by the Network Management System (NMS) in place of cnpdAllStatInPkts is cnpdHCInPkts.
- cnpdAllStatsOutPkts—The cnpdAllStatsOutPks value represents the number of packets for traffic of a specific protocol or application that have left a specific interface. This value is given as a 32-bit variable; if the 32-bit variable is too small to handle the value (because SNMPv2 can support a 64-bit counter), the value used by the NMS in place of cnpdAllStatOutPkts is cnpdHCOutPkts.
- cnpdAllStatsInBytes—The cnpdAllStatsInBytes represent the total number of bytes for traffic of a specific protocol or application that have entered a specific interface. This value is given as a 32-bit variable; if the 32-bit variable is too small to handle the value (because SNMPv2 can support a 64-bit counter), the value used by the NMS in place of cnpdAllStatInBytes is cnpdHCInBytes.
- cnpdAllStatsOutBytes—The cnpdAllStatsOutBytes represent the total number of bytes for traffic of a specific protocol or application that have left a specific interface. This value is given as a 32-bit variable; if the 32-bit variable is unable to handle the value (because SNMPv2 can support a 64-bit counter), the value used by the NMS in place of cnpdAllStatOutBytes is cnpdHCOutBytes.
- cnpdAllStatsHCInPkts—The cnpdAllStatsHCInPkts represent the total number of packets for traffic of a specific protocol or application that have entered a specific interface. This value is given as a 64-bit variable. If the cnpdAllStatsInPkts value could fit into the 32-bit counter, the cnpdAllStatsHCInPkts value will match the cnpdInPkts value. If the cnpdAllStatsInPkts value could not fit into the 64-bit counter, the NMS will use the cnpdAllStatsHCInPkts value in place of the cnpdAllStatsInPkts value.
- cnpdAllStatsHCOutPkts—The cnpdAllStatsHCOutPkts represent the total number of packets for traffic of a specific protocol or application that have left a specific interface. This value is given as a 64-bit variable. If the cnpdAllStatsOutPkts value could fit into the 32-bit counter, the cnpdAllStatsHCOutPkts value will match the cnpdOutPkts value. If the cnpdAllStatsOutPkts value could not fit into the 64-bit counter, the NMS will use the cnpdAllStatsHCOutPkts value in place of the cnpdAllStatsOutPkts value.
- cnpdAllStatsHCInBytes—The cnpdAllStatsHCInBytes represent the total number of bytes for traffic of a specific protocol or application that have entered a specific interface. This value is given in hexadecimal format. This value is given as a 64-bit variable. If the cnpdAllStatsInBytes value could fit into the 32-bit counter, the cnpdAllStatsHCInBytes value will match the cnpdInBytes value. If the cnpdAllStatsInBytes value could not fit into the 64-bit counter, the NMS will use the cnpdAllStatsHCInBytes value in place of the cnpdAllStatsInBytes value.
- cnpdAllStatsHCOutBytes—The cnpdAllStatsHCOutBytes represent the total number of bytes for traffic of a specific protocol or application that have left a specific interface. This value is given in hexadecimal format. This value is given as a 64-bit variable. If the cnpdAllStatsOutBytes value could fit into the 32-bit counter, the cnpdAllStatsHCOutBytes value will match the cnpdAllStatsOutBytes value. If the cnpdAllStatsOutBytes value could not fit into the 64-bit counter, the NMS will use the cnpdAllStatsHCOutBytes value in place of the cnpdAllStatsOutBytes value.

- cnpdAllStatsInBitRate—The cnpdAllStatsInBitRate represents the bit rate for traffic entering a specific interface.
- cnpdAllStatsOutBitRate—The cnpdAllStatsOutBitRate represents the bit rate for traffic leaving a specific interface.

cnpdTopNConfigTable

The cnpdTopNConfigTable is used to request a top-n list of protocols and their statitistics.

The cnpdTopNConfigTable contains the following elements:

- cnpdTopNConfigIndex—The cnpdTopNConfigIndex is a number that represents a set of configuration parameters that will result in as a single row in the entire table.
- cnpdTopNConfigIfIndex—The cnpdTopNConfigIfIndex is a number that identifies a specific interface.
- cnpdTopNConfigStatsSelect—The cnpdTopNConfigStatsSelect determines the statistic used to calculate the order of precedence of the top-n protocols in the cnpdTopNStatsTable. The following statistics can be chosen:
  - bitRateIn(1)—Incoming bit rate
  - bitRateOut(2)—Outgoing bit rate
  - bitRateSum(3)—The sum of the incoming and outgoing bit rates
  - byteCountIn(4)—Incoming byte count
  - byteCountOut(5)—Outgoing byte count
  - byteCountSum(6)—Sum of incoming and outgoing byte counts
  - packetCountIn(7)—Incoming packet count
  - packetCountOut(8)—Outgoing packet count
  - packetCountSum(9)—Sum of incoming and outgoing packet counts
- cnpdTopNConfigSampleTime—For statistics based on bit rates only, the cnpdTopNConfigSampleTime determines the intervals during which the bit rate is sampled.
- cnpdTopNConfigRequestedSize—The cnpdTopNConfigRequestedSize selects the number of protocols or applications shown in the top-n table (in other words, it represent the *n* variable in the term “top-n”). In some cases, the cnpdTopNConfigRequestedSize may not show as many statistics as selected due to memory restrictions. The cnpdTopNConfigGrantedSize represents the actual number of protocols or applications displayed in the top-n table.
- cnpdTopNConfigGrantedSize—The cnpdTopNConfigGrantedSize is the actual number of protocols or applications that are shown in the top-n table. The number of protocols or applications in a top-n table does not always match the cnpdTopNConfigRequestedSize due to memory restrictions.
- cnpdTopNConfigTime—The cnpdTopNConfigTime represents the time in timeticks that a particular row entry was made active.

- cnpdTopNConfigStatus—The cnpdTopNConfigStatus creates and deletes rows in the cnpdTopNConfigTable.

  It can have the following values:

  – Active—A TopNStats entry has been generated for this row.

  – NotInService—A TopNStats entry has not been generated for this row because cnpdTopNConfigStatus has been manually set to NotInService.

  – NotReady—A TopNStats entry has not been generated for this row.

  – createAndGo—create a table with default values for empty fields.

  – createAndWait—create a table and set values for all fields.

  – Destroy—destroy table.

  All of these values can be set at an appropriate time, but Active, NotInService, and NotReady are the only states that this object can be converted to after the values are initially set.

  Each row in the cnpdTopNConfigStatus contains a default value. These default values are used when a top-n table is created using createAndGo. If createAndGo is used to create a table from scratch, default values are used for each table row. The actual default values are defined in the MIB.

  A user creating a top-n table should be aware that when a table is created using createAndGo, no default value can be assigned to an interface and the row status will therefore be NotReady.

cnpdTopNStatsTable

The cnpdTopNStatsTable contains an overall view of the TopNStats. In particular, this table takes the values of TopNConfigTable and the cnpdTopNConfigGrantedSize and produces an overall dynamic top-n table that monitors these objects.

This cnpdTopNStatsTable contains the following elements:

- cnpdTopNConfigIndex—The cnpdTopNConfigIndex represents the value of the index in the associated cnpdTopNConfigTable.
- cnpdTopNStatsIndex—The cnpdTopNStatsIndex indicates a position of a specific protocol in the top-n table.
- cnpdTopNStatsIfIndex—The cnpdTopNStatsIfIndex specifies the interface where the statistic of the top-n table is being monitored.
- cnpdTopNStatsProtocolName—The name of the protocol being measured.
- cnpdTopNStatsRate—measures the rate of the measured statistic. This value is given as a 32-bit variable. If this value cannot fit into the 32-bit variable, the cnpdTopNHCRate variable is used to provide the rate.
- cnpdTopNHCRate—measures the rate of the measured statistic. This value is given as a 64-bit variable. If this value can fit into the 32-bit variable, this value will match cnpdTopNStatsRate. If this value cannot fit into the 32-bit variable, the cnpdTopNHCRate is used to provide this rate.

cnpdThresholdConfigTable

The cnpdThresholdConfigTable is used to configure thresholds based on Protocol Discovery statistics.

The cnpdThresholdConfigTable contains the following elements:

- cnpdThresholdConfigIndex—Represents a threshold entry or notification if configured.
- cnpdThresholdConfigIfIndex—Represents the interface on which the protocol or application will be measured.
- cndpThresholdConfigInterval—Represents the number of seconds that elapse before polling for the application or protocol.
- cnpdThresholdConfigSampleType—This value determines how statistics are sampled for the threshold.
  - If the cnpdThresholdConfigSampleType is set at absoluteValue(1), the value at the end of the sampling interval cnpdThresholdConfigInterval will be compared with the cnpdThresholdConfigRising and cnpdThresholdConfigFalling thresholds.
  - If the cnpdThresholdConfigSampleType is set at deltaValue(2), the difference between the samples at the beginning and at the end of the cnpdThresholdConfigInterval will be comparted with the cnpdThresholdConfigRising and cnpdThresholdConfigFalling thresholds.
- cnpdThresholdConfigProtocol—The application or protocol that the thresholds for which the thresholds are being set.
- cnpdThresholdConfigAny—This setting determines if “any” protocol is being monitored for thresholds or if a particular protocol is being monitored for thresholds.
  - If cnpdThresholdConfigAny is set to true, “any” protocol is being monitored for thresholds. When any protocol is set, all protocols are monitored to see if they breach the thresholds and an SNMP trap is sent if any individual protocol breaches the threshold.
  - If cnpdThresholdConfigAny is set to false, a particular protocol is monitored to see if it crosses the configured thresholds. The cnpdThresholdConfigProtocol is used to set the particular protocol that is being monitored for threshold crossing.
- cnpdThresholdConfigProtocol—The cnpdThresholdConfigProtocol is used to set the particular protocol that is being monitored for threshold crossing.
- cnpdThresholdConfigStatsSelect—The cnpdThresholdConfigStatsSelect specifies the statistic that is being measured.
- cnpdThresholdConfigStartup—The cnpdThresholdConfigStartup controls if and when a notification should be generated the first time a statistic is measured. The following values may appear:
  - rising(1)—If cnpdThresholdConfigStartup is set to rising, no threshold breaches will be reported if the first measure of a statistic is above the rising threshold.
  - falling(2)—If CnpdThresholdConfigStartup is set to falling, no threshold breaches will be reported if the first measure of a statistic is below the falling threshold.
  - risingOrFalling(3)—The default setting. If the cnpdThresholdConfigStartup is set to risingOrFalling, no threshold breaches will be reported when the first measure of a statistic is reported.
- cnpdThresholdConfigRising—The cnpdThresholdConfigRising specifies the high value of the statistic being monitored that needs to be breached for a notification to be sent.
- cnpdThresholdConfigFalling—The cnpdThresholdConfigFalling specifies the low value of the statistic being monitored that needs to be breached for a notification to be sent.

- cnpdThresholdConfigStatus—Specifies if the row on the table is Active or NotReady.

cnpdThresholdHistoryTable

The cnpdThresholdHistoryTable keeps a history of all thresholds that have been breached. It is a two-dimensional table that tracks each particular breached threshold (which is numbered using cnpdThresholdHistoryIndex) and the entire cnpdThresholdConfigTable.

The cnpdThresholdHistoryTable contains the following elements:

- cnpdThresholdHistoryIndex—The cnpdThresholdHistoryIndex is a number that increments each time a threshold is breached.
- cnpdThresholdHistoryConfigIndex—represents a threshold entry or notification if configured. Same value as in the cnpdThresholdConfigTable.
- cnpdThresholdHistoryValue—The value of the sample at the time of the breach.
- cnpdThresholdHistoryType—indicates when the rising or falling threshold was breached in timeticks.
- cnpdThresholdHistoryTime—The time of the breach.
- cnpdThresholdHistoryProtocol—indicates the protocol that has breached the threshold.
- cnpdThresholdHistoryStatsSelect—indicates the statistic that was being monitored when this threshold was breached.

How to Use the NBAR Protocol Discovery MIB

The following sections provide information on configuring elements of the Protocol Discovery MIB and contains the following sections:

- Querying the Supported Protocols Table, page 9
- Enabling and Disabling NBAR Protocol Discovery on an Interface, page 9
- Searching the AllStats Table, page 10
- Creating a Top-N Table, page 11
- Setting Protocol Thresholds, page 13

Note Throughout this document, the **setany** and **getmany** commands used with some MIB tools are used in the examples. In these cases, the **setany** command is equivalent to the SNMP set command and the **getmany** command is equivalent to the SNMP getbulk command.

Note For detailed configurations and outputs, we strongly suggest viewing the "Configuration Examples For NBAR Protocol Discovery MIBs" section on page 14 of this document.

Querying the Supported Protocols Table

The Supported Protocols Table is used to see which protocols are supported by the NBAR Protocol Discovery MIB and to display which object identifier (OID) is assigned to each NBAR-supported application or protocol. The OID is a number that is used in the NBAR Protocol Discovery MIB to identify protocols and applications and to configure and read thresholds.

SUMMARY STEPS

1. **getmany -v2c IP-address public cnpdSupported Protocols**

Examples

To view the supported protocols table, enter the following command in SNMP:

>$ getmany -v2c a.b.c.d public cnpdSupportedProtocols

(where a.b.c.d is the IP address of the router).

Enabling and Disabling NBAR Protocol Discovery on an Interface

The Status Enabled Table is used to either enable or disable NBAR Protocol Discovery on an interface or to view whether NBAR Protocol Discovery is enabled.

SUMMARY STEPS for Enabling NBAR Protocol Discovery on an Interface

1. **getmany -v2c** *IP-address* **public cnpdStatusTable** (optional)
2. **setany -v2c** *IP-address* **public cnpdStatusPdEnable.***interface-number* **-i 1**

SUMMARY STEPS for Enabling NBAR Protocol Discovery on an Interface

1. **getmany -v2c** *IP-address* **public cnpdStatusTable** (optional)
2. **setany -v2c** *IP-address* **public cnpdStatusPdEnable.***interface-number* **-i 2**

Examples

To view the Status Enabled Table, enter the following SNMP command:

>$ getmany -v2c a.b.c.d public cnpdStatusTable

(where a.b.c.d is the IP address of the router).

To enable Protocol Discovery on an interface, enter the following SNMP command:

>$ setany -v2c a.b.c.d public cnpdStatusPdEnable.14 -i 1

(where a.b.c.d is the IP address of the router, 14 is the number of the interface where Protocol Discovery is being enabled, and 1 sets the cnpdStatusPdEnable object to true(1) to enable Protocol Discovery).

To disable Protocol Discovery on an interface, enter the following SNMP command:

>$ setany -v2c a.b.c.d public cnpdStatusPdEnable.14 -i 2

(where a.b.c.d is the IP address of the router, 14 is the number of the interface where Protocol Discovery is being disabled, and 2 sets the cnpdStatusPdEnable object to false(2) to disable Protocol Discovery).

Searching the AllStats Table

The AllStats Table stores all of the statistics currently stored by NBAR Protocol Discovery. If many interfaces have enabled NBAR Protocol Discovery, the All Stats Table can get incredibly large.

To search the All Stats table, enter the following SNMP command:

>$ getmany -v2c a.b.c.d public cnpdAllStats

(where a.b.c.d is the IP address of the router).

SUMMARY STEPS

1. **getmany -v2c** *IP-address* **public cnpdAllStats**

Creating a Top-N Table

A top-n table in NBAR is a table that displays the most frequently classified NBAR-supported protocols for a specified statistic on a specified interface.

The top-n functionality in the NBAR Protocol Discovery MIB involves two tables. The first table is a configuration table in which each row represents a group of objects that will create a unique top-n report. The other table is a Statistics Results Table, which records the outputs of each row of the configuration table. The Statistics Results Table can be regenerated by the correct use of the rowStatus object in the configuration table.

Summary Steps

1. **setany -v2c** *IP-address* **public cnpdTopNConfigIfIndex.***interface-number* **-i** *OID-number*
2. **getmany -v2c** *IP-address* **public cnpdTopNConfig**
3. **setany -v2c** *IP-address* **public cnpdTopNConfigStatsSelect.***interface-number* **-i** *stats-value* **-i** *IfIndex*
4. **getmany -v2c** *IP-address* **public cnpdTopNConfig**
5. **setany -v2c** *IP-address* **public cnpdTopNConfigSampleTime.***interface-number* **-g** *number-of-timeticks*
6. **getmany -v2c** *IP-address* **public cnpdTopNConfig**
7. **setany -v2c** *IP-address* **public cnpdTopNConfigRequestedSize.***interface-number* **-g** *requested-size*
8. **getmany -v2c** *IP-address* **public cnpdTopNConfig**
9. **setany -v2c** *IP-address* **public cnpdTopNConfigStatus.***interface-number* **-i** *config-status*
10. **getmany -v2c** *IP-address* **public cnpdTopNConfig**
11. **getmany -v2c** *IP-address* **public cnpdTopNStats**

Examples

To create a Top-n Table, enter the following SNMP commands:

>$ setany -v2c a.b.c.d public cnpdTopNConfigIfIndex.1 -i 13

(where a.b.c.d is the IP address of the router, 1 is the interface on the router, and 13 is the If-Index r of the protocol being monitored).

>$ getmany -v2c a.b.c.d public cnpdTopNConfig

(where a.b.c.d is the IP address of the router).

>$ setany -v2c a.b.c.d public cnpdTopNConfigStatsSelect.1 -i 7

(where a.b.c.d is the IP address of the router, 1 is the interface on the router, and 7 is the OID number of the protocol being monitored).

>$ getmany -v2c a.b.c.d public cnpdTopNConfig

(where a.b.c.d is the IP address of the router).

>$ setany -v2c a.b.c.d public cnpdTopNConfigSampleTime.1 -g 13

(where a.b.c.d is the IP address of the router, 1 is the interface on the router, and 13 is the number of timeticks between when samples are taken for the table).

>$ getmany -v2c a.b.c.d public cnpdTopNConfig
(where a.b.d.c is the IP address of the router).

>$ setany -v2c a.b.c.d public cnpdTopNConfigRequestedSize.1 -g 5

(where a.b.c.d is the IP address of the router, 1 is the interface on the router, and 5 is the number of statistics to appear on the table [in this case, the top-5 statistics will be monitored by this top-n table assuming this object request is granted]).

>$ getmany -v2c a.b.c.d public cnpdTopNConfig

(where a.b.c.d is the IP address of the router).

>$ setany -v2c a.b.c.d public cnpdTopNConfigStatus.1 -i 4

(where a.b.c.d is the IP address of the router, 1 is the interface on the router, and 4 is the number that corresponds to a particular status).

>$ getmany -v2c a.b.c.d public cnpdTopNConfig

(where a.b.c.d is the IP address of the router).

>$ getmany -v2c a.b.c.d public cnpdTopNStats

(where a.b.c.d is the IP address of the router).

Setting Protocol Thresholds

The NBAR Protocol Discovery MIB can be used to set two types of thresholds—a threshold that sends a trap when thresholds for an individual protocol are crossed and a threshold that sends a trap when all NBAR-classifiable protocols or applications are crossed.

Setting a Threshold for a Particular Application or Protocol

A specific protocol threshold is a threshold that monitors the traffic of a specific protocol and the breaches of these thresholds are reported and stored when they occur.

The following items should be noted when configuring a specific protocol threshold:

- The cnpdThresholdConfigProtocolAny must be set to FALSE(2) and cnpdThresholdConfigProtocol must be set to a value that indicates the OID of the protocol being monitored. The OID for each protocol can be seen by querying cnpdSupportedProtocolsTable.
- Use cnpdThresholdConfigIfIndex to select the interface and cnpdThresholdConfigRising and cnpdThresholdConfigFalling to set the rising and falling thresholds.
- Set the cnpdThresholdConfigInterval to configure the frequency with which thresholds should be checked.
- Set cnpdThresholdConfigStatus to 4 (createAndGo) and this config member will become active.
- A hysterisis mechanism is used with thresholds using the NBAR Protocol Discovery MIB to avoid the reporting of multiple breaches.

Summary Steps

1. **setany -v2c** *IP-address* **public cnpdThresholdConfigProtocolAny.***interface-number* **-i 2**
2. **setany -v2c** *IP-address* **public cnpdThresholdConfigProtocol.***interface-number* **-g** *protocol-OID*
3. **setany -v2c** *IP-address* **public cnpdThresholdConfigStatsSelect.***interface-number* **-i** *statistic*
4. **setany -v2c** *IP-address* **public cnpdThresholdConfigIfIndex.***interface-number* **-i** *interface*
5. **setany -v2c** *IP-address* **public cnpdThresholdConfigRising.***interface-number* **-g** *rising-threshold*
6. **setany -v2c** *IP-address* **public cnpdThresholdConfigFalling.***interface-number* **-g** *falling-threshold*
7. **setany -v2c** *IP-address* **public cnpdThresholdConfigStatus.***interface-number* **-i** *status*

Examples

```
setany -v2c a.b.c.d public cnpdThresholdConfigProtocolAny.2 -i 2
setany -v2c a.b.c.d public cnpdThresholdConfigProtocol.2 -g 5
setany -v2c a.b.c.d public cnpdThresholdConfigStatsSelect.2 -i 7
setany -v2c a.b.c.d public cnpdThresholdConfigIfIndex.2 -i 13
setany -v2c a.b.c.d public cnpdThresholdConfigRising.2 -g 2000
setany -v2c a.b.c.d public cnpdThresholdConfigFalling.2 -g 1000
setany -v2c a.b.c.d public cnpdThresholdConfigStatus.2 -i 4
```

Setting the Any Protocol Threshold

The Any Protocol Threshold setting is used to send SNMP traps and create history entries if ANY protocol crosses a rising or falling threshold. This threshold does have a mechanism to report a breach only once in a given time period. For instance, if HTTP breaches a rising threshold, a trap is sent only for the first breach. If another protocol like DHCP breaches a rising threshold, a separate trap is sent for that breach.

The setup for any protocol is identical to a specific protocol except cnpdThresholdConfigProtocolAny does not have to be configured and it is unnecessary to specify a cnpdThresholdConfigProtocol.

Summary Steps

1. **setany -v2c** *IP-address* **public cnpdThresholdConfigStatsSelect.***interface-number* **-i** *statistic*
2. **setany -v2c** *IP-address* **public cnpdThresholdConfigIfIndex.***interface-number* **-i** *interface*
3. **setany -v2c** *IP-address* **public cnpdThresholdConfigRising.***interface-number* **-g** *rising-threshold*
4. **setany -v2c** *IP-address* **public cnpdThresholdConfigFalling.***interface-number* **-g** *falling-threshold*
5. **setany -v2c** *IP-address* **public cnpdThresholdConfigStatus.***interface-number* **-i** *status*

Examples

```
setany -v2c a.b.c.d public cnpdThresholdConfigStatsSelect.1 -i 7
setany -v2c a.b.c.d public cnpdThresholdConfigIfIndex.1 -i 13
setany -v2c a.b.c.d public cnpdThresholdConfigRising.1 -g 30
setany -v2c a.b.c.d public cnpdThresholdConfigFalling.1 -g 20
setany -v2c a.b.c.d public cnpdThresholdConfigStatus.1 -i 4
```

Configuration Examples For NBAR Protocol Discovery MIBs

This section provides the following configuration examples:

- Querying Supported Protocols Table, page 15
- Query Status Enabled Table, page 16
- Enabling and Disabling Protocol Discovery, page 18
- Disabling Protocol Discovery on an Interface, page 19
- Searching the All Stats Table, page 21
- Creating Top-N Tables Using the NBAR Protocol Discovery MIB, page 36
- Configuring Thresholds, page 39
- Threshold Options, page 40

Note Throughout this document, the **setany** and **getmany** commands used with some MIB tools are used in the examples. In these cases, the **setany** command is equivalent to the SNMP **set** command and the **getmany** command is equivalent to the SNMP getbulk command.

Querying Supported Protocols Table

The Querying Supported Protocols Table serves two purposes:

- Lists all the protocols and applications that can be classified by NBAR.
- Gives an OID per application or protocol that can be used in the NBAR Protocol Discovery MIB to identify protocols and applications and to configure and read thresholds.

```
>$ getmany -v2c  a.b.c.d public cnpdSupportedProtocols

cnpdSupportedProtocolsName.1 = rtp
cnpdSupportedProtocolsName.2 = fasttrack
cnpdSupportedProtocolsName.3 = napster
cnpdSupportedProtocolsName.4 = citrix
cnpdSupportedProtocolsName.5 = http
cnpdSupportedProtocolsName.6 = custom-10
cnpdSupportedProtocolsName.7 = custom-09
cnpdSupportedProtocolsName.8 = custom-08
cnpdSupportedProtocolsName.9 = custom-07
cnpdSupportedProtocolsName.10 = custom-06
cnpdSupportedProtocolsName.11 = custom-05
cnpdSupportedProtocolsName.12 = custom-04
cnpdSupportedProtocolsName.13 = custom-03
cnpdSupportedProtocolsName.14 = custom-02
cnpdSupportedProtocolsName.15 = custom-01
cnpdSupportedProtocolsName.16 = gnutella
cnpdSupportedProtocolsName.17 = streamwork
cnpdSupportedProtocolsName.18 = sunrpc
cnpdSupportedProtocolsName.19 = netshow
cnpdSupportedProtocolsName.20 = rcmd
cnpdSupportedProtocolsName.21 = sqlnet
cnpdSupportedProtocolsName.22 = vdolive
cnpdSupportedProtocolsName.23 = realaudio
cnpdSupportedProtocolsName.24 = exchange
cnpdSupportedProtocolsName.25 = tftp
cnpdSupportedProtocolsName.26 = novadigm
cnpdSupportedProtocolsName.27 = printer
cnpdSupportedProtocolsName.28 = xwindows
cnpdSupportedProtocolsName.29 = secure-ftp
cnpdSupportedProtocolsName.30 = secure-telnet
cnpdSupportedProtocolsName.31 = telnet
cnpdSupportedProtocolsName.32 = syslog
cnpdSupportedProtocolsName.33 = ssh
cnpdSupportedProtocolsName.34 = socks
cnpdSupportedProtocolsName.35 = snmp
cnpdSupportedProtocolsName.36 = smtp
cnpdSupportedProtocolsName.37 = rsvp
cnpdSupportedProtocolsName.38 = rip
cnpdSupportedProtocolsName.39 = pptp
cnpdSupportedProtocolsName.40 = secure-pop3
cnpdSupportedProtocolsName.41 = pop3
cnpdSupportedProtocolsName.42 = pcanywhere
```

```
cnpdSupportedProtocolsName.43 = ntp
cnpdSupportedProtocolsName.44 = notes
cnpdSupportedProtocolsName.45 = secure-nntp
cnpdSupportedProtocolsName.46 = nntp
cnpdSupportedProtocolsName.47 = nfs
cnpdSupportedProtocolsName.48 = netbios
cnpdSupportedProtocolsName.49 = sqlserver
cnpdSupportedProtocolsName.50 = secure-ldap
cnpdSupportedProtocolsName.51 = ldap
cnpdSupportedProtocolsName.52 = l2tp
cnpdSupportedProtocolsName.53 = kerberos
cnpdSupportedProtocolsName.54 = secure-irc
cnpdSupportedProtocolsName.55 = irc
cnpdSupportedProtocolsName.56 = secure-imap
cnpdSupportedProtocolsName.57 = imap
cnpdSupportedProtocolsName.58 = secure-http
cnpdSupportedProtocolsName.59 = gopher
cnpdSupportedProtocolsName.60 = finger
cnpdSupportedProtocolsName.61 = dns
cnpdSupportedProtocolsName.62 = dhcp
cnpdSupportedProtocolsName.63 = cuseeme
cnpdSupportedProtocolsName.64 = bgp
cnpdSupportedProtocolsName.65 = ipsec
cnpdSupportedProtocolsName.66 = ipinip
cnpdSupportedProtocolsName.67 = eigrp
cnpdSupportedProtocolsName.68 = icmp
cnpdSupportedProtocolsName.69 = gre
cnpdSupportedProtocolsName.70 = egp
cnpdSupportedProtocolsName.71 = ftp
cnpdSupportedProtocolsName.72 = unknown
```

Query Status Enabled Table

The Query Status Enabled Table has two purposes:

- Each row represents an interface on the router and whether NBAR Protocol Discovery is enabled or disabled on that interface.
- Each row can be configured to enable or disable NBAR Protocol Discovery using SNMP set.

```
>$ getmany -v2c  a.b.c.d public cnpdStatusTable

cnpdStatusPdEnable.1 = false(2)
cnpdStatusPdEnable.2 = false(2)
cnpdStatusPdEnable.3 = false(2)
cnpdStatusPdEnable.4 = false(2)
cnpdStatusPdEnable.5 = false(2)
cnpdStatusPdEnable.6 = false(2)
cnpdStatusPdEnable.7 = false(2)
cnpdStatusPdEnable.8 = false(2)
cnpdStatusPdEnable.9 = false(2)
cnpdStatusPdEnable.10 = false(2)
cnpdStatusPdEnable.11 = false(2)
```

```
cnpdStatusPdEnable.12 = false(2)
cnpdStatusPdEnable.13 = true(1)
cnpdStatusPdEnable.14 = false(2)
cnpdStatusPdEnable.15 = false(2)
cnpdStatusPdEnable.16 = false(2)
cnpdStatusPdEnable.17 = false(2)
cnpdStatusPdEnable.18 = false(2)
cnpdStatusPdEnable.19 = false(2)
cnpdStatusPdEnable.20 = false(2)
cnpdStatusPdEnable.21 = false(2)
cnpdStatusPdEnable.22 = false(2)
cnpdStatusPdEnable.23 = false(2)
cnpdStatusPdEnable.24 = false(2)
cnpdStatusPdEnable.25 = false(2)
cnpdStatusPdEnable.26 = false(2)
cnpdStatusPdEnable.27 = false(2)
cnpdStatusPdEnable.28 = false(2)
cnpdStatusPdEnable.29 = false(2)
cnpdStatusPdEnable.30 = false(2)
cnpdStatusPdEnable.31 = false(2)
cnpdStatusPdEnable.32 = false(2)
cnpdStatusPdEnable.33 = false(2)
cnpdStatusPdEnable.34 = false(2)
cnpdStatusPdEnable.35 = false(2)
cnpdStatusLastUpdateTime.1 = 0
cnpdStatusLastUpdateTime.2 = 0
cnpdStatusLastUpdateTime.3 = 0
cnpdStatusLastUpdateTime.4 = 0
cnpdStatusLastUpdateTime.5 = 0
cnpdStatusLastUpdateTime.6 = 0
cnpdStatusLastUpdateTime.7 = 0
cnpdStatusLastUpdateTime.8 = 0
cnpdStatusLastUpdateTime.9 = 0
cnpdStatusLastUpdateTime.10 = 0
cnpdStatusLastUpdateTime.11 = 0
cnpdStatusLastUpdateTime.12 = 0
cnpdStatusLastUpdateTime.13 = 1111
cnpdStatusLastUpdateTime.14 = 0
cnpdStatusLastUpdateTime.15 = 0
cnpdStatusLastUpdateTime.16 = 0
cnpdStatusLastUpdateTime.17 = 0
cnpdStatusLastUpdateTime.18 = 0
cnpdStatusLastUpdateTime.19 = 0
cnpdStatusLastUpdateTime.20 = 0
cnpdStatusLastUpdateTime.21 = 0
cnpdStatusLastUpdateTime.22 = 0
cnpdStatusLastUpdateTime.23 = 0
cnpdStatusLastUpdateTime.24 = 0
cnpdStatusLastUpdateTime.25 = 0
cnpdStatusLastUpdateTime.26 = 0
cnpdStatusLastUpdateTime.27 = 0
cnpdStatusLastUpdateTime.28 = 0
cnpdStatusLastUpdateTime.29 = 0
```

```
cnpdStatusLastUpdateTime.30 = 0
cnpdStatusLastUpdateTime.31 = 0
cnpdStatusLastUpdateTime.32 = 0
cnpdStatusLastUpdateTime.33 = 0
cnpdStatusLastUpdateTime.34 = 0
cnpdStatusLastUpdateTime.35 = 0
```

Enabling and Disabling Protocol Discovery

The following examples show how to enable Protocol Discovery using the NBAR Protocol Discovery MIB using the SNMP set.

```
>$ setany -v2c  a.b.c.d public cnpdStatusPdEnable.14 -i 1


cnpdStatusPdEnable.14 = true(1)


>$ getmany -v2c  a.b.c.d public cnpdStatusTable


cnpdStatusPdEnable.1 = false(2)
cnpdStatusPdEnable.2 = false(2)
cnpdStatusPdEnable.3 = false(2)
cnpdStatusPdEnable.4 = false(2)
cnpdStatusPdEnable.5 = false(2)
cnpdStatusPdEnable.6 = false(2)
cnpdStatusPdEnable.7 = false(2)
cnpdStatusPdEnable.8 = false(2)
cnpdStatusPdEnable.9 = false(2)
cnpdStatusPdEnable.10 = false(2)
cnpdStatusPdEnable.11 = false(2)
cnpdStatusPdEnable.12 = false(2)
cnpdStatusPdEnable.13 = true(1)
cnpdStatusPdEnable.14 = true(1)
cnpdStatusPdEnable.15 = false(2)
cnpdStatusPdEnable.16 = false(2)
cnpdStatusPdEnable.17 = false(2)
cnpdStatusPdEnable.18 = false(2)
cnpdStatusPdEnable.19 = false(2)
cnpdStatusPdEnable.20 = false(2)
cnpdStatusPdEnable.21 = false(2)
cnpdStatusPdEnable.22 = false(2)
cnpdStatusPdEnable.23 = false(2)
cnpdStatusPdEnable.24 = false(2)
cnpdStatusPdEnable.25 = false(2)
cnpdStatusPdEnable.26 = false(2)
cnpdStatusPdEnable.27 = false(2)
cnpdStatusPdEnable.28 = false(2)
cnpdStatusPdEnable.29 = false(2)
cnpdStatusPdEnable.30 = false(2)
cnpdStatusPdEnable.31 = false(2)
cnpdStatusPdEnable.32 = false(2)
```

```
cnpdStatusPdEnable.33 = false(2)
cnpdStatusPdEnable.34 = false(2)
cnpdStatusPdEnable.35 = false(2)
cnpdStatusLastUpdateTime.1 = 0
cnpdStatusLastUpdateTime.2 = 0
cnpdStatusLastUpdateTime.3 = 0
cnpdStatusLastUpdateTime.4 = 0
cnpdStatusLastUpdateTime.5 = 0
cnpdStatusLastUpdateTime.6 = 0
cnpdStatusLastUpdateTime.7 = 0
cnpdStatusLastUpdateTime.8 = 0
cnpdStatusLastUpdateTime.9 = 0
cnpdStatusLastUpdateTime.10 = 0
cnpdStatusLastUpdateTime.11 = 0
cnpdStatusLastUpdateTime.12 = 0
cnpdStatusLastUpdateTime.13 = 1111
cnpdStatusLastUpdateTime.14 = 44042577
cnpdStatusLastUpdateTime.15 = 0
cnpdStatusLastUpdateTime.16 = 0
cnpdStatusLastUpdateTime.17 = 0
cnpdStatusLastUpdateTime.18 = 0
cnpdStatusLastUpdateTime.19 = 0
cnpdStatusLastUpdateTime.20 = 0
cnpdStatusLastUpdateTime.21 = 0
cnpdStatusLastUpdateTime.22 = 0
cnpdStatusLastUpdateTime.23 = 0
cnpdStatusLastUpdateTime.24 = 0
cnpdStatusLastUpdateTime.25 = 0
cnpdStatusLastUpdateTime.26 = 0
cnpdStatusLastUpdateTime.27 = 0
cnpdStatusLastUpdateTime.28 = 0
cnpdStatusLastUpdateTime.29 = 0
cnpdStatusLastUpdateTime.30 = 0
cnpdStatusLastUpdateTime.31 = 0
cnpdStatusLastUpdateTime.32 = 0
cnpdStatusLastUpdateTime.33 = 0
cnpdStatusLastUpdateTime.34 = 0
cnpdStatusLastUpdateTime.35 = 0
```

Disabling Protocol Discovery on an Interface

The following example shows how to disable Protocol Discovery on an interface using SNMP set.

```
>$ setany -v2c  a.b.c.d public cnpdStatusPdEnable.14 -i 2

cnpdStatusPdEnable.14 = false(2)

>$ getmany -v2c  a.b.c.d public cnpdStatusTable

cnpdStatusPdEnable.1 = false(2)
cnpdStatusPdEnable.2 = false(2)
cnpdStatusPdEnable.3 = false(2)
cnpdStatusPdEnable.4 = false(2)
```

```
cnpdStatusPdEnable.5 = false(2)
cnpdStatusPdEnable.6 = false(2)
cnpdStatusPdEnable.7 = false(2)
cnpdStatusPdEnable.8 = false(2)
cnpdStatusPdEnable.9 = false(2)
cnpdStatusPdEnable.10 = false(2)
cnpdStatusPdEnable.11 = false(2)
cnpdStatusPdEnable.12 = false(2)
cnpdStatusPdEnable.13 = true(1)
cnpdStatusPdEnable.14 = false(2)
cnpdStatusPdEnable.15 = false(2)
cnpdStatusPdEnable.16 = false(2)
cnpdStatusPdEnable.17 = false(2)
cnpdStatusPdEnable.18 = false(2)
cnpdStatusPdEnable.19 = false(2)
cnpdStatusPdEnable.20 = false(2)
cnpdStatusPdEnable.21 = false(2)
cnpdStatusPdEnable.22 = false(2)
cnpdStatusPdEnable.23 = false(2)
cnpdStatusPdEnable.24 = false(2)
cnpdStatusPdEnable.25 = false(2)
cnpdStatusPdEnable.26 = false(2)
cnpdStatusPdEnable.27 = false(2)
cnpdStatusPdEnable.28 = false(2)
cnpdStatusPdEnable.29 = false(2)
cnpdStatusPdEnable.30 = false(2)
cnpdStatusPdEnable.31 = false(2)
cnpdStatusPdEnable.32 = false(2)
cnpdStatusPdEnable.33 = false(2)
cnpdStatusPdEnable.34 = false(2)
cnpdStatusPdEnable.35 = false(2)
cnpdStatusLastUpdateTime.1 = 0
cnpdStatusLastUpdateTime.2 = 0
cnpdStatusLastUpdateTime.3 = 0
cnpdStatusLastUpdateTime.4 = 0
cnpdStatusLastUpdateTime.5 = 0
cnpdStatusLastUpdateTime.6 = 0
cnpdStatusLastUpdateTime.7 = 0
cnpdStatusLastUpdateTime.8 = 0
cnpdStatusLastUpdateTime.9 = 0
cnpdStatusLastUpdateTime.10 = 0
cnpdStatusLastUpdateTime.11 = 0
cnpdStatusLastUpdateTime.12 = 0
cnpdStatusLastUpdateTime.13 = 1111
cnpdStatusLastUpdateTime.14 = 0
cnpdStatusLastUpdateTime.15 = 0
cnpdStatusLastUpdateTime.16 = 0
cnpdStatusLastUpdateTime.17 = 0
cnpdStatusLastUpdateTime.18 = 0
cnpdStatusLastUpdateTime.19 = 0
cnpdStatusLastUpdateTime.20 = 0
cnpdStatusLastUpdateTime.21 = 0
cnpdStatusLastUpdateTime.22 = 0
```

```
cnpdStatusLastUpdateTime.23 = 0
cnpdStatusLastUpdateTime.24 = 0
cnpdStatusLastUpdateTime.25 = 0
cnpdStatusLastUpdateTime.26 = 0
cnpdStatusLastUpdateTime.27 = 0
cnpdStatusLastUpdateTime.28 = 0
cnpdStatusLastUpdateTime.29 = 0
cnpdStatusLastUpdateTime.30 = 0
cnpdStatusLastUpdateTime.31 = 0
cnpdStatusLastUpdateTime.32 = 0
cnpdStatusLastUpdateTime.33 = 0
cnpdStatusLastUpdateTime.34 = 0
cnpdStatusLastUpdateTime.35 = 0
```

Searching the All Stats Table

The All Stats Table stores all of the statistics currently stored by NBAR Protocol Discovery. If many interfaces have enabled NBAR Protocol Discovery, the All Stats Table can get incredibly large. In the following example, NBAR Protocol Discovery is enabled on one interface only.

```
>$ getmany -v2c  a.b.c.d public cnpdAllStats


cnpdAllStatsProtocolName.13.1 = rtp
cnpdAllStatsProtocolName.13.2 = fasttrack
cnpdAllStatsProtocolName.13.3 = napster
cnpdAllStatsProtocolName.13.4 = citrix
cnpdAllStatsProtocolName.13.5 = http
cnpdAllStatsProtocolName.13.6 = custom-10
cnpdAllStatsProtocolName.13.7 = custom-09
cnpdAllStatsProtocolName.13.8 = custom-08
cnpdAllStatsProtocolName.13.9 = custom-07
cnpdAllStatsProtocolName.13.10 = custom-06
cnpdAllStatsProtocolName.13.11 = custom-05
cnpdAllStatsProtocolName.13.12 = custom-04
cnpdAllStatsProtocolName.13.13 = custom-03
cnpdAllStatsProtocolName.13.14 = custom-02
cnpdAllStatsProtocolName.13.15 = custom-01
cnpdAllStatsProtocolName.13.16 = gnutella
cnpdAllStatsProtocolName.13.17 = streamwork
cnpdAllStatsProtocolName.13.18 = sunrpc
cnpdAllStatsProtocolName.13.19 = netshow
cnpdAllStatsProtocolName.13.20 = rcmd
cnpdAllStatsProtocolName.13.21 = sqlnet
cnpdAllStatsProtocolName.13.22 = vdolive
cnpdAllStatsProtocolName.13.23 = realaudio
cnpdAllStatsProtocolName.13.24 = exchange
cnpdAllStatsProtocolName.13.25 = tftp
cnpdAllStatsProtocolName.13.26 = novadigm
cnpdAllStatsProtocolName.13.27 = printer
cnpdAllStatsProtocolName.13.28 = xwindows
cnpdAllStatsProtocolName.13.29 = secure-ftp
```

```
cnpdAllStatsProtocolName.13.30 = secure-telnet
cnpdAllStatsProtocolName.13.31 = telnet
cnpdAllStatsProtocolName.13.32 = syslog
cnpdAllStatsProtocolName.13.33 = ssh
cnpdAllStatsProtocolName.13.34 = socks
cnpdAllStatsProtocolName.13.35 = snmp
cnpdAllStatsProtocolName.13.36 = smtp
cnpdAllStatsProtocolName.13.37 = rsvp
cnpdAllStatsProtocolName.13.38 = rip
cnpdAllStatsProtocolName.13.39 = pptp
cnpdAllStatsProtocolName.13.40 = secure-pop3
cnpdAllStatsProtocolName.13.41 = pop3
cnpdAllStatsProtocolName.13.42 = pcanywhere
cnpdAllStatsProtocolName.13.43 = ntp
cnpdAllStatsProtocolName.13.44 = notes
cnpdAllStatsProtocolName.13.45 = secure-nntp
cnpdAllStatsProtocolName.13.46 = nntp
cnpdAllStatsProtocolName.13.47 = nfs
cnpdAllStatsProtocolName.13.48 = netbios
cnpdAllStatsProtocolName.13.49 = sqlserver
cnpdAllStatsProtocolName.13.50 = secure-ldap
cnpdAllStatsProtocolName.13.51 = ldap
cnpdAllStatsProtocolName.13.52 = l2tp
cnpdAllStatsProtocolName.13.53 = kerberos
cnpdAllStatsProtocolName.13.54 = secure-irc
cnpdAllStatsProtocolName.13.55 = irc
cnpdAllStatsProtocolName.13.56 = secure-imap
cnpdAllStatsProtocolName.13.57 = imap
cnpdAllStatsProtocolName.13.58 = secure-http
cnpdAllStatsProtocolName.13.59 = gopher
cnpdAllStatsProtocolName.13.60 = finger
cnpdAllStatsProtocolName.13.61 = dns
cnpdAllStatsProtocolName.13.62 = dhcp
cnpdAllStatsProtocolName.13.63 = cuseeme
cnpdAllStatsProtocolName.13.64 = bgp
cnpdAllStatsProtocolName.13.65 = ipsec
cnpdAllStatsProtocolName.13.66 = ipinip
cnpdAllStatsProtocolName.13.67 = eigrp
cnpdAllStatsProtocolName.13.68 = icmp
cnpdAllStatsProtocolName.13.69 = gre
cnpdAllStatsProtocolName.13.70 = egp
cnpdAllStatsProtocolName.13.71 = ftp
cnpdAllStatsProtocolName.13.72 = unknown
cnpdAllStatsInPkts.13.1 = 0
cnpdAllStatsInPkts.13.2 = 0
cnpdAllStatsInPkts.13.3 = 108392
cnpdAllStatsInPkts.13.4 = 0
cnpdAllStatsInPkts.13.5 = 3681501
cnpdAllStatsInPkts.13.6 = 0
cnpdAllStatsInPkts.13.7 = 0
cnpdAllStatsInPkts.13.8 = 0
cnpdAllStatsInPkts.13.9 = 0
cnpdAllStatsInPkts.13.10 = 0
```

```
cnpdAllStatsInPkts.13.11 = 0
cnpdAllStatsInPkts.13.12 = 0
cnpdAllStatsInPkts.13.13 = 0
cnpdAllStatsInPkts.13.14 = 0
cnpdAllStatsInPkts.13.15 = 0
cnpdAllStatsInPkts.13.16 = 0
cnpdAllStatsInPkts.13.17 = 0
cnpdAllStatsInPkts.13.18 = 0
cnpdAllStatsInPkts.13.19 = 0
cnpdAllStatsInPkts.13.20 = 0
cnpdAllStatsInPkts.13.21 = 0
cnpdAllStatsInPkts.13.22 = 0
cnpdAllStatsInPkts.13.23 = 1348424
cnpdAllStatsInPkts.13.24 = 0
cnpdAllStatsInPkts.13.25 = 0
cnpdAllStatsInPkts.13.26 = 0
cnpdAllStatsInPkts.13.27 = 0
cnpdAllStatsInPkts.13.28 = 0
cnpdAllStatsInPkts.13.29 = 0
cnpdAllStatsInPkts.13.30 = 0
cnpdAllStatsInPkts.13.31 = 0
cnpdAllStatsInPkts.13.32 = 0
cnpdAllStatsInPkts.13.33 = 0
cnpdAllStatsInPkts.13.34 = 0
cnpdAllStatsInPkts.13.35 = 0
cnpdAllStatsInPkts.13.36 = 106086
cnpdAllStatsInPkts.13.37 = 0
cnpdAllStatsInPkts.13.38 = 0
cnpdAllStatsInPkts.13.39 = 0
cnpdAllStatsInPkts.13.40 = 0
cnpdAllStatsInPkts.13.41 = 17941
cnpdAllStatsInPkts.13.42 = 0
cnpdAllStatsInPkts.13.43 = 0
cnpdAllStatsInPkts.13.44 = 0
cnpdAllStatsInPkts.13.45 = 0
cnpdAllStatsInPkts.13.46 = 562337
cnpdAllStatsInPkts.13.47 = 0
cnpdAllStatsInPkts.13.48 = 81449
cnpdAllStatsInPkts.13.49 = 0
cnpdAllStatsInPkts.13.50 = 0
cnpdAllStatsInPkts.13.51 = 366922
cnpdAllStatsInPkts.13.52 = 0
cnpdAllStatsInPkts.13.53 = 0
cnpdAllStatsInPkts.13.54 = 0
cnpdAllStatsInPkts.13.55 = 0
cnpdAllStatsInPkts.13.56 = 0
cnpdAllStatsInPkts.13.57 = 0
cnpdAllStatsInPkts.13.58 = 0
cnpdAllStatsInPkts.13.59 = 17318
cnpdAllStatsInPkts.13.60 = 0
cnpdAllStatsInPkts.13.61 = 36182
cnpdAllStatsInPkts.13.62 = 12134
cnpdAllStatsInPkts.13.63 = 0
```

```
cnpdAllStatsInPkts.13.64 = 0
cnpdAllStatsInPkts.13.65 = 0
cnpdAllStatsInPkts.13.66 = 0
cnpdAllStatsInPkts.13.67 = 0
cnpdAllStatsInPkts.13.68 = 9046
cnpdAllStatsInPkts.13.69 = 0
cnpdAllStatsInPkts.13.70 = 0
cnpdAllStatsInPkts.13.71 = 311978
cnpdAllStatsInPkts.13.72 = 27150
cnpdAllStatsOutPkts.13.1 = 0
cnpdAllStatsOutPkts.13.2 = 0
cnpdAllStatsOutPkts.13.3 = 0
cnpdAllStatsOutPkts.13.4 = 0
cnpdAllStatsOutPkts.13.5 = 0
cnpdAllStatsOutPkts.13.6 = 0
cnpdAllStatsOutPkts.13.7 = 0
cnpdAllStatsOutPkts.13.8 = 0
cnpdAllStatsOutPkts.13.9 = 0
cnpdAllStatsOutPkts.13.10 = 0
cnpdAllStatsOutPkts.13.11 = 0
cnpdAllStatsOutPkts.13.12 = 0
cnpdAllStatsOutPkts.13.13 = 0
cnpdAllStatsOutPkts.13.14 = 0
cnpdAllStatsOutPkts.13.15 = 0
cnpdAllStatsOutPkts.13.16 = 0
cnpdAllStatsOutPkts.13.17 = 0
cnpdAllStatsOutPkts.13.18 = 0
cnpdAllStatsOutPkts.13.19 = 0
cnpdAllStatsOutPkts.13.20 = 0
cnpdAllStatsOutPkts.13.21 = 0
cnpdAllStatsOutPkts.13.22 = 0
cnpdAllStatsOutPkts.13.23 = 0
cnpdAllStatsOutPkts.13.24 = 0
cnpdAllStatsOutPkts.13.25 = 0
cnpdAllStatsOutPkts.13.26 = 0
cnpdAllStatsOutPkts.13.27 = 0
cnpdAllStatsOutPkts.13.28 = 0
cnpdAllStatsOutPkts.13.29 = 0
cnpdAllStatsOutPkts.13.30 = 0
cnpdAllStatsOutPkts.13.31 = 0
cnpdAllStatsOutPkts.13.32 = 0
cnpdAllStatsOutPkts.13.33 = 0
cnpdAllStatsOutPkts.13.34 = 0
cnpdAllStatsOutPkts.13.35 = 0
cnpdAllStatsOutPkts.13.36 = 0
cnpdAllStatsOutPkts.13.37 = 0
cnpdAllStatsOutPkts.13.38 = 0
cnpdAllStatsOutPkts.13.39 = 0
cnpdAllStatsOutPkts.13.40 = 0
cnpdAllStatsOutPkts.13.41 = 0
cnpdAllStatsOutPkts.13.42 = 0
cnpdAllStatsOutPkts.13.43 = 0
cnpdAllStatsOutPkts.13.44 = 0
```

```
cnpdAllStatsOutPkts.13.45 = 0
cnpdAllStatsOutPkts.13.46 = 0
cnpdAllStatsOutPkts.13.47 = 0
cnpdAllStatsOutPkts.13.48 = 0
cnpdAllStatsOutPkts.13.49 = 0
cnpdAllStatsOutPkts.13.50 = 0
cnpdAllStatsOutPkts.13.51 = 0
cnpdAllStatsOutPkts.13.52 = 0
cnpdAllStatsOutPkts.13.53 = 0
cnpdAllStatsOutPkts.13.54 = 0
cnpdAllStatsOutPkts.13.55 = 0
cnpdAllStatsOutPkts.13.56 = 0
cnpdAllStatsOutPkts.13.57 = 0
cnpdAllStatsOutPkts.13.58 = 0
cnpdAllStatsOutPkts.13.59 = 0
cnpdAllStatsOutPkts.13.60 = 0
cnpdAllStatsOutPkts.13.61 = 0
cnpdAllStatsOutPkts.13.62 = 0
cnpdAllStatsOutPkts.13.63 = 0
cnpdAllStatsOutPkts.13.64 = 0
cnpdAllStatsOutPkts.13.65 = 0
cnpdAllStatsOutPkts.13.66 = 0
cnpdAllStatsOutPkts.13.67 = 0
cnpdAllStatsOutPkts.13.68 = 0
cnpdAllStatsOutPkts.13.69 = 0
cnpdAllStatsOutPkts.13.70 = 0
cnpdAllStatsOutPkts.13.71 = 0
cnpdAllStatsOutPkts.13.72 = 0
cnpdAllStatsInBytes.13.1 = 0
cnpdAllStatsInBytes.13.2 = 0
cnpdAllStatsInBytes.13.3 = 20157945
cnpdAllStatsInBytes.13.4 = 0
cnpdAllStatsInBytes.13.5 = 2255639168
cnpdAllStatsInBytes.13.6 = 0
cnpdAllStatsInBytes.13.7 = 0
cnpdAllStatsInBytes.13.8 = 0
cnpdAllStatsInBytes.13.9 = 0
cnpdAllStatsInBytes.13.10 = 0
cnpdAllStatsInBytes.13.11 = 0
cnpdAllStatsInBytes.13.12 = 0
cnpdAllStatsInBytes.13.13 = 0
cnpdAllStatsInBytes.13.14 = 0
cnpdAllStatsInBytes.13.15 = 0
cnpdAllStatsInBytes.13.16 = 0
cnpdAllStatsInBytes.13.17 = 0
cnpdAllStatsInBytes.13.18 = 0
cnpdAllStatsInBytes.13.19 = 0
cnpdAllStatsInBytes.13.20 = 0
cnpdAllStatsInBytes.13.21 = 0
cnpdAllStatsInBytes.13.22 = 0
cnpdAllStatsInBytes.13.23 = 464526676
cnpdAllStatsInBytes.13.24 = 0
cnpdAllStatsInBytes.13.25 = 0
```

```
cnpdAllStatsInBytes.13.26 = 0
cnpdAllStatsInBytes.13.27 = 0
cnpdAllStatsInBytes.13.28 = 0
cnpdAllStatsInBytes.13.29 = 0
cnpdAllStatsInBytes.13.30 = 0
cnpdAllStatsInBytes.13.31 = 0
cnpdAllStatsInBytes.13.32 = 0
cnpdAllStatsInBytes.13.33 = 0
cnpdAllStatsInBytes.13.34 = 0
cnpdAllStatsInBytes.13.35 = 0
cnpdAllStatsInBytes.13.36 = 10280188
cnpdAllStatsInBytes.13.37 = 0
cnpdAllStatsInBytes.13.38 = 0
cnpdAllStatsInBytes.13.39 = 0
cnpdAllStatsInBytes.13.40 = 0
cnpdAllStatsInBytes.13.41 = 1298760
cnpdAllStatsInBytes.13.42 = 0
cnpdAllStatsInBytes.13.43 = 0
cnpdAllStatsInBytes.13.44 = 0
cnpdAllStatsInBytes.13.45 = 0
cnpdAllStatsInBytes.13.46 = 149193275
cnpdAllStatsInBytes.13.47 = 0
cnpdAllStatsInBytes.13.48 = 14181177
cnpdAllStatsInBytes.13.49 = 0
cnpdAllStatsInBytes.13.50 = 0
cnpdAllStatsInBytes.13.51 = 25078176
cnpdAllStatsInBytes.13.52 = 0
cnpdAllStatsInBytes.13.53 = 0
cnpdAllStatsInBytes.13.54 = 0
cnpdAllStatsInBytes.13.55 = 0
cnpdAllStatsInBytes.13.56 = 0
cnpdAllStatsInBytes.13.57 = 0
cnpdAllStatsInBytes.13.58 = 0
cnpdAllStatsInBytes.13.59 = 1128158
cnpdAllStatsInBytes.13.60 = 0
cnpdAllStatsInBytes.13.61 = 5164765
cnpdAllStatsInBytes.13.62 = 4214412
cnpdAllStatsInBytes.13.63 = 0
cnpdAllStatsInBytes.13.64 = 0
cnpdAllStatsInBytes.13.65 = 0
cnpdAllStatsInBytes.13.66 = 0
cnpdAllStatsInBytes.13.67 = 0
cnpdAllStatsInBytes.13.68 = 633220
cnpdAllStatsInBytes.13.69 = 0
cnpdAllStatsInBytes.13.70 = 0
cnpdAllStatsInBytes.13.71 = 21057032
cnpdAllStatsInBytes.13.72 = 2339425
cnpdAllStatsOutBytes.13.1 = 0
cnpdAllStatsOutBytes.13.2 = 0
cnpdAllStatsOutBytes.13.3 = 0
cnpdAllStatsOutBytes.13.4 = 0
cnpdAllStatsOutBytes.13.5 = 0
cnpdAllStatsOutBytes.13.6 = 0
```

```
cnpdAllStatsOutBytes.13.7 = 0
cnpdAllStatsOutBytes.13.8 = 0
cnpdAllStatsOutBytes.13.9 = 0
cnpdAllStatsOutBytes.13.10 = 0
cnpdAllStatsOutBytes.13.11 = 0
cnpdAllStatsOutBytes.13.12 = 0
cnpdAllStatsOutBytes.13.13 = 0
cnpdAllStatsOutBytes.13.14 = 0
cnpdAllStatsOutBytes.13.15 = 0
cnpdAllStatsOutBytes.13.16 = 0
cnpdAllStatsOutBytes.13.17 = 0
cnpdAllStatsOutBytes.13.18 = 0
cnpdAllStatsOutBytes.13.19 = 0
cnpdAllStatsOutBytes.13.20 = 0
cnpdAllStatsOutBytes.13.21 = 0
cnpdAllStatsOutBytes.13.22 = 0
cnpdAllStatsOutBytes.13.23 = 0
cnpdAllStatsOutBytes.13.24 = 0
cnpdAllStatsOutBytes.13.25 = 0
cnpdAllStatsOutBytes.13.26 = 0
cnpdAllStatsOutBytes.13.27 = 0
cnpdAllStatsOutBytes.13.28 = 0
cnpdAllStatsOutBytes.13.29 = 0
cnpdAllStatsOutBytes.13.30 = 0
cnpdAllStatsOutBytes.13.31 = 0
cnpdAllStatsOutBytes.13.32 = 0
cnpdAllStatsOutBytes.13.33 = 0
cnpdAllStatsOutBytes.13.34 = 0
cnpdAllStatsOutBytes.13.35 = 0
cnpdAllStatsOutBytes.13.36 = 0
cnpdAllStatsOutBytes.13.37 = 0
cnpdAllStatsOutBytes.13.38 = 0
cnpdAllStatsOutBytes.13.39 = 0
cnpdAllStatsOutBytes.13.40 = 0
cnpdAllStatsOutBytes.13.41 = 0
cnpdAllStatsOutBytes.13.42 = 0
cnpdAllStatsOutBytes.13.43 = 0
cnpdAllStatsOutBytes.13.44 = 0
cnpdAllStatsOutBytes.13.45 = 0
cnpdAllStatsOutBytes.13.46 = 0
cnpdAllStatsOutBytes.13.47 = 0
cnpdAllStatsOutBytes.13.48 = 0
cnpdAllStatsOutBytes.13.49 = 0
cnpdAllStatsOutBytes.13.50 = 0
cnpdAllStatsOutBytes.13.51 = 0
cnpdAllStatsOutBytes.13.52 = 0
cnpdAllStatsOutBytes.13.53 = 0
cnpdAllStatsOutBytes.13.54 = 0
cnpdAllStatsOutBytes.13.55 = 0
cnpdAllStatsOutBytes.13.56 = 0
cnpdAllStatsOutBytes.13.57 = 0
cnpdAllStatsOutBytes.13.58 = 0
cnpdAllStatsOutBytes.13.59 = 0
```

```
cnpdAllStatsOutBytes.13.60 = 0
cnpdAllStatsOutBytes.13.61 = 0
cnpdAllStatsOutBytes.13.62 = 0
cnpdAllStatsOutBytes.13.63 = 0
cnpdAllStatsOutBytes.13.64 = 0
cnpdAllStatsOutBytes.13.65 = 0
cnpdAllStatsOutBytes.13.66 = 0
cnpdAllStatsOutBytes.13.67 = 0
cnpdAllStatsOutBytes.13.68 = 0
cnpdAllStatsOutBytes.13.69 = 0
cnpdAllStatsOutBytes.13.70 = 0
cnpdAllStatsOutBytes.13.71 = 0
cnpdAllStatsOutBytes.13.72 = 0
cnpdAllStatsHCInPkts.13.1 = 0x000000000
cnpdAllStatsHCInPkts.13.2 = 0x000000000
cnpdAllStatsHCInPkts.13.3 = 0x00001a768
cnpdAllStatsHCInPkts.13.4 = 0x000000000
cnpdAllStatsHCInPkts.13.5 = 0x000382cdd
cnpdAllStatsHCInPkts.13.6 = 0x000000000
cnpdAllStatsHCInPkts.13.7 = 0x000000000
cnpdAllStatsHCInPkts.13.8 = 0x000000000
cnpdAllStatsHCInPkts.13.9 = 0x000000000
cnpdAllStatsHCInPkts.13.10 = 0x000000000
cnpdAllStatsHCInPkts.13.11 = 0x000000000
cnpdAllStatsHCInPkts.13.12 = 0x000000000
cnpdAllStatsHCInPkts.13.13 = 0x000000000
cnpdAllStatsHCInPkts.13.14 = 0x000000000
cnpdAllStatsHCInPkts.13.15 = 0x000000000
cnpdAllStatsHCInPkts.13.16 = 0x000000000
cnpdAllStatsHCInPkts.13.17 = 0x000000000
cnpdAllStatsHCInPkts.13.18 = 0x000000000
cnpdAllStatsHCInPkts.13.19 = 0x000000000
cnpdAllStatsHCInPkts.13.20 = 0x000000000
cnpdAllStatsHCInPkts.13.21 = 0x000000000
cnpdAllStatsHCInPkts.13.22 = 0x000000000
cnpdAllStatsHCInPkts.13.23 = 0x000149348
cnpdAllStatsHCInPkts.13.24 = 0x000000000
cnpdAllStatsHCInPkts.13.25 = 0x000000000
cnpdAllStatsHCInPkts.13.26 = 0x000000000
cnpdAllStatsHCInPkts.13.27 = 0x000000000
cnpdAllStatsHCInPkts.13.28 = 0x000000000
cnpdAllStatsHCInPkts.13.29 = 0x000000000
cnpdAllStatsHCInPkts.13.30 = 0x000000000
cnpdAllStatsHCInPkts.13.31 = 0x000000000
cnpdAllStatsHCInPkts.13.32 = 0x000000000
cnpdAllStatsHCInPkts.13.33 = 0x000000000
cnpdAllStatsHCInPkts.13.34 = 0x000000000
cnpdAllStatsHCInPkts.13.35 = 0x000000000
cnpdAllStatsHCInPkts.13.36 = 0x000019e66
cnpdAllStatsHCInPkts.13.37 = 0x000000000
cnpdAllStatsHCInPkts.13.38 = 0x000000000
cnpdAllStatsHCInPkts.13.39 = 0x000000000
cnpdAllStatsHCInPkts.13.40 = 0x000000000
```

```
cnpdAllStatsHCInPkts.13.41 = 0x000004615
cnpdAllStatsHCInPkts.13.42 = 0x000000000
cnpdAllStatsHCInPkts.13.43 = 0x000000000
cnpdAllStatsHCInPkts.13.44 = 0x000000000
cnpdAllStatsHCInPkts.13.45 = 0x000000000
cnpdAllStatsHCInPkts.13.46 = 0x0000894a1
cnpdAllStatsHCInPkts.13.47 = 0x000000000
cnpdAllStatsHCInPkts.13.48 = 0x000013e29
cnpdAllStatsHCInPkts.13.49 = 0x000000000
cnpdAllStatsHCInPkts.13.50 = 0x000000000
cnpdAllStatsHCInPkts.13.51 = 0x00005994a
cnpdAllStatsHCInPkts.13.52 = 0x000000000
cnpdAllStatsHCInPkts.13.53 = 0x000000000
cnpdAllStatsHCInPkts.13.54 = 0x000000000
cnpdAllStatsHCInPkts.13.55 = 0x000000000
cnpdAllStatsHCInPkts.13.56 = 0x000000000
cnpdAllStatsHCInPkts.13.57 = 0x000000000
cnpdAllStatsHCInPkts.13.58 = 0x000000000
cnpdAllStatsHCInPkts.13.59 = 0x0000043a6
cnpdAllStatsHCInPkts.13.60 = 0x000000000
cnpdAllStatsHCInPkts.13.61 = 0x000008d56
cnpdAllStatsHCInPkts.13.62 = 0x000002f66
cnpdAllStatsHCInPkts.13.63 = 0x000000000
cnpdAllStatsHCInPkts.13.64 = 0x000000000
cnpdAllStatsHCInPkts.13.65 = 0x000000000
cnpdAllStatsHCInPkts.13.66 = 0x000000000
cnpdAllStatsHCInPkts.13.67 = 0x000000000
cnpdAllStatsHCInPkts.13.68 = 0x000002356
cnpdAllStatsHCInPkts.13.69 = 0x000000000
cnpdAllStatsHCInPkts.13.70 = 0x000000000
cnpdAllStatsHCInPkts.13.71 = 0x00004c2aa
cnpdAllStatsHCInPkts.13.72 = 0x000006a0e
cnpdAllStatsHCOutPkts.13.1 = 0x000000000
cnpdAllStatsHCOutPkts.13.2 = 0x000000000
cnpdAllStatsHCOutPkts.13.3 = 0x000000000
cnpdAllStatsHCOutPkts.13.4 = 0x000000000
cnpdAllStatsHCOutPkts.13.5 = 0x000000000
cnpdAllStatsHCOutPkts.13.6 = 0x000000000
cnpdAllStatsHCOutPkts.13.7 = 0x000000000
cnpdAllStatsHCOutPkts.13.8 = 0x000000000
cnpdAllStatsHCOutPkts.13.9 = 0x000000000
cnpdAllStatsHCOutPkts.13.10 = 0x000000000
cnpdAllStatsHCOutPkts.13.11 = 0x000000000
cnpdAllStatsHCOutPkts.13.12 = 0x000000000
cnpdAllStatsHCOutPkts.13.13 = 0x000000000
cnpdAllStatsHCOutPkts.13.14 = 0x000000000
cnpdAllStatsHCOutPkts.13.15 = 0x000000000
cnpdAllStatsHCOutPkts.13.16 = 0x000000000
cnpdAllStatsHCOutPkts.13.17 = 0x000000000
cnpdAllStatsHCOutPkts.13.18 = 0x000000000
cnpdAllStatsHCOutPkts.13.19 = 0x000000000
cnpdAllStatsHCOutPkts.13.20 = 0x000000000
cnpdAllStatsHCOutPkts.13.21 = 0x000000000
```

```
cnpdAllStatsHCOutPkts.13.22 = 0x000000000
cnpdAllStatsHCOutPkts.13.23 = 0x000000000
cnpdAllStatsHCOutPkts.13.24 = 0x000000000
cnpdAllStatsHCOutPkts.13.25 = 0x000000000
cnpdAllStatsHCOutPkts.13.26 = 0x000000000
cnpdAllStatsHCOutPkts.13.27 = 0x000000000
cnpdAllStatsHCOutPkts.13.28 = 0x000000000
cnpdAllStatsHCOutPkts.13.29 = 0x000000000
cnpdAllStatsHCOutPkts.13.30 = 0x000000000
cnpdAllStatsHCOutPkts.13.31 = 0x000000000
cnpdAllStatsHCOutPkts.13.32 = 0x000000000
cnpdAllStatsHCOutPkts.13.33 = 0x000000000
cnpdAllStatsHCOutPkts.13.34 = 0x000000000
cnpdAllStatsHCOutPkts.13.35 = 0x000000000
cnpdAllStatsHCOutPkts.13.36 = 0x000000000
cnpdAllStatsHCOutPkts.13.37 = 0x000000000
cnpdAllStatsHCOutPkts.13.38 = 0x000000000
cnpdAllStatsHCOutPkts.13.39 = 0x000000000
cnpdAllStatsHCOutPkts.13.40 = 0x000000000
cnpdAllStatsHCOutPkts.13.41 = 0x000000000
cnpdAllStatsHCOutPkts.13.42 = 0x000000000
cnpdAllStatsHCOutPkts.13.43 = 0x000000000
cnpdAllStatsHCOutPkts.13.44 = 0x000000000
cnpdAllStatsHCOutPkts.13.45 = 0x000000000
cnpdAllStatsHCOutPkts.13.46 = 0x000000000
cnpdAllStatsHCOutPkts.13.47 = 0x000000000
cnpdAllStatsHCOutPkts.13.48 = 0x000000000
cnpdAllStatsHCOutPkts.13.49 = 0x000000000
cnpdAllStatsHCOutPkts.13.50 = 0x000000000
cnpdAllStatsHCOutPkts.13.51 = 0x000000000
cnpdAllStatsHCOutPkts.13.52 = 0x000000000
cnpdAllStatsHCOutPkts.13.53 = 0x000000000
cnpdAllStatsHCOutPkts.13.54 = 0x000000000
cnpdAllStatsHCOutPkts.13.55 = 0x000000000
cnpdAllStatsHCOutPkts.13.56 = 0x000000000
cnpdAllStatsHCOutPkts.13.57 = 0x000000000
cnpdAllStatsHCOutPkts.13.58 = 0x000000000
cnpdAllStatsHCOutPkts.13.59 = 0x000000000
cnpdAllStatsHCOutPkts.13.60 = 0x000000000
cnpdAllStatsHCOutPkts.13.61 = 0x000000000
cnpdAllStatsHCOutPkts.13.62 = 0x000000000
cnpdAllStatsHCOutPkts.13.63 = 0x000000000
cnpdAllStatsHCOutPkts.13.64 = 0x000000000
cnpdAllStatsHCOutPkts.13.65 = 0x000000000
cnpdAllStatsHCOutPkts.13.66 = 0x000000000
cnpdAllStatsHCOutPkts.13.67 = 0x000000000
cnpdAllStatsHCOutPkts.13.68 = 0x000000000
cnpdAllStatsHCOutPkts.13.69 = 0x000000000
cnpdAllStatsHCOutPkts.13.70 = 0x000000000
cnpdAllStatsHCOutPkts.13.71 = 0x000000000
cnpdAllStatsHCOutPkts.13.72 = 0x000000000
cnpdAllStatsHCInBytes.13.1 = 0x000000000
cnpdAllStatsHCInBytes.13.2 = 0x000000000
```

```
cnpdAllStatsHCInBytes.13.3 = 0x0013395f9
cnpdAllStatsHCInBytes.13.4 = 0x000000000
cnpdAllStatsHCInBytes.13.5 = 0x086725280
cnpdAllStatsHCInBytes.13.6 = 0x000000000
cnpdAllStatsHCInBytes.13.7 = 0x000000000
cnpdAllStatsHCInBytes.13.8 = 0x000000000
cnpdAllStatsHCInBytes.13.9 = 0x000000000
cnpdAllStatsHCInBytes.13.10 = 0x000000000
cnpdAllStatsHCInBytes.13.11 = 0x000000000
cnpdAllStatsHCInBytes.13.12 = 0x000000000
cnpdAllStatsHCInBytes.13.13 = 0x000000000
cnpdAllStatsHCInBytes.13.14 = 0x000000000
cnpdAllStatsHCInBytes.13.15 = 0x000000000
cnpdAllStatsHCInBytes.13.16 = 0x000000000
cnpdAllStatsHCInBytes.13.17 = 0x000000000
cnpdAllStatsHCInBytes.13.18 = 0x000000000
cnpdAllStatsHCInBytes.13.19 = 0x000000000
cnpdAllStatsHCInBytes.13.20 = 0x000000000
cnpdAllStatsHCInBytes.13.21 = 0x000000000
cnpdAllStatsHCInBytes.13.22 = 0x000000000
cnpdAllStatsHCInBytes.13.23 = 0x01bb01d54
cnpdAllStatsHCInBytes.13.24 = 0x000000000
cnpdAllStatsHCInBytes.13.25 = 0x000000000
cnpdAllStatsHCInBytes.13.26 = 0x000000000
cnpdAllStatsHCInBytes.13.27 = 0x000000000
cnpdAllStatsHCInBytes.13.28 = 0x000000000
cnpdAllStatsHCInBytes.13.29 = 0x000000000
cnpdAllStatsHCInBytes.13.30 = 0x000000000
cnpdAllStatsHCInBytes.13.31 = 0x000000000
cnpdAllStatsHCInBytes.13.32 = 0x000000000
cnpdAllStatsHCInBytes.13.33 = 0x000000000
cnpdAllStatsHCInBytes.13.34 = 0x000000000
cnpdAllStatsHCInBytes.13.35 = 0x000000000
cnpdAllStatsHCInBytes.13.36 = 0x0009cdcfc
cnpdAllStatsHCInBytes.13.37 = 0x000000000
cnpdAllStatsHCInBytes.13.38 = 0x000000000
cnpdAllStatsHCInBytes.13.39 = 0x000000000
cnpdAllStatsHCInBytes.13.40 = 0x000000000
cnpdAllStatsHCInBytes.13.41 = 0x00013d148
cnpdAllStatsHCInBytes.13.42 = 0x000000000
cnpdAllStatsHCInBytes.13.43 = 0x000000000
cnpdAllStatsHCInBytes.13.44 = 0x000000000
cnpdAllStatsHCInBytes.13.45 = 0x000000000
cnpdAllStatsHCInBytes.13.46 = 0x008e4823b
cnpdAllStatsHCInBytes.13.47 = 0x000000000
cnpdAllStatsHCInBytes.13.48 = 0x000d86339
cnpdAllStatsHCInBytes.13.49 = 0x000000000
cnpdAllStatsHCInBytes.13.50 = 0x000000000
cnpdAllStatsHCInBytes.13.51 = 0x0017ea9a0
cnpdAllStatsHCInBytes.13.52 = 0x000000000
cnpdAllStatsHCInBytes.13.53 = 0x000000000
cnpdAllStatsHCInBytes.13.54 = 0x000000000
cnpdAllStatsHCInBytes.13.55 = 0x000000000
```

```
cnpdAllStatsHCInBytes.13.56 = 0x000000000
cnpdAllStatsHCInBytes.13.57 = 0x000000000
cnpdAllStatsHCInBytes.13.58 = 0x000000000
cnpdAllStatsHCInBytes.13.59 = 0x0001136de
cnpdAllStatsHCInBytes.13.60 = 0x000000000
cnpdAllStatsHCInBytes.13.61 = 0x0004ecedd
cnpdAllStatsHCInBytes.13.62 = 0x000404e8c
cnpdAllStatsHCInBytes.13.63 = 0x000000000
cnpdAllStatsHCInBytes.13.64 = 0x000000000
cnpdAllStatsHCInBytes.13.65 = 0x000000000
cnpdAllStatsHCInBytes.13.66 = 0x000000000
cnpdAllStatsHCInBytes.13.67 = 0x000000000
cnpdAllStatsHCInBytes.13.68 = 0x00009a984
cnpdAllStatsHCInBytes.13.69 = 0x000000000
cnpdAllStatsHCInBytes.13.70 = 0x000000000
cnpdAllStatsHCInBytes.13.71 = 0x001414e08
cnpdAllStatsHCInBytes.13.72 = 0x00023b261
cnpdAllStatsHCOutBytes.13.1 = 0x000000000
cnpdAllStatsHCOutBytes.13.2 = 0x000000000
cnpdAllStatsHCOutBytes.13.3 = 0x000000000
cnpdAllStatsHCOutBytes.13.4 = 0x000000000
cnpdAllStatsHCOutBytes.13.5 = 0x000000000
cnpdAllStatsHCOutBytes.13.6 = 0x000000000
cnpdAllStatsHCOutBytes.13.7 = 0x000000000
cnpdAllStatsHCOutBytes.13.8 = 0x000000000
cnpdAllStatsHCOutBytes.13.9 = 0x000000000
cnpdAllStatsHCOutBytes.13.10 = 0x000000000
cnpdAllStatsHCOutBytes.13.11 = 0x000000000
cnpdAllStatsHCOutBytes.13.12 = 0x000000000
cnpdAllStatsHCOutBytes.13.13 = 0x000000000
cnpdAllStatsHCOutBytes.13.14 = 0x000000000
cnpdAllStatsHCOutBytes.13.15 = 0x000000000
cnpdAllStatsHCOutBytes.13.16 = 0x000000000
cnpdAllStatsHCOutBytes.13.17 = 0x000000000
cnpdAllStatsHCOutBytes.13.18 = 0x000000000
cnpdAllStatsHCOutBytes.13.19 = 0x000000000
cnpdAllStatsHCOutBytes.13.20 = 0x000000000
cnpdAllStatsHCOutBytes.13.21 = 0x000000000
cnpdAllStatsHCOutBytes.13.22 = 0x000000000
cnpdAllStatsHCOutBytes.13.23 = 0x000000000
cnpdAllStatsHCOutBytes.13.24 = 0x000000000
cnpdAllStatsHCOutBytes.13.25 = 0x000000000
cnpdAllStatsHCOutBytes.13.26 = 0x000000000
cnpdAllStatsHCOutBytes.13.27 = 0x000000000
cnpdAllStatsHCOutBytes.13.28 = 0x000000000
cnpdAllStatsHCOutBytes.13.29 = 0x000000000
cnpdAllStatsHCOutBytes.13.30 = 0x000000000
cnpdAllStatsHCOutBytes.13.31 = 0x000000000
cnpdAllStatsHCOutBytes.13.32 = 0x000000000
cnpdAllStatsHCOutBytes.13.33 = 0x000000000
cnpdAllStatsHCOutBytes.13.34 = 0x000000000
cnpdAllStatsHCOutBytes.13.35 = 0x000000000
cnpdAllStatsHCOutBytes.13.36 = 0x000000000
```

```
cnpdAllStatsHCOutBytes.13.37 = 0x000000000
cnpdAllStatsHCOutBytes.13.38 = 0x000000000
cnpdAllStatsHCOutBytes.13.39 = 0x000000000
cnpdAllStatsHCOutBytes.13.40 = 0x000000000
cnpdAllStatsHCOutBytes.13.41 = 0x000000000
cnpdAllStatsHCOutBytes.13.42 = 0x000000000
cnpdAllStatsHCOutBytes.13.43 = 0x000000000
cnpdAllStatsHCOutBytes.13.44 = 0x000000000
cnpdAllStatsHCOutBytes.13.45 = 0x000000000
cnpdAllStatsHCOutBytes.13.46 = 0x000000000
cnpdAllStatsHCOutBytes.13.47 = 0x000000000
cnpdAllStatsHCOutBytes.13.48 = 0x000000000
cnpdAllStatsHCOutBytes.13.49 = 0x000000000
cnpdAllStatsHCOutBytes.13.50 = 0x000000000
cnpdAllStatsHCOutBytes.13.51 = 0x000000000
cnpdAllStatsHCOutBytes.13.52 = 0x000000000
cnpdAllStatsHCOutBytes.13.53 = 0x000000000
cnpdAllStatsHCOutBytes.13.54 = 0x000000000
cnpdAllStatsHCOutBytes.13.55 = 0x000000000
cnpdAllStatsHCOutBytes.13.56 = 0x000000000
cnpdAllStatsHCOutBytes.13.57 = 0x000000000
cnpdAllStatsHCOutBytes.13.58 = 0x000000000
cnpdAllStatsHCOutBytes.13.59 = 0x000000000
cnpdAllStatsHCOutBytes.13.60 = 0x000000000
cnpdAllStatsHCOutBytes.13.61 = 0x000000000
cnpdAllStatsHCOutBytes.13.62 = 0x000000000
cnpdAllStatsHCOutBytes.13.63 = 0x000000000
cnpdAllStatsHCOutBytes.13.64 = 0x000000000
cnpdAllStatsHCOutBytes.13.65 = 0x000000000
cnpdAllStatsHCOutBytes.13.66 = 0x000000000
cnpdAllStatsHCOutBytes.13.67 = 0x000000000
cnpdAllStatsHCOutBytes.13.68 = 0x000000000
cnpdAllStatsHCOutBytes.13.69 = 0x000000000
cnpdAllStatsHCOutBytes.13.70 = 0x000000000
cnpdAllStatsHCOutBytes.13.71 = 0x000000000
cnpdAllStatsHCOutBytes.13.72 = 0x000000000
cnpdAllStatsInBitRate.13.1 = 0
cnpdAllStatsInBitRate.13.2 = 0
cnpdAllStatsInBitRate.13.3 = 0
cnpdAllStatsInBitRate.13.4 = 0
cnpdAllStatsInBitRate.13.5 = 0
cnpdAllStatsInBitRate.13.6 = 0
cnpdAllStatsInBitRate.13.7 = 0
cnpdAllStatsInBitRate.13.8 = 0
cnpdAllStatsInBitRate.13.9 = 0
cnpdAllStatsInBitRate.13.10 = 0
cnpdAllStatsInBitRate.13.11 = 0
cnpdAllStatsInBitRate.13.12 = 0
cnpdAllStatsInBitRate.13.13 = 0
cnpdAllStatsInBitRate.13.14 = 0
cnpdAllStatsInBitRate.13.15 = 0
cnpdAllStatsInBitRate.13.16 = 0
cnpdAllStatsInBitRate.13.17 = 0
```

```
cnpdAllStatsInBitRate.13.18 = 0
cnpdAllStatsInBitRate.13.19 = 0
cnpdAllStatsInBitRate.13.20 = 0
cnpdAllStatsInBitRate.13.21 = 0
cnpdAllStatsInBitRate.13.22 = 0
cnpdAllStatsInBitRate.13.23 = 0
cnpdAllStatsInBitRate.13.24 = 0
cnpdAllStatsInBitRate.13.25 = 0
cnpdAllStatsInBitRate.13.26 = 0
cnpdAllStatsInBitRate.13.27 = 0
cnpdAllStatsInBitRate.13.28 = 0
cnpdAllStatsInBitRate.13.29 = 0
cnpdAllStatsInBitRate.13.30 = 0
cnpdAllStatsInBitRate.13.31 = 0
cnpdAllStatsInBitRate.13.32 = 0
cnpdAllStatsInBitRate.13.33 = 0
cnpdAllStatsInBitRate.13.34 = 0
cnpdAllStatsInBitRate.13.35 = 0
cnpdAllStatsInBitRate.13.36 = 0
cnpdAllStatsInBitRate.13.37 = 0
cnpdAllStatsInBitRate.13.38 = 0
cnpdAllStatsInBitRate.13.39 = 0
cnpdAllStatsInBitRate.13.40 = 0
cnpdAllStatsInBitRate.13.41 = 0
cnpdAllStatsInBitRate.13.42 = 0
cnpdAllStatsInBitRate.13.43 = 0
cnpdAllStatsInBitRate.13.44 = 0
cnpdAllStatsInBitRate.13.45 = 0
cnpdAllStatsInBitRate.13.46 = 0
cnpdAllStatsInBitRate.13.47 = 0
cnpdAllStatsInBitRate.13.48 = 0
cnpdAllStatsInBitRate.13.49 = 0
cnpdAllStatsInBitRate.13.50 = 0
cnpdAllStatsInBitRate.13.51 = 0
cnpdAllStatsInBitRate.13.52 = 0
cnpdAllStatsInBitRate.13.53 = 0
cnpdAllStatsInBitRate.13.54 = 0
cnpdAllStatsInBitRate.13.55 = 0
cnpdAllStatsInBitRate.13.56 = 0
cnpdAllStatsInBitRate.13.57 = 0
cnpdAllStatsInBitRate.13.58 = 0
cnpdAllStatsInBitRate.13.59 = 0
cnpdAllStatsInBitRate.13.60 = 0
cnpdAllStatsInBitRate.13.61 = 0
cnpdAllStatsInBitRate.13.62 = 0
cnpdAllStatsInBitRate.13.63 = 0
cnpdAllStatsInBitRate.13.64 = 0
cnpdAllStatsInBitRate.13.65 = 0
cnpdAllStatsInBitRate.13.66 = 0
cnpdAllStatsInBitRate.13.67 = 0
cnpdAllStatsInBitRate.13.68 = 0
cnpdAllStatsInBitRate.13.69 = 0
cnpdAllStatsInBitRate.13.70 = 0
```

```
cnpdAllStatsInBitRate.13.71 = 0
cnpdAllStatsInBitRate.13.72 = 0
cnpdAllStatsOutBitRate.13.1 = 0
cnpdAllStatsOutBitRate.13.2 = 0
cnpdAllStatsOutBitRate.13.3 = 0
cnpdAllStatsOutBitRate.13.4 = 0
cnpdAllStatsOutBitRate.13.5 = 0
cnpdAllStatsOutBitRate.13.6 = 0
cnpdAllStatsOutBitRate.13.7 = 0
cnpdAllStatsOutBitRate.13.8 = 0
cnpdAllStatsOutBitRate.13.9 = 0
cnpdAllStatsOutBitRate.13.10 = 0
cnpdAllStatsOutBitRate.13.11 = 0
cnpdAllStatsOutBitRate.13.12 = 0
cnpdAllStatsOutBitRate.13.13 = 0
cnpdAllStatsOutBitRate.13.14 = 0
cnpdAllStatsOutBitRate.13.15 = 0
cnpdAllStatsOutBitRate.13.16 = 0
cnpdAllStatsOutBitRate.13.17 = 0
cnpdAllStatsOutBitRate.13.18 = 0
cnpdAllStatsOutBitRate.13.19 = 0
cnpdAllStatsOutBitRate.13.20 = 0
cnpdAllStatsOutBitRate.13.21 = 0
cnpdAllStatsOutBitRate.13.22 = 0
cnpdAllStatsOutBitRate.13.23 = 0
cnpdAllStatsOutBitRate.13.24 = 0
cnpdAllStatsOutBitRate.13.25 = 0
cnpdAllStatsOutBitRate.13.26 = 0
cnpdAllStatsOutBitRate.13.27 = 0
cnpdAllStatsOutBitRate.13.28 = 0
cnpdAllStatsOutBitRate.13.29 = 0
cnpdAllStatsOutBitRate.13.30 = 0
cnpdAllStatsOutBitRate.13.31 = 0
cnpdAllStatsOutBitRate.13.32 = 0
cnpdAllStatsOutBitRate.13.33 = 0
cnpdAllStatsOutBitRate.13.34 = 0
cnpdAllStatsOutBitRate.13.35 = 0
cnpdAllStatsOutBitRate.13.36 = 0
cnpdAllStatsOutBitRate.13.37 = 0
cnpdAllStatsOutBitRate.13.38 = 0
cnpdAllStatsOutBitRate.13.39 = 0
cnpdAllStatsOutBitRate.13.40 = 0
cnpdAllStatsOutBitRate.13.41 = 0
cnpdAllStatsOutBitRate.13.42 = 0
cnpdAllStatsOutBitRate.13.43 = 0
cnpdAllStatsOutBitRate.13.44 = 0
cnpdAllStatsOutBitRate.13.45 = 0
cnpdAllStatsOutBitRate.13.46 = 0
cnpdAllStatsOutBitRate.13.47 = 0
cnpdAllStatsOutBitRate.13.48 = 0
cnpdAllStatsOutBitRate.13.49 = 0
cnpdAllStatsOutBitRate.13.50 = 0
cnpdAllStatsOutBitRate.13.51 = 0
```

```
cnpdAllStatsOutBitRate.13.52 = 0
cnpdAllStatsOutBitRate.13.53 = 0
cnpdAllStatsOutBitRate.13.54 = 0
cnpdAllStatsOutBitRate.13.55 = 0
cnpdAllStatsOutBitRate.13.56 = 0
cnpdAllStatsOutBitRate.13.57 = 0
cnpdAllStatsOutBitRate.13.58 = 0
cnpdAllStatsOutBitRate.13.59 = 0
cnpdAllStatsOutBitRate.13.60 = 0
cnpdAllStatsOutBitRate.13.61 = 0
cnpdAllStatsOutBitRate.13.62 = 0
cnpdAllStatsOutBitRate.13.63 = 0
cnpdAllStatsOutBitRate.13.64 = 0
cnpdAllStatsOutBitRate.13.65 = 0
cnpdAllStatsOutBitRate.13.66 = 0
cnpdAllStatsOutBitRate.13.67 = 0
cnpdAllStatsOutBitRate.13.68 = 0
cnpdAllStatsOutBitRate.13.69 = 0
cnpdAllStatsOutBitRate.13.70 = 0
cnpdAllStatsOutBitRate.13.71 = 0
cnpdAllStatsOutBitRate.13.72 = 0
```

Creating Top-N Tables Using the NBAR Protocol Discovery MIB

The top-n functionality in the NBAR Protocol Discovery MIB involves two tables. The first table is a configuration table in which each row represents a group of objects that will create a unique top-n report. The other table is a Statistics Results Table that records the outputs of each row of the configuration table. The Statistics ResultsTable can be regenerated by the correct use of the rowStatus object in the configuration table.

Create a new TopNConfig Entry

```
>$ setany -v2c a.b.c.d public cnpdTopNConfigIfIndex.1 -i 13
cnpdTopNConfigIfIndex.1 = 13

>$ getmany -v2c  a.b.c.d public cnpdTopNConfig


cnpdTopNConfigIfIndex.1 = 13
cnpdTopNConfigStatsSelect.1 = byteCountSum(6)
cnpdTopNConfigSampleTime.1 = 0
cnpdTopNConfigRequestedSize.1 = 0
cnpdTopNConfigGrantedSize.1 = 0
cnpdTopNConfigTime.1 = 0
cnpdTopNConfigStatus.1 = notReady(3)
```

Change StatsSelect

```
>$ setany -v2c a.b.c.d public cnpdTopNConfigStatsSelect.1 -i 7

cnpdTopNConfigStatsSelect.1 = packetCountIn(7)


>$ getmany -v2c  a.b.c.d public cnpdTopNConfig

cnpdTopNConfigIfIndex.1 = 13
cnpdTopNConfigStatsSelect.1 = packetCountIn(7)
cnpdTopNConfigSampleTime.1 = 0
cnpdTopNConfigRequestedSize.1 = 0
cnpdTopNConfigGrantedSize.1 = 0
cnpdTopNConfigTime.1 = 0
cnpdTopNConfigStatus.1 = notReady(3)
```

Change SampleTime

```
>$ setany -v2c a.b.c.d public cnpdTopNConfigSampleTime.1 -g 13

cnpdTopNConfigSampleTime.1 = 13


>$ getmany -v2c  a.b.c.d public cnpdTopNConfig

cnpdTopNConfigIfIndex.1 = 13
cnpdTopNConfigStatsSelect.1 = byteCountOut(5)
cnpdTopNConfigSampleTime.1 = 13
cnpdTopNConfigRequestedSize.1 = 0
cnpdTopNConfigGrantedSize.1 = 0
cnpdTopNConfigTime.1 = 0
cnpdTopNConfigStatus.1 = notReady(3)
```

Change RequestedSize

```
>$ setany -v2c a.b.c.d public cnpdTopNConfigRequestedSize.1 -g 5

cnpdTopNConfigRequestedSize.1 = 5

>$ getmany -v2c  a.b.c.d public cnpdTopNConfig

cnpdTopNConfigIfIndex.1 = 13
cnpdTopNConfigStatsSelect.1 = byteCountOut(5)
cnpdTopNConfigSampleTime.1 = 13
cnpdTopNConfigRequestedSize.1 = 5
cnpdTopNConfigGrantedSize.1 = 5
cnpdTopNConfigTime.1 = 0
cnpdTopNConfigStatus.1 = notReady(3)
```

Set Status to createAndGo and Display Results

```
>$ setany -v2c a.b.c.d public cnpdTopNConfigStatus.1 -i 4

cnpdTopNConfigStatus.1 = createAndGo(4)

>$ getmany -v2c  a.b.c.d public cnpdTopNConfig

cnpdTopNConfigIfIndex.1 = 13
cnpdTopNConfigStatsSelect.1 = packetCountIn(7)
cnpdTopNConfigSampleTime.1 = 13
cnpdTopNConfigRequestedSize.1 = 5
cnpdTopNConfigGrantedSize.1 = 5
cnpdTopNConfigTime.1 = 44093748
cnpdTopNConfigStatus.1 = active(1)

>$ getmany -v2c  a.b.c.d public cnpdTopNStats

cnpdTopNStatsProtocolName.1.1 = http
cnpdTopNStatsProtocolName.1.2 = realaudio
cnpdTopNStatsProtocolName.1.3 = nntp
cnpdTopNStatsProtocolName.1.4 = ldap
cnpdTopNStatsProtocolName.1.5 = ftp
cnpdTopNStatsRate.1.1 = 3681501
cnpdTopNStatsRate.1.2 = 1348424
cnpdTopNStatsRate.1.3 = 562337
cnpdTopNStatsRate.1.4 = 366922
cnpdTopNStatsRate.1.5 = 311978
cnpdTopNStatsHCRate.1.1 = 0x000382cdd
cnpdTopNStatsHCRate.1.2 = 0x000149348
```

```
cnpdTopNStatsHCRate.1.3 = 0x0000894a1
cnpdTopNStatsHCRate.1.4 = 0x00005994a
cnpdTopNStatsHCRate.1.5 = 0x00004c2aa
```

Configuring Thresholds

The NBAR Protocol Discovery MIB can be used to set two types of thresholds—a threshold that reports a breach when thresholds for an individual protocol are crossed and a threshold that reports a breach when all NBAR-classifiable protocols or applications are crossed.

Specific Protocol Threshold

A specific protocol threshold is a threshold that monitors the traffic of a specific protocol. An SNMP trap is sent when this threshold is crossed.

The following items should be noted when configuring a specific protocol threshold:

- The cnpdThresholdConfigProtocolAny must be set to FALSE(2) and cnpdThresholdConfigProtocol must be set to a value that indicates the OID of the protocol being monitored. The OID for each protocol can be seen by querying cnpdSupportedProtocolsTable (for instance, use 5 to monitor HTTP).
- Use cnpdThresholdConfigIfIndex to select the interface and cnpdThresholdConfigRising and cnpdThresholdConfigFalling to set the rising and falling thresholds.
- Set the cnpdThresholdConfigInterval to configure the frequency with which thresholds should be checked.
- Set cnpdThresholdConfigStatus to 4 (createAndGo) and this config member will become active.

Example:

```
setany -v2c a.b.c.d public cnpdThresholdConfigProtocolAny.2 -i 2
setany -v2c a.b.c.d public cnpdThresholdConfigProtocol.2 -g 5
setany -v2c a.b.c.d public cnpdThresholdConfigStatsSelect.2 -i 7
setany -v2c a.b.c.d public cnpdThresholdConfigIfIndex.2 -i 13
setany -v2c a.b.c.d public cnpdThresholdConfigRising.2 -g 2000
setany -v2c a.b.c.d public cnpdThresholdConfigFalling.2 -g 1000
setany -v2c a.b.c.d public cnpdThresholdConfigStatus.2 -i 4
```

Any Protocol Threshold

The Any Protocol Threshold setting is used to report breaches and create history entries if ANY protocol crosses a rising or falling threshold. This threshold does have a mechanism to report a breach only once (in a given time period). For instance, if HTTP breaches a rising threshold, a trap is sent only for the first breach. If another protocol, such as DHCP, breaches a rising threshold, a separate breach is reported.

The setup for any protocol is identical to a specific protocol except cnpdThresholdConfigProtocolAny does not have to be configured and it is unnecessary to specify a cnpdThresholdConfigProtocol.

Example:

```
setany -v2c a.b.c.d public cnpdThresholdConfigStatsSelect.1 -i 7
setany -v2c a.b.c.d public cnpdThresholdConfigIfIndex.1 -i 13
setany -v2c a.b.c.d public cnpdThresholdConfigRising.1 -g 30
setany -v2c a.b.c.d public cnpdThresholdConfigFalling.1 -g 20
setany -v2c a.b.c.d public cnpdThresholdConfigStatus.1 -i 4
```

Threshold Options

The cnpdThresholdConfigSampleType is used to change the way a threshold value is sampled and can be set to either of the following values:

- absoluteValue(1)
- deltaValue(2)

When the cnpdThresholdConfigSampleType is set to absoluteValue, the sampled statistic is compared to cnpdThresholdConfigRising and cnpdThresholdConfigFalling and a trap is sent if either of these thresholds are crossed. A ThresholdHistory event is also registered when either of these thresholds are crossed.

Note Most Protocol Discovery statistics are aggregates and only rise. Therefore, only one threshold is likely to be breached.

When the cnpdThresholdConfigSampleType is set to deltaValue, the difference between the current sample and the previous sample is compared to cnpdThresholdConfigRising and cnpdThresholdConfigFalling. A ThresholdHistory event is registered only if this difference is greater or less than the compared values. This deltaValue therefore measures the gradient of statistic change is the value being tested and can lead to several breach events and traps.

Examples:

Any Protocol

Note The cnpdThresholdConfigProtocolAny object is set to TRUE by default so it does not need to set the value manually.

```
>$ setany -v2c a.b.c.d public cnpdThresholdConfigStatsSelect.1 -i 7

cnpdThresholdConfigStatsSelect.1 = packetCountIn(7)

>$ setany -v2c a.b.c.d public cnpdThresholdConfigIfIndex.1 -i 13

cnpdThresholdConfigIfIndex.1 = 13

>$ setany -v2c a.b.c.d public cnpdThresholdConfigRising.1 -g 30

cnpdThresholdConfigRising.1 = 30

>$ setany -v2c a.b.c.d public cnpdThresholdConfigFalling.1 -g 20

cnpdThresholdConfigFalling.1 = 20

>$ setany -v2c a.b.c.d public cnpdThresholdConfigStatus.1 -i 4

cnpdThresholdConfigStatus.1 = createAndGo(4)

>$ getmany -v2c  a.b.c.d public cnpdThresholdConfig

cnpdThresholdConfigIfIndex.1 = 13
cnpdThresholdConfigInterval.1 = 10
cnpdThresholdConfigSampleType.1 = absoluteValue(1)
cnpdThresholdConfigProtocol.1 = 62
cnpdThresholdConfigProtocolAny.1 = true(1)
cnpdThresholdConfigStatsSelect.1 = packetCountIn(7)
cnpdThresholdConfigStartup.1 = risingOrFalling(3)
cnpdThresholdConfigRising.1 = 30
cnpdThresholdConfigFalling.1 = 20
cnpdThresholdConfigStatus.1 = active(1)

>$ getmany -v2c  a.b.c.d public cnpdThresholdHistory

cnpdThresholdHistoryConfigIndex.1 = 1
cnpdThresholdHistoryConfigIndex.2 = 1
cnpdThresholdHistoryConfigIndex.3 = 1
cnpdThresholdHistoryConfigIndex.4 = 1
cnpdThresholdHistoryConfigIndex.5 = 1
cnpdThresholdHistoryConfigIndex.6 = 1
cnpdThresholdHistoryConfigIndex.7 = 1
cnpdThresholdHistoryConfigIndex.8 = 1
cnpdThresholdHistoryConfigIndex.9 = 1
cnpdThresholdHistoryConfigIndex.10 = 1
cnpdThresholdHistoryConfigIndex.11 = 1
cnpdThresholdHistoryConfigIndex.12 = 1
cnpdThresholdHistoryConfigIndex.13 = 1
cnpdThresholdHistoryConfigIndex.14 = 1
cnpdThresholdHistoryValue.1 = 73906
cnpdThresholdHistoryValue.2 = 26985
cnpdThresholdHistoryValue.3 = 11250
```

```
cnpdThresholdHistoryValue.4 = 7380
cnpdThresholdHistoryValue.5 = 6552
cnpdThresholdHistoryValue.6 = 2184
cnpdThresholdHistoryValue.7 = 2160
cnpdThresholdHistoryValue.8 = 1620
cnpdThresholdHistoryValue.9 = 724
cnpdThresholdHistoryValue.10 = 364
cnpdThresholdHistoryValue.11 = 360
cnpdThresholdHistoryValue.12 = 181
cnpdThresholdHistoryValue.13 = 2
cnpdThresholdHistoryValue.14 = 31
cnpdThresholdHistoryType.1 = risingBreach(1)
cnpdThresholdHistoryType.2 = risingBreach(1)
cnpdThresholdHistoryType.3 = risingBreach(1)
cnpdThresholdHistoryType.4 = risingBreach(1)
cnpdThresholdHistoryType.5 = risingBreach(1)
cnpdThresholdHistoryType.6 = risingBreach(1)
cnpdThresholdHistoryType.7 = risingBreach(1)
cnpdThresholdHistoryType.8 = risingBreach(1)
cnpdThresholdHistoryType.9 = risingBreach(1)
cnpdThresholdHistoryType.10 = risingBreach(1)
cnpdThresholdHistoryType.11 = risingBreach(1)
cnpdThresholdHistoryType.12 = risingBreach(1)
cnpdThresholdHistoryType.13 = fallingBreach(2)
cnpdThresholdHistoryType.14 = risingBreach(1)
cnpdThresholdHistoryTime.1 = 8579
cnpdThresholdHistoryTime.2 = 8579
cnpdThresholdHistoryTime.3 = 8579
cnpdThresholdHistoryTime.4 = 8579
cnpdThresholdHistoryTime.5 = 8579
cnpdThresholdHistoryTime.6 = 8579
cnpdThresholdHistoryTime.7 = 8579
cnpdThresholdHistoryTime.8 = 8579
cnpdThresholdHistoryTime.9 = 8579
cnpdThresholdHistoryTime.10 = 8579
cnpdThresholdHistoryTime.11 = 8579
cnpdThresholdHistoryTime.12 = 8579
cnpdThresholdHistoryTime.13 = 8579
cnpdThresholdHistoryTime.14 = 97579
cnpdThresholdHistoryProtocol.1 = 5
cnpdThresholdHistoryProtocol.2 = 23
cnpdThresholdHistoryProtocol.3 = 46
cnpdThresholdHistoryProtocol.4 = 51
cnpdThresholdHistoryProtocol.5 = 71
cnpdThresholdHistoryProtocol.6 = 36
cnpdThresholdHistoryProtocol.7 = 3
cnpdThresholdHistoryProtocol.8 = 48
cnpdThresholdHistoryProtocol.9 = 61
cnpdThresholdHistoryProtocol.10 = 59
cnpdThresholdHistoryProtocol.11 = 41
cnpdThresholdHistoryProtocol.12 = 68
cnpdThresholdHistoryProtocol.13 = 62
cnpdThresholdHistoryProtocol.14 = 62
```

```
cnpdThresholdHistoryStatsSelect.1 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.2 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.4 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.5 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.6 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.7 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.8 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.9 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.10 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.11 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.12 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.13 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.14 = packetCountIn(7)
```

SpecificProtocol (HTTP)

```
>$ setany -v2c a.b.c.d public cnpdThresholdConfigProtocolAny.2 -i 2

cnpdThresholdConfigProtocolAny.2 = false(2)

>$ setany -v2c a.b.c.d public cnpdThresholdConfigProtocol.2 -g 5

cnpdThresholdConfigProtocol.2 = 5

>$ setany -v2c a.b.c.d public cnpdThresholdConfigStatsSelect.2 -i 7

cnpdThresholdConfigStatsSelect.2 = packetCountIn(7)

>$ setany -v2c a.b.c.d public cnpdThresholdConfigIfIndex.2 -i 13

cnpdThresholdConfigIfIndex.2 = 13

>$ setany -v2c a.b.c.d public cnpdThresholdConfigRising.2 -g 2000

cnpdThresholdConfigRising.2 = 2000

>$ setany -v2c a.b.c.d public cnpdThresholdConfigFalling.2 -g 1000

cnpdThresholdConfigFalling.2 = 1000

>$ setany -v2c a.b.c.d public cnpdThresholdConfigStatus.2 -i 4

cnpdThresholdConfigStatus.2 = createAndGo(4)

>$ getmany -v2c  a.b.c.d public cnpdThresholdConfig

cnpdThresholdConfigIfIndex.1 = 13
cnpdThresholdConfigIfIndex.2 = 13
cnpdThresholdConfigInterval.1 = 10
```

```
cnpdThresholdConfigInterval.2 = 10
cnpdThresholdConfigSampleType.1 = absoluteValue(1)
cnpdThresholdConfigSampleType.2 = absoluteValue(1)
cnpdThresholdConfigProtocol.1 = 62
cnpdThresholdConfigProtocol.2 = 5
cnpdThresholdConfigProtocolAny.1 = true(1)
cnpdThresholdConfigProtocolAny.2 = false(2)
cnpdThresholdConfigStatsSelect.1 = packetCountIn(7)
cnpdThresholdConfigStatsSelect.2 = packetCountIn(7)
cnpdThresholdConfigStartup.1 = risingOrFalling(3)
cnpdThresholdConfigStartup.2 = risingOrFalling(3)
cnpdThresholdConfigRising.1 = 30
cnpdThresholdConfigRising.2 = 2000
cnpdThresholdConfigFalling.1 = 20
cnpdThresholdConfigFalling.2 = 1000
cnpdThresholdConfigStatus.1 = active(1)
cnpdThresholdConfigStatus.2 = active(1)

>$ getmany -v2c  a.b.c.d public cnpdThresholdHistory

cnpdThresholdHistoryConfigIndex.1 = 1
cnpdThresholdHistoryConfigIndex.2 = 1
cnpdThresholdHistoryConfigIndex.3 = 1
cnpdThresholdHistoryConfigIndex.4 = 1
cnpdThresholdHistoryConfigIndex.5 = 1
cnpdThresholdHistoryConfigIndex.6 = 1
cnpdThresholdHistoryConfigIndex.7 = 1
cnpdThresholdHistoryConfigIndex.8 = 1
cnpdThresholdHistoryConfigIndex.9 = 1
cnpdThresholdHistoryConfigIndex.10 = 1
cnpdThresholdHistoryConfigIndex.11 = 1
cnpdThresholdHistoryConfigIndex.12 = 1
cnpdThresholdHistoryConfigIndex.13 = 1
cnpdThresholdHistoryConfigIndex.14 = 1
cnpdThresholdHistoryConfigIndex.15 = 2
cnpdThresholdHistoryValue.1 = 73906
cnpdThresholdHistoryValue.2 = 26985
cnpdThresholdHistoryValue.3 = 11250
cnpdThresholdHistoryValue.4 = 7380
cnpdThresholdHistoryValue.5 = 6552
cnpdThresholdHistoryValue.6 = 2184
cnpdThresholdHistoryValue.7 = 2160
cnpdThresholdHistoryValue.8 = 1620
cnpdThresholdHistoryValue.9 = 724
cnpdThresholdHistoryValue.10 = 364
cnpdThresholdHistoryValue.11 = 360
cnpdThresholdHistoryValue.12 = 181
cnpdThresholdHistoryValue.13 = 2
cnpdThresholdHistoryValue.14 = 31
cnpdThresholdHistoryValue.15 = 3681501
cnpdThresholdHistoryType.1 = risingBreach(1)
cnpdThresholdHistoryType.2 = risingBreach(1)
cnpdThresholdHistoryType.3 = risingBreach(1)
```

```
cnpdThresholdHistoryType.4 = risingBreach(1)
cnpdThresholdHistoryType.5 = risingBreach(1)
cnpdThresholdHistoryType.6 = risingBreach(1)
cnpdThresholdHistoryType.7 = risingBreach(1)
cnpdThresholdHistoryType.8 = risingBreach(1)
cnpdThresholdHistoryType.9 = risingBreach(1)
cnpdThresholdHistoryType.10 = risingBreach(1)
cnpdThresholdHistoryType.11 = risingBreach(1)
cnpdThresholdHistoryType.12 = risingBreach(1)
cnpdThresholdHistoryType.13 = fallingBreach(2)
cnpdThresholdHistoryType.14 = risingBreach(1)
cnpdThresholdHistoryType.15 = risingBreach(1)
cnpdThresholdHistoryTime.1 = 8579
cnpdThresholdHistoryTime.2 = 8579
cnpdThresholdHistoryTime.3 = 8579
cnpdThresholdHistoryTime.4 = 8579
cnpdThresholdHistoryTime.5 = 8579
cnpdThresholdHistoryTime.6 = 8579
cnpdThresholdHistoryTime.7 = 8579
cnpdThresholdHistoryTime.8 = 8579
cnpdThresholdHistoryTime.9 = 8579
cnpdThresholdHistoryTime.10 = 8579
cnpdThresholdHistoryTime.11 = 8579
cnpdThresholdHistoryTime.12 = 8579
cnpdThresholdHistoryTime.13 = 8579
cnpdThresholdHistoryTime.14 = 97579
cnpdThresholdHistoryTime.15 = 44261817
cnpdThresholdHistoryProtocol.1 = 5
cnpdThresholdHistoryProtocol.2 = 23
cnpdThresholdHistoryProtocol.3 = 46
cnpdThresholdHistoryProtocol.4 = 51
cnpdThresholdHistoryProtocol.5 = 71
cnpdThresholdHistoryProtocol.6 = 36
cnpdThresholdHistoryProtocol.7 = 3
cnpdThresholdHistoryProtocol.8 = 48
cnpdThresholdHistoryProtocol.9 = 61
cnpdThresholdHistoryProtocol.10 = 59
cnpdThresholdHistoryProtocol.11 = 41
cnpdThresholdHistoryProtocol.12 = 68
cnpdThresholdHistoryProtocol.13 = 62
cnpdThresholdHistoryProtocol.14 = 62
cnpdThresholdHistoryProtocol.15 = 5
cnpdThresholdHistoryStatsSelect.1 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.2 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.4 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.5 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.6 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.7 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.8 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.9 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.10 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.11 = packetCountIn(7)
```

cnpdThresholdHistoryStatsSelect.12 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.13 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.14 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.15 = packetCountIn(7)

Note The traps shown below are sample traps that could be generated by traffic patterns that breach the above configuration. The traps are shown only for the sake of the example.

Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 1
Time Ticks: 44261817
cnpdThresholdConfigIfIndex.2 = 13
cnpdThresholdConfigStatsSelect.2 = packetCountIn(7)
cnpdThresholdHistoryValue.2 = 3681501
cnpdThresholdConfigRising.2 = 2000
cnpdThresholdHistoryProtocol.2 = 5
cnpdThresholdHistoryTime.2 = 44261817

Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 1
Time Ticks: 44278817
cnpdThresholdConfigIfIndex.2 = 13
cnpdThresholdConfigStatsSelect.2 = packetCountIn(7)
cnpdThresholdHistoryValue.2 = 3681501
cnpdThresholdConfigRising.2 = 2000
cnpdThresholdHistoryProtocol.2 = 5
cnpdThresholdHistoryTime.2 = 44278817

Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 1
Time Ticks: 44284817
cnpdThresholdConfigIfIndex.2 = 13
cnpdThresholdConfigStatsSelect.2 = packetCountIn(7)
cnpdThresholdHistoryValue.2 = 3681501

```
cnpdThresholdConfigRising.2 = 2000
cnpdThresholdHistoryProtocol.2 = 5
cnpdThresholdHistoryTime.2 = 44284817
```

AnyProtocol=TRUE and SampleType=Delta

```
>$ setany -v2c a.b.c.d public cnpdThresholdConfigStatsSelect.3 -i 7

cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)

>$ setany -v2c a.b.c.d public cnpdThresholdConfigIfIndex.3 -i 13

cnpdThresholdConfigIfIndex.3 = 13

>$ setany -v2c a.b.c.d public cnpdThresholdConfigRising.3 -g 30

cnpdThresholdConfigRising.3 = 30

>$ setany -v2c a.b.c.d public cnpdThresholdConfigFalling.3 -g 20

cnpdThresholdConfigFalling.3 = 20

>$ setany -v2c a.b.c.d public cnpdThresholdConfigSampleType.3 -i 2

cnpdThresholdConfigSampleType.3 = deltaValue(2)

>$ setany -v2c a.b.c.d public cnpdThresholdConfigStatus.3 -i 4

cnpdThresholdConfigStatus.3 = createAndGo(4)

>$ getmany -v2c  a.b.c.d public cnpdThresholdConfig

cnpdThresholdConfigIfIndex.1 = 13
cnpdThresholdConfigIfIndex.2 = 13
cnpdThresholdConfigIfIndex.3 = 13
cnpdThresholdConfigInterval.1 = 10
cnpdThresholdConfigInterval.2 = 10
cnpdThresholdConfigInterval.3 = 10
cnpdThresholdConfigSampleType.1 = absoluteValue(1)
cnpdThresholdConfigSampleType.2 = absoluteValue(1)
cnpdThresholdConfigSampleType.3 = deltaValue(2)
cnpdThresholdConfigProtocol.1 = 62
cnpdThresholdConfigProtocol.2 = 5
cnpdThresholdConfigProtocol.3 = 62
cnpdThresholdConfigProtocolAny.1 = true(1)
cnpdThresholdConfigProtocolAny.2 = false(2)
cnpdThresholdConfigProtocolAny.3 = true(1)
cnpdThresholdConfigStatsSelect.1 = packetCountIn(7)
cnpdThresholdConfigStatsSelect.2 = packetCountIn(7)
cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)
cnpdThresholdConfigStartup.1 = risingOrFalling(3)
```

```
cnpdThresholdConfigStartup.2 = risingOrFalling(3)
cnpdThresholdConfigStartup.3 = risingOrFalling(3)
cnpdThresholdConfigRising.1 = 30
cnpdThresholdConfigRising.2 = 2000
cnpdThresholdConfigRising.3 = 30
cnpdThresholdConfigFalling.1 = 20
cnpdThresholdConfigFalling.2 = 1000
cnpdThresholdConfigFalling.3 = 20
cnpdThresholdConfigStatus.1 = active(1)
cnpdThresholdConfigStatus.2 = active(1)
cnpdThresholdConfigStatus.3 = active(1)

>$ getmany -v2c  a.b.c.d public cnpdThresholdHistory

cnpdThresholdHistoryConfigIndex.1 = 1
cnpdThresholdHistoryConfigIndex.2 = 1
cnpdThresholdHistoryConfigIndex.3 = 1
cnpdThresholdHistoryConfigIndex.4 = 1
cnpdThresholdHistoryConfigIndex.5 = 1
cnpdThresholdHistoryConfigIndex.6 = 1
cnpdThresholdHistoryConfigIndex.7 = 1
cnpdThresholdHistoryConfigIndex.8 = 1
cnpdThresholdHistoryConfigIndex.9 = 1
cnpdThresholdHistoryConfigIndex.10 = 1
cnpdThresholdHistoryConfigIndex.11 = 1
cnpdThresholdHistoryConfigIndex.12 = 1
cnpdThresholdHistoryConfigIndex.13 = 1
cnpdThresholdHistoryConfigIndex.14 = 1
cnpdThresholdHistoryConfigIndex.15 = 2
cnpdThresholdHistoryConfigIndex.16 = 2
cnpdThresholdHistoryConfigIndex.17 = 2
cnpdThresholdHistoryConfigIndex.18 = 3
cnpdThresholdHistoryConfigIndex.19 = 3
cnpdThresholdHistoryConfigIndex.20 = 3
cnpdThresholdHistoryConfigIndex.21 = 3
cnpdThresholdHistoryConfigIndex.22 = 3
cnpdThresholdHistoryConfigIndex.23 = 3
cnpdThresholdHistoryConfigIndex.24 = 3
cnpdThresholdHistoryConfigIndex.25 = 3
cnpdThresholdHistoryConfigIndex.26 = 3
cnpdThresholdHistoryConfigIndex.27 = 3
cnpdThresholdHistoryConfigIndex.28 = 3
cnpdThresholdHistoryConfigIndex.29 = 3
cnpdThresholdHistoryConfigIndex.30 = 3
cnpdThresholdHistoryConfigIndex.31 = 3
cnpdThresholdHistoryValue.1 = 73906
cnpdThresholdHistoryValue.2 = 26985
cnpdThresholdHistoryValue.3 = 11250
cnpdThresholdHistoryValue.4 = 7380
cnpdThresholdHistoryValue.5 = 6552
cnpdThresholdHistoryValue.6 = 2184
cnpdThresholdHistoryValue.7 = 2160
cnpdThresholdHistoryValue.8 = 1620
```

```
cnpdThresholdHistoryValue.9 = 724
cnpdThresholdHistoryValue.10 = 364
cnpdThresholdHistoryValue.11 = 360
cnpdThresholdHistoryValue.12 = 181
cnpdThresholdHistoryValue.13 = 2
cnpdThresholdHistoryValue.14 = 31
cnpdThresholdHistoryValue.15 = 3681501
cnpdThresholdHistoryValue.16 = 3681501
cnpdThresholdHistoryValue.17 = 3681501
cnpdThresholdHistoryValue.18 = 3681501
cnpdThresholdHistoryValue.19 = 1348424
cnpdThresholdHistoryValue.20 = 562337
cnpdThresholdHistoryValue.21 = 366922
cnpdThresholdHistoryValue.22 = 311978
cnpdThresholdHistoryValue.23 = 108392
cnpdThresholdHistoryValue.24 = 106086
cnpdThresholdHistoryValue.25 = 81449
cnpdThresholdHistoryValue.26 = 36182
cnpdThresholdHistoryValue.27 = 17941
cnpdThresholdHistoryValue.28 = 17318
cnpdThresholdHistoryValue.29 = 12197
cnpdThresholdHistoryValue.30 = 9046
cnpdThresholdHistoryValue.31 = 1
cnpdThresholdHistoryType.1 = risingBreach(1)
cnpdThresholdHistoryType.2 = risingBreach(1)
cnpdThresholdHistoryType.3 = risingBreach(1)
cnpdThresholdHistoryType.4 = risingBreach(1)
cnpdThresholdHistoryType.5 = risingBreach(1)
cnpdThresholdHistoryType.6 = risingBreach(1)
cnpdThresholdHistoryType.7 = risingBreach(1)
cnpdThresholdHistoryType.8 = risingBreach(1)
cnpdThresholdHistoryType.9 = risingBreach(1)
cnpdThresholdHistoryType.10 = risingBreach(1)
cnpdThresholdHistoryType.11 = risingBreach(1)
cnpdThresholdHistoryType.12 = risingBreach(1)
cnpdThresholdHistoryType.13 = fallingBreach(2)
cnpdThresholdHistoryType.14 = risingBreach(1)
cnpdThresholdHistoryType.15 = risingBreach(1)
cnpdThresholdHistoryType.16 = risingBreach(1)
cnpdThresholdHistoryType.17 = risingBreach(1)
cnpdThresholdHistoryType.18 = risingBreach(1)
cnpdThresholdHistoryType.19 = risingBreach(1)
cnpdThresholdHistoryType.20 = risingBreach(1)
cnpdThresholdHistoryType.21 = risingBreach(1)
cnpdThresholdHistoryType.22 = risingBreach(1)
cnpdThresholdHistoryType.23 = risingBreach(1)
cnpdThresholdHistoryType.24 = risingBreach(1)
cnpdThresholdHistoryType.25 = risingBreach(1)
cnpdThresholdHistoryType.26 = risingBreach(1)
cnpdThresholdHistoryType.27 = risingBreach(1)
cnpdThresholdHistoryType.28 = risingBreach(1)
cnpdThresholdHistoryType.29 = risingBreach(1)
cnpdThresholdHistoryType.30 = risingBreach(1)
```

```
cnpdThresholdHistoryType.31 = fallingBreach(2)
cnpdThresholdHistoryTime.1 = 8579
cnpdThresholdHistoryTime.2 = 8579
cnpdThresholdHistoryTime.3 = 8579
cnpdThresholdHistoryTime.4 = 8579
cnpdThresholdHistoryTime.5 = 8579
cnpdThresholdHistoryTime.6 = 8579
cnpdThresholdHistoryTime.7 = 8579
cnpdThresholdHistoryTime.8 = 8579
cnpdThresholdHistoryTime.9 = 8579
cnpdThresholdHistoryTime.10 = 8579
cnpdThresholdHistoryTime.11 = 8579
cnpdThresholdHistoryTime.12 = 8579
cnpdThresholdHistoryTime.13 = 8579
cnpdThresholdHistoryTime.14 = 97579
cnpdThresholdHistoryTime.15 = 44261817
cnpdThresholdHistoryTime.16 = 44278817
cnpdThresholdHistoryTime.17 = 44284817
cnpdThresholdHistoryTime.18 = 44288471
cnpdThresholdHistoryTime.19 = 44288471
cnpdThresholdHistoryTime.20 = 44288471
cnpdThresholdHistoryTime.21 = 44288471
cnpdThresholdHistoryTime.22 = 44288471
cnpdThresholdHistoryTime.23 = 44288471
cnpdThresholdHistoryTime.24 = 44288471
cnpdThresholdHistoryTime.25 = 44288471
cnpdThresholdHistoryTime.26 = 44288471
cnpdThresholdHistoryTime.27 = 44288471
cnpdThresholdHistoryTime.28 = 44288471
cnpdThresholdHistoryTime.29 = 44288471
cnpdThresholdHistoryTime.30 = 44288471
cnpdThresholdHistoryTime.31 = 44289471
cnpdThresholdHistoryProtocol.1 = 5
cnpdThresholdHistoryProtocol.2 = 23
cnpdThresholdHistoryProtocol.3 = 46
cnpdThresholdHistoryProtocol.4 = 51
cnpdThresholdHistoryProtocol.5 = 71
cnpdThresholdHistoryProtocol.6 = 36
cnpdThresholdHistoryProtocol.7 = 3
cnpdThresholdHistoryProtocol.8 = 48
cnpdThresholdHistoryProtocol.9 = 61
cnpdThresholdHistoryProtocol.10 = 59
cnpdThresholdHistoryProtocol.11 = 41
cnpdThresholdHistoryProtocol.12 = 68
cnpdThresholdHistoryProtocol.13 = 62
cnpdThresholdHistoryProtocol.14 = 62
cnpdThresholdHistoryProtocol.15 = 5
cnpdThresholdHistoryProtocol.16 = 5
cnpdThresholdHistoryProtocol.17 = 5
cnpdThresholdHistoryProtocol.18 = 5
cnpdThresholdHistoryProtocol.19 = 23
cnpdThresholdHistoryProtocol.20 = 46
cnpdThresholdHistoryProtocol.21 = 51
```

```
cnpdThresholdHistoryProtocol.22 = 71
cnpdThresholdHistoryProtocol.23 = 3
cnpdThresholdHistoryProtocol.24 = 36
cnpdThresholdHistoryProtocol.25 = 48
cnpdThresholdHistoryProtocol.26 = 61
cnpdThresholdHistoryProtocol.27 = 41
cnpdThresholdHistoryProtocol.28 = 59
cnpdThresholdHistoryProtocol.29 = 62
cnpdThresholdHistoryProtocol.30 = 68
cnpdThresholdHistoryProtocol.31 = 62
cnpdThresholdHistoryStatsSelect.1 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.2 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.4 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.5 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.6 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.7 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.8 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.9 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.10 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.11 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.12 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.13 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.14 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.15 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.16 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.17 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.18 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.19 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.20 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.21 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.22 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.23 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.24 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.25 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.26 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.27 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.28 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.29 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.30 = packetCountIn(7)
cnpdThresholdHistoryStatsSelect.31 = packetCountIn(7)
```

Note The traps shown below are sample traps that could be generated by traffic patterns that breach the above configuration. The traps are shown only for the sake of the example.

```
Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
```

```
Enterprise Specific trap: 1
Time Ticks: 44288471
cnpdThresholdConfigIfIndex.3 = 13
cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryValue.3 = 366922
cnpdThresholdConfigRising.3 = 30
cnpdThresholdHistoryProtocol.3 = 51
cnpdThresholdHistoryTime.3 = 44288471

Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 1
Time Ticks: 44288471
cnpdThresholdConfigIfIndex.3 = 13
cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryValue.3 = 311978
cnpdThresholdConfigRising.3 = 30
cnpdThresholdHistoryProtocol.3 = 71
cnpdThresholdHistoryTime.3 = 44288471

Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 1
Time Ticks: 44288471
cnpdThresholdConfigIfIndex.3 = 13
cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryValue.3 = 108392
cnpdThresholdConfigRising.3 = 30
cnpdThresholdHistoryProtocol.3 = 3
cnpdThresholdHistoryTime.3 = 44288471

Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 1
Time Ticks: 44288471
cnpdThresholdConfigIfIndex.3 = 13
cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryValue.3 = 106086
cnpdThresholdConfigRising.3 = 30
cnpdThresholdHistoryProtocol.3 = 36
cnpdThresholdHistoryTime.3 = 44288471

Received SNMPv1 Trap:
Community: public
```

```
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 1
Time Ticks: 44288471
cnpdThresholdConfigIfIndex.3 = 13
cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryValue.3 = 81449
cnpdThresholdConfigRising.3 = 30
cnpdThresholdHistoryProtocol.3 = 48
cnpdThresholdHistoryTime.3 = 44288471

Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 1
Time Ticks: 44288471
cnpdThresholdConfigIfIndex.3 = 13
cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryValue.3 = 36182
cnpdThresholdConfigRising.3 = 30
cnpdThresholdHistoryProtocol.3 = 61
cnpdThresholdHistoryTime.3 = 44288471

Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 1
Time Ticks: 44288471
cnpdThresholdConfigIfIndex.3 = 13
cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryValue.3 = 17941
cnpdThresholdConfigRising.3 = 30
cnpdThresholdHistoryProtocol.3 = 41
cnpdThresholdHistoryTime.3 = 44288471

Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 1
Time Ticks: 44288472
cnpdThresholdConfigIfIndex.3 = 13
cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryValue.3 = 17318
cnpdThresholdConfigRising.3 = 30
cnpdThresholdHistoryProtocol.3 = 59
cnpdThresholdHistoryTime.3 = 44288471
```

```
Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 1
Time Ticks: 44288472
cnpdThresholdConfigIfIndex.3 = 13
cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryValue.3 = 12197
cnpdThresholdConfigRising.3 = 30
cnpdThresholdHistoryProtocol.3 = 62
cnpdThresholdHistoryTime.3 = 44288471

Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 1
Time Ticks: 44288472
cnpdThresholdConfigIfIndex.3 = 13
cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryValue.3 = 9046
cnpdThresholdConfigRising.3 = 30
cnpdThresholdHistoryProtocol.3 = 68
cnpdThresholdHistoryTime.3 = 44288471

Received SNMPv1 Trap:
Community: public
Enterprise: ciscoNbarProtocolDiscoveryMIB
Agent-addr: a.b.c.d
Enterprise Specific trap.
Enterprise Specific trap: 2
Time Ticks: 44289471
cnpdThresholdConfigIfIndex.3 = 13
cnpdThresholdConfigStatsSelect.3 = packetCountIn(7)
cnpdThresholdHistoryValue.3 = 1
cnpdThresholdConfigFalling.3 = 20
cnpdThresholdHistoryProtocol.3 = 62
cnpdThresholdHistoryTime.3 = 44289471
```

Additional References

For additional information related to <module feature>, refer to the following references:

Related Documents

Related Topic	Document Title
Network-Based Application Recognition	Network-Based Application Recognition and Distributed Network-Based Application Recognition
SNMP	Configuring SNMP Support Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2

Standards

This feature does not address any new Standards.

For information on Standards supported by the general NBAR and dNBAR features, see the *Network-Based Application Recognition and Distributed Network-Based Application Recognition* document.

MIBs

This document documents the CISCO-NBAR-PROTOCOL-DISCOVERY MIB.

For information on other MIBs supported by the general NBAR and dNBAR features, see the *Network-Based Application Recognition and Distributed Network-Based Application Recognition* document.

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

This feature does not address any new RFCs.

For information on RFCs supported by the general NBAR and dNBAR features, see the *Network-Based Application Recognition and Distributed Network-Based Application Recognition* document.

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

This section documents modified commands. All other commands used with this feature are documented in the Cisco IOS Release 12.2 command reference publications.

- **snmp-server enable traps cnpd**
- **snmp-server host**

Note This command reference does not document the SNMP-related commands shown elsewhere in this document. This command reference only documents the new and changed commands for the router.

snmp-server enable traps cnpd

To enable NBAR Protocol Discovery MIB notifications, use the **snmp-server enable traps cnpd** command in global configuration mode. To disable NBAR Protocol Discovery MIB notifications, use the **no** form of this command.

snmp-server enable traps cnpd

no snmp-server enable traps cnpd

Syntax Description

This command has no arguments or keywords.

Defaults

SNMP notifications are disabled by default.

Command Modes

Global configuration

Command History

Release	Modification
12.2(15)T	This command was introduced.

Usage Guidelines

SNMP notifications can be sent as traps or inform requests. This command enables both traps and inform requests.

This command controls (enables or disables) cnpd notifications. The cnpd notifications are used with the Protocol Discovery MIB to provide various Protocol Discovery information.

The **snmp-server enable traps cnpd** command is used in conjunction with the **snmp-server host** command. Use the **snmp-server host** command to specify which host or hosts receive SNMP notifications. To send SNMP notifications, you must configure at least one **snmp-server host** command.

Examples

The following example enables the router to send cnpd notifications:

```
Router(config)# snmp-server enable traps cnpd
```

Related Commands

Command	Description
snmp-server host	Specifies the recipient of an SNMP notification operation.

snmp-server host

To specify the recipient of an SNMP notification operation, use the **snmp-server host** command in global configuration mode. To remove the specified host from the configuration, use the **no** form of this command.

snmp-server host *host-addr* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] [**vrf** *vrf-name*]

no snmp-server host *host-addr* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] [**vrf** *vrf-name*]

Syntax Description

host-addr	Name or Internet address of the host (the targeted recipient).
traps	(Optional) Specifies that notifications should be sent as traps. This is the default.
informs	(Optional) Specifies that notifications should be sent as informs.
version	(Optional) Version of the SNMP used to send the traps. Version 3 is the most secure model, because it allows packet encryption with the **priv** keyword. If you use the **version** keyword, one of the following keywords must be specified: • **1** —SNMPv1. This option is not available with informs. • **2c** —SNMPv2C. • **3** —SNMPv3. One of the following three optional keywords can follow the version 3 keyword: – **auth**—Enables Message Digest 5 (MD5) and Secure Hash Algorithm (SHA) packet authentication. – **noauth**—Specifies that the noAuthNoPriv security level applies to this host. This is the default security level for SNMPv3. – **priv**—Enables Data Encryption Standard (DES) packet encryption (also called 'privacy').
community-string	Password-like community string sent with the notification operation. Though you can set this string using the **snmp-server host** command by itself, we recommend you define this string using the **snmp-server community** command prior to using the **snmp-server host** command.
udp-port *port*	(Optional) User Datagram Protocol (UDP) port of the host to use. The default is 162.

notification-type	(Optional) Type of notification to be sent to the host. If no type is specified, all available notifications are sent. The notification type can be one or more of the following keywords: • **bgp**—Sends Border Gateway Protocol (BGP) state change notifications. • **calltracker**—Sends Call Tracker call-start/call-end notifications. • **cnpd—**Sends Cisco NBAR Protocol Discovery notifications. • **config**—Sends configuration change notifications. • **director**—Sends DistributedDirector-related notifications. • **dspu**—Sends downstream physical unit (DSPU) notifications. • **entity**—Sends Entity MIB modification notifications. • **envmon**—Sends Cisco enterprise-specific environmental monitor notifications when an environmental threshold is exceeded. • **frame-relay**—Sends Frame Relay notifications. • **hsrp**—Sends Hot Standby Routing Protocol (HSRP) notifications. • **ipmobile**—Sends Mobile IP notifications. • **ipsec**—Sends IP Security (IPSec) notifications. • **isdn**—Sends ISDN notifications. • **llc2**—Sends Logical Link Control, type 2 (LLC2) notifications. • **mpls-ldp**—Sends MPLS Label Distribution Protocol (LDP) notifications indicating status changes in LDP sessions. • **mpls-traffic-eng**—Sends MPLS traffic engineering notifications indicating changes in the status of MPLS traffic engineering tunnels. • **mpls-vpn**—Sends MPLS VPN notifications. • **pim**—Sends Protocol Independent Multicast (PIM) notifications. • **repeater**—Sends standard repeater (hub) notifications. • **rsrb**—Sends remote source-route bridging (RSRB) notifications. • **rsvp**—Sends Resource Reservation Protocol (RSVP) notifications. • **rtr**—Sends Service Assurance Agent (RTR) notifications. • **sdlc**—Sends Synchronous Data Link Control (SDLC) notifications. • **sdllc**—Sends SDLC Logical Link Control (SDLLC) notifications. • **snmp**—Sends any enabled RFC 1157 SNMP linkUp, linkDown, authenticationFailure, warmStart, and coldStart notifications. • **srp**—Sends Spatial Reuse Protocol (SRP) notifications. • **stun**—Sends serial tunnel (STUN) notifications. • **syslog**—Sends error message notifications (Cisco Syslog MIB). Specify the level of messages to be sent with the **logging history level** command.

notification-type (Continued)	• **tty**—Sends Cisco enterprise-specific notifications when a TCP connection closes. • **voice**—Sends SNMP poor quality of voice traps, when used with the **snmp enable peer-trap poor qov** command. • **vsimaster—**Sends VSI Master notifications. • **x25**—Sends X.25 event notifications.
vrf *vrf-name*	(Optional) Specifies the Virtual Private Network (VPN) routing/forwarding (VRF) table that should be used to send SNMP notifications.

Defaults

This command is disabled by default. No notifications are sent.

If you enter this command with no keywords, the default is to send all trap types to the host. No informs will be sent to this host.

If no **version** keyword is present, the default is version 1. If version 3 is specified, but the security level is not specified, the default security level is **noauth**.

The **no snmp-server host** command with no keywords will disable traps, but not informs, to the host. In order to disable informs, use the **no snmp-server host informs** command.

The default UDP port is 162.

Note If the *community-string* is not defined using the **snmp-server community** command prior to using this command, the default form of the **snmp-server community** command will automatically be inserted into the configuration. The password (*community-string*) used for this automatic configuration of the **snmp-server community** will be the same as specified in the **snmp-server host** command. This is the default behavior for Cisco IOS Release 12.0(3) and later.

Command Modes

Global configuration

Command History

Release	Modification
10.0	This command was introduced.
12.0(3)T	The following keywords were added: • **version 3** [**auth** | **noauth** | **priv**] • **hsrp**
11.3(1) MA, 12.0(3)T	The **voice** notification-type keyword was added.
12.1(3)T	The **calltracker** notification-type keyword was added for the Cisco AS5300 and AS5800 platforms.
12.2(2)T	• The **vrf** *vrf-name* keyword/argument combination was added. • The **ipmobile** notification-type keyword was added. • Support for the **vsimaster** notification-type keyword was added for the Cisco 7200 and Cisco 7500 series.
12.2(4)T	• The **pim** notification-type keyword was added. • The **ipsec** notification-type keyword was added.

Release	Modification
12.2(8)T	• The **mpls-traffic-eng** notification-type keyword was added. (Also in 12.0(17)ST) • The **director** notification-type keyword was added.
12.2(13)T	• The **srp** notification-type keyword was added. • The **mpls-vpn** notification-type keyword was added. (Also in 12.0(22)S) • The **mpls-ldp** notification-type keyword was added.
12.2(15)T	• The **cnpd** notification-type keyword was added.

Usage Guidelines

SNMP notifications can be sent as traps or inform requests. Traps are unreliable because the receiver does not send acknowledgments when it receives traps. The sender cannot determine if the traps were received. However, an SNMP entity that receives an inform request acknowledges the message with an SNMP response PDU. If the sender never receives the response, the inform request can be sent again. Thus, informs are more likely to reach their intended destination.

However, informs consume more resources in the agent and in the network. Unlike a trap, which is discarded as soon as it is sent, an inform request must be held in memory until a response is received or the request times out. Also, traps are sent only once, while an inform may be retried several times. The retries increase traffic and contribute to a higher overhead on the network.

If you do not enter an **snmp-server host** command, no notifications are sent. In order to configure the router to send SNMP notifications, you must enter at least one **snmp-server host** command. If you enter the command with no keywords, all trap types are enabled for the host.

In order to enable multiple hosts, you must issue a separate **snmp-server host** command for each host. You can specify multiple notification types in the command for each host.

When multiple **snmp-server host** commands are given for the same host and kind of notification (trap or inform), each succeeding command overwrites the previous command. Only the last **snmp-server host** command will be in effect. For example, if you enter an **snmp-server host inform** command for a host and then enter another **snmp-server host inform** command for the same host, the second command will replace the first.

The **snmp-server host** command is used in conjunction with the **snmp-server enable** command. Use the **snmp-server enable** command to specify which SNMP notifications are sent globally. For a host to receive most notifications, at least one **snmp-server enable** command and the **snmp-server host** command for that host must be enabled.

However, some notification types cannot be controlled with the **snmp-server enable** command. For example, some notification types are always enabled. Other notification types are enabled by a different command. For example, the linkUpDown notifications are controlled by the **snmp trap link-status** command. These notification types do not require an **snmp-server enable** command.

A notification-type option's availability depends on the router type and Cisco IOS software features supported on the router. For example, the **envmon** notification-type is available only if the environmental monitor is part of the system. To see what notification types are available on your system, use the command help **?** at the end of the **snmp-server host** command.

The **vrf** keyword allows you to specify the notifications being sent to a specified IP address over a specific VRF. The VRF defines a VPN membership of a customer so data is stored using the VPN.

Regarding Notification Type Keywords

The *notification-type* keywords used in the **snmp-server host** command do not always match the keywords used in the corresponding **snmp-server enable traps** command. For example, the notification keyword applicable to MPLS traffic engineering tunnels is specified as **mpls-traffic-eng** (containing two dashes and no intervening spaces). The corresponding parameter in the **snmp-server enable traps** command is specified as **mpls traffic-eng** (containing an intervening space and a dash).

This syntax difference is necessary to ensure that the CLI interprets the *notification-type* keyword of the **snmp-server host** command as a unified, single-word construct, which preserves the capability of the **snmp-server host** command to accept multiple *notification-type* keywords in the CLI command line. The **snmp-server enable traps** commands, however, often use two-word constructs in order to provide hierarchical configuration options and to maintain consistency with the command syntax of related commands. Table 1 maps **snmp-server enable traps** commands to the keywords used in the **snmp-server host** command.

Table 1 Notification Keywords and Corresponding SNMP Enable Traps Commands

SNMP Enable Traps Command	SNMP Host Command Keyword
snmp-server enable traps mpls ldp	**mpls-ldp**
snmp-server enable traps mpls traffic-eng[1]	**mpls-traffic-eng**
snmp-server enable traps mpls vpn	**mpls-vpn**

1. See the *Cisco IOS Switching Services Command Reference* for documentation of this command.

Examples

If you want to configure a unique SNMP community string for traps, but you want to prevent SNMP polling access with this string, the configuration should include an access-list. In the following example, the community string is named “comaccess” and the access list is numbered 10:

```
Router(config)# snmp-server community comaccess ro 10
Router(config)# snmp-server host 172.20.2.160 comaccess
Router(config)# access-list 10 deny any
```

The following example sends RFC 1157 SNMP traps to the host specified by the name myhost.cisco.com. Other traps are enabled, but only SNMP traps are sent because only **snmp** is specified in the **snmp-server host** command. The community string is defined as comaccess.

```
Router(config)# snmp-server enable traps
Router(config)# snmp-server host myhost.cisco.com comaccess snmp
```

The following example sends the SNMP and Cisco environmental monitor enterprise-specific traps to address 172.30.2.160:

```
Router(config)# snmp-server enable traps snmp
Router(config)# snmp-server enable traps envmon
Router(config)# snmp-server host 172.30.2.160 public snmp envmon
```

The following example enables the router to send all traps to the host myhost.cisco.com using the community string public:

```
Router(config)# snmp-server enable traps
Router(config)# snmp-server host myhost.cisco.com public
```

The following example will not send traps to any host. The BGP traps are enabled for all hosts, but only the ISDN traps are enabled to be sent to a host.

```
Router(config)# snmp-server enable traps bgp
Router(config)# snmp-server host bob public isdn
```

The following example enables the router to send all inform requests to the host myhost.cisco.com using the community string public:

```
Router(config)# snmp-server enable traps
Router(config)# snmp-server host myhost.cisco.com informs version 2c public
```

The following example sends HSRP MIB informs to the host specified by the name myhost.cisco.com. The community string is defined as public.

```
Router(config)# snmp-server enable traps hsrp
Router(config)# snmp-server host myhost.cisco.com informs version 2c public hsrp
```

The following example sends all SNMP notifications to xyz.com over the VRF named “trap-vrf”:

```
Router(config)# snmp-server host xyz.com vrf trap-vrf
```

Related Commands

Command	Description
snmp-server enable peer-trap poor qov	Enable poor quality of voice notifications for applicable calls associated with a specific voice dial peer.
snmp-server enable traps	Enables SNMP notifications (traps and informs).
snmp-server informs	Specifies inform request options.
snmp-server trap-source	Specifies the interface (and hence the corresponding IP address) that an SNMP trap should originate from.
snmp-server trap-timeout	Defines how often to try resending trap messages on the retransmission queue.