



Configuring Simple Network Management Protocol (SNMP)

This chapter provides information on configuring Simple Network Management Protocol (SNMP) features on your CSS. It also provides a brief overview of SNMP, an Application Layer protocol used extensively in the communications industry. Information in this chapter applies to all CSS models except where noted.

This chapter includes the following sections:

- SNMP Overview
- Configuring SNMP on the CSS
- Displaying the SNMP Configuration
- Managing SNMP on the CSS
- CSS MIBs

SNMP Overview

Simple Network Management Protocol (SNMP) is a set of network management standards for IP-based internetworks. It includes a protocol, a database-structure specification, and a set of management data objects. SNMP implementations typically consist of a management application, running on one or more Network Management Systems (NMSs), and agent applications, usually executing in firmware on various network devices.

SNMP has two major standard revisions, SNMPv1 and SNMPv2. Your CSS supports SNMPv2C (SNMP version 2C), known as “community-based SNMP”, and standard Management Information Base (MIB-II) objects, along with an extensive set of enterprise objects. (MIBs are discussed later in this chapter in the section “Management Information Base (MIB)”.)

This overview contains the following sections:

- Managers and Agents
- Manager/Agent Communication
- Management Information Base (MIB)
- SNMP Communities

**Note**

By default, SNMP access to the CSS is enabled in software through the **no restrict snmp** command. Refer to “Controlling SNMP Access to the CSS” for details.

Managers and Agents

SNMP uses software entities called *managers* and *agents* to manage network devices:

- The *manager* monitors and controls all other SNMP-managed devices (network nodes) in the network. There must be at least one SNMP Manager in a managed network. The manager is installed on a workstation somewhere in the network.
- An *agent* resides in a managed device (a network node). The agent receives instructions from the SNMP Manager, and also sends management information back to the SNMP Manager as events occur. The agent can reside on routers, bridges, hubs, workstations, or printers, to name just a few network devices.

There are many different SNMP management applications, but they all perform the same basic task: they allow SNMP managers to communicate with agents to monitor, configure, and receive alerts from the network devices. You can use any SNMP-compatible network management system to monitor and control a CSS.

Manager/Agent Communication

There are several ways that the SNMP manager and the agent communicate.

- The manager can:
 - Retrieve a value (a *get* action).

The SNMP manager requests information from the agent, such as the number of users logged on to the agent device, or the status of a critical process on that device. The agent gets the value of the requested MIB variable and sends the value back to the manager.
 - Retrieve the value immediately after the variable you name (a *get-next* action).

The SNMP manager retrieves values from within a MIB. Using the *get-next* function, you do not need to know the exact variable instance you are looking for; the SNMP manager takes the variable you name and then uses a sequential search to find the desired variables.

- Retrieve a number of values (a *get-bulk* action).

The SNMP manager performs a number of get-next actions that you specify.

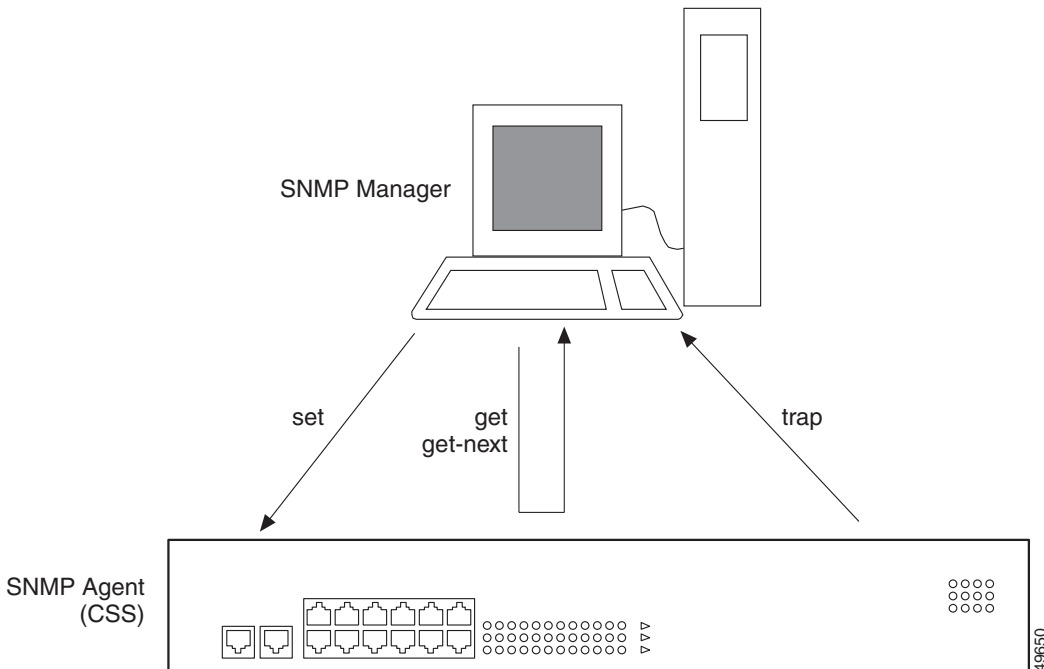
- Change a setting on the agent (a *set* action).

The SNMP manager requests the agent to change the value of the MIB variable. For example, you could run a script or an application on a remote device with a set action.

- An agent can send an unsolicited message to the manager at any time if a significant, predetermined event takes place on the agent. This message is called a *trap*.

When a trap condition occurs, the SNMP agent sends an SNMP trap message to the device specified as the *trap receiver* or *trap host*. The SNMP Administrator configures the trap host (usually the SNMP management station) to perform the action needed when a trap is detected. Figure 9-1 illustrates manager/agent communication.

Figure 9-1 SNMP Manager/Agent Interaction



Management Information Base (MIB)

SNMP obtains information from the network through a Management Information Base (MIB). The MIB is a database of code blocks called *MIB objects*. Each MIB object controls one specific function, such as counting how many bytes are transmitted through an agent's port. The MIB object comprises *MIB variables*, which define the MIB object name, description, default value, and so forth.

The collection of MIB objects is structured hierarchically. The MIB hierarchy is referred to as the *MIB tree*. The MIB tree is defined by the International Standards Organization (ISO). The MIB is installed on the manager, and is present within each agent in the SNMP network.

At the top of the tree is the broadest information about a network. Each branch and sub-branch of the tree gets progressively more specific, and the lowest branches of the tree contain the most specific MIB objects; the leaves contain the actual data. See Figure 9-2 for an example of how the MIB tree objects become more specific as the tree expands.



Note

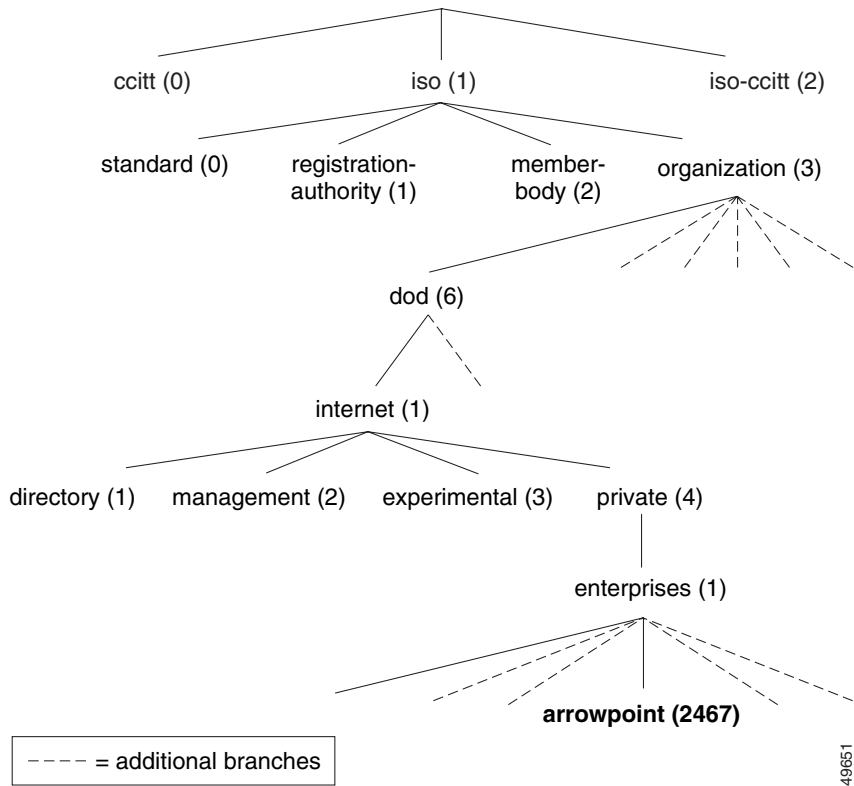
There are two versions of the MIB tree as defined by ISO: MIB-I and MIB-II, which has more variables than MIB-I. Refer to the MIB-II standard in RFC 1213, "Management Information Base for Network Management of TCP/IP-based Internets: MIB-II."

MIB Variables

There are two types of MIB variables:

- **Scalar** - Variables that define an object with a single representation. This means that an object describes a particular characteristic of the entire system. An example of a scalar variable is **SysDescr**, which provides a system-wide description of the CSS.
- **Tabular** - Variables that define an object with multiple representations. This means that an object can have different values, depending on the qualifier. For example, one tabular object could show bytes per interface, temperature per board, or hits per service.

Figure 9-2 Top of the MIB Tree



As shown in Figure 9-2, a number is associated with a MIB object name. This number is called the *object identifier* (or *object ID*), and it uniquely identifies the MIB object in the MIB tree. (The dotted lines represent other branches not relevant to this discussion.)

For example, the MIB object labeled *arrowpoint* (which contains the MIB objects specific to CSSs) in Figure 9-2 can be labeled:

```
iso.organization.dod.internet.private.enterprises.arrowpoint
or
```

```
1.3.6.1.4.1.2467
```

MIB Extensions (Enterprise MIBs)

The MIB tree has a special branch set aside for specific vendors to build their own extensions; this is called the *enterprise MIB branch*. The MIB files in this branch, included on your CSS Documentation and System Software CD, comprise the CSS Enterprise MIBs. (This is the highlighted MIB object in Figure 9-2.) The enterprise MIB files are categorized along functional boundaries.

For a list of MIB branches under the Cisco CSS Enterprise MIB, refer to Table 9-5 later in this chapter.

SNMP Communities

Each SNMP device or member is part of a *community*. An SNMP community determines the access that each SNMP device has.

You supply a name to the community. After that, all SNMP devices that are assigned to that community as *members* have the same access rights. The access rights that the CSS supports are:

- **read** - Allows read-only access to the MIB tree for devices included in this community
- **read-write** - Allows both read and write access to the MIB tree for devices included in this community

Configuring SNMP on the CSS

Once you have set up the SNMP management software (SNMP version 2C) on the network devices, you are ready to configure SNMP settings on the CSS. You can configure two basic areas of SNMP functionality on the CSS: SNMP functions and RMON functions.

The following sections describe how to configure SNMP on the CSS. For information on configuring RMON, refer to Chapter 10, Configuring Remote Monitoring (RMON).

- Controlling SNMP Access to the CSS
- Planning Your SNMP Configuration
- Defining the CSS as an SNMP Agent
- Configuring Denial of Service (DoS)

Controlling SNMP Access to the CSS

To control SNMP access to the CSS, use the **no restrict snmp** and **restrict snmp** commands. Access through SNMP is enabled by default. The options for this global configuration mode command are:

- **no restrict snmp** - Enable SNMP access to the CSS (default setting)
- **restrict snmp** - Disable SNMP access to the CSS

Planning Your SNMP Configuration

Consider the following information before you set up SNMP on your network:

- Decide which types of information the SNMP Manager needs (if your application is using an SNMP Manager). Choose the particular MIB variables that you want through the management software.
- Decide how many trap hosts you need. In some network configurations, you may want to have a primary trap host with one other workstation also receiving traps for redundancy. In a distributed or segmented network, you may want to have more trap hosts enabled. You can configure up to five trap hosts per SNMP agent; that is, one agent can report to a maximum of five hosts.
- Designate a management station or stations. The CSS is an agent in the SNMP network scheme. The agent is already embedded in the CSS when you boot up the device; all you need to do is configure the SNMP parameters on the CSS.

Defining the CSS as an SNMP Agent

The following sections describe how to define the CSS as an SNMP agent. Read these sections for a complete description of the commands associated with this procedure. If you are familiar with this procedure, refer to Table 9-1 as a quick start configuration reference for this task.

- Configuring an SNMP Community
- Configuring an SNMP Contact
- Configuring an SNMP Location
- Configuring an SNMP Name
- Configuring an SNMP Trap-Host
- Configuring SNMP Generic Traps
- Configuring SNMP Auth-Traps
- Configuring SNMP Enterprise Traps
- Configuring SNMP Reload-Enable

Table 9-1 Quick Start for Defining the CSS as an SNMP Agent

Task and Command Example

1. Define the SNMP community strings for each access type, read-only (for a GET action) or read-write (for a GET and SET action). This step is required for using SNMP on the CSS.

```
(config)# snmp community public read-only
(config)# snmp community private read-write
```

2. Provide the SNMP contact name (optional).

```
(config)# snmp contact "fred n mandy"
```

3. Provide an SNMP contact location (optional).

```
(config)# snmp location "Operations"
```

4. Provide the SNMP device name (optional).

```
(config)# snmp name "arrowpoint.com"
```

5. Turn on generic traps (optional).

```
(config)# snmp trap-type generic
```

Table 9-1 Quick Start for Defining the CSS as an SNMP Agent (continued)

Task and Command Example

6. Assign trap receivers and community (required if configuring SNMP traps). You can specify a maximum of five trap hosts. By default, all traps are disabled. The trap-host IP address must correspond to a management station that is monitoring for traps. The community information provided at the end of the **trap-host** command is included in the trap, and may be used by the management station to filter incoming traps.

```
(config)# snmp trap-host 172.16.3.6 trap
(config)# snmp trap-host 172.16.8.4 trap
```

7. Turn on authentication failure traps (optional). An authentication failure occurs if an unauthorized SNMP manager sends an invalid or incorrect community name to an SNMP agent. If this occurs, the agent sends an authentication trap to the trap host (or hosts depending on how many trap hosts are configured).

```
(config)# snmp auth-traps
```

8. Enable global enterprise traps (optional).

```
(config)# snmp trap-type enterprise
```

Then enable a specific enterprise trap type. For example, you can set a trap to notify the trap host of failed login attempts. Login failure traps provide the username and source IP address of the person who failed to log in.

```
(config)# snmp trap-type enterprise login-failure
```

9. Configure the trap host for reload enable ability (optional). Reload enable allows a management station with the proper WRITE community privilege to reboot the CSS.

```
(config)# snmp reload-enable 100
```

10. Configure special enterprise trap thresholds to notify the trap host of Denial of Service (DoS) attacks on your system (optional). For example, you can set a trap threshold to notify the trap host of DoS attacks with illegal addresses, either source or destination.

```
(config)# snmp trap-type enterprise dos-illegal-attack
trap-threshold 1
```

Configuring an SNMP Community

Use the **snmp community** command to set or modify SNMP community names and access properties. You may specify as many community names as you wish.



Caution

It is required that you define the community strings for each access type (read-only or read-write) before you use SNMP on the CSS. The CSS is inaccessible until a read community string is specified.

The syntax for this global configuration mode command is:

```
snmp community community_name [read-only|read-write]
```

The variables and options are:

- *community_name* - The SNMP community name for this system. Enter an unquoted text string with no space and a maximum length of 12 characters.
- **read-only** - Allow read-only access for this community.
- **read-write** - Allow read-write access for this community.

For example:

```
(config)# snmp community sqa read-write
```

To remove a community name, enter:

```
(config)# no snmp community sqa
```

Configuring an SNMP Contact

Use the **snmp contact** command to set or modify the contact name for the SNMP system. You can specify only one contact name. The syntax for this global configuration mode command is:

```
snmp contact "contact_name"
```

Enter the contact name as an unquoted text string with a maximum of 255 characters including spaces. You can also include information on how to contact the person; for example, a phone number or email address.

For example:

```
(config)# snmp contact "Fred N. Mandy"
```

To remove the contact name, enter:

```
(config)# no snmp contact
```

Configuring an SNMP Location

Use the **snmp location** command to set or modify the SNMP system location. You can specify only one location. The syntax for this global configuration mode command is:

```
snmp location "location"
```

Enter the location as the physical location of the system. Enter a quoted text string with a maximum length of 255 characters.

For example:

```
(config)# snmp location "sqa_lab1"
```

To remove the location, enter:

```
(config)# no snmp location
```

Configuring an SNMP Name

Use the **snmp name** command to set or modify the SNMP name for this system. You can specify only one name. The syntax for this global configuration mode command is:

```
snmp name "name"
```

Enter the SNMP name as the unique name assigned to a system by the administrator. Enter a quoted text string with a maximum of 255 characters. The standard name convention is the system's fully-qualified domain name (for example, sqa@arrowpoint.com).

For example:

```
(config)# snmp name "sqa@arrowpoint.com"
```

To remove the SNMP name for a system, enter:

```
(config)# no snmp name
```

Configuring SNMP Generic Traps

Use the **snmp trap-type generic** command to enable SNMP generic trap types. The generic SNMP traps consist of cold start, warm start, link down, and link up.

For example:

```
(config)# snmp trap-type generic
```

To disable a generic trap, enter:

```
(config)# no snmp trap-type generic
```

Configuring an SNMP Trap-Host

Use the **snmp trap-host** command to set or modify the SNMP host to receive traps from a CSS. You can specify a maximum of five hosts. The syntax for this global configuration mode command is:

```
snmp trap-host ip_address or host community_name
```

The variables are:

- *ip_address* or *host* - The IP address or host name of an SNMP host that has been configured to receive traps. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1) or in mnemonic host-name format (for example, myhost.mydomain.com).
- *community_name* - The community name to use when sending traps to the specified SNMP host. Enter an unquoted text string with no spaces and a maximum length of 12 characters.

For example:

```
(config)# snmp trap-host 172.16.3.6 sqa@arrowpoint.com
```

To remove a specified trap host, enter:

```
(config)# no snmp trap-host 172.16.3.6
```

Configuring SNMP Auth-Traps

Use the **snmp auth-traps** command to enable reception of SNMP authentication traps. The CSS generates these traps when an SNMP management station attempts to access your system with invalid community names.

For example:

```
(config)# snmp auth-traps
```

To disable reception of authentication traps, enter:

```
(config)# no snmp auth-traps
```

Configuring SNMP Enterprise Traps

Use the **snmp trap-type enterprise** command to enable SNMP enterprise trap types. You can enable the CSS to generate enterprise traps when denial of service attack events occur, a login fails, or a CSS service transitions state.



Note

For information on configuring Denial of Service enterprise traps, refer to “Configuring Denial of Service (DoS)” later in this chapter.

The options for this global configuration mode command are:

- **snmp trap-type enterprise** - Enable enterprise traps. You must enable enterprise traps before you configure an enterprise trap option.
- **snmp trap-type enterprise login-failure** - Generate SNMP enterprise traps when a CSS login failure occurs. The CSS also generates an alert-level log message.
- **snmp trap-type enterprise reload** - Generate SNMP enterprise traps when a CSS reboot occurs. The CSS also generates a trap when a reboot is initiated directly through SNMP.
- **snmp trap-type enterprise redundancy-transition** - Generate SNMP enterprise traps when the CSS redundancy transitions state.
- **snmp trap-type enterprise service-transition** - Generate SNMP enterprise traps when a CSS service transitions state. A trap is generated when a service fails and when a failed service resumes proper operation.

For example, to enable enterprise traps, enter:

```
(config)# snmp trap-type enterprise
```

To disable all enterprise traps, enter:

```
(config)# no snmp trap-type enterprise
```

To prevent the CSS from generating traps when a login fails, enter:

```
(config)# no snmp trap-type enterprise login-failure
```

To prevent the CSS from generating traps when a CSS reload occurs, enter:

```
(config)# no snmp trap-type enterprise reload
```

To prevent the CSS from generating traps when the service transitions state, enter:

```
(config)# no snmp trap-type enterprise service-transition
```

To prevent the CSS from generating traps when a redundant CSS transitions state, enter:

```
(config)# no snmp trap-type enterprise redundancy-transition
```

Configuring SNMP Reload-Enable

Use the **snmp reload-enable** command to reboot the CSS using SNMP. The syntax and options for this global configuration mode command are:

- **snmp reload-enable** - Allow any SNMP write to the apSnmExtReloadSet object to force a CSS reboot. The reload object, apSnmExtReloadSet, is located at 1.3.6.1.4.1.2467.1.22.7. You can find this object in the CSS Enterprise MIB, snmpext.mib.
- **snmp reload-enable reload_value** - Allow an SNMP write equal to the *reload_value* to force a CSS reboot.

Enter the *reload_value* as the object used to control apSnmExtReloadSet, providing the SNMP-based reboot. When the object is set to 0, an SNMP reboot is not allowed. When the object is set between 1 to 232, a reboot may be caused with any write value to apSnmExtReloadSet. For security purposes, this object always returns 0 when read.

For example:

```
(config)# snmp reload-enable
```

To prevent users from rebooting the CSS using SNMP (default behavior), enter:

```
(config)# no snmp reload-enable
```

Configuring Denial of Service (DoS)

You can configure special enterprise traps to notify the trap host of Denial of Service (DoS) attacks on your system. You can also use the CLI to display detailed information about DoS attacks and reset the DoS statistics for your CSS to zero. This section describes how to configure DoS traps. If you are familiar with this procedure, use Table 9-2 as a quick start configuration reference.



Note

Ensure you first enable SNMP enterprise traps using the **snmp trap-type enterprise** command before you configure the CSS to generate SNMP enterprise traps when a DoS attack event occurs. For information, refer to “Configuring SNMP Enterprise Traps” earlier in this chapter.

Table 9-2 Denial of Service Configuration Quick Start

Task and Command Example

1. Set the trap threshold to notify the trap host of DoS attacks with illegal addresses, either source or destination.

```
(config)# snmp trap-type enterprise dos-illegal-attack  
trap-threshold 1
```

2. Set the trap threshold to notify the trap host of DoS LAND attacks.

```
(config)# snmp trap-type enterprise dos-land-attack  
trap-threshold 1
```

3. Set the trap threshold to notify the trap host of DoS smurf attacks.

```
(config)# snmp trap-type enterprise dos-smurf-attack  
trap-threshold 1
```

Table 9-2 Denial of Service Configuration Quick Start (continued)

Task and Command Example
<p>4. Set the trap threshold to notify the trap host of DoS SYN attacks.</p> <pre>(config)# snmp trap-type enterprise dos-syn-attack trap-threshold 10</pre>
<p>5. Display information about DoS attacks.</p> <pre>(config)# show dos summary (config)# show dos</pre>
<p>6. As required, reset the DoS statistics for a CSS to zero.</p> <pre>(config)# zero dos statistics</pre>

Defining a DoS SNMP Trap-Type

Use the **snmp trap-type enterprise** command to enable the CSS to generate SNMP enterprise traps when a denial of service (DoS) attack event occurs. One trap is generated each second when the number of attacks during that second exceeds the threshold for the configured DoS attack type.



Note

Ensure you first enable SNMP enterprise traps using the **snmp trap-type enterprise** command before you configure the CSS to generate SNMP enterprise traps when a DoS attack event occurs. For information, refer to “Configuring SNMP Enterprise Traps” earlier in this chapter.

The syntax for this global configuration mode command is:

```
snmp trap-type enterprise dos_attack_type {trap-threshold threshold_value}
```

The *dos_attack_type* variable is the type of denial of service attack event to trap. The options are:

- **dos-illegal-attack** - Generates traps for illegal addresses, either source or destination. Illegal addresses are loopback source addresses, broadcast source addresses, loopback destination addresses, multicast source addresses, or source addresses that you own. The default trap threshold for this type of attack is 1 per second.

- **dos-land-attack** - Generates traps for packets that have identical source and destination addresses. The default trap threshold for this type of attack is 1 per second.
- **dos-smurf-attack** - Generates traps when the number of pings with a broadcast destination address exceeds the threshold value. The default trap threshold for this type of attack is 1 per second.
- **dos-syn-attack** - Generates traps when the number of TCP connections that are initiated by a source, but not followed with an acknowledgment (ACK) frame to complete the three-way TCP handshake, exceeds the threshold value. The default trap threshold for this type of attack is 10 per second.

**Note**

You can override a default trap threshold by using the **trap-threshold** option. For the *threshold_value*, enter a number from 1 to 65535.

For example, to enable the CSS to generate traps for packets that have identical source and destination addresses, enter:

```
(config)# snmp trap-type enterprise dos-land-attack
```

For example, to prevent the CSS from generating denial of service attack event traps, enter:

```
(config)# no snmp trap-type enterprise dos_attack_type
```

Displaying Denial of Service Configurations

Use the **show dos summary** command to display a summary of information about DoS attacks. To display more detailed information, use the **show dos** command.

For example:

```
(config)# show dos summary
```

Table 9-3 describes the fields in the **show dos** output.

Table 9-3 Field Descriptions for the show dos Command


Field	Description
Total Attacks	<p>The total number of DOS attacks detected since the CSS was booted. The type of attacks that are listed along with their number of occurrences are:</p> <ul style="list-style-type: none"> • SYN Attacks - The TCP connections that are initiated by a source but are not followed with an ACK frame to complete the three way TCP handshake • LAND Attacks - Packets that have identical source and destination addresses • Zero Port Attacks - Frames that contain source or destination TCP or UDP ports equal to zero <p> Note Older SmartBits software may send frames containing source or destination ports equal to zero. The CSS logs them as DOS attacks and drops these frames.</p> <ul style="list-style-type: none"> • Illegal Src Attacks - Illegal source addresses • Illegal Dst Attacks - Illegal destination addresses • Smurf Attacks - Pings with a broadcast destination address
Maximum per second	<p>The maximum number of events per second. Use the maximum events per second information to set SNMP trap threshold values. Note that the maximum number of events per second is the maximum per SFP. For a CSS 11800, which may have up to 4 SFPs, the maximum rate per second may be as high as four times that which is displayed.</p>
First Attack Detected	<p>The first time an attack was detected.</p>

Table 9-3 *Field Descriptions for the show dos Command (continued)*

Field	Description
Last Attack Detected	The last time an attack was detected.
DOS Attack Event	Details for each detected attack event, up to a maximum of 50 events per SFP.
First Attack	The first time that the attack event occurred.
Last Attack	The last time that the attack event occurred.
Source/Destination Address	The source and destination addresses for the attack event.
Event Type	The type of event.
Total Attacks	The total number of attack occurrences for the event.

Displaying the SNMP Configuration

After you configure SNMP, display the SNMP configuration. For example:

```
(config)# show running-config global
```

For details on the **show running-config** command and its output, refer to Chapter 1, Logging in and Getting Started.

Managing SNMP on the CSS

The main tasks you need to do to manage SNMP on the CSS are:

- Enabling SNMP Manager Access to the CSS
- Using the CSS to Look Up MIB Objects
- Reading Logs
- Setting Alarms

Enabling SNMP Manager Access to the CSS

By default, the CSS enables SNMP access to its command base, but you must first create community strings using the **snmp community** command before you can use SNMP in the CSS. Refer to “Configuring an SNMP Community” earlier in this chapter for details.

**Note**

SNMP is not a secure network environment. Do not use SNMP by itself to provide security for your network.

Using the CSS to Look Up MIB Objects

To look up a MIB object, including the variables that make up the object:

1. Access global configuration mode by entering:

```
# config
```

2. Access rmon-alarm mode by entering:

```
(config)# rmon-alarm index_number
```

where *index_number* is the index of the alarm.

3. Display the MIB object by entering:

```
(config-rmonalarm[1])# lookup object
```

where *object* is the name of the MIB object.

You can look up a specific object, or you can use the question mark (?) character as a wildcard to help you complete your request.

For example, you want to look up a MIB object, but you are not sure of its exact name. You already know that the MIB you want is part of the apFlowMgrExt group of objects. In this case, issue the **lookup** command with the question mark (?) character, as shown below.

```
(config-rmonalarm[1])# lookup apFlowMgrExt?
```

```
apFlowMgrExtDoSAttackEventType  
apFlowMgrExtDoSAttackEventCount  
apFlowMgrExtDoSAttackIndex  
apFlowMgrExtDosTotalSmurfAttacks  
apFlowMgrExtDosTotalIllegalSourceAttacks  
apFlowMgrExtDosTotalZeroPortAttacks  
apFlowMgrExtDosTotalLandAttacks  
apFlowMgrExtDosTotalSynAttacks  
apFlowMgrExtDosTotalAttacks  
apFlowMgrExtIdleTimer  
apFlowMgrExtPortIdleValue  
apFlowMgrExtPortIdle  
apFlowMgrExtReserveCleanTimer  
apFlowMgrExtPermanentPort4  
apFlowMgrExtPermanentPort3  
apFlowMgrExtPermanentPort2  
apFlowMgrExtPermanentPort1  
apFlowMgrExtFlowTraceDuration  
apFlowMgrExtFlowTraceMaxFileSize  
apFlowMgrExtFlowTraceState
```

The previous example shows that using the question mark (?) character as a wildcard returns information about the apFlowMgrExt MIB object. You can also issue the **lookup** command on the exact MIB you want and view its description without using the question mark (?) character. For example:

```
(config-rmonalarm[1])# lookup apFlowMgrExtDOSAttackEventCount

ASN Name:          apFlowMgrExtDOSAttackEventCount
MIB:              flowmgrext
Object Identifier: 1.3.6.1.4.1.2467.1.36.27.1.6
Argument Type:    Integer
Range:           0-4294967295
Description:
  This is the number of times this DoS attack had occurred.
```

You can also display a list of all the Enterprise MIBs by using the **lookup** command without any MIB object names, as in the following example:

```
(config-rmonalarm[1])# lookup ?
```

**Note**

This command omits MIB objects of type *string* and *MAC address*.

Useful MIB Statistics

Table 9-4 contains some of the MIB groups that provide useful statistics.

Table 9-4 CSS MIB Statistics

MIB Name	Description
RFC-1398	Ethernet statistics
RFC-1493	Bridge information
RFC-1757	RMON statistics
svcExt.mib	Service variables (including TCP connections)
cntExt.mib	Content rule variables (including frame statistics)
ownExt.mib	Owner statistics (including frame and bytes counts)
cntsvcExt.mib	Services per content rule statistics (including frames, bytes, hits)

Reading Logs

The traplog file contains all of the traps, both generic and enterprise, that have occurred. The network device writes to the traplog file whether or not the SNMP trap configuration is enabled.

To show the trap log since the last CSS reboot, issue the **show log** command as shown:

```
# show log traplog
```

By default, the following events generate level critical-2 messages:

- Link Up
- Link Down
- Cold Start
- Warm Start
- Service Down
- Service Suspended

All other SNMP traps generate level notice-5 messages by default.

Setting Alarms

For information about commands available in this mode, refer to Chapter 10, Configuring Remote Monitoring (RMON).

CSS MIBs

Table 9-5 describes the CSS MIB objects directly under the CSS Enterprise MIB (1.3.6.1.4.1.2467). The MIBs listed in this table are a representation of the CSS content-specific MIB objects. To find out how you can look up object information, see the section “Using the CSS to Look Up MIB Objects” in this chapter.

Table 9-5 MIB Branches Under the CSS Enterprise MIB

MIB File Name	MIB Module Description	Related CLI Commands
aclExt.mib	The CSS Access Control List clause table.	(config-acl)# ?
ap64Stats.mib	The 64 bit statistical aggregation of RMON (RFC1757), MIB-II (RFC1213) and EtherErrors (RFC1398).	# show rmon ? # show mibii ? # show ether-errors ?
apent.mib	CSS Enterprise MIB branch hierarchy.	_____
apIpv4.mib	MIB support for IPv4 Global Information.	(config)# ip ?
apIpv4Arp.mib	MIB support for IPv4 ARP.	(config)# arp ?
apIpv4Dns.mib	MIB support for IPv4 DNS resolver configuration.	(config)# dns ?
apIpv4Host.mib	MIB support for IPv4 Host table.	(config)# host ?
apIpv4Interface.mib	MIB support for IPv4 Interfaces.	(config-ip)# ?
apIpv4Ospf.mib	MIB support for the Open Shortest Path First (OSPF).	(config)# ospf ?
apIpv4Redundancy.mib	MIB support for IPv4 Redundancy.	(config-ip)# redundancy ?
apIpv4Rip.mib	MIB support for the Routing Information Protocol (RIP).	(config-ip)# rip ?
apIpv4Sntp.mib	MIB support for the Simple Network Time Protocol.	(config)# sntp ?
apIpv4StaticRoutes.mib	MIB support for IPv4 Static Routes.	(config)# ip route ?

Table 9-5 MIB Branches Under the CSS Enterprise MIB (continued)

MIB File Name	MIB Module Description	Related CLI Commands
appExt.mib	MIB support for APP configurations.	(config)# app ?
boomClientExt.mib	Configuration and monitoring of Content Routing Agent (CRA) parameters.	(config)# dns-boomerang client ?
bootExt.mib	MIB support for system boot administration.	(config-boot)# ?
bridgeExt.mib	Configuration and monitoring of bridge-related parameters.	(config)# bridge ?
cappUdpExt.mib	Application Peering Protocol-User Datagram Protocol (APP-UDP) global statistical information and security configuration settings.	(config)# app-udp ?
cctExt.mib	CSS circuit information.	(config)# circuit ?
chassisMgrExt.mib	MIB for the CSS chassis manager.	# show chassis ?
cntdnsExt.mib	Content rule DNS statistics.	(config)# dns hotlist ?
cntExt.mib	Content rule table.	(config-owner-content)# ?
cnthotExt.mib	Content rule hot list.	(config-owner-content)# hotlist ?
cntsvcExt.mib	Monitoring of services attached to content rules.	(config-owner-content)# add service ? (config-owner-content)# remove service ?
csaExt.mib	Configuration and monitoring of Client Side Accelerator (CSA) parameters on a CSS.	(config)# dns-server ?
dnshotExt.mib	DNS hot list.	(config)# domain hotlist ?
dnsServerExt.mib	MIB support for DNS Server.	(config)# dns-server ?
domainCacheExt.mib	Configuration management for the domain cache on the Client Side Accelerator (CSA) in the CSS.	(config)# dns-server domain-cache ?
dqlExt.mib	Domain Qualifier Lists (DQLs).	(config-dql [name])# ?

Table 9-5 MIB Branches Under the CSS Enterprise MIB (continued)

MIB File Name	MIB Module Description	Related CLI Commands
enetExt.mib	Configuration of the PHY state for Ethernet ports.	(config-interface)# phy ?
eqlExt.mib	Extension Qualification Lists (EQLs).	(config-eql [name])#
fileExt.mib	File extensions to support Network Management movement to/from the CSS, and to examine and modify the existing file structure.	_____
flowMgrExt.mib	MIB for the flow manager module.	(config)# flow ?
ftpExt.mib	MIB support for FTP transfer administration records.	(config)# ftp-record ?
grpExt.mib	Configuration of all group-related parameters.	(config-group)# ?
grpsvcExt.mib	Groups attached to services.	(config-group)# add service ? (config-group)# remove service ?
httpExt.mib	MIB support for HTTP transfer administration records.	_____
kalExt.mib	Configuration of keepalive mode.	(config-keepalive)# ?
logExt.mib	CSS logging functionality.	(config)# logging ?
nqlExt.mib	Describes the CSS Network Qualifier Lists (NQLs).	(config-nql [name])# ?
ownExt.mib	Web Host Owner information.	(config-owner)# ?
plucExt.mib	Proximity Lookup Client functionality.	(config)# proximity cache ?
probeRttExt.mib	Tiered Proximity Service RTT Probe Module functionality.	(config)# proximity probe rtt ?
proxDbExt.mib	Tiered Proximity Database functionality. This MIB contains all configuration, statistic, and metric objects.	(config)# proximity db ?
publishExt.mib	Publisher and subscriber services.	(config-service)# publisher ?

Table 9-5 MIB Branches Under the CSS Enterprise MIB (continued)

MIB File Name	MIB Module Description	Related CLI Commands
qosExt.mib	CSS MIB module QOS class definitions (the QOS class of this known piece of content).	
radiusClientExt.mib	CSS extensions to the client side of the Remote Access Dial-in User Service (RADIUS) authentication protocol.	(config)# radius-server ?
schedExt.mib	MIB support for CLI command scheduler records.	(config)# cmd-scheduler ?
securityMgrExt.mib	CSS MIB objects for the Network Security manager.	(config)# username ?
snmpExt.mib	SNMP traps and communities.	(config)# snmp ?
sshdExt.mib	MIB support for the Secure Shell Host server (SSHD).	(config)# sshd ?
subscribeExt.mib	CSS Enterprise subscriber.	(config-service)# subscriber ?
svcExt.mib	Configuration and monitoring of all service-related parameters.	(config-service)# ?
tagExt.mib	Content Tag Lists.	(config)# header-field-group ?
terminalMgmt.mib	MIB support for terminal options.	# terminal ? # restrict ?
urqlExt.mib	Uniform Resource Locator Qualifier Lists (URQL).	(config-urql [name])# ?

