



Manual de Estado de Puertos

Red IIMAS

5 de febrero de 2010

Contenido

1 Introducción.....	3
2 Pre-requisitos.....	3
2.1 Configuración de switch.....	3
3 Recuperar información de puertos.....	4

1 Introducción

El presente documento muestra como se puede obtener la información del estado de los puertos sobre un equipo de la familia ESW 500 Series desde una línea de comandos en un servidor linux usando Ubuntu.

2 Pre-requisitos

Para lograr obtener la información necesaria se debe contar con la suite de herramientas net-snmp disponible en:

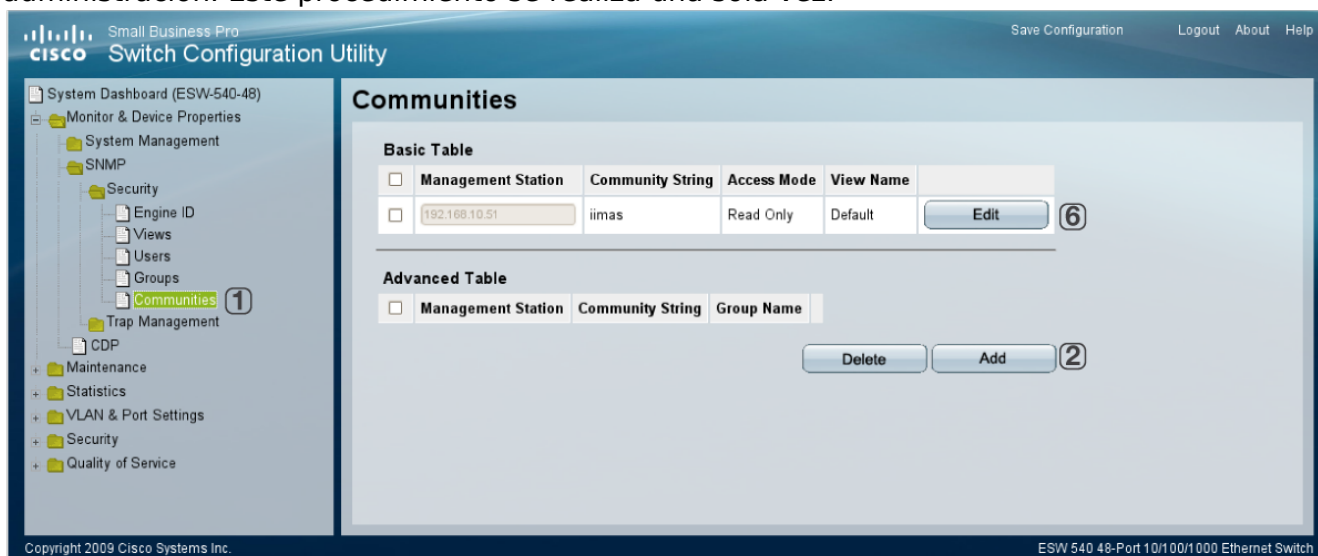
<http://net-snmp.sourceforge.net/>

O mediante el paquete **snmp** en la distribución de Ubuntu y puede ser instalado usando cualquiera de las herramientas de administración de paquetes del sistema (e.g. Synaptic)

Este paquete hace que estén disponibles entre otros los comandos `snmpget` y `snmpwalk`, estos serán los comandos que usaremos para recuperar la información necesaria.

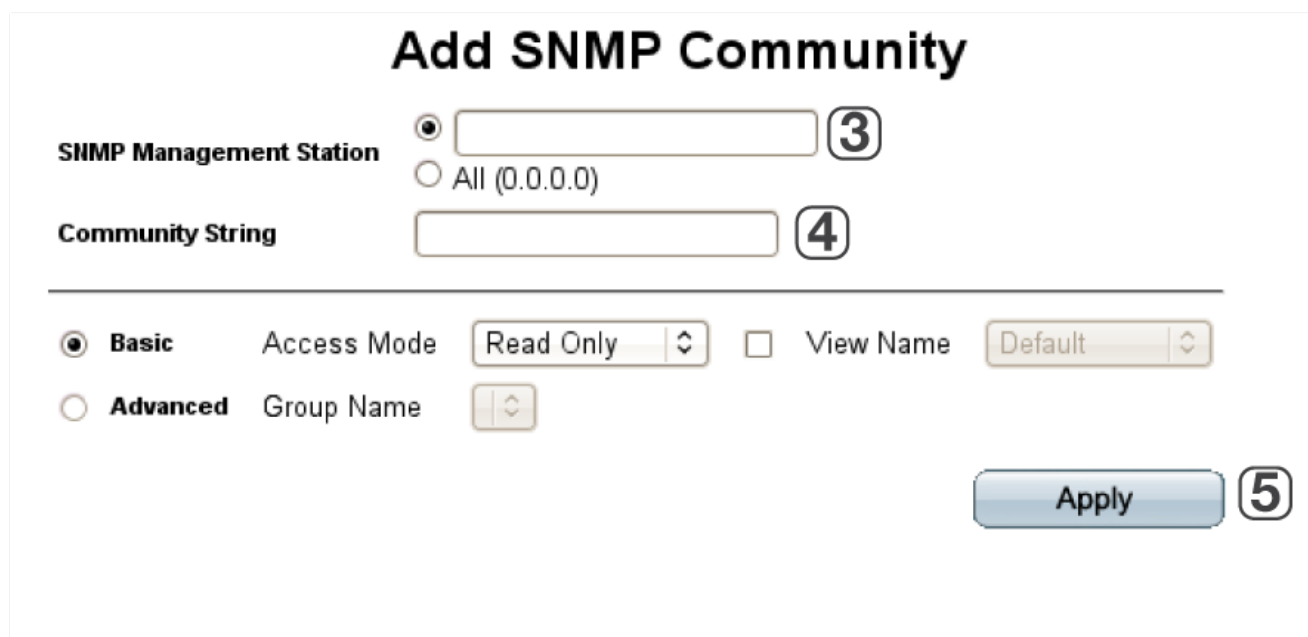
2.1 Configuración de switch

Para poder utilizar SNMP para recuperar el estado de los puertos y otros datos útiles debemos de configurar previamente el dispositivo de nuestro interés. Para realizar esto tenemos que configurar una cadena de comunidad mediante la interfaz gráfica de administración. Este procedimiento se realiza una sola vez.



The screenshot shows the Cisco Small Business Pro Switch Configuration Utility interface. The left sidebar contains a tree view with categories like Monitor & Device Properties, System Management, SNMP, Security, and Trap Management. The 'Communities' option under SNMP is highlighted with a circled '1'. The main content area is titled 'Communities' and contains two tables: 'Basic Table' and 'Advanced Table'. The 'Basic Table' has columns for Management Station, Community String, Access Mode, and View Name. It contains one entry with Management Station '192.168.10.51', Community String 'iimas', Access Mode 'Read Only', and View Name 'Default'. An 'Edit' button is next to this entry, with a circled '6' next to the button. The 'Advanced Table' has columns for Management Station, Community String, and Group Name. It is currently empty, with 'Delete' and 'Add' buttons at the bottom right, and a circled '2' next to the 'Add' button. The top right of the interface has links for 'Save Configuration', 'Logout', 'About', and 'Help'. The bottom of the interface shows 'Copyright 2009 Cisco Systems Inc.' and 'ESW 540 48-Port 10/100/1000 Ethernet Switch'.

El primer paso después de ingresar a la interfaz de administración es seleccionar la pantalla de communities navegando en el arbol de configuración: Monitor & device properties → SNMP → security → Communities (1) una vez ahí debemos de presionar el botón Add (2) y aparecerá el diálogo siguiente:



The image shows a web-based configuration interface titled "Add SNMP Community". It contains the following elements:

- SNMP Management Station:** A radio button selected next to a text input field (labeled 3), and another radio button labeled "All (0.0.0.0)".
- Community String:** A text input field (labeled 4).
- Basic/Advanced:** Two radio buttons, with "Basic" selected.
- Access Mode:** A dropdown menu showing "Read Only".
- View Name:** A checkbox and a dropdown menu showing "Default".
- Group Name:** A dropdown menu (only visible when "Advanced" is selected).
- Apply:** A button (labeled 5) to save the configuration.

En el campo SNMP Management Station debemos de agregar la IP del servidor donde se ejecutará el script de monitoreo (3), en community string (4) pondremos la cadena con la que identificaremos esta comunidad esta debe de corresponder con la que usamos en el comando. Dentro de este manual usaremos iimas como cadena de comunidad. Al terminar presionamos el boton apply (5) y la nueva comunidad configurada aparecerá en la lista de la pantalla anterior (6).

3 Recuperar información de puertos

Para recuperar información del estado de los puertos solo hace falta ejecutar el comando snmpget con los siguientes parámetros:

```
snmpget -v 2c -c <community> <ip-switch> IF-MIB::ifOperStatus.<puerto>
```

Donde <community> es el identificador de la comunidad usado en el paso (4), <ip-switch> es la IP del switch para el que se desea saber el estado y <puerto> el número del puerto cuyo estado se desea conocer. Por ejemplo:

```
snmpget -v 2c -c iimas 192.168.10.2 IF-MIB::ifOperStatus.25
```

Obtiene el estado del puerto 25 en el switch con ip 192.168.10.2 usando la comunidad configurada como iimas.

La respuesta a este comando es

```
IF-MIB::ifOperStatus.25 = INTEGER: down(2)
```

o

```
IF-MIB::ifOperStatus.25 = INTEGER: up(1)
```

dependiendo del estado del puerto, como se muestra en la siguiente captura del comando antes y después de desconectar el dispositivo en el puerto correspondiente.

```
vaz@varda:~$ snmpget -v 2c -c iimas 192.168.10.2 IF-MIB::ifOperStatus.25
IF-MIB::ifOperStatus.25 = INTEGER: up(1)
vaz@varda:~$ snmpget -v 2c -c iimas 192.168.10.2 IF-MIB::ifOperStatus.25
IF-MIB::ifOperStatus.25 = INTEGER: down(2)
vaz@varda:~$ █
```

Si deseamos recuperar la información del estado de todos los puertos podemos usar el comando:

```
snmpwalk -v 2c -c iimas 192.168.10.2 IF-MIB::ifOperStatus
```

Que produce la salida mostrada en la captura de la siguiente página. Se debe tener precaución ya que se muestran interfaces adicionales correspondientes a los puertos SFP y vlans configuradas.

```
vaz@varda:~$ snmpwalk -v 2c -c iimas 192.168.10.2 IF-MIB::ifOperStatus
IF-MIB::ifOperStatus.1 = INTEGER: up(1)
IF-MIB::ifOperStatus.2 = INTEGER: down(2)
IF-MIB::ifOperStatus.3 = INTEGER: down(2)
IF-MIB::ifOperStatus.4 = INTEGER: down(2)
IF-MIB::ifOperStatus.5 = INTEGER: down(2)
IF-MIB::ifOperStatus.6 = INTEGER: down(2)
IF-MIB::ifOperStatus.7 = INTEGER: down(2)
IF-MIB::ifOperStatus.8 = INTEGER: down(2)
IF-MIB::ifOperStatus.9 = INTEGER: down(2)
IF-MIB::ifOperStatus.10 = INTEGER: down(2)
IF-MIB::ifOperStatus.11 = INTEGER: down(2)
IF-MIB::ifOperStatus.12 = INTEGER: down(2)
IF-MIB::ifOperStatus.13 = INTEGER: down(2)
IF-MIB::ifOperStatus.14 = INTEGER: down(2)
IF-MIB::ifOperStatus.15 = INTEGER: down(2)
IF-MIB::ifOperStatus.16 = INTEGER: down(2)
IF-MIB::ifOperStatus.17 = INTEGER: down(2)
IF-MIB::ifOperStatus.18 = INTEGER: down(2)
IF-MIB::ifOperStatus.19 = INTEGER: down(2)
IF-MIB::ifOperStatus.20 = INTEGER: down(2)
IF-MIB::ifOperStatus.21 = INTEGER: down(2)
IF-MIB::ifOperStatus.22 = INTEGER: down(2)
IF-MIB::ifOperStatus.23 = INTEGER: down(2)
IF-MIB::ifOperStatus.24 = INTEGER: down(2)
IF-MIB::ifOperStatus.25 = INTEGER: down(2)
IF-MIB::ifOperStatus.26 = INTEGER: up(1)
IF-MIB::ifOperStatus.27 = INTEGER: down(2)
IF-MIB::ifOperStatus.28 = INTEGER: down(2)
IF-MIB::ifOperStatus.29 = INTEGER: down(2)
IF-MIB::ifOperStatus.30 = INTEGER: down(2)
IF-MIB::ifOperStatus.31 = INTEGER: down(2)
IF-MIB::ifOperStatus.32 = INTEGER: down(2)
IF-MIB::ifOperStatus.33 = INTEGER: down(2)
IF-MIB::ifOperStatus.34 = INTEGER: down(2)
IF-MIB::ifOperStatus.35 = INTEGER: down(2)
IF-MIB::ifOperStatus.36 = INTEGER: down(2)
IF-MIB::ifOperStatus.37 = INTEGER: down(2)
IF-MIB::ifOperStatus.38 = INTEGER: down(2)
IF-MIB::ifOperStatus.39 = INTEGER: down(2)
IF-MIB::ifOperStatus.40 = INTEGER: down(2)
IF-MIB::ifOperStatus.41 = INTEGER: down(2)
IF-MIB::ifOperStatus.42 = INTEGER: down(2)
IF-MIB::ifOperStatus.43 = INTEGER: down(2)
IF-MIB::ifOperStatus.44 = INTEGER: down(2)
IF-MIB::ifOperStatus.45 = INTEGER: down(2)
IF-MIB::ifOperStatus.46 = INTEGER: down(2)
IF-MIB::ifOperStatus.47 = INTEGER: down(2)
IF-MIB::ifOperStatus.48 = INTEGER: down(2)
IF-MIB::ifOperStatus.1000 = INTEGER: notPresent(6)
IF-MIB::ifOperStatus.1001 = INTEGER: notPresent(6)
IF-MIB::ifOperStatus.1002 = INTEGER: notPresent(6)
IF-MIB::ifOperStatus.1003 = INTEGER: notPresent(6)
IF-MIB::ifOperStatus.1004 = INTEGER: notPresent(6)
IF-MIB::ifOperStatus.1005 = INTEGER: notPresent(6)
IF-MIB::ifOperStatus.1006 = INTEGER: notPresent(6)
IF-MIB::ifOperStatus.1007 = INTEGER: notPresent(6)
IF-MIB::ifOperStatus.100000 = INTEGER: up(1)
vaz@varda:~$ █
```