



## Cisco Remote Inventory Module

## ***Remote Inventory Module***

OpenOffice/PDF Version

1<sup>st</sup> Edition , 21 Oct 2011

© Artica Soluciones Tecnológicas 2005-2011

© Javier Lanz <[javier.lanz@artica.es](mailto:javier.lanz@artica.es)>

These remote inventory modules are meant to gather information from a Cisco machine through ICMP requests.

## 2 REQUERIMENTS

As every system is different regarding the specific packages and the mibs functionality, the user should refer to his/her distribution manual or contact with the system administrator.

## 2.1. Ubuntu

Due licenses reasons, it is a must to edit `/etc/snmp/snmp.conf` file, commenting the first line, letting the file as below.



## 3 PANDORA FMS CONFIGURATION

---

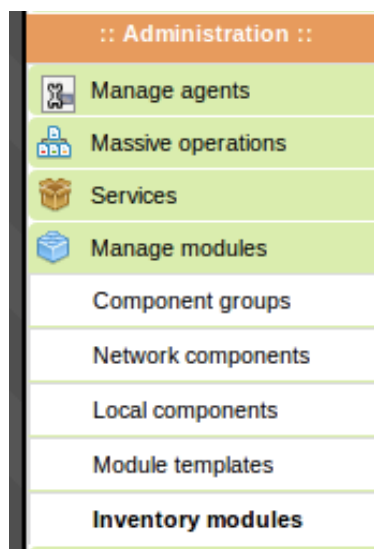
First of all, it is needed to define the remote inventory module for each checking to do.

Once defined, it will be associated to an agent, so the information will be gathered.

### 3.1. Creating a inventory module

---

Administration > Manage modules > Inventory modules



Once in this screen, it will be displayed every inventory modules already defined on Pandora FMS.

Module management » Inventory modules

<b>Name</b>	<input type="text" value="Cisco CPU"/>
<b>Description</b>	<input type="text" value="Gets CPU info from a Cisco Swit"/>
<b>OS</b>	<input type="text" value="Cisco"/>
<b>Interpreter</b>	<input type="text" value="/usr/bin/perl"/>
<b>Format</b> ★	<input type="text" value="IP;Hardware;SerialNumber;IOS"/>
<b>Code</b>	<pre>#!/usr/bin/perl #Informacion del Equipo #IP;Hardware;SerialNumber;IOS  my \$target_ip = \$ARGV[0]; my \$community = \$ARGV[1];  \$result = ""; #\$result .= `snmpget -v2c -c \$community -Ovq \$target_ip sysName.0` . "\n"; \$result .= \$target_ip . "\n";  \$error = ""; \$error .= `snmpget -v2c -c \$community -Ovq \$target_ip mib-2.47.1.1.1.2.1   grep exists`;  \$serial = ""; \$serial .= `snmpget -v2c -c \$community -Ovq \$target_ip mib-2.47.1.1.1.11.1`; \$serial =~ s/^\n &lt; &gt; &amp; \\ / /g;  if (\$error eq "" &amp;&amp; \$serial ne "") {     \$result .= `snmpget -v2c -c \$community -Ovq \$target_ip mib-2.47.1.1.1.2.1` . "\n";     \$result .= `snmpget -v2c -c \$community -Ovq \$target_ip mib-2.47.1.1.1.11.1` . "\n"; } else {  \$VSS = ""; \$VSS = `snmpget -v2c -c \$community -Ovq \$target_ip mib-2.47.1.1.1.2.1   grep VSS`;</pre>

Clicking in the Create button, we will access the next screen, in which it is possible to create an new remote inventory module.

- Name: Module name
- Description: Just a description
- OS: Operating system in which to get the information
- Interpreter: Interpreter with witch to execute the script
- Format: Define here the headers to display the output
- Code: the script code itself.

### 3.2. CPU Remote Inventory module

---

Here is described which data is required in every field to define this module and so get the proper information from a Cisco machine.

- Name: Whichever you want
- Description: Just a description.
- OS: **Must be: Cisco**
- Interpreter: **Must be exactly: /usr/bin/perl**
- Format: **Must be: IP;Hardware;SerialNumber;IOS**
- Code: get the code from the file **cisco\_CPU.pl**

### 3.3. Hardware Remote Inventory module

---

Here is described which data is required in every field to define this module and so get the proper information from a Cisco machine.

- Name: Whichever you want
- Description: Just a description.
- OS: **Must be: Cisco**
- Interpreter: **Must be exactly: /usr/bin/perl**
- Format: **Must be: Slot;PartNumber;SerialNumber**
- Code: get the code from the file **cisco\_Hardware.pl**

### 3.4. Ports Remote Inventory module

---

Here is described which data is required in every field to define this module and so get the proper information from a Cisco machine.

- Name: Whichever you want
- Description: Just a description.
- OS: **Must be: Cisco**
- Interpreter: **Must be exactly: /usr/bin/perl**
- Format: **Must be: Interface;Description;AdminStatus;OperStatus**

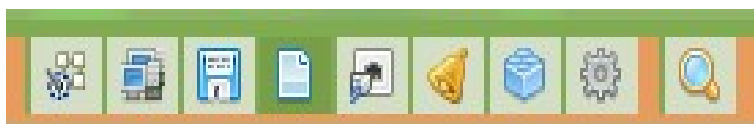
- Code: get the code from the file **cisco\_Ports.pl**

### 3.5. Adding the remote inventory module to an agent

Once defined the remote inventory module, is time to associate it to an agent, in order to get the information from the Cisco machine.

**The selected agent must be a Cisco agent**, in order to be able to select the previously defined remote inventory modules.

In the Administration view from the agent, we need to select the Inventory tab to get the screen in which to associate them.



So we will access the next screen

The next configuration is suitable for the three remote inventory modules, CPU, Hardware and Ports remote inventory module.

The fields are:

- Module: Select here the previously defined remote inventory module.
- Target: Here is written the **IP** of the machine to point at. Should detect automatically the IP of the Agent selected. If not, check it that agent has an IP associated.
- Interval: Select here how often you wish to execute this module.
- Username: in that field we will define the required **community**.
- Password: for our case, we will leave that field **in black**.

## 4 INVENTORY VIEW

Accessing through Operation > Inventory it is possible to check the result from the changes we have made.

Next is displayed an screenshot for our configuration.

Cisco CPU - (2011-10-21 12:38:43)				
Agent	IP	Hardware	SerialNumber	IOS
Switch Cisco	192.168.50.250	Cisco Catalyst c2950 switch with 24 10/100 BaseTX ports	FOC0644X0WV	IOS (tm) C2950 Software (C2950-I6Q4L2-M), Version 12.1(22)EA2, RELEASE SOFTWARE (fc1)

Cisco Hardware - (2011-10-21 12:38:43)				
Agent	Slot	PartNumber	SerialNumber	
Switch Cisco	almendra	WS-C2950-24	FOC0644X0WV	

Cisco Ports - (2011-10-21 12:38:43)				
Agent	Interface	Description	AdminStatus	OperStatus
Switch Cisco	FastEthernet0/1		up	up
	FastEthernet0/2		up	up
	FastEthernet0/3		up	down
	FastEthernet0/4		up	up
	FastEthernet0/5		up	up
	FastEthernet0/6		up	down
	FastEthernet0/7		up	down
	FastEthernet0/8		up	up
	FastEthernet0/9		up	down
	FastEthernet0/10		up	up
	FastEthernet0/11		up	down
	FastEthernet0/12		up	down
	FastEthernet0/13		up	down
	FastEthernet0/14		up	up
	FastEthernet0/15		up	down
	FastEthernet0/16		up	up

As we can see, in every Remote inventory module, it is shown the agent in which the data is stored, and the next columns are the specific headers that we defined previously in every remote inventory module.