

PANDORAFMS
E N T E R P R I S E

Pandora FMS
Administrator Manual
Monitoring Solaris Systems



Administrator Manual Monitoring Solaris Systems

© Artica Soluciones Tecnológicas 2005-2012

Index

1Changes.....	3
2Introduction.....	4
3Compatibility Matrix.....	5
4Compulsory documentation to hand in by the area that requires the monitoring	6
5Monitoring Modules.....	7
6Monitoring Modules Definition.....	8
6.1.General Modules.....	8
6.1.1.CPU User.....	8
6.1.2.CPU System.....	8
6.1.3.Process Running.....	9
6.1.4.Physical Free Memory.....	9
6.1.5.Used space	9
6.1.6. Number processes.....	10
6.1.7.Zombie processes.....	10
6.1.8.Process memory.....	10
6.1.9.Process CPU.....	11
6.1.10.NTPD Instances.....	11



1 CHANGES

Date	Autor	Change	Version
17/03/12	Dario	Plugin first version	v1r1

2 INTRODUCTION

This document has as main objective the monitoring description of any machine with the Solaris Operative system.

A serie of modules have been chosen according to our experience in system monitoring and the requirements of some of our clients.

To extract the information we use the recollection layer for the Pandora FMS modules that are are susceptible to it, reducing this way the load of the systems to monitor. Or they have been added through *module_exec* if it is a command or through *module_plugin* if we talk about an agent plugin.

3 COMPATIBILITY MATRIX

The compatibility matrix for the plugin is shown here:

Systems where it has been tested	<ul style="list-style-type: none"> • SunOs 5.x
---	---

Systems where it should work	<ul style="list-style-type: none"> • Same system or higher
-------------------------------------	---



4 COMPULSORY DOCUMENTATION TO HAND IN BY THE AREA THAT REQUIRES THE MONITORING

- The user that executes the Pandora FMS agent should have the necessary permissions to execute the commands and have access to the necessary files without any problems.
- The requirements to this monitoring works right are the following:
 - Agente Pandora FMS instalado
- Information for the modules:
 - Services to monitor
 - Disks to monitor

5 MONITORING MODULES

- **CPU User:** This module returns the CPU percentage that is occupied by user processes
- **CPU System:** This module returns the CPU percentage occupied by system processes.
- **Process Running:** The module searches if an specific process is running.
- **Physical Free Memory:** The module returns the percentage of free physical memory.
- **Used space:**The module returns the percentage of occupied directory.
- **Number processes:** Returns the number of processes that are running in the machine.
- **Zombie processes:** Returns the number of zombies processes.
- **Process memory:** The module returns the percentage of use memory by one process.
- **Process CPU:** The module returns the CPU use percentage by one process.
- **NTPD Instances:** The module returns the number of instances of the ntpd process that is running.

6 MONITORING MODULES DEFINITION

Next are described the different monitoring modules, and also the code of each one of them.

6.1. General Modules

6.1.1. CPU User

This module returns the percentage of CPU occupied by user processes. The syntax for the modules would be this:

```
module_begin
module_name CPU User
module_type generic_data
module_exec vmstat 1 2 | tail -1 | awk '{ print $20 }' | tr -d "\n"
module_max 100
module_min 0
module_description CPU (%) used by user
module_end
```

6.1.2. CPU System

This module returns the percentage of CPU occupied by system processes. The syntax for the module would be this:

```
module_begin
module_name CPU System
module_type generic_data
module_cpusage all
module_max 100
module_min 0
module_description CPU (%) used by system
module_end
```


6.1.3. *Process Running*

The module searches if a specific process is running. The module syntax is the following:

```
module_begin
module_name Process Running xxx
module_type generic_proc
module_service xxx
module_description Proces xxx Running
module_end
```

We should replace *xxx* by the specific process that we want to monitor.

6.1.4. *Physical Free Memory*

The module returns the percentage of free physical memory. The module syntax will be the following:

```
module_begin
module_name Physical Free Memory
module_type generic_data
module_freepcentmemory
module_description Porcentaje Memoria física libre
module_end
```

6.1.5. *Used space*

The module returns the percentage of occupied directory, the syntax is the following:

```
module_begin
module_name Used space xxx
module_type generic_data
module_occupiedpercentdisk /xxx
module_description % Directorio ocupado
module_end
```

We should change *xxx* by the volume to monitor.

6.1.6. *Number processes*

It returns the number of processes that are running in the machine. The module syntax is the following:

```
module_begin
module_name Number processes
module_type generic_data
module_exec ps -e | wc -l | tr -d "\n" | tr -d " "
module_description Number processes
module_end
```

6.1.7. *Zombie processes*

It returns the number of zombie processes. The syntax is the following:

```
module_begin
module_name Zombie processes
module_type generic_data
module_exec ps -el | awk '$2~/\"Z\"/{print $2}' | wc -l | tr -d "\n" | tr -d " "
module_description Zombie processes
module_end
```

6.1.8. *Process memory*

The module returns the percentage of memory in use by one process. The syntax is the following:

```
module_begin
module_name Process memory xxx
module_type generic_data
module_memproc xxx
module_description % used memory of process
module_end
```

We should change xxx by the process to monitor.

6.1.9. Process CPU

The module returns the percentage of CPU used by one process. The syntax is the following:

```
module_begin
module_name Process CPU xxx
module_type generic_data
module_cpuproc xxx
module_description % CPU used by process
module_end
```

We change xxx by the process to monitor

6.1.10. NTPD Instances

The module returns the number of instances of the ntpd process that are running.

```
module_begin
module_name NTPD Instances
module_type generic_data
module_exec ps -Af | grep ntpd | grep -v "grep" | wc -l | tr -d " " | tr -d "\n"
module_description Instancias NTPD en ejecución
module_end
```