PANDORAFMS
ENTERPRISE

Pandora FMS Administrator Manual Monitoring Windows Systems

Administrator Manual Monitoring Windows Systems

Index

1 Changes

Fecha	Autor	Cambio	Versión
17/03/12	Dario	Plugin First Version	v1r1

2 Introduction

This document has as main objective the description of the monitoring of any machine with the Windows Operative System.

Some modules have been chosen according with our experience in system monitoring and the requirements of some of our clients.

To extract the information we use these:

- Collection layer for the Pandora FMS modules that are susceptible to it, reducing the load of the systems to monitor.
- Modules kind *module_exec* if it is command.
- Modules kind *module_plugin* if if is an agent plugin.

3 Compatibility Matrix

The matrix compatibility for the plugin is shown next:

Systems where it has been tested	• Windows 2003 Server • Windows 2008 Server • Windows XP

Systems where it should work	• Same system or higher

4 Compulsory documentation to hand in by the area that requires the monitoring

The information the area that requires the monitoring should hand in, is the following one:

- 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 installed.
- Information for the modules:
  - Services to monitor
  - Disks to monitor

5 Plugin modules

- **CPU Load:** This module returns the percentage of the CPU in use.
- **Number processes:** Module that returns the number of system processes.
- **Free Memory:** Module that returns the percentage of the system free memory
- **Disk discovery:** Plugin that does a disk selfdiscovery and monitor its occupied space.
- **WMI Service:** Module that checks if the WMI service is active.
- **SNMP Service:** Module that checks if the SNMP service is active.
- **Domain Auth. Fail:** This module search the event that informs about one failure in the domain authentication.
- **Disk Structure Corrupted:** This module searchs the event that informs about one fail in the disk structure.
- **No domain controller:** This module search the event that informs that no controler has been found for the domain.
- **Account blocked:** This module serachs the event that informs that the account is blocked.
- **Account deleted:** This module searchs the event that informs that the account has been deleted.
- **Account disabled:** This module search the event that informs that the account has been disabled.
- **Password reset:** Thsi module searchs the event that informs that the password has been reseted
- **Password change:** This module searchs the event that informs that the the password has been changed.
- **Account enabled:** This module searchs the event that informs that the account has been enabled.

6 Modules Definition

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

6.1. General Modules

6.1.1. CPU Load

This module returns the percentage of the CPU in use. The syntaxis for the module would be the following:

```
module_begin
module_name CPU Load
module_type generic_data
module_wmiquery SELECT LoadPercentage FROM Win32_Processor
module_wmicolumn LoadPercentage
module_description CPU Load (%)
module_min_warning 80
module_max_warning 90
module_min_critical 91
module_max_critical 100
module_end
```

6.1.2. Number processes

Module that returns the number of processes of the system. Its syntax is this:

```
module_begin
module_name Number processes
module_type generic_data
module_exec tasklist | gawk "NR > 3 {print$0}" | wc -l
module_description Number of processes running
module_min_warning 175
module_max_warning 249
module_min_critical 250
module_max_critical 300
module_end
```

6.1.3. Free Memory

Module that returns the percentage of the system free memory. Its syntax is the following:

```
module_begin
module_name Free Memory
module_type generic_data
module_freepercentmemory
module_description Free memory (%).
module_min_warning 21
module_max_warning 30
module_min_critical 0
module_max_critical 20
module_end
```

6.1.4. Disk discovery

Plugin that does a disk self discovery and monitor its occupied space. Its syntax is the following one:

```
module_plugin cscript.exe //B "%ProgramFiles%\Pandora_Agent\util\df.vbs"
```

6.1.5. WMI Service

Module that checks if the WMI service is active. Its syntax is the following:

```
module_begin
module_name WMI Service
module_type generic_proc
module_service winmgmt
module_description WMI Service enabled
module_end
```

6.1.6. SNMP Service

Module that checks if the SNMP service is active. Its syntax is the following:

module_begin
module_name SNMP Service
module_type generic_proc
module_service SNMP
module_description SNMP Service enabled
module_end

6.2. Event Modules

6.2.1. Domain Auth. Fail

This module searchs the event that informs about one failure in the domain authentication. The syntax for the module would be the following:

module_begin
module_name Domain Auth Fail
module_type async_string
module_logevent
module_source System
module_eventcode 3210
module_description Domain Authentication Failure
module_end

6.2.2. Disk Structure Corrupted

This module search the event that informs about a failure in the disk structure. The syntaxis for the module would be this:

module_begin
module_name Disk Structure Corrupted
module_type async_string
module_logevent
module_source System
module_eventcode 55
module_application Ntfs
module_description Disk structure corrupted
module_end

6.2.3. No domain controller

This module searchs the event that informs that no controler has been found for the domain.The syntax for for the module would be this:

module_begin
module_name No domain controller
module_type async_string
module_logevent
module_source System
module_eventcode 5719
module_aplication NETLOGON
module_description Domain controller not found
module_end

6.2.4. Account blocked

This module searchs the event that informs that the account is blocked. The syntax for the module would be the following:

module_begin
module_name Account blocked
module_type async_string
module_logevent
module_source Security
module_eventcode 4740
module_description Account blocked
module_end

6.2.5. Account deleted

This module searchs the event that informs that the account has been deleted. The syntax for the module would be this:

module_begin
module_name Account deleted
module_type async_string
module_logevent
module_source Security
module_eventcode 4726
module_description Account deleted
module_end

6.2.6. Account disabled

This module searchs the event that informs that the account has been disabled. The syntax for the module would be this:

```
module_begin
module_name Account disabled
module_type async_string
module_logevent
module_source Security
module_eventcode 4725
module_description Account disabled
module_end
```

6.2.7. Password reset

This module searchs the event that informs that the password has been reseted. The syntax for the module would be the following:

```
module_begin
module_name Password reset
module_type async_string
module_logevent
module_source Security
module_eventcode 4724
module_description Password reset
module_end
```

6.2.8. Password change

This module searchs the event that informs that the password has been changed. The syntax for the modules would be this:

```
module_begin
module_name Password change
module_type async_string
module_logevent
module_source Security
module_eventcode 4723
module_description Password change
module_end
```

6.2.9. Account enabled

This module searchs the event that informs that the account has been enabled. The syntax for the module would be the following:

module_begin
module_name Account enabled
module_type async_string
module_logevent
module_source Security
module_eventcode 4722
module_description Account enabled
module_end