

PANDORAFMS
E N T E R P R I S E

Pandora FMS

FTP Monitoring Administrator Manual



FTP Monitoring Administration Manual

© Artica Soluciones Tecnológicas 2005-2012

Index

1Changelog.....	3
2Introduction.....	4
3Requirements.....	5
4Compatibility Matrix.....	6
5Software agent modules generated.....	7
6Installation.....	8
7Monitoring.....	9



1 CHANGELOG

Date	Author	Change	Version
12/12/12	Mario P.	First Version	v1r1
24/06/13	Tomas Palacios	Revision	v1r2

2 INTRODUCTION

This document has as main objective the description of the generic monitoring of downloads and uploads times and rates, besides the file modification in FTP servers.

3 REQUIREMENTS

In order to function properly:

- To do the configuration in a configuration file, to which the plugin would have access (is given to it as parameter).
- To have access from the agent where plugin would be installed to the Ftp server that is going to be monitored.
- To have installed Perl in the system and the libraries that are needed: Net::FTP y Time::HiRes(installed by default in the Perl and ActivePerl last versions).

4 COMPATIBILITY MATRIX

The agent compatibility matrix is the following:

Systems where it has been checked	<ul style="list-style-type: none"> • Linux (SUSE, Ubuntu...) • Windows 2003 Server with ActivePerl
Systems where it should work	<ul style="list-style-type: none"> • Same system or higher

5 SOFTWARE AGENT MODULES GENERATED

The Plugin will create 5 modules in its execution

- *PUT_file_transfer_time*
- *PUT_file_transfer_rate*
- *GET_file_transfer_time*
- *GET_file_transfer_time*
- *FTP_Maching_files*

6 INSTALLATION

Copy the plugin to the agent plugin directory, or distribute it with file collections. Do the same with the conf.file. The call from the agent will be similar to this, but using the paths where is installed the plugin and the conf.

Administered Mode:

```
module_plugin perl /etc/pandora/plugins/plugin_ftp.pl /etc/pandora/plugins/ftp.conf
```

Managed Mode:

```
module_plugin perl /etc/pandora/plugins/plugin_ftp_gest.pl  
/etc/pandora/plugins/ftp_gest.conf <ftp_server_ip> <ftp_user> <ftp_password>  
<transfer_file> <file_client_path> <file_server_path> <OS (Unix, Windows)>
```

In the managed mode, each one of the required parameters could be provided through the use of the fields of the corresponding module when auto provisioning the plugin from the metaconsole.

In the plugin configuration file given as example at the end of this document, we describe each one of the definable parameters both of the managed parts and the administered ones of the plugin.

It is important to say, that in the managed mode, the plugin configuration file is built automatically to meet the needs that we have previously mentioned with the provided parameters, but in the conf shown path there should be one file with this name before starting overwriting.

7 MONITORING

The plugin has as objective the monitoring of a FTP server from a Unix or a Windows agent.

The modules that we are going to extract from the plugin are five:

- *PUT_file_transfer_time*

It shows the time that takes to upload to the FTP server the file that we have introduced in the configuration file. To do this it uses a file that we should locate on the agent system and with the permissions required in order to could send it to the FTP server.

- *PUT_file_transfer_rate*

It shows the uploading B/s from the local file to the FTP server.

Same as with the previous one, we should locate in on the agent system and with the permissions required to could send it to the FTP server.

- *GET_file_transfer_time*

It shows the amount of time that it takes to download from the FTP server the file that we have configured in the configuration file.

- We should introduce in the configuration file the data of the file that we want to download and the name with which we want to save it in the configuration file

- *GET_file_transfer_time*

It shows the download B/s from the FTP server to the agent.

- *FTP_Maching_files*

It compares two files, one in the agent system and the other in the FTP server and it shows if it has been done changes in it or not. In that same changes have been made in the configuration file, we should specify if we want that the file that is located in local would be overwritten, or, on the contrary, we want that the module would be in critical status until we modify it manually.

The plugin comes parameterized by an external configuration file. This configuration file has several parameters that are explained in the following example:

```
# FTP Plugin Config file example

#=====
#----- FTP parameters -----
#=====

# User name of FTP server
conf_ftp_user username

# FTP user password
conf_ftp_pass password

# IP FTP server
conf_ftp_host localhost

# File name to upload at FTP server (Complete PATH)
conf_ftp_putfile /home/user/example.zip

# File name upload in FTP server
conf_ftp_putname example.zip

# # File name to download at FTP server
conf_ftp_getfile test.zip

# Download file name in our agent
conf_ftp_getname test.zip

# Operating system (Unix or Windows)
conf_operating_system Unix

# Options for the files to be compared.
#      1.- To change the file name to download the new one in the event that
#changed write---> write
#      2.- To not change the file name to download---> notwrite
conf_ftp_compare write

# Filename to compare in FTP server
conf_ftp_compare_file prueba.zip

# Filename to compare in our system (Complete PATH)
conf_local_comp_file prueba.zip

# Name that we will give the file you downloaded to compare with the file you have
locally. Advisable to introduce it in the / tmp folder. (Full Path)
conf_local_downcomp_file /tmp/prueba.zip
```