Pandora v1.2

***Pandora* v1.2**

First Edition(v1.2) Edition

Published October 12th, 2006

Revision History

Revision 0.1 12 Oct 2006
First draft for review.

Table of Contents

Chapter 1. Documentation for developers: pandora log engine

1.1. Notes for this chapter

This chapter is documentation for developers of the Pandora project, and its main objective is to share information about the inners workings of the code.

1.2. Introduction

Till version 1.2 beta2, pandora was able to treat several types of data, including numeric, incremetal numeric and strings. However, string data type lacks some funcionalities for monitoring log files.

The aim of this pack of code, or development branch, is to strength the capabilities of Pandora for monitoring text files that are increased and rotated with the time, like log files.

In particular, the main objectives of this development branch are:

- pandora agents should be able to analize new lines of a text log file, and only the new ones
- pandora agents should treat each new line as a independent element/data unit
- pandora agents need to be aware about rotation of log files in order to not lose information
- alerts capabilities on string data types has to be improved
- graphic representation capabilites on string data modules has to be improved

1.3. Architecture

Funcionalities described in the Introduction chapter can be implemented in several ways. The approach presented in this document propose that the pandora agent can control log files, detect which new lines are created since its last execution, and process differents modules on EACH of the lines.

On this branch of development, pandora_agent.sh calls to another script just before copying data files to the server. This script is a simple perl script named pandora_agent_log.pl, that uses another file as a configuration file, pandora_agent_log.conf.

pandora_agent_log.pl and pandora_agent_log.conf are very similar to pandora_agent.sh and pandora_agent.conf. In the future, its funcionalities are intended to be included in the agent code. Now, they are presented sepparated just for clarity in the development.

1.4. pandora_agent_log.pl

Some of the code used for this script, mainly the index files manegement, is based in the part of the code of pandora_server/bin/ pandora_snmpconsole.pl

this script performs the following actions in the following order:

- load the pandora_agent_log.conf configuration file, that is described later in this chapter
- for each log file to be monitored loops through the next steps
- loads or creates an index file. Each index file stores information regarding the state of the log file in the last execution of the agent. Indexes are stored in $PANDORA_HOME
- the script make some checks over the log file to see if it has been rewritten and/or rotated. If it has been rotated, rotated log file is processed first to recover last lines.
- loops for the NEW lines of the log (since last agent's execution). For each line, a data file in data_out is created.
- each module associated with that log file is executed against the new log line, and data is written in the corresponding data file
- checksum is performed on data files

1.5. pandora_agent_log.conf

the structure of this file is the same as the structure of pandora_agent.conf, but some extensions has been added.

- module_log [LOG FILE] : only the modules with "module_log" are considered. [LOG FILE] is the file, path included, of the log to be analysed. Different modules can be associated to the same log file. A module can only be associated (for now) to a log file.
- module_log_timestamp : timestamp of the data file can be rewritten using this module. The overriding timestamp is the result of processing 'module_exec' on the log line. Modules with module_log_timestamp are not further considered as pandora modules. So, they require no name, description, data type, ...
- module_log_rotated [LOG FILE] : tells the agent what wil be the name of the rotated log file. Everytime that the agent detects a rotation in the main log file, it will analize the last lines of [LOG FILE]. You don't need to put it in every module associated with a log file: in one is enough.

- module_exec [EXPR] : expression to be executed on every new line for this module. For now, EXPR is a perl expression. You can use the variable $LINE to represent the new log line. F.ex., 'module_exec $LINE =~ y/A-Z/a-z/; return $LINE;' lowercase the log lines before be stored in pandora database.
- module_store_all_data : the default behaviour of pandora is not to store in the database repeated values captures by the agent. This options override that behaviour, forcing pandora to store ALL data. NOTE: please note that this parameter can be used with all kind of modules, not just module_log ones.

1.6. alert configuration on generic_data_string modules

From Pandora 1.2 beta 3, it is going to be possible to configure alerts on generic_data_string modules using perl regular expressions.

NOTE: please note that, although this feature has been development in this development branch, it can be used in all generic_data_string modules, not only in the module_log ones.

For the configuration of this feature, a new field has been added to the "Alert association form" of the pandora console: "Perl expression". A succesful matching of the generic_data_string data will trigger an alert.

Regular expressions has to have Perl syntax, for example:

- word : matches the word "word"
- ^# : matches lines beginning with '#'
- (\d{1,3}\.){3}\d{1,3} : matches IP addresses

1.7. Example

Next pandora_agent_log.conf performs the following actions:

- monitors log file /tmp/log1.log with two modules. First one, returns the line, and second one makes a simple substitution.
- all log lines of /tmp/log1.log are stored and displayed in database, even repeated and consecutives ones.
- for /tmp/log1.log, a rotated file is configured, /tmp/log1.log.0
- /tmp/log2.log log file is also monitored. A generic_data module is configured.

- for /tmp/log2.log, a fixed timestamp is forced, so all data is registered in database as captured '2006/9/25 1:1:1'

1.7.1. pandora_agent_log.conf, an example

module_begin

module_name log1

module_descripcion log 1 log file

module_log /tmp/log1.log

module_log_rotated /tmp/log1.log.0

module_store_all_data

module_type generic_data_string

module_exec return $LINE

module_end

module_begin

module_name log1_subst

module_descripcion simple substitution in log 1

module_log /tmp/log1.log

module_type generic_data_string

module_exec $LINE =~ s/o/X/g; return $LINE

module_end

module_begin

module_name log2

module_description log2 - generic_data

module_log /tmp/log2.log

module_type generic_data

module_exec return $LINE;

module_end

module_begin

module_log /tmp/log2.log

module_log_timestamp

module_exec return ’2006/9/25 1:1:1’;

module_end

1.8. todo list

any ideas?