PANDORAFMS

Considerations on Plugins Development

18-09-2021

Considerations on Plugins Development

Go back to Pandora FMS documentation index

We are working on the translation of the Pandora FMS documentation. Sorry for any inconvenience.

Considerations on Plugin Development

Introduction

The plugins allows to Pandora get information that requires a complex process or that requires the use of complex systems or APIs. Examples of plugins could be the Oracle database monitoring that requires a complex process for the monitoring and also some auto-discovery tasks. Other example could be a simple HTML parse, but that requires some that Goliat can't do.

Differences in Implementation and Performance

Pandora offers two possibilities when executing plugins: execution in the agent or in the server.

The server plugins do independent executions to collect each information piece. The server plugin execution is very difficult so it is only possible for plugins that aren't heavy, this is, that doesn' t need several queries to get a single piece of information. A server plugin could be an specific HTML parse plugin that doesn't requires lot of queries and so it will not overload the server.

The agent plugins allow to obtain several modules at the same time and for this reason they are much more flexible than the server plugins. They are perfects for plugins that need several queries to get an information piece, so they allow more flexibility to the programmer so it's possible to return several modules at the same time.

Recon Tasks

To do recon tasks on plugins that need it, we have two possibilities:

The first one consists on using the the Pandora server Recon server. To do this, it will be necessary to create the ad-hoc code for the specific technology or situation.The Recon Tasks loads the Pandora server, so,if for doing the recon task are necessary lot of data requests, this option should't be consider.

It is also possible to create a recon task using an agent plugin. Usually, the agent plugins returns modules that are attached to the XML that the agent sends to the Pandora server. But, consider that when installing the agent in a machine with it, Tentacle is also installed, and this allow to send XML to the Pandora server. To do a recon task from an agent plugin, it is possible to use this, and besides adding the modules to the agent as a common plugin does, to give our plugin the capacity to send XMLs to Pandora with the information of other agents updated as a recon task would do.

The idea is that the plugin, besides creating the average modules,collects the information and create and send the XML simulating other agents installed if necessary.

The reason to create a plugin that sends data through XML and besides does recon task is to could distribute the monitoring load in different machines and not centralize it in the server.

Server Plugin or Agent Plugin?

A server plugin should be used when:

- The load of each execution is small, for example, simple queries.
- If the Recon Task requires lot of data process.
- If the Recon Task execution intervals are large, for example, once a week

An agent plugin will be used when:

- The information collection requires lot of process or lot of queries.
- The associated Recon Task requires a high process load or lot of queries.
- The Recon Task execution intervals are close to the common execution intervals for agents, for example, every 5 minutes.

Standardization in Development

In order that all plugins would be the more standard possible, and that they have similar features, you should consider the following aspects:

Plugin and Extension Versioning

In Pandora FMS we follow a system of versions for the plugins that has the following

format:

```
v1r1
```

Being:

- vX: plugin version, the step of one version to another is made when a new important functionality is added or an error that makes impossible the correct working of the plugin is corrected.The first version is the v1.
- rY: Plugin revisión,

The change to one revision to another is done when any bug is fixed or a minor feature is implemented. The first revision is the r1. el paso de una revisión a otra se produce cuando se arregla algún bug o se implementa una feature menor. La primera revisión es la r1.

Always that there would be a change to a new version, should be started by the first revision, that is, if we have a plugin in the version v1r5 and we want to get a higher number of version, then we will have v2r1.

Usage and Plugin version

All plugins should respond to a call without parameters, or also with an option type -h or –help, showing the command for its execution and the different parameters of it.Besides, it will be necessary to show the version of the plugin. For example:

```
$ ./myplugin

myplugin version: v1r1

Usage myplugin <param1> <param2> <param3>

   param1: este parametro es una cosa
   param2: este parametro es otra cosa
```

Go back to Pandora FMS documentation index