Consideraciones en el desarrollo de plugins

Latest:
https://pandorafms.com/manual/!current/
Permanent link:
https://pandorafms.com/manual/!current/es/documentation/pandorafms/technical_reference/04_anexo_plugins_considerations
2023/12/03 19:32

Consideraciones en el desarrollo de plugins

Introducción

Los *plugins* (complementos) permiten a Pandora FMS obtener información que requiere un procesamiento complejo o que requiere el uso de sistemas o API complejas. Ejemplos de *plugins* pueden ser la monitorización de bases de datos Oracle® que requiere un proceso múltiple para la monitorización y además ciertas tareas de auto descubrimiento; otro ejemplo podría ser un *plugin* de *parseo* de HTML simple que requiera algo que no puede hacer el Servidor de chequeos WEB.

Diferencias en la implementación y rendimiento

Pandora FMS ofrece dos posibilidades a la hora de ejecutar *plugins*: ejecución en el Agente Software o en el servidor.

- Los *plugin* de servidor realizan ejecuciones independientes para recoger cada pieza de información. La ejecución de *plugin* de servidor es muy costosa por lo que sólo es viable para *plugins* que sean "ligeros", es decir, que requieran pocas o muy pocas consultas para obtener una única pieza de información. Un *plugin* de servidor podría ser un *plugin* de *parseo* HTML específico que requiere 2 ó 3 consultas y por lo tanto cargará poco trabajo al servidor.
- Los *plugin* de Agente Software permiten obtener varios módulos de una vez y por ello son mucho más flexibles que los *plugins* de servidor. Son ideales para *plugins* que requieran varias consultas para obtener una pieza de información: permiten más flexibilidad al programador ya que se pueden devolver varios módulos a la vez.

Tareas de reconocimiento

Para realizar tareas de reconocimiento en *plugins* que así lo requieran se tienen de nuevo dos posibilidades.

La primera consiste en usar el servidor Recon Task del servidor de Pandora FMS. Para ello será necesario crear el código *ad-hoc* para la tecnología o situación concreta. De nuevo las Recon Task cargar el servidor de Pandora FMS por lo que si para realizar la tarea de reconocimiento son necesarias muchas peticiones de datos esta opción deberá ser descartada.

Como alternativa es posible crear una tarea de Recon usando un *plugin* de agente. Normalmente los *plugins* de agente devuelven unos módulos que se adjuntan al XML que envía el agente al servidor PFMS. Ahora bien, al instalar el agente en una máquina con el se instala Tentacle lo que

permite enviar XML al servidor PFMS. Para realizar una Recon task desde un *plugin* de agente es posible aprovechar esto y, además de añadir los módulos al agente como hace un *plugin* normal, dotar al *plugin* de la capacidad de enviar los XML a Pandora FMS con la información de otros agentes actualizada como lo haría una Recon task.

La idea es que el *plugin* además de crear los módulos normales, recolecte la información, monte y envíe los XML simulando otros agentes instalados si fuera necesario.

La razón para realizar un *plugin* que envíe datos por XML y además realice tareas de reconocimiento es para poder distribuir la carga de la monitorización en distintas máquinas y no centralizar en el servidor.

¿Plugin de servidor o plugin de agente?

Se usará un plugin de servidor cuando:

- La carga de cada ejecución sea poca, por ejemplo consultas simples.
- La Recon Task requiera poco procesamiento de datos.
- Los intervalos de ejecución de la Recon Task sean espaciados en tiempo, como por ejemplo una vez a la semana.

Se usará un plugin de agente cuando:

- La recogida de información requiera de mucho proceso o muchas consultas.
- La Recon Task asociada requiera una alta carga de proceso o de muchas consultas.
- Los intervalos de ejecución de la Recon Task están cerca de los intervalos de ejecución típicos para agentes, por ejemplo cada 5 minutos.

Estandarización en el desarrollo

Con el fin de que todos los *plugins* sean lo más estándar posible y que dispongan de características similares hay que tener en cuenta los siguientes aspectos que se detallan a continuación.

Versionado de plugins y extensiones

En Pandora FMS se sigue un sistema de versiones para los *plugin* que tiene el siguiente formato:

```
v1r1
```

Siendo:

- vX: versión del *plugin*, el paso de una versión a otra se produce cuando se añade una nueva funcionalidad importante o se corrige un error que impide el correcto funcionamiento del *plugin*. La primera versión siempre es `v1`.
- `rY`: revisión del *plugin*, el paso de una revisión a otra se produce cuando se arregla algún *bug* o se implementa una funcionalidad menor. La primera revisión siempre es `r1`.

Siempre que se pase a una nueva versión se comenzará por la primera revisión, es decir, si un *plugin* está en la versión `v1r5` y se debe aumentar el número de versión entonces se pasará a `v2r1`.

Ejecución y parámetros

Todos los *plugins* deberán responder ante una llamada sin parámetros o con una opción corta tipo `-h` o larga tipo `-help` mostrando el comando para su ejecución y los diferentes parámetros del mismo y además es necesario mostrar la versión del *plugin*.

```
./myplugin

 myplugin version: v1r1

 Usage myplugin <param1> <param2> <param3>
```

Volver al índice de documentación de Pandora FMS.