



Éléments à prendre en compte pour le développement d'un plugin



<https://pandorafms.com/manual/!780/>

Permanent link:

https://pandorafms.com/manual/!780/fr/documentation/pandorafms/technical_reference/04_anexo_plugins_considerations

2025/03/04 21:22



Éléments à prendre en compte pour le développement d'un plugin

Introduction

Les *plugins* (modules complémentaires) permettent à Pandora FMS d'obtenir des informations qui nécessitent un traitement complexe ou l'utilisation de systèmes ou d'API complexes. Des exemples de *plugins* pourraient être la supervision des bases de données Oracle® qui nécessite un processus multiple pour la supervision et également certaines tâches de découverte automatique; un autre exemple pourrait être un *plugin* correctif de HTML simple qui nécessite quelque chose qui ne peut pas être fait par le [WEB check server](#).

Différences de mise en œuvre et de performance

Pandora FMS offre deux possibilités lors de l'exécution des *plugins* : l'exécution sur le [Software Agent](#) ou sur le serveur.

- Les *plugins* de serveur effectuent des exécutions indépendantes pour collecter chaque élément d'information. L'exécution d'un *plugin* de serveur est très coûteuse et n'est donc viable que pour les *plugins* qui sont « légers », c'est-à-dire qui nécessitent peu ou très peu de requêtes pour obtenir un seul élément d'information. Un *plugin* côté serveur pourrait être un *plugin* HTML spécifique correctif qui ne nécessite que 2 ou 3 requêtes et qui, par conséquent, demandera peu de travail au serveur.
- Les *plugins* d'agent logiciel permettent de récupérer plusieurs modules à la fois et sont donc beaucoup plus flexibles que les *plugins* de serveur. Ils sont idéaux pour les *plugins* qui nécessitent plusieurs requêtes pour obtenir une information : ils offrent une plus grande flexibilité au programmeur car plusieurs modules peuvent être renvoyés en même temps.

Tâches de reconnaissance

Pour effectuer les tâches de reconnaissance dans les *plugins* qui ont besoin, il y a à nouveau deux possibilités.

La première consiste à utiliser le serveur [Recon Task](#) du serveur Pandora FMS. Pour cela, il sera nécessaire de créer le code *ad-hoc* pour la technologie ou la situation spécifique. Une fois de plus, les tâches de reconnaissance chargent le serveur Pandora FMS, de sorte que si de nombreuses demandes de données sont nécessaires pour effectuer la tâche de reconnaissance, cette option doit être écartée.

Il est également possible de créer une tâche Recon à l'aide d'un agent *plugin*. Normalement, les agents *plugins* renvoient des modules qui sont attachés au XML que l'agent envoie au serveur PFMS. Cependant, lors de l'installation de l'agent sur une machine qui en est équipée, **Tentacle** est installé, ce qui permet d'envoyer du XML au serveur PFMS. Pour faire une tâche Recon à partir d'un agent *plugin* il est possible de profiter de cela et, en plus d'ajouter les modules à l'agent comme le fait un *plugin* normal, de fournir au *plugin* la capacité d'envoyer du XML à Pandora FMS avec les informations des autres agents mises à jour comme le ferait une tâche Recon.

L'idée est que le *plugin*, en plus de créer les modules normaux, recueille les informations, assemble et envoie les XML en simulant d'autres agents installés si nécessaire.

La raison de la création d'un *plugin* qui envoie des données via XML et effectue également des tâches de reconnaissance est de pouvoir répartir la charge de supervision sur différentes machines et de ne pas la centraliser sur le serveur.

Plugin serveur ou plugin agent?

Un **server plugin** doit être utilisé lorsque :

- La charge de chaque exécution est faible, par exemple des requêtes simples.
- Recon Task nécessite peu de traitement de données.
- Les intervalles d'exécution de la tâche Recon sont espacés dans le temps, par exemple une fois par semaine.

Un **plugin d'agent** est utilisé lorsque :

- La collecte d'informations nécessite un important travail de traitement ou de consultation.
- La tâche de reconnaissance associée nécessite une charge de traitement élevée ou de nombreuses requêtes.
- Les intervalles d'exécution des Recon Task sont proches des intervalles d'exécution typiques des agents, par exemple toutes les 5 minutes.

Normalisation en cours de développement

Afin de garantir que tous les *plugins* soient aussi standard que possible et présentent des caractéristiques similaires, les aspects suivants doivent être pris en compte.

Versionnement des plugins et des extensions

Dans Pandora FMS, un système de version est suivi pour *plugin* qui a le format suivant :

```
v1r1
```

Soit :

- vX : Le passage d'une version à l'autre se produit lorsqu'une nouvelle fonctionnalité importante est ajoutée ou qu'un bogue empêchant le bon fonctionnement du *plugin* est corrigé. La première version est toujours v1.
- rY : Le passage d'une révision à une autre se produit lorsqu'un *bug* est corrigé ou qu'une fonctionnalité mineure est implémentée. La première révision est toujours r1.

Lorsque vous passez à une nouvelle version, vous commencez par la première révision, c'est-à-dire que si un *plugin* est en version v1r5 et que vous devez augmenter le numéro de version, vous passez à v2r1.

Exécution et paramètres

Tous les *plugins* doivent répondre à un appel sans paramètres ou avec une option courte -h ou longue -help indiquant la commande à exécuter et les différents paramètres de la commande, et la version du *plugin* doit également être indiquée.

```
./myplugin  
  
myplugin version: v1r1  
  
Usage myplugin <param1> <param2> <param3>
```

[Retour à l'index de la documentation de Pandora FMS.](#)