Le Receiver

Log de la version du démon

Tous les jours et à chaque démarrage du démon, on log la version du démon avec le dernier patch s'il y a.

Log de la version au démarrage du démon

```
[YYYY-MM-DD HH:MM:SS] INFO   : [ demon_type ] [START-DAEMON] The daemon in version VXX.XX.XX-
XXX_centos_redhat_X_X_AND_OPTIONS_local_repository_added.fr - VXX.XX.XX-CumulativePatch-
XX_FR_Linux_FULL_YYYY-MM-DD is now started as a daemon (detached from any shell) with pid=XXPID
```

Log de la version à minuit

```
[YYYY-MM-DD HH:MM:SS] INFO   : [ demon_type ] Daemon version is : VXX.XX.XX-
XXX_centos_redhat_X_X_AND_OPTIONS_local_repository_added.fr - VXX.XX.XX-CumulativePatch-
XX_FR_Linux_FULL_YYYY-MM-DD
```

Décalage d'heure entre le démon et le système détecté

Les démons effectuent régulièrement des vérifications sur le système sur lequel il est lancé tant qu'il est en vie (*environ toutes les deux secondes*).

Une de ces vérifications est le changement d'heure du serveur. Si un démon est décalé par rapport à la date que garde le démon en mémoire, alors celui-ci nous informe qu'il détecte un changement d'heure de la machine, et va se mettre à jour automatiquement à la nouvelle heure connue :

```
[YYYY-MM-DD HH:MM:SS] WARNING: [ demon_type ] System time change detected. The last daemon time was [ YYYY-
MM-DD HH:YY:YY ], but the system time is [ YYYY-MM-DD HH:XX:XX ] which makes a difference of X seconds.
Compensating...
```

Gestion des ressources

Libération périodique des ressources

Les démons effectuent régulièrement une collecte de mémoire (*"garbage collection"*) afin de libérer l'espace mémoire inutilisé (*environ toutes les minutes*).

- Log affiché que si le temps "X.XXXs" est supérieur à une seconde
- generation X, le X peut prendre la valeur 0,1 ou 2. Une génération représente une zone mémoire de Python (*voir cet article pour plus d'explications:* Garbage Collection in Python)

```
[YYYY-MM-DD HH:MM:SS] WARNING: [ daemon_or_module_name ] [ MEMORY ] [ PERF ] [ X.XXXs ] PERIODIC python
garbage collection for memory area ( generation X ) in process with pid XXXXX
```

Libération des ressources requise suite à une opération spécifique

Lorsqu'une opération demandant beaucoup de ressources est effectuée (*exemple: le chargement d'une nouvelle configuration de supervision*), une collecte de mémoire arbitraire peut être lancée en dehors de celle périodique pour libérer l'espace mémoire inutilisé à l'issue immédiatement.

```
[YYYY-MM-DD HH:MM:SS] WARNING: [ daemon_or_module_name ] [ MEMORY ] [ PERF ] [ X.XXXs ] REQUESTED python
garbage collection ( __REASON__ ) for memory area ( generation X ) in process with pid XXXXX
```

Exemple

```
[2023-04-13 10:04:52] WARNING: [ WebUI   ]  [ MEMORY ] [ PERF ] [ 6.038s ] REQUESTED python garbage
collection ( because the module >WebUI< have received a new monitoring configuration ) for memory area (
generation 2 ) in process with pid 9299
```

Restitution des ressources inutilisées au système d'exploitation

La libération de l'espace mémoire est généralement effectuée au sein du processus afin que l'allocation d'un nouvel espace mémoire réutilise celui déjà acquis par le processus, dans le but de ne pas demander plus d'espace au système d'exploitation.

Mais souvent, une partie de cet espace mémoire ne sera vraiment plus utilisée (*pour diverses raisons*), auquel cas on restitue l'espace mémoire en question au système d'exploitation pour qu'il puisse le donner à un autre processus.

Cette opération est souvent quasi-instantanée, mais si elle prend un temps anormal d'exécution, on affiche dans les logs le temps d'exécution de la restitution.

```
[YYYY-MM-DD HH:MM:SS] WARNING: [ daemon_or_module_name ] [ MEMORY ] [ PERF ] [ X.XXXs ] Return of freed
memory space to the OS ( malloc_trim ) in process with pid XXXXX
```