

Shinken-healthcheck : Vérifier le bon fonctionnement de Shinken Entreprise

Sommaire

- Qu'est-ce que le shinken-healthcheck ?
 - Usage
 - Principales options
- Lancement centralisé de toute l'architecture ou local uniquement
- Informations de version
- Vérification de l'architecture
 - Etat des démons
 - Cas spécifique au Synchronizer - Vérification du statut du chiffrement des propriétés et données protégées
 - Royaumes et sous-royaumes
- Vérification de la licence
- Vérification des librairies externes
- Vérification des espaces de stockage
 - Graphite
 - Vérification de la configuration de la sauvegarde des données de métrologie
 - Vérification de la configuration pour la lecture des données de métrologie
 - Vérification du bon fonctionnement des serveurs graphite
 - Affichage lorsque la gestion des données de performance est désactivé
- Vérification des addons
- Structure des royaumes
- Historique des installations et de la configuration
 - Historique des mises à jour
 - Historique des données
 - Historique des paramètres de chiffrement des données
- Comment interpréter les informations et erreurs courantes du shinken-healthcheck
 - Le démon est configuré avec l'adresse "localhost"
 - Le démon est injoignable
 - Le démon est en cours d'arrêt
 - Le démon n'a pas encore reçu sa configuration de l'Arbiter
 - Le démon et son Arbiter ont des versions différentes
 - Conflit d'Arbiter sur un démon
 - La dernière connexion de l'Arbiter remonte à trop longtemps
 - Les serveurs ne sont pas à la même heure
 - La configuration de l'Arbiter n'a pas été trouvée
 - Décalage de temps entre 2 démons
 - Un démon est bloqué et doit être redémarré
 - Un démon est configuré en tant que Spare
 - Un démon master a un démon spare de désigné (seulement pour les Brokers)
 - Un démon master autorise son spare à ne pas avoir la même liste de modules
 - Un démon master n'a pas de démon spare de désigné (seulement pour les Brokers)
 - Un démon spare qui n'a pas de démon master (seulement pour les Brokers)
 - Informations et détections dans le cadre de virtualisation Vmware
 - Un module a redémarré de manière imprévue
 - Erreur de communication entre les démons
 - Un scheduler n'a pas de broker ou de poller ou de reactionner
 - Un module a mis trop de temps à répondre aux demandes d'informations/statistiques
 - Erreurs de configuration concernant les données de métrologie
 - Les erreurs d'écriture
 - Cas 1 : Les données d'un royaume ne sont pas sauvegardées
 - Cas 2 : Plusieurs broker du même royaume écrivent dans une même base graphite
 - Cas 3 : Le module graphite sauvegarde uniquement des royaumes différents de celui du broker
 - Cas 4 : le paramètre `realm_store_only` contient un royaume qui n'existe pas
 - Cas 5 : Le port de Graphite-Perfdata est invalide
 - Cas 6 : Le nom de l'hôte de Graphite-Perfdata est invalide
 - Les erreurs de lecture
 - Cas 1 : La configuration du module `webui.cfg` précise un royaume qui n'existe pas
 - Cas 2 : La configuration du module `webui.cfg` précise un royaume non géré par le broker
 - Cas 3 : Le serveur graphite précisé dans la configuration de `webui` ne gère pas les données de ce royaume
 - Cas 4 : Le serveur graphite précisé dans la configuration de `webui` n'est pas un serveur graphite connu
 - Cas 5 : Le broker gère des royaumes qui ne sont pas configurés dans les `graphite_backends` de la `webui`
 - Cas 6 : Le broker rencontre des erreurs pour lire les données sur les serveurs graphite
 - Cas 7 : Le port de Graphite n'est pas correct
 - Cas 8 : L'adresse locale des serveurs de lecture ou d'écriture de Graphite n'a pas pu être résolue
 - Cas 9 : Aucune adresse n'a été fournie dans les `graphite_backends`
 - Cas 10 : Aucun royaume n'a été précisé dans les `graphite_backends`
 - Cas 11 : Le protocole utilisé dans le Graphite backend n'est pas correct
 - Le démon a bloqué une tentative de chargement d'objet malveillant

Qu'est-ce que le shinken-healthcheck ?

Le shinken-healthcheck est une commande présente dans toute installation Shinken Entreprise qui permet de vérifier le bon fonctionnement de Shinken Entreprise.

Cet outil est utilisé pour vérifier :

- L'état de l'installation de Shinken Entreprise (version des démons).
- L'état des principales options de configuration réseau (ports, adresses).
- L'état des modules et sous-modules activés sur les démons.
- L'état des connexions réseau et la synchronisation d'horloge entre les démons.

Le shinken-healthcheck est donc un outil de diagnostic général qui peut détecter les problèmes les plus importants. Cependant, il ne fournit pas autant d'informations et de détail que les checks fournis par Shinken pour sa propre supervision, par exemple des indicateurs de performance.

Usage

```
shinken-healthcheck
```

Principales options

Option	Option longue	Description
-h	--help	Affiche le message d'aide
-v	--version	Affiche la version de Shinken Entreprise installée
-l	--local	Effectue une vérification des démons locaux seulement
-g	--global	Effectue une vérification complète des démons (doit être lancé depuis la machine comportant l'Arbiter et le Synchronizer). Par défaut sur une machine avec un Arbiter et un Synchronizer, un shinken-healthcheck global est effectué sauf si un shinken-healthcheck local est explicitement demandé.
	--debug	Active l'affichage des données de debug dans la sortie de la commande. Cette option est utile seulement dans le cas d'un envoi de ces données aux équipes de support de Shinken Solutions.
-f	--file	Écrit la sortie de la commande dans un fichier. La sortie de la commande est également affichée.
	--output-directory	Dossier dans lequel sera placé le fichier de sortie. Par défaut, le dossier courant est utilisé.
	--output-name	Fichier dans lequel sera placé le fichier de sortie. Valeur par défaut : shinken-healthcheck_\$(DATE).txt
	--timeout	Temps en secondes à partir duquel un démon sera considéré comme injoignable. Par défaut: 3 secondes
	--modules-warning-expire	Temps en minutes pendant lequel un redémarrage de module génère une alerte. Par défaut : 120 (2 heures), valeur maximale : 1440 (24 heures)
	--show-history	Affiche l'historique des installations et données de Shinken Entreprise sur ce serveur

La commande shinken-healthcheck sépare sa vérification en plusieurs parties qui sont décrites dans les sections suivantes.

Lancement centralisé de toute l'architecture ou local uniquement

La commande shinken-healthcheck peut être utilisée dans deux modes de fonctionnements différents :

- Vérification globale de toute l'architecture
 - Ce mode est le mode par défaut de la commande
 - La commande vérifie l'ensemble des serveurs à distance et local, telle que définie dans ses fichiers .cfg
 - Peut être lancé avec l'option **--global** ou **-g**
- Vérification locale
 - Ce mode vérifie que les démons locaux à la machine, sans vérifier les notions de royaumes
 - Ce mode est automatiquement sélectionné quand :
 - On est sur le serveur de l'Arbiter spare
 - On est sur un serveur qui n'a pas d'Arbiter
 - Peut être lancé avec l'option **--local** ou **-l**

Informations de version

L'option `--version` ou `-v` nous donne les informations suivantes :

- **Original installed version** : C'est la toute première version de Shinken qui a été installée
- **Updated version** : C'est la version actuelle de Shinken
- **Updated patch** : C'est le nom du *patch* actuellement installé. Si aucun patch est installé, alors cette ligne ne s'affiche pas

```
Original installed version : 2.12.0-1
Updated version           : 2.12.0-1
Updated patch             :
```

Vérification de l'architecture

Etat des démons

La commande `shinken-healthcheck` affiche ensuite pour tous les démons activés dans la configuration, différentes informations indiquant le bon fonctionnement du démon :

- Le type et le nom du démon. Si celui est un Spare, une mention "SPARE" est présente à la suite du nom du démon.
- La configuration est-elle valide ?
- Le démon est-il joignable sur le port trouvé dans la configuration ?
- La version actuelle du démon. S'il y a une différence de version entre l'Arbiter et le démon, un message d'erreur indique cette différence.
- Connection avec l'Arbiter, ainsi que le décalage de temps entre le démon et l'Arbiter.
- Si plusieurs Arbiters envoient une configuration au démon, un message d'erreur indique qu'un ou plusieurs Arbiters sont en conflit.
- Liste des autres démons à contacter pour pouvoir fonctionner correctement (section "Talk to").
- État des modules, et le cas échéant des sous-modules. Si un module a redémarré récemment, un avertissement sera affiché en indiquant la liste des derniers redémarrages du module ou sous-modules.
- Champs spécifiques au démon
 - Liste des tags pour le Poller et Reactionner, ainsi que les éventuelles erreurs sur les workers.

```
[ brokers ]
[broker: broker-master]
OK: Configuration seems valid
OK: Connection to daemon is OK at port 7772
OK: Daemon version is: 2.12.0-1
OK: Correct connection from arbiter "arbiter-master" ( and no time shift )
Modules:
OK: Name: Simple-log           Type: simple-log
OK: Name: broker-module-livedata Type: broker_module_livedata
OK: Name: WebUI                 Type: webui
OK: Name: sla                   Type: sla
OK: Name: Livestatus            Type: livestatus
OK: Name: Graphite-Perfdata     Type: graphite_perfdata
Module WebUI
OK: Auth_secret is a custom variable
OK: Connection to daemon is OK at port 7767
OK: Mongodb server is available at: mongodb://localhost/?safe=true
Submodules:
OK: Name: Mongodb              Type: mongodb
OK: Name: Cfg_password         Type: cfg_password_webui
OK: Name: webui-enterprise     Type: webui_enterprise
OK: Name: sla                   Type: sla
Talk to:
OK: Reachable receiver satellite (receiver-master) at http://192.168.1.72:7773
OK: Reachable poller satellite (poller-master) at http://192.168.1.72:7771
OK: Reachable reactionner satellite (reactionner-master) at http://192.168.1.72:776
OK: Reachable scheduler satellite (scheduler-master) at http://192.168.1.72:7768
```

Dans l'affichage de l'état des démons, ainsi que dans les sections suivantes, plusieurs états peuvent être retournés:

- **OK** : Tout va bien
- **AT RISK** : Problème pouvant potentiellement nuire au fonctionnement du système.
- **ERROR** : Une erreur bloquante a été détectée.

Cas spécifique au Synchronizer - Vérification du statut du chiffrement des propriétés et données protégées

La sous-section "Encryption status" donne des informations sur le chiffrement des propriétés et données protégées :

- Protection activée ou pas (et nom de la clé si elle est activée)
- Si la clé n'a pas été sauvegardée ou si la clé ne peut pas être chargée, une erreur est affichée
- S'il y a une incohérence de configuration du chiffrement (voir la page [Mécanismes de sécurisation du chiffrement](#)), le message d'erreur correspondant sera affiché.

```
[synchronizer: dev75]
OK:      Mongoddb server is available at: mongoddb://localhost/?safe=false
OK:      Auth_secret is a custom variable
OK:      Connection to daemon is OK at port 7766
OK:      Configuration seems valid
OK:      Connection to daemon is OK at port 7765
OK:      Daemon version is: 2.0.0-RC1-2012-07-19
[Encryption status]
ERROR:   The key has never been exported. Run shinken-protected-fields-keyfile-export and follow instructions
OK:      Encryption ENABLED with key named : validation2
Modules:
OK:      Name: module-syncuf                               Type: syncuf-import
OK:      Name: module-no-encryption-user-right             Type: cfg-file-import
```

```
[Encryption status]
ERROR:   Unable to load keyfile /etc/shinken/secrets/protected_fields_key : No such file or directory ;you need to restore the key test, using shinken-protected-fields-keyfile-restore before the synchronizer can run.
```

Royaumes et sous-royaumes

Dans une configuration de Shinken Entreprise, les démons peuvent être répartis sur plusieurs machines.

Dans la commande shinken-healthcheck, les démons sont regroupés en fonction de la machine sur laquelle ils sont installés.

On voit dans l'exemple ci-dessous qu'un Poller est installé sur la machine d'adresse 192.168.1.35, et qu'un Arbiter et un Broker sont installés et activés sur la machine vm3 (172.16.0.3).

```
- 192.168.1.35 (192.168.1.35):
*****
[poller: poller-spare] - SPARE
ERROR:   Cannot contact daemon 192.168.1.35:7771 ( timed out )
OK:      Configuration seems valid

- vm3 (172.16.0.3):
*****
[arbiter: arbiter-vm3]
OK:      Configuration seems valid
OK:      Connection to daemon is OK at port 7770
OK:      Daemon version is: 2.0.0-RC1-2012-07-19
OK:      This element is defined as the master arbiter
Modules:
OK:      Name: synchronizer-import       Type: synchronizer-import
Talk to:
ERROR:   Unreachable poller satellite (poller-spare) at http://192.168.1.35:7771 (timed out)
OK:      Reachable scheduler satellite (scheduler-vm3) at http://vm3:7768
OK:      Reachable reactionner satellite (reactionner-vm3) at http://vm3:7769
OK:      Reachable poller satellite (poller-vm3) at http://vm3:7771
OK:      Reachable broker satellite (broker-vm3) at http://vm3:7772
OK:      Reachable receiver satellite (receiver-vm3) at http://vm3:7773
[broker: broker-vm3]
OK:      Configuration seems valid
```

Si plusieurs royaumes sont définis, la sortie de shinken-healthcheck organise les machines par royaume et sous-royaumes, puis les démons sont répartis par machine d'installation.

Dans l'exemple ci-contre, on voit que la configuration comporte 4 royaumes, agencés comme suivant:

- Un royaume principal: **France**
 - Un sous Royaume: **Corse**

Exemple d'architecture

```
-----
| Realm /France |
-----
```

```
-----
| In France/ |
-----
```

- Un sous Royaume: **Sud Ouest**
 - Un sous Royaume: **Bordeaux**

On voit aussi, pour chaque royaume, les démons activés ainsi que la machine sur laquelle ils sont installés. Dans l'exemple de shinken-healthcheck, on peut faire le récapitulatif suivant:

- Royaume **France** : 8 démons répartis sur 2 machines
 - Machine d'adresse a.a.a.a: Poller
 - Machine master1 (b.b.b.b): Arbiter, Broker, Poller, Reactionner, Synchronizer, Receiver, Scheduler
- Royaume **France/Corse** : 3 démons installés sur une seule machine
 - Machine master2 (d.d.d.d): Broker, Poller, Scheduler
- Royaume **France/Sud Ouest** : Un démon installé sur une machine
 - Machine master2: Broker
- Royaume **France/Sud Ouest/Bordeaux** : 2 démons installés
 - Machine master2: Poller, Scheduler

```

-----
- a.a.a.a (a.a.a.a):
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
    [poller: poller-windows1]
    ...

- master1 (b.b.b.b):
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
    [arbiter: arbiter-master]
    ...
    [broker: broker-master]
    ...
    [poller: poller-master1]
    ...
    [reactionner: reactionner-master]
    ...
    [synchronizer: synchronizer-master]
    ...
    [receiver: receiver-1]
    ...
    [scheduler: scheduler-master]
    ...

-----
| Realm /France/Corse |
-----

-----
| In Corse/ |
-----

- master3 (d.d.d.d):
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
    [broker: broker-master-3]
    ...
    [poller: poller-master3]
    ...
    [scheduler: scheduler-master3]
    ...

-----
| Realm /France/Sud Ouest |
-----

-----
| In Sud Ouest/ |
-----

- master2 (c.c.c.c):
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
    [broker: broker-master-2]
    ...

-----
| Realm /France/Sud Ouest/Bordeaux |
-----

-----
| In Bordeaux/ |
-----

- master2 (c.c.c.c):
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
    [poller: poller-master2]
    ...
    [scheduler: scheduler-
master2]
    ...

```

Vérification de la licence

Cette section du shinken-healthcheck affiche des informations sur la licence en cours.

A noter que cette section ne s'affiche que sur les serveurs hébergeant un Arbitre ou un Broker.

Elle affiche le propriétaire de la licence, le type de licence, et la date d'expiration de la licence.

Si le nombre de nœuds est dépassé ou que la licence est expirée, une erreur sera affichée dans cette section.

```
License key
OK: The license key is valid
OK: The license key (Customer: ██████████ Node limit:500) is valid (start: ██████████ end: ██████████ => 111 days remaining)
```

Vérification des bibliothèques externes

Shinken Entreprise utilise de nombreuses bibliothèques externes pour fonctionner.

Cette partie du shinken-healthcheck vérifie que toutes les bibliothèques nécessaires au bon fonctionnement de Shinken Entreprise sont installées sur la machine, ainsi que leur version.

En cas d'erreur sur une des bibliothèques, une erreur est affichée indiquant la nature de l'erreur.

```
Local libraries
[gevent]
OK: Library gevent is available. Version: 0.13.8
[ldap]
OK: Library ldap is available. Version: 2.3.10
[pycurl]
OK: Library pycurl is available. Version: libcurl/7.52.1 OpenSSL/1.0.1e zlib/1.2.3 c-ares/1.12.0 libssh2/1.8.0 nghttp2/1.6.
[pymongo]
OK: Library pymongo is available. Version: 2.9.2
```

Vérification des espaces de stockage

Graphite

Shinken Entreprise sauvegarde les données de métrologie dans un serveur Graphite. Si la configuration de la sauvegarde des données de métrologies est simple pour une configuration basique, elle devient rapidement compliquée lorsque la configuration comporte plusieurs royaumes avec plusieurs Brokers.

Lorsque plusieurs Brokers sont dans un même royaume,

- Il faut s'assurer que les Brokers n'écrivent pas les données de métrologie au même endroit (sur le ou les mêmes serveurs Graphite).
- Il faut également vérifier que toutes les interfaces de Visualisation d'un broker lisent les données de métrologie de tous les royaumes et sous-royaume que le broker gère, afin d'assurer une cohérence des données.
- Aussi, il faut s'assurer de ne pas avoir oublié de configurer au moins un broker d'un royaume pour écrire les données de métrologie dans la base Graphite, sans quoi aucune donnée de métrologie ne sera sauvegardée pour ce royaume.

La commande shinken-healthcheck possède une section dédiée à la vérification des espaces de stockage des données de métrologie, qui vérifie que les brokers de chaque royaume sont configurés pour sauvegarder les données, que tous les modules Webui sont configurés correctement pour lire les données stockées dans les serveurs graphite de chaque royaume ou sous-royaume et que les serveurs de métrologie sont effectivement joignables et aptes à sauvegarder des données.

Pour cette partie, nous allons utiliser une configuration avec un royaume par défaut (Metropole) et deux sous royaumes (Corse et Reunion).

Vérification de la configuration de la sauvegarde des données de métrologie

La première section de la vérification vérifie qu'il y a au moins un broker pour chaque royaume ou sous-royaume configuré pour sauvegarder les données de métrologie.

Dans l'exemple ci-contre, nous pouvons voir qu'il y a un broker configuré par royaume, pour sauvegarder les données métrologie sur deux serveurs graphite différents :

- Les données métrologie du royaume **France** sont stockées par le broker **broker-france** sur le serveur graphite **192.168.1.23**
- Les données métrologie du royaume **Bordeaux** sont stockées par le broker **broker-bordeaux** sur le serveur graphite **192.168.1.131**

Si un royaume n'est configuré sur aucun Broker sauvegardant les données de métrologie, une erreur sera affichée.

On note que s'il n'y a aucun hôte à superviser dans le royaume, le Broker n'a pas besoin de sauvegarder de données de métrologie, et donc ne sera pas indiqué en erreur s'il ne sauvegarde pas de données dans Graphite.

```
Storage
[graphite]
  Realm(s)
    Writing - Data location(s):
      OK: Realm [ France ] is stored by the module [ Graphite-Perfdata-France ] of the broker [ broker-france ] on graphite backend 192.168.1.23
      OK: Realm [ Bordeaux ] is stored by the module [ Graphite-Perfdata-Bordeaux ] of the broker [ broker-bordeaux ] on graphite backend 192.168.1.131
      ...
    Reading - UI Visualisation configuration(s): The * means that webui will read all realms that it's broker manage
      broker-bordeaux (Bordeaux):
        WebUI_Bordeaux
          OK: Realm [ Bordeaux ] (*) : The Graphite server on http://192.168.1.131:80 will be used
      broker-france (France):
        WebUI_France
          OK: Realm [ France ] (*) : The Graphite server on http://192.168.1.23:80 (itself) will be used
      ...
```

Vérification de la configuration pour la lecture des données de métrologie

Cette section vérifie que chaque module webui a configuré pour chaque royaume (*ou sous royaume*) un serveur graphite que gère le broker.

Dans l'exemple ci-dessous, nous pouvons voir que la vérification de configuration de la lecture s'effectue pour module webui de chaque broker et que pour chaque module, la commande shinken-healthcheck teste la configuration pour les différents royaumes présents sur le broker.

Si un royaume est inconnu ou non géré par le broker, la vérification retournera une erreur pour ce royaume.

```
Storage
[graphite]
  Realm(s)
    Writing - Data location(s):
      OK: Realm [ France ] is stored by the module [ Graphite-Perfdata-France ] of the broker [ broker-france ] on graphite backend 192.168.1.23
      OK: Realm [ Bordeaux ] is stored by the module [ Graphite-Perfdata-Bordeaux ] of the broker [ broker-bordeaux ] on graphite backend 192.168.1.131
      ...
    Reading - UI Visualisation configuration(s): The * means that webui will read all realms that it's broker manage
      broker-bordeaux (Bordeaux):
        WebUI_Bordeaux
          OK: Realm [ Bordeaux ] (*) : The Graphite server on http://192.168.1.131:80 will be used
      broker-france (France):
        WebUI_France
          OK: Realm [ France ] (*) : The Graphite server on http://192.168.1.23:80 (itself) will be used
      ...
```

Vérification du bon fonctionnement des serveurs graphite

Cette section présente tous les serveurs graphite répertoriés lors de la lecture de la configuration. Un serveur est déterminé par les modules de type "graphite_perfdata" utilisés et également par les graphite_backend de chaque module de type "webui" utilisées.

Chaque serveur graphite est identifié de manière unique par son adresse IP : il est possible que le module "graphite_perfdata" utilise un nom DNS et le module "Webui" une adresse IP. Si les modules utilisent l'adresse de boucle locale (*127.0.0.1*), elle sera également remplacée par l'adresse IP du broker sur lequel il tourne.

Pour chaque serveur deux catégories sont affichées : la partie écriture et la partie lecture.

La partie écriture (*Write connexion status*) affiche :

- Le port utilisé pour la communiquer avec le service carbon-cache.
- L'état de la connexion pour chaque module de type "graphite_perfdata" qui est configuré pour écrire sur ce serveur.
- Le port utilisé par ce serveur est affiché en information si les connexions des modules sont **OK** (*sinon il sera absent*).

- Les royaumes qui sont écrits sur ce serveur sont affichés en information si les connexions des modules sont **OK** (*sinon il sera absent*).
- Si aucun module n'écrit d'information, cela sera affiché en **AT RISK** .

La partie lecture (*Read connection status*) affiche :

- Le port utilisé pour la communiquer avec le service Graphite.
- L'état de la connexion pour chaque module de type "Webui" qui est configuré pour lire des données sur ce serveur.
- Le nombre d'hôtes qui est accessible sur ce serveur est affiché en information si les connexions des modules sont **OK** (*sinon il sera absent*).
- Si aucun module ne lit d'informations sur ce serveur, cela sera affiché en **AT RISK** .

```

Server(s)
192.168.1.131:
  Write connection status:
  Port 2003:
    OK: Module Graphite-Perfdata-Bordeaux on broker broker-bordeaux can write data
    INFO: Write data for realms : Bordeaux
  Read connection status:
  Port 80:
    OK: The webui [ WebUI Bordeaux ] on broker [ broker-bordeaux ] can access the graphite server [ *:192.168.1.131 ]
    INFO: On this server, 11 hosts (with metrics) are present(s), all realm(s) combined
192.168.1.23:
  Write connection status:
  Port 2003:
    OK: Module Graphite-Perfdata-France on broker broker-france can write data
    INFO: Write data for realms : France
  Read connection status:
  Port 80:
    OK: The webui [ WebUI France ] on broker [ broker-france ] can access the graphite server [ *:192.168.1.23:80 ]
    INFO: On this server, 11 hosts (with metrics) are present(s), all realm(s) combined

```

Affichage lorsque la gestion des données de performance est désactivé

Lorsque dans la configuration (voir la page [Configuration avancée \(shinken.cfg\)](#)), l'option de gestion des données de performance (*process_performance_data*) est désactivée, la section "graphite" de la commande shinken-healthcheck va :

- Informer que la fonctionnalité de gestion des données de performance est désactivée
- Mettre un **AT RISK** pour chaque broker ayant un module " graphite_perfdata" configuré

```

Storage
[graphite]
  INFO: Process performance data are disable
Brokers:
  AT RISK: Broker broker-master have a Graphite-Perfdata module

```

Vérification des addons

La section suivante du shinken-healthcheck affiche l'état des différents addons actifs sur la machine.

La liste des addons actuellement activés est affichés, avec pour chacun, leur statut ainsi que l'état de différentes vérifications.

Puisque les addons peuvent avoir chacun un fonctionnement différent, les vérifications effectuées diffèrent selon l'addon.

```

Local addons
[nagvis-shinken-architecture]
  OK: Module configuration file found (/etc/shinken/modules/architecture-export.cfg)
  OK: NagVis installation found (/etc/shinken/external/nagvis)
  OK: 'nagvis-shinken-architecture' addon is running correctly at http://localhost/shinken-core-map
[nagvis]
  OK: NagVis installation found (/opt/nagvis)
  OK: 'nagvis' addon is running correctly at http://localhost/shinken-map

```

Structure des royaumes

La dernière section du shinken-healthcheck l'état de la structure des royaumes.

Si la configuration des royaumes est erronée, les autres vérifications ne sont pas faites et seule cette erreur est affichée.

La copie d'écran montre une erreur, car deux royaumes ont été définis comme royaumes par défaut.

```
#####
This tool is used to check the state of your Shinken Enterprise (企业, 企业) installation and configuration
Note: This check is a global healthcheck as if launched from an Arbiter master server
#####
Healthcheck report 08/09/2022 09:28:03
-----
shinken-healthcheck versions:
Original installed version : V90.077.04-rc1[16664_7.0].FC
Updated version           : V90.077.04-rc1[16664_7.0].FC
Updated patch             : V90.077.04-rc1[16664_7.0].FC
[.....] 100%
Realms Structure
ERROR: There must be one default realm defined. Please set one by setting the 'default' property to '1' for one realm
Healthcheck report 08/09/2022 09:28:03
```

Historique des installations et de la configuration

Le shinken-healthcheck affiche par défaut la version initiale d'installation ainsi que la version actuellement installée.

Il est possible d'obtenir des informations supplémentaires sur l'historique de l'installation en utilisant l'option "--show-history" de la commande "shinken-healthcheck":

```
shinken-healthcheck --show-history
```

Ce paramètre affiche des informations sur l'évolution de Shinken Enterprise depuis son installation initiale.

L'utilisation de cette option est surtout pratique lorsque vous communiquez avec le support Shinken, qui utilisera les informations remontées pour plus facilement retrouver l'origine de votre problème.

Historique des mises à jour

L'option --show-history affiche d'abord une liste triée chronologiquement des différentes versions de Shinken installées et désinstallées sur la machine courante.

On peut donc voir facilement, pour une date donnée, quelle était la version de Shinken installée.

Cette option est également utile pour communiquer avec le support Shinken, pour détecter des erreurs de configuration liées à d'anciennes versions de Shinken.

Description des champs de la sortie de l'historique :

Action	Description	Détails
INSTALLATION	Première installation de Shinken.	<ul style="list-style-type: none">version : Nom de la version de l'installation d'origine.date : Date d'installation au format YYYY-MM-DD HH:MM:SS.
PATCH	Mise à jour de Shinken.	<ul style="list-style-type: none">on version : Nom de la version d'origine de Shinken.date : Date d'installation de la mise à jour au format YYYY-MM-DD HH:MM:SS. <p>Le nom de la version à la ligne correspond à la version vers la quelle la mise à jour est faite.</p>
UNPATCH	Désinstalle une mise à jour.	<ul style="list-style-type: none">on version : Nom de la version d'origine de Shinken.date : Date de désinstallation de la mise à jour au format YYYY-MM-DD HH:MM:SS. <p>Le nom de la version à la ligne correspond à la version du patch désinstallé.</p>


```

CONFIGURATION HISTORY: (can be different from installation history if you did backup/restore data from another server)
- INSTALLATION : [redacted]
- UPDATE : [redacted]
- UPDATE : [redacted]
* sanitize : [redacted]
- RESTORE : [redacted]
- UPDATE : [redacted]
* sanitize : [redacted]

SLA HISTORY: (can be different from installation history if you did backup/restore data from another server)
- INSTALLATION : [redacted]
- UPDATE : [redacted]
- UPDATE : [redacted]
- RESTORE : [redacted]
- UPDATE : [redacted]

METROLOGY HISTORY: (can be different from installation history if you did backup/restore data from another server)
- INSTALLATION : [redacted]
- UPDATE : [redacted]
- UPDATE : [redacted]
- RESTORE : [redacted]
- UPDATE : [redacted]

USER HISTORY: (can be different from installation history if you did backup/restore data from another server)
- INSTALLATION : [redacted]
- UPDATE : [redacted]
- UPDATE : [redacted]
- RESTORE : [redacted]
- UPDATE : [redacted]

MODULES HISTORY: (can be different from installation history if you did backup/restore data from another server)
- INSTALLATION : [redacted]
- UPDATE : [redacted]
- UPDATE : [redacted]
* sanitize : [redacted]
- RESTORE : [redacted]
- UPDATE : [redacted]
* sanitize : [redacted]

```

Historique des paramètres de chiffrement des données

Le dernier type d'information remonté par l'option --show-history est l'historique des paramètres de chiffrement des données sensibles.

À chaque fois que les paramètres de chiffrement sont modifiés et déclenchent un changement dans le chiffrement des données (activation, désactivation, changement des champs chiffrés), une entrée sera ajoutée dans la liste des modifications.

L'état d'activation, le nom de la clé utilisée, l'état de sauvegarde de la clé ainsi que la liste des définitions des données à chiffrer sont affichés pour chaque entrée de la liste.

Notez que le statut de sauvegarde de la clé peut être différent du statut affiché dans la section **[Encryption Status]** :

- Dans la section **[Encryption Status]**, shinken-healthcheck affiche l'état actuel.
- Dans l'historique, shinken-healthcheck affiche l'état au moment de la modification de la configuration.

Seuls les cinq derniers changements sont conservés.

```

PROTECTED FIELDS DATABASE MIGRATIONS HISTORY:
Date : 2018-04-18
      From:           To:
Encrypted:           True           False
Key Name:            test2          test2
Backupted:           True           True
(note that the backup status may be different from the one displayed in the
"Encryption Status" section as this one is the status at migration time.)
Unchanged key hash : 5ad843b335d709cee546874efd3a3d266ad60de1d2b07527b621b040bf9faae
Unchanged substrings : DOMAINUSER LOGIN MSSQLUSER MYSQLUSER ORACLE_USER PASSE PASSPHRASE PASSWORD SSH_USER

Date : 2018-04-18
      From:           To:
Encrypted:           False          True
Key Name:            test2          test
Backupted:           True           False
(note that the backup status may be different from the one displayed in the
"Encryption Status" section as this one is the status at migration time.)
From key hash :      5ad843b335d709cee546874efd3a3d266ad60de1d2b07527b621b040bf9faae
To key hash :        819d6ad057610805cc73c638df5a02cf73c05ceb6fec79d50cec8be6c4b4da4e
Unchanged substrings : DOMAINUSER LOGIN MSSQLUSER MYSQLUSER ORACLE_USER PASSE PASSPHRASE PASSWORD SSH_USER

```

Comment interpréter les informations et erreurs courantes du shinken-healthcheck

Le shinken-healthcheck peut afficher de nombreux cas d'erreur différents. Aussi, certaines informations importantes sont présentées régulièrement. Pour faciliter l'utilisation du shinken-healthcheck, cette section présente ces différentes informations et erreurs courantes, accompagnées de leur interprétation.

Le démon est configuré avec l'adresse "localhost"

Puisque les démons constituant une installation de Shinken Entreprise ont pour vocation à être utilisés dans une architecture distribuée, il ne faut pas définir les démons avec des adresses locales. Les démons vont devoir communiquer entre eux, il faut donc les configurer avec une adresse précise pour éviter tout problème dans une architecture distribuée.

Cette erreur est très fréquemment rencontrée dans le shinken-healthcheck puisque dans l'installation par défaut de Shinken Entreprise, les démons utilisent l'adresse localhost. Dès que des démons seront utilisés dans la même configuration sur plusieurs machines différentes, il est impératif d'utiliser des adresses précises.

```
- localhost (127.0.0.1):
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
[scheduler: scheduler-vm3]
  AT RISK: scheduler-vm3 is defined with the localhost address. It is a problem in distributed mode. Please configure it with the LAN IP/FQDN address instead
  OK:      Connection to daemon is OK at port 7768
  OK:      Daemon version is: 02.04.01-005_BUILD01.fr
  OK:      Correct connection from arbiter "arbiter-vm3" ( and no time shift )
Modules:
  OK:      Name: MongoddbRetention      Type: mongoddb_retention
```

Le démon est injoignable

Le Scheduler " scheduler-vm3 " est injoignable.

Dans ce cas, on exécute le shinken-healthcheck sur la machine vm3. On peut donc déduire que le démon est éteint.

Sur un démon hébergé sur une machine distante, il faudra d'abord déterminer si la machine est joignable avant de pouvoir affirmer que le démon est éteint. Un shinken-healthcheck local sur la machine hébergeant le démon peut confirmer cette hypothèse.

```
[scheduler: scheduler-vm3]
  ERROR:   Cannot contact daemon vm3:7768 ( [Errno 111] Connection refused )
  OK:      Configuration seems valid
```

Le démon est en cours d'arrêt

Lorsque le démon est en cours d'arrêt, shinken-healthcheck le signale, et les informations relatives aux modules ne sont plus disponibles

```
[ brokers ]
  [broker: broker-master]
    AT RISK: The daemon is performing a shutdown.
    OK:      Configuration seems valid
    OK:      Connection to daemon is OK at port 7772
```

Le démon n'a pas encore reçu sa configuration de l'Arbiter

Lorsque de l'on démarre ou redémarre un démon, celui-ci demande sa configuration à l'Arbiter. Il est possible que l'Arbiter mette du temps à l'envoyer.

Dans ce cas un message "AT RISK" sera affiché indiquant que le démon n'a pas encore été contacté par l'Arbiter.

```
[ brokers ]
  [broker: broker-master]
    AT RISK: Daemon has not yet been contacted by an arbiter.
    OK:      Configuration seems valid
    OK:      Connection to daemon is OK at port 7772
    OK:      Daemon version is: 02.04.01-005_BUILD01.fr
    INFO:    This daemon do not have any spare defined
```

Le démon et son Arbiter ont des versions différentes

Pour garantir un bon fonctionnement, il faut que tous les démons présents dans la configuration possèdent la même version de Shinken Entreprise.

Dans ce cas, le Poller " poller-domtom " et l'Arbiter n'ont pas la même version installée, ce qui peut provoquer des dysfonctionnements.

```
[poller: poller-daemon]
ERROR: Daemon version is 2.0.0 and arbiter version is 2.0.0. Please update
OK: Configuration seems valid
OK: Connection to daemon is OK at port 7771
OK: Correct connection from arbiter "arbiter-vm3" ( and no time shift )
Tag list:
None [This poller will handle all untagged checks (the "None" tag)]
Talk to:
```

Conflit d'Arbiter sur un démon

Lors d'une mauvaise configuration, il arrive que 2 installations différentes de Shinken Entreprise pointent vers un même démon. Chaque Arbiter va détecter des incohérences, et ils vont constamment renvoyer des configurations (*contradictoires entre elles*) afin de les corriger, ce qui fait que le démon ne va jamais réussir à travailler correctement.

Dans ce cas, une erreur est affichée dans le shinken-healthcheck indiquant le conflit d'Arbiter.

Pour chaque Arbiter, est donné son nom, son uuid, son adresse IP ainsi que de quelle installation Shinken (Architecture) il provient afin de les retrouver et corriger immédiatement l'erreur de configuration.

```
ERROR: Arbiters CONFLICT :
1.Architecture : Production [IP: 192.168.1.100] :
ERROR: Arbiter arbiter-master : last connection 8s ago. Defined on the server with uuid d2a358b0-cae3-4ce8-bc3f-edd241627bac (/var/lib/shinken/server.uuid)
2.Architecture : Testing [IP: 192.168.1.101] :
ERROR: Arbiter arbiter-master : last connection 14s ago. Defined on the server with uuid 785e0227-455a-449d-b8f1-51689700e3b (/var/lib/shinken/server.uuid)
```

La dernière connexion de l'Arbiter remonte à trop longtemps

Si la dernière connexion de l'Arbiter remonte à trop de temps, le démon va lever un **AT RISK** . Ceci peut être dû:

- Les Arbiters MASTER et SPARE sont réellement éteints
- Les Arbiter MASTER et SPARE sont en train d'envoyer des configurations à d'autres démons, et ne peuvent donc pas contacter ce démon pour l'instant

Le temps pris en compte comme limite de dernière connexion est de $check_interval * max_check_attempts$ du démon (*définis dans sa configuration*). Les valeurs par défauts sont de $60s * 3$, soit 3 minutes.

```
Arbiters connection
Architecture [IP: 192.168.1.100]
AT RISK: Arbiter arbiter-master: Missed connection from arbiter since 17m 54s ( > daemon check_interval * max_check_attempts)
```

Les serveurs ne sont pas à la même heure

Si le serveur n'est pas à la même heure que le serveur Arbiter (qui fait office de référence), une erreur **CRITICAL** sera levée, car des temps différents sur les différents serveurs vont avoir des effets **désastreux** sur la cohérence des données de supervision.

```
Arbiters connection
Architecture [IP: 192.168.1.100]
ERROR: Arbiter arbiter-spare: server times are different, time shift of 1 days 2h
```

La configuration de l'Arbiter n'a pas été trouvé

Ce message apparaît quand la commande shinken-healthcheck utilise l'option --locale ou -l et que la propriété *host_name* n'a été renseignée dans aucun fichier de configuration de ou des Arbiter(s).

Pour résoudre ce problème, il faut dans le fichier de configuration de l'Arbiter qui est présent sur cette machine situé dans */etc/shinken/arbiters/* changer la propriété *host_name* avec le nom de la machine (*qui est trouvable grâce à la commande hostname*)

```
[ arbiters ]
[arbiter: --]
ERROR: No configuration for this arbiter was found. Check if host name in arbiter configuration has the machine host name
```

Décalage de temps entre 2 démons

Puisque Shinken Entreprise repose sur une architecture distribuée composée de plusieurs démons, la synchronisation de l'horloge entre les démons est un point important pour un bon fonctionnement du système.

Dans le shinken-healthcheck, si un démon ne possède pas la même heure que l'Arbiter, une erreur est affichée et indique le décalage horaire.

Dans l'exemple, le poller "poller-domtom" possède un décalage horaire de 4160 secondes avec l'Arbiter "arbiter-vm3".

```
[poller: poller-domtom]
ERROR: Correct connection from arbiter "arbiter-vm3" but a time shift of -4160 seconds
OK: Configuration seems valid
OK: Connection to daemon is OK at port 7771
OK: Daemon version is: 2.11.11-RELEASE_0.0.0
Tag list:
None [This poller will handle all untagged checks (the "None" tag)]
Talk to:
```

Un démon est bloqué et doit être redémarré

Si un démon arrive dans un état bloqué à cause d'un bug par exemple, il ne fonctionnera plus correctement et doit être redémarré. Vous aurez l'erreur suivante:

```
ERROR: The daemon have a lock, it's not working and MUST be restarted. Please contact your support to analyse the daemon log
"Main loop" was locked more than 3600s, and detected at 2022-01-13 10:58:48 [ WATCH DOG ]
```

Un démon est configuré en tant que Spare

Un démon peut être configuré en tant que Spare. Un démon Spare est par défaut inactif, et devient actif lorsqu'un démon principal n'est plus disponible, afin d'assurer une continuité du service.

Pour pouvoir identifier facilement les démons Spare dans l'architecture, une mention "SPARE" est présente à côté du nom du démon.

Si le démon prend la main afin d'assurer la continuité de service, alors une mention supplémentaire "RUNNING" est affichée.

Dans le cas d'un démon Broker, une ligne supplémentaire va être affichée afin d'indiquer de quel démon master il est le spare.

```
[broker: broker-spare] (SPARE)
OK: We are the SPARE of the master daemon => broker-master
OK: Configuration seems valid
OK: Connection to daemon is OK at port 9772
OK: Daemon version is: 2.11.11-RELEASE_0.0.0
OK: Correct connection from arbiter "arbiter-master" ( and no time shift )
```

```
[broker: broker-spare] (SPARE) (RUNNING)
OK: We are the SPARE of the master daemon => broker-master
OK: Configuration seems valid
OK: Connection to daemon is OK at port 9772
OK: Daemon version is: 2.11.11-RELEASE_0.0.0
OK: Correct connection from arbiter "arbiter-master" ( and no time shift )
```

Un démon master a un démon spare de désigné (seulement pour les Brokers)

Quand un démon master a un démon spare de désigné (via sa propriété `spare_daemon`) alors il sera listé.

```
[broker: broker-master]
OK: Our SPARE => broker-spare
OK: Configuration seems valid
OK: Connection to daemon is OK at port 7772
OK: Daemon version is: 2.11.11-RELEASE_0.0.0
OK: Correct connection from arbiter "arbiter-master" ( and no time shift )
```

Un démon master autorise son spare à ne pas avoir la même liste de modules

Quand un master autorise son spare à ne pas avoir la même liste de modules, via la propriété `broker__manage_spare__spare_must_have_the_same_list_of_module_type`.

Le texte entre parenthèses le précise (voir la page [Le Broker](#)).

```
[ brokers ]
[broker: broker-master]
  OK: My SPARE => broker-spare ( the broker configuration doesn't require the same list of modules on its spare )
  OK: Configuration seems valid
  OK: Connection to daemon is OK at port 7772
  OK: Daemon version is: v
  OK: Correct connection from arbiter "arbiter-master" ( and no time shift )
```

Un démon master n'a pas de démon spare de désigné (seulement pour les Brokers)

Quand un démon master n'a pas de démon spare, ceci sera affiché.

```
[broker: broker-secondary]
  OK: Configuration seems valid
  OK: Connection to daemon is OK at port 8772
  OK: Daemon version is:
  OK: Correct connection from arbiter "arbiter-master" ( and no time shift )
  INFO: This daemon do not have any spare defined
```

Un démon spare qui n'a pas de démon master (seulement pour les Brokers)

Quand un démon spare n'est le spare d'aucun autre démon, il est noté comme inutilisé.

```
[broker: broker-spare-useless] (SPARE)
  ERROR: No master is using this spare dameon => UNUSED
  OK: Configuration seems valid
  OK: Connection to daemon is OK at port 11772
  OK: Daemon version is:
  OK: Correct connection from arbiter "arbiter-master" ( and no time shift
```

Informations et détections dans le cadre de virtualisation Vmware

Il se peut que vous décidiez de virtualiser votre architecture Shinken, ou simplement quelques-uns des démons, avec **VMWare**.

Si c'est le cas, une ligne sera ajoutée dans la sortie du "shinken-healthcheck" et précisera que votre satellite fonctionne sur une architecture VMWare.

```
Local daemons
[broker]
  OK: Connection to daemon is OK at port 7772
  OK: Daemon version is:
  OK: This satellite runs on a VMWare VM
  OK: Correct connection from arbiter "arbiter-master" ( and no time shift )
```

Une vérification sera faite sur la présence ou non des "VMWare tools" sur votre VM (Machine virtuelle). Si ces derniers ne sont pas installés, alors une information sera affichée.

Nous vous conseillons de toujours avoir vos "VM tools" installés, et à jour, sur l'ensemble de votre parc de VM.

Ces utilitaires peuvent être installés avec le paquet " `open-vm-tools` ":

```
yum install open-vm-tools
```

```
[broker: broker-master]
AT RISK: This satellite runs on a VMWare VM but the VMWare tools are not installed ; please install them and reboot the server.
OK: Configuration seems valid
OK: Connection to daemon is OK at port 7772
OK: Daemon version is: 1.0.0
OK: Correct connection from arbiter "arbiter-master" ( and no time shift )
```

Enfin, si votre configuration de virtualisation ne permet pas à votre machine virtuelle une utilisation correcte des CPU physiques de son hôte via les VCPU que vous avez associés à celle-ci, alors une information sera affichée. Vos VCPU n'arrivent pas à traiter assez rapidement toutes les demandes d'exécution, ou bien il passe trop de temps à lire les fichiers snapshots.

La valeur "CPU Stolen" en pourcentage est utilisée pour la détection. Si cette valeur:

- est située entre 5% et 10%, alors un message de type "AT RISK" apparaît
- si elle est supérieure ou égale à 10% alors un message de type "ERROR" apparaît.

Si vous recevez un de ces messages, plusieurs options sont possibles :

- Réduisez le nombre d'allocations de VCPU sur votre VM
- Augmentez vos ressources CPU physiques de vos ESXi
- Déplacer vos VM sur un autre serveur physique moins sollicité en terme CPU
- Réduisez le nombre de snapshots présents sur vos VMs

Pour plus de détail sur cet indicateur et des solutions pour réduire le CPU Stolen se référer à la page [Machine VMWare avec un fort taux de CPU Stolen \(%ready + %costop\)](#).

```
[poller: poller-master]
ERROR: Your machine got 20% of CPU STOLEN from the Hypervisor (Type VMWare) - On the VCenter search for the data CPU %ready+%costop & please have a look at the Shinken Enterprise documentation about advices to reduce it
OK: Configuration seems valid
```

La récupération des données de VMWare sont faites via le Gatherer qui écrit le fichier

/dev/shm/vmware_stats_export.dat que chaque démon va lire pour répondre au shinken-healthcheck.

Si le Gatherer est éteint (il est automatiquement démarré lorsqu'on démarre un démon shinken), alors ce fichier ne sera plus mis à jour et les informations ne seront plus fiables.

Si le fichier n'est plus mis à jour depuis plus d'une minute, alors un "AT RISK" sera affiché sur le Shinken healthcheck.

Vérifier dans les logs du Gatherer ce qui est la source du problème, et ensuite, si besoin, relancer le Gatherer sur le serveur distant via la commande :

```
service shinken-gatherer restart
```

```
[scheduler: scheduler-master]
AT RISK: The VMWare stats file '/dev/shm/vmware_stats_export.dat' seems to be too old (not update since 0d 2h 2m 49s > 60s)
The shinken-gatherer daemon should be stopped. Please restart it with 'service shinken-gatherer start'.
```

Un module a redémarré de manière imprévue

Certains démons de Shinken Entreprise utilisent des modules qui permettent d'étendre leurs fonctionnalités. Il arrive que ces modules redémarrent pour des raisons diverses et variées.

Lorsqu'un module a redémarré dans les 2 dernières heures, un avertissement est affiché pour le module en question, car il peut s'agir d'un problème récurrent et/ou potentiellement gênant. Une analyse des logs peut permettre d'avoir plus d'informations sur les erreurs du module.

```
[scheduler: scheduler-vm3]
AT RISK: The scheduler started without loading retention
OK: Configuration seems valid
OK: Connection to daemon is OK at port 7768
OK: Daemon version is: 1.0.0
OK: Correct connection from arbiter "arbiter-vm3" ( and no time shift )
Modules:
ERROR: Name: MongodbRetention Type: mongodb_retention
This module have restarted [1 times] in the last 120 minutes. A restart is removed from count after 120 minutes.
Last restart:
- 2017-09-29 10:33:46 : Cannot start a ssh tunnel for a mongodb connection to server 172.16.0.2 because your local ssh key (~shinken/.ssh/id_rsa) is not authorized on the distant server. You can use the command "ssh-copy-id -i ~shinken/.ssh/id_rsa shinkenuser@172.16.0.2" from this local server to fix this.
=> Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).
No connection to Mongo Database.
```

Erreur de communication entre les démons

Les différents démons de Shinken Entreprise communiquent entre eux pour fonctionner. Il arrive que des erreurs de communications soient remarquées et génèrent des erreurs, qui sont alors notifiées dans le shinken-healthcheck dans la section du démon concerné.

Il est à noter que ces erreurs ne sont pas forcément fatales au bon fonctionnement de Shinken Entreprise, mais qu'elles nécessitent une attention particulière.

i Les erreurs de communication peuvent ne pas être graves et ne pas avoir d'incidence sur votre architecture Shinken. Cependant, si vous recevez des erreurs et que vous avez des doutes sur l'origine de ces problèmes de communication, par prévention, envoyez-nous votre log pour analyse.

```
[poller: poller-vm3]
ERROR: Some API calls between daemons failed in the last 24 hours (2 errors). Please look at your daemon logs for more details about these errors
OK: Configuration seems valid
OK: Connection to daemon is OK at port 7771
OK: Daemon version is: 3.14.0-1.el6.noarch
OK: Correct connection from arbiter "arbiter-vm3" ( and no time shift )
Tag list:
None [This poller will handle all untagged checks (the "None" tag)]
Talk to:
OK: Reachable scheduler satellite (scheduler-vm3) at http://vm3:7768
```

Un scheduler n'a pas de broker ou de poller ou de réagir

Les Schedulers produisent des jobs à effectuer pour les pollers/réagir, ainsi que des données pour les brokers (*par exemple pour l'interface de visualisation*). Ils ont donc besoin de ces autres démons pour fonctionner.

En cas de mauvaise configuration de démons et/ou de royaumes, un Scheduler peut ne pas être défini comme cible d'un autre démon (*poller/réagir/broker*).

Haute disponibilité de la base de métrologie (Graphite)

i Si le Scheduler est dans un sous royaume et qu'un Poller/Réagir/broker est présent dans un royaume supérieur, ceci signifie que ce dernier n'a pas l'option `manage_sub_realms` activée.

```
- localhost (192.168.56.102):
*****
[ schedulers ]
[scheduler: scheduler-master]
ERROR: The scheduler has no reactionner in its realm (or upper realm with the manage_sub_realms option enabled)
OK: Configuration seems valid
OK: Connection to daemon is OK at port 7768
OK: Daemon version is: 3.14.0-1.el6.noarch
OK: Correct connection from arbiter "arbiter-master" ( and no time shift )
Modules:
OK: Name: PickleRetentionFile Type: pickle_retention_file
Talk to:
OK: Reachable reactionner satellite (reactionner-master) at http://localhost:7769
OK: Reachable scheduler satellite (scheduler-master) at http://192.168.56.102:7768
```

Un module a mis trop de temps à répondre aux demandes d'informations/statistiques

En cas de surcharge d'un module, ce dernier peut mettre trop de temps pour répondre aux demandes d'informations (*2 secondes, retardées une fois*), ce qui va faire une erreur au niveau du module.

```
Modules:
OK: Name: sla Type: sla
Module WebUI:
ERROR: [MODULE-INFO ] Fail to send command call get_module_info for module WebUI because the module did timeout (2s)
OK: Connection to WebUI is OK at port 7767
OK: Auth_secret is a custom variable
OK: Mongodb server is available at: mongodb://localhost/?safe=true
```

Erreurs de configuration concernant les données de métrologie

Dans une configuration avec plusieurs royaumes, il est plus difficile de vérifier que les données de métrologie sont bien écrites et peuvent être lues pour chaque royaume.

Il existe 2 types d'erreurs:

- Les erreurs d'écriture
- Les erreurs de lecture

Les erreurs d'écriture

Cas 1 : Les données d'un royaume ne sont pas sauvegardées

```
[*] There is no broker configured to write data for this realm. This realm might produce data that are not stored anywhere (no Graphite backend configured)
```

Exemple : Un broker dans le royaume Toulouse qui n'a pas de module graphite, mais il a un module webui, il n'y a donc pas de broker qui sauvegarde les données du royaume **Toulouse** .

Correction : Ajouter le module Graphite-Perfdata sur le broker

```
Storage
[graphite]
  Realm(s)
    Writing - Data location(s):
      AT RISK: [ Toulouse ] There is no broker configured to write data for this realm. This realm might produce data that are not stored anywhere (no Graphite backend configured)
      OK:      Realm [ France ] is stored by the module [ Graphite-Perfdata ] of the broker [ broke-master ] on graphite backend 192.168.1.23
```

Cas 2 : Plusieurs broker du même royaume écrivent dans une même base graphite

```
[*] This realm is stored on same backends by multiple brokers : by broker * on graphite *(*) and by broker * on graphite * (*)
```

Exemple : Deux brokers sur le même royaume (*France*) ont un même module graphite qui va écrire les données sur le même serveur graphite (192.168.1.23)

Toutes les métriques sont reçues et gérées par le broker. Chaque broker écrira donc les données dans le serveur graphite. Il y aura donc des données en double sur le serveur graphite.

Correction : Un seul broker doit écrire les données d'un royaume sur un serveur graphite. Vous pouvez :

- Enlever le module graphite_perfdata d'un des deux brokers
- Créer un deuxième module graphite_perfdata qui écrira les données sur un serveur graphite différent. Afin de mettre en place de la haute disponibilité pour la base de métrologie ; consultez la page [Haute disponibilité de la base de métrologie \(Graphite\)](#)

```
Storage
[graphite]
  Realm(s)
    Writing - Data location(s):
      ERROR: [ France ] This realm is stored on same backends by multiple brokers : by the module [ Graphite-Perfdata ] of the broker [ broker-1 ] on graphite backend 192.168.1.23 and by the module [ Graphite-Perfdata ] of the broker [ broke-master ] on graphite backend 192.168.1.23
```

Cas 3 : Le module graphite sauvegarde uniquement des royaumes différents de celui du broker

```
Broker * has module* with realm [*] in parameter realm_store_only, but this broker handle only realms : *
```

Le broker à un module graphite_perfdata qui sauvegarde les données uniquement pour un royaume qui est différent de celui ou ceux que le broker gère. Les royaumes gérés par le module sont cités dans le paramètre "realm_store_only". Si ce paramètre est commenté ou absent, le module prend en compte tous les royaumes que le broker gère.

Correction : Il faut changer le module de type graphite_perfdata pour un autre module qui gère le/les mêmes royaumes que le broker. Il est possible de mettre un module graphite_perfdata qui prend tous les royaumes (*paramètre realm_store_only commenté*).

```
Storage
[graphite]
  Realm(s)
    Writing - Data location(s):
      ERROR: Broker broke-master has module Graphite-Perfdata with realm [ Toulouse ] in parameter realm_store_only, but this broker handle only realm Paris, Corse, Bordeaux, France
```

Cas 4 : le paramètre realm_store_only contient un royaume qui n'existe pas

```
Module * has realm [*] in parameter realm_store_only but this realm is unknown
```

Le paramètre "realm_store_only" contient un royaume qui n'est pas défini dans la configuration Shinken.

Correction : vérifiez qu'il n'y a pas d'erreur dans le ou les noms de royaumes qui sont définis dans le paramètre "realm_store_only" dans le module graphite_perfdata cité dans la ligne d'erreur.

```
Storage
[graphite]
  Realm(s)
    Writing - Data location(s):
      ERROR: Module Graphite-Perfdata has realm [ Toulon ] in parameter realm store only but this realm is unknow
```

Cas 5 : Le port de Graphite-Perfdata est invalide

```
The url of Graphite perfdata [ URL ] is invalid : The port [ HTTP_PORT ] is not valid. Valid values are integers from 0 to 65535.
```

Lors que le module Graphite Perfdata est configuré avec un port qui n'est pas un port HTTP valide, l'erreur est remontée dans le shinken-healthcheck.

Correction : Changer le port de Graphite Perfdata (*par défaut dans /etc/shinken/modules/graphite.cfg*) par un port HTTP valide.

```
Storage
[graphite]
  Realm(s)
    Writing - Data location(s):
      ERROR: The url of Graphite perfdata [ http://192.168.1.23:invalid ] is invalid : The port [ invalid ] is not valid
Valid values are integers from 0 to 65535.
```

Cas 6 : Le nom de l'hôte de Graphite-Perfdata est invalide

```
The url of Graphite perfdata [ URL ] is invalid : The hostname or IP address is empty or not found.
```

Lorsque le module Graphite Perfdata est configuré avec un hôte vide, une erreur est remontée dans le shinken-healthcheck

Correction : Changer le nom d'hôte dans la configuration de Graphite Perfdata (*par défaut dans /etc/shinken/modules/graphite.cfg*) par un nom d'hôte ou une adresse IP valide

```
Storage
[graphite]
  Realm(s)
    Writing - Data location(s):
      ERROR: The url of Graphite perfdata [ http://:2003 ] is invalid : The hostname or IP address is empty or not found
```

Les erreurs de lecture

Cas 1 : La configuration du module webui.cfg précise un royaume qui n'existe pas

```
Realm [*] : This realm is an unknown realm. Know realms are : *
```

Exemple : Dans le fichier */etc/shinken/module/webui.cfg* la configuration précise un royaume qui n'existe pas.

```
graphite_backends      Bordeaux:192.168.1.23, France:192.168.1.23
```

Correction : Préciser un nom de royaume qui existe

```
graphite_backends          *:192.168.1.23:80
```

```
Reading - UI Visualisation configuration(s): The * means that webui will read all realms that it's broker manage
broker-master (France):
WebUI
  AT RISK: Realm [Paris]      : This realm does not have neither hosts nor Graphite backend. If hosts are added, their graphics will not be display
  OK:     Realm [Bordeaux]   : The Graphite server on http://192.168.1.23:80 (itself) will be used
  OK:     Realm [France]    : The Graphite server on http://192.168.1.23:80 (itself) will be used
WebUI_01
  OK:     Realm [Bordeaux] (*) : The Graphite server on http://192.168.1.23:80 (itself) will be used
  OK:     Realm [France] (*)  : The Graphite server on http://192.168.1.23:80 (itself) will be used
  OK:     Realm [Paris] (*)   : The Graphite server on http://192.168.1.23:80 (itself) will be used
```

Cas 2 : La configuration du module webui.cfg précise un royaume non géré par le broker

```
Realm [*] : This realm can't be accessed by the broker *. It can access to : *
```

Il y a deux cas possibles :

Exemple 1 : le broker ne gère pas les sous royaume, mais dans la configuration du module webui est précisée l'adresse d'un royaume fils

Exemple 2 : le broker broker-reunion reçoit un module Webui_Reunion est configuré pour le royaume Corse dont il ne s'occupe pas.

Correction :

- Le royaume qui ne peut être accessible est un sous royaume du royaume qui est géré par le broker. Dans ce cas, il suffit de dire au broker de gérer les sous-royaumes : ajouter la gestion des sous-royaumes par le broker dans `/etc/shinken/brokers/broker-master3.cfg`

```
manage_sub_realms          1
```

- Le royaume n'est pas un sous royaume, il faut enlever le royaume non géré du module webui. Il est possible de spécifier le caractère * dans la définition des backend afin de dire que ce module webui prendra tous les royaumes et sous-royaumes du broker vers le même graphite_backend

Realm * can't be accessed by the broker *

```
Reading - UI Visualisation configuration(s): The * means that webui will read all realms that it's broker manage
broker-corse (Corse):
WebUI
  OK:     Realm [Corse] (*)   : The Graphite server on 192.168.1.53 (itself) will be used
broker-metropole-01 (Metropole):
webui-01-validation16-01
  OK:     Realm [Corse]      : The Graphite server on 192.168.1.53 will be used
  OK:     Realm [Metropole]  : The Graphite server on 172.16.0.181 will be used
  OK:     Realm [Reunion]   : The Graphite server on 192.168.1.148 will be used
webui-02-validation16-01
  OK:     Realm [Corse]      : The Graphite server on 192.168.1.53 will be used
  OK:     Realm [Metropole]  : The Graphite server on 172.16.0.181 will be used
  OK:     Realm [Reunion]   : The Graphite server on 192.168.1.148 will be used
webui-03-validation16-01
  OK:     Realm [Corse]      : The Graphite server on 192.168.1.53 will be used
  OK:     Realm [Metropole]  : The Graphite server on 172.16.0.181 will be used
  OK:     Realm [Reunion]   : The Graphite server on 192.168.1.148 will be used
broker-reunion (Reunion):
WebUI
  OK:     Realm [Reunion] (*) : The Graphite server on 192.168.1.148 (itself) will be used
WebUI_Reunion
  ERROR:  Realm [Corse]      : This realm can't be accessed by the broker broker-reunion. It can access to : Reunion
  OK:     Realm [Reunion]   : The Graphite server on 192.168.1.148 (itself) will be used
```

Cas 3 : Le serveur graphite précisé dans la configuration de webui ne gère pas les données de ce royaume

```
Realm [*] : The Graphite server on * does not contain data for this realm
```

Exemple : Le serveur graphite qui est sur 192.168.1.53 sauvegarde uniquement les données du royaume Corse. Cependant, le broker broker-metropole-01 est configuré avec une webui qui va lire les données pour le royaume Metropole sur le serveur graphite 192.168.1.53.

```
graphite_backends          Corse:192.168.1.53, Reunion:192.168.1.148, Metropole:192.168.1.53
```

Correction : Préciser un serveur qui gère les données du royaume Metropole :

```
graphite_backends          Corse:192.168.1.53, Reunion:192.168.1.148, Metropole:172.16.0.181
```

Realm [*] : The Graphite server on * does not contain data for this realm

```
Reading - UI Visualisation configuration(s): The * means that webui will read all realms that it's broker manage
broker-corse (Corse):
  WebUI
    OK:      Realm [Corse]      : There are no hosts in this realm.
broker-metropole-01 (Metropole):
  webui-01-validation16-01
    ERROR:   Realm [Metropole]  : The Graphite server on 192.168.1.53 does not contain data for this realm
    OK:      Realm [Corse]      : There are no hosts in this realm.
    OK:      Realm [Reunion]    : There are no hosts in this realm.
  webui-02-validation16-01
    OK:      Realm [Corse]      : There are no hosts in this realm.
    OK:      Realm [Metropole]  : The Graphite server on 172.16.0.181 will be used
    OK:      Realm [Reunion]    : There are no hosts in this realm.
  webui-03-validation16-01
    OK:      Realm [Corse]      : There are no hosts in this realm.
    OK:      Realm [Metropole]  : The Graphite server on 172.16.0.181 will be used
    OK:      Realm [Reunion]    : There are no hosts in this realm.
broker-reunion (Reunion):
  WebUI
    OK:      Realm [Corse]      : There are no hosts in this realm.
    OK:      Realm [Reunion]    : There are no hosts in this realm
```

Cas 4 : Le serveur graphite précisé dans la configuration de webui n'est pas un serveur graphite connu

The server on * is not a known Graphite server : There is no module graphite_perfdata that writes data on backend *

Exemple : La configuration de webui précise que les données du royaume Metropole sont sur 172.16.0.182 qui n'est pas un serveur graphite connu (aucun module graphite_perfdata n'est configuré pour écrire des données sur ce serveur)

```
graphite_backends          Corse:192.168.1.53, Reunion:192.168.1.148, Metropole:172.16.0.182
```

Correction : Préciser un serveur qui gère les données du royaume Metropole :

```
graphite_backends          Corse:192.168.1.53, Reunion:192.168.1.148, Metropole:172.16.0.181
```

The server on * is not a known Graphite server

```
Reading - UI Visualisation configuration(s): The * means that webui will read all realms that it's broker manage
broker-corse (Corse):
  WebUI
    OK:      Realm [Corse]      : There are no hosts in this realm.
broker-metropole-01 (Metropole):
  webui-01-validation16-01
    ERROR:   Realm [Metropole]  : The server on 172.16.0.182 is not a known Graphite server : There is no module graphite_perfdata that writes data on backend 172.16.0.182
    OK:      Realm [Corse]      : There are no hosts in this realm.
    OK:      Realm [Reunion]    : There are no hosts in this realm.
  webui-02-validation16-01
    OK:      Realm [Corse]      : There are no hosts in this realm.
    OK:      Realm [Metropole]  : The Graphite server on 172.16.0.181 will be used
    OK:      Realm [Reunion]    : There are no hosts in this realm.
  webui-03-validation16-01
    OK:      Realm [Corse]      : There are no hosts in this realm.
    OK:      Realm [Metropole]  : The Graphite server on 172.16.0.181 will be used
    OK:      Realm [Reunion]    : There are no hosts in this realm.
broker-reunion (Reunion):
  WebUI
    OK:      Realm [Corse]      : There are no hosts in this realm.
    OK:      Realm [Reunion]    : There are no hosts in this realm
```

Cas 5 : Le broker gère des royaumes qui ne sont pas configurés dans les graphite_backends de la webui

Realm [*] : No graphics will be displayed as this realm is not present in "graphite_backends" parameter

Exemple : Le broker broker-metropole-01 gère le royaume Metropole et ses deux sous-royaume Reunion et Corse. Dans la définition de la webui webui-02-validation16-01, le royaume Reunion est manquant. On ne pourra donc pas accéder aux métriques sur les hôtes du royaume Reunion sur la webui webui-02-validation16-01

```
graphite_backends          Corse:192.168.1.53, Metropole:172.16.0.181
```

Correction : Il faut ajouter le royaume manquant spécifiquement, ou bien mettre le caractère * pour indiquer que tous les royaumes se situent sur le même serveur

```
graphite_backends          Corse:192.168.1.53, Reunion:192.168.1.148, Metropole:172.16.0.181
```

i Cette erreur n'est présente que s'il y a des hôtes dans ce royaume. Si le royaume n'a pas d'hôte, aucune métrique ne sera lue et donc il n'y aura pas d'erreurs.

Realm [*] : No graphics will be displayed as this realm is not present in "graphite_backends" parameter

```
Reading - UI Visualisation configuration(s): The * means that webui will read all realms that it's broker manage
broker-corse (Corse):
  WebUI
    OK: Realm [Corse] (*) : The Graphite server on 192.168.1.53 (itself) will be used
broker-metropole-01 (Metropole):
  webui-01-validation16-01
    OK: Realm [Corse] : The Graphite server on 192.168.1.53 will be used
    OK: Realm [Metropole] : The Graphite server on 172.16.0.181 will be used
    OK: Realm [Reunion] : The Graphite server on 192.168.1.148 will be used
  webui-02-validation16-01
    ERROR: Realm [Reunion] : No graphics will be displayed as this realm is not present in "graphite_backends" parameter
    OK: Realm [Corse] : The Graphite server on 192.168.1.53 will be used
    OK: Realm [Metropole] : The Graphite server on 172.16.0.181 will be used
  webui-03-validation16-01
    OK: Realm [Corse] : The Graphite server on 192.168.1.53 will be used
    OK: Realm [Metropole] : The Graphite server on 172.16.0.181 will be used
    OK: Realm [Reunion] : The Graphite server on 192.168.1.148 will be used
broker-reunion (Reunion):
  WebUI
    OK: Realm [Reunion] (*) : The Graphite server on 192.168.1.148 (itself) will be used
  WebUI_Reunion
    OK: Realm [Reunion] : The Graphite server on 192.168.1.148 (itself) will be used
```

Dans le cas où le royaume n'a pas de graphite backend mais qu'il n'a pas d'hôtes, alors le message change et devient un " **AT RISK** " car il est nécessaire d'avoir des hôtes pour l'affichage des graphiques :

```
Storage
[graphite]
  Realm(s)
    Writing - Data location(s):
      OK: Realm [All] is stored by the module [Graphite-Perfdata] of the broker [broker-master] on graphite backend 192.168.1.72
      OK: Realm [France] is stored by the module [Graphite-Perfdata] of the broker [broker-master] on graphite backend 192.168.1.72
    Reading - UI Visualisation configuration(s): The * means that webui will read all realms that it's broker manage
    broker-master (All):
      WebUI
        AT RISK: Realm [France] : This realm does not have neither hosts nor graphite backend. If hosts are added, their graphics will not be displayed
        OK: Realm [All] : The Graphite server on http://192.168.1.72:80 will be used
```

Cas 6 : Le broker rencontre des erreurs pour lire les données sur les serveurs graphite

```
Graphite read error happened * times during the last 24h, check the /opt/graphite/storage/whisper read access on the graphite host
```

Le broker a rencontré des erreurs lors des dernières 24 heures pour lire des métriques.

Correction : Il faut consulter les logs du broker concerné afin de connaître la nature de l'erreur (problème de droits sur les dossiers de stockage du serveur graphite).

```
Server(s)
192.168.1.59:
  Write connexion status
    OK: success: broker-prod
  Read connexion status
    ERROR: failure: broker-prod - Graphite read error happened 1 times during the last 24h, check the /opt/graphite/storage/whisper/ read access on the graphite host
all addons
```

Cas 7 : Le port de Graphite n'est pas correct

```
The Graphite backend [ GRAPHITE_BACKEND ] is_incorrect : The port [ HTTP_PORT ] is not valid. Valid values are integers from 0 to 65535.
```

Quand un port incorrect est utilisé dans le graphite backend, une erreur est remontée dans le shinken-healthcheck pour nous l'indiquer.

```
graphite_backends      *:192.168.1.23:invalid
```

Correction : Il faut changer le port du "graphite_backend" concerné par un port qui existe (*un entier compris entre 0 et 65535 inclus*)

```
graphite_backends      *:192.168.1.23:8080
```



Faites attention aux ports inférieurs à 1024, ceux-ci sont généralement réservés pour certains usages comme par exemple les ports 22 pour le protocole SSH ou 443 pour HTTPS.

```
Storage
[graphite]
  Realm(s)
    Writing - Data location(s):
      OK: Realm [ France ] is stored by the module [ Graphite-Perfdata-France ] of the broker [ broker-master ] on graphite backend 192.168.1.23
      OK: Realm [ Bordeaux ] is stored by the module [ Graphite-Perfdata-Bordeaux ] of the broker [ broker-bordeaux ] on graphite backend 192.168.1.128
    Reading - UI Visualisation configuration(s): The * means that weui will read all realms that it's broker manage
      broker-bordeaux (Bordeaux):
        WebUI Bordeaux
          OK: Realm [ Bordeaux ] (*) : The Graphite server on http://192.168.1.128:80 will be used
      broker-master (France):
        WebUI_France
          ERROR: Realm [ France ] (*) : The Graphite backend [ 127.0.0.1:invalid ] is incorrect : The port [ invalid ] is not valid. Valid values are integers from 0 to 65535.
```

Cas 8 : L'adresse locale des serveurs de lecture ou d'écriture de Graphite n'a pas pu être résolu

Lorsqu'un graphite_backends d'un module WebUI ou qu'un module Graphite-Perfdata utilise une adresse locale (*localhost ou 127.0.0.1*), il est possible que le shinken-healthcheck n'arrive pas à résoudre son adresse IP (*remplacer localhost ou 127.0.0.1 par sa réelle IP*)

Configuration du graphite_backends en local d'une WebUI

```
graphite_backends      *:localhost
```

Configuration d'un module graphite_perfdata sur une adresse locale

```
host                   localhost
```

Dans ce cas, on remplace l'adresse par `_127.0.0.1_` et l'on averti que son IP locale n'a pas pu être résolue :

```
Storage
[graphite]
  Realm(s)
    Writing - Data location(s):
      OK: Realm [ All ] is stored by the module [ Graphite-Perfdata ] of the broker [ broker-master ] on graphite backend 127.0.0.1 (192.168.1.23)
    Reading - UI Visualisation configuration(s): The * means that weui will read all realms that it's broker manage
      broker-master (All):
        WebUI
          OK: Realm [All] (*) : The Graphite server on http://192.168.1.23:80 (itself) will be used
      Server(s)
        127.0.0.1:
          AT RISK: The local address could not be resolved
```

Correction : L'erreur provient du fait que le shinken-healthcheck a eu un problème lors de l'ouverture de socket. Cela peut provenir de la configuration réseau du serveur.

Cas 9 : Aucune adresse n'a été fournie dans le graphite_backends

```
The Graphite backend [ GRAPHITE_BACKEND ] is incorrect : The hostname or IP address is empty or not found.
```

Si dans un graphite_backends, une adresse n'a pas de nom d'hôte ou d'adresse IP, une erreur est remontée dans le shinken-healthcheck :

Configuration du graphite_backends sans adresse

```
graphite_backends      *::80
```

Correction : Corriger l'adresse en question pour rajouter une adresse IP ou on nom de l'hôte.

```
Storage
[graphite]
  Realm(s)
    Writing - Data location(s):
      OK:      Realm [ France ] is stored by the module [ Graphite-Perfdata-France ] of the broker [ broker-master ] on graphite backend 1
168.1.23
      OK:      Realm [ Bordeaux ] is stored by the module [ Graphite-Perfdata-Bordeaux ] of the broker [ broker-bordeaux ] on graphite back
d 192.168.1.128
    Reading - UI Visualisation configuration(s): The * means that webui will read all realms that it's broker manage
      broker-bordeaux (Bordeaux):
        WebUI_Bordeaux
          OK:      Realm [ Bordeaux ] (*) : The Graphite server on http://192.168.1.128:80 will be used
      broker-master (France):
        WebUI_France
          ERROR:   Realm [ France ] (*) : The Graphite backend [ :80 ] is incorrect : The hostname or IP address is empty or not four
```

Cas 10 : Aucun royaume n'a été précisé dans le graphite_backends

```
The Graphite backend [ GRAPHITE_BACKEND ] is incorrect : Format error detected. It needs at least a realm and a hostname : <REALM>:<HOSTNAME>
```

Si un des backends renseigné dans la WebUI n'a pas le bon format, un message d'erreur nous averti qu'un backend n'est pas correct en plus de montrer un rappel du format.

Exemple de configuration incorrecte :

Configuration du graphite_backends avec un backend mal formé

```
graphite_backends      *:162.168.1.23:80, 192.168.1.23
```

Correction : Enlever le backend problématique ou rajouter les informations manquantes (*Royaume ou adresse*)

```
Storage
[graphite]
  Realm(s)
    Writing - Data location(s):
      OK:      Realm [ France ] is stored by the module [ Graphite-Perfdata-France ] of the broker [ broker-france ] on graphite backend 1
168.1.23
      OK:      Realm [ Bordeaux ] is stored by the module [ Graphite-Perfdata-France ] of the broker [ broker-france ] on graphite backend
2.168.1.23
    Reading - UI Visualisation configuration(s): The * means that webui will read all realms that it's broker manage
      broker-france (France):
        WebUI_France
          ERROR:   Realm [ ] : The Graphite backend [ 192.168.1.23 ] is incorrect : Format error detected. It needs at le
a realm and a hostname : <REALM>:<HOSTNAME>
          OK:      Realm [ Bordeaux ] (*) : The Graphite server on http://192.168.1.23:80 (itself) will be used
          OK:      Realm [ France ] (*) : The Graphite server on http://192.168.1.23:80 (itself) will be used
```

Cas 11 : Le protocole utilisé dans le Graphite backend n'est pas correct

```
The Graphite backend [ GRAPHITE_BACKEND ] is incorrect : The protocol [ PROTOCOL ] is unknown
```

Lorsque le protocole du Graphite backend n'est pas valide, un message d'erreur est remonté dans la catégorie correspondante.

Exemple de configuration incorrecte :

Configuration du graphite_backends avec un backend mal formé

```
graphite_backends      *:192.168.1.23:80, France:htt://192.168.1.23:80
```

Correction : Changer le protocole pour un protocole valide : *http* ou *https*

```
Storage
[graphite]
  Realm(s)
    Writing - Data location(s):
      OK: Realm [ France ] is stored by the module [ Graphite-Perfdata-France ] of the broker [ broker-france ] on graphite backend 192.168.1.23
      OK: Realm [ Bordeaux ] is stored by the module [ Graphite-Perfdata-France ] of the broker [ broker-france ] on graphite backend 192.168.1.23
    Reading - UI Visualisation configuration(s): The * means that webui will read all realms that it's broker manage
      broker-france (France):
        WebUI France
          ERROR: Realm [ France ] : The Graphite backend [ htt://192.168.1.23:80 ] is incorrect : The protocol [ htt ] is unknown
          OK: Realm [ Bordeaux ] (*) : The Graphite server on http://192.168.1.23:80 (itself) will be used
```

Le démon a bloqué une tentative de chargement d'objet malveillant

There were [X] security breaches blocked today (last 3):

```
AT RISK: ( OBJECT_NAME ) by "HTTP_ROUTE_NAME" by IP=ATTACKER_IP" at YYYY-MM-DD HH:MM:SS
AT RISK: ( OBJECT_NAME ) by "HTTP_ROUTE_NAME" by IP=ATTACKER_IP" at YYYY-MM-DD HH:MM:SS
AT RISK: ( OBJECT_NAME ) by "HTTP_ROUTE_NAME" by IP=ATTACKER_IP" at YYYY-MM-DD HH:MM:SS
```

Il est possible qu'un démon puisse détecter et bloquer une tentative d'injection d'objet malveillant par le biais de l'une de ses routes.

Un message est remonté :

- le nombre total de ces tentatives que le démon a bloqué ce jour (*le compte commence à minuit*) ;
- pour chacune des tentatives (*maximum 3*) :
 - descriptif de l'objet que l'attaquant essaye de charger,
 - sa provenance de l'attaque, par exemple le nom de la route utilisée, et l'IP à la source de l'attaque,
 - sa date.

```
[pollers]
[poller: poller-master]
at 2022-02-08 15:48:03 poller-master is defined with the localhost address. It is a problem to distributed mode. Please configure it with the LAN IP/VPN address 'ip:127.0.0.1'
at 2022-02-08 15:48:03 poller-master is up at port 8080
at 2022-02-08 15:48:03 poller-master local_repository added to
arbiters connections:
  arbiters: [poller-master, poller-slave]
  poller-master: [poller-master]
  poller-slave: [poller-slave]
at 2022-02-08 15:48:03 poller-master last connection: 2022-02-08 15:48:03
Tag [None]
Name: [None] (This poller will handle all untagged checks (like "None" tag))
Poller ID: [None]
at 2022-02-08 15:48:03 poller-master received configuration from poller-master at http://localhost:8080
There were [ 3 ] security breaches blocked today (last 3):
AT RISK: ( hacker/attack ) by "HTTP(s) call "configuration reception from an Arbiter server" by IP=127.0.0.1" at 2022-02-08 15:48:03
AT RISK: ( hacker/attack ) by "HTTP(s) call "configuration reception from an Arbiter server" by IP=127.0.0.1" at 2022-02-08 15:49:56
AT RISK: ( hacker/attack ) by "HTTP(s) call "configuration reception from an Arbiter server" by IP=127.0.0.1" at 2022-02-08 15:49:57
```