Comment interpréter les informations et les erreurs courantes du Healthcheck

Sommaire

- 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 a chargé une configuration enregistrée
- 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
- La configuration de l'Arbiter n'a pas été trouvé
- Les serveurs ne sont pas à la même heure
- 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
  - Détection de la présence des "VMWare tools"
  - Détection d'un pourcentage élevé de "CPU Stolen"
  - Le démon Gatherer n'est pas disponible
- Un module a redémarré de manière imprévue
- Erreur de communication entre les démons
- Erreur d'encodage dans les fichiers de configuration d'un démon
- 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 brokers 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 résolu
  - Erreurs de format dans le paramètre graphite_backends de la WebUI
    - Cas 1 : Pas de royaume dans le backend
    - Cas 2 : Trop de séparateur de royaume
    - Cas 3 : Pas de protocole
    - Cas 4 : Pas de port HTTP
    - Cas 5 : Pas de nom d'hôte ou d'adresse IP
  - Erreurs dans le contenu de la valeur de la clé graphite_backend
    - Cas 1 : Port invalide
    - Cas 2 : Protocole non supporté
  - Le démon a bloqué une tentative de chargement d'objet malveillant

Le healthcheck peut afficher de nombreux cas d'erreur différents. Aussi, certaines informations importantes sont présentées régulièrement.

Pour faciliter l'utilisation du 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 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):
^^^^^^^^^^^^^^^^^^^^^^^^
    [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: [illegible]
        OK:      Correct connection from arbiter "arbiter-vm3" ( and no time shift )
        Modules:
            OK:      Name: MongodbRetention        Type: mongodb_retention
```

Le démon est injoignable

Le Scheduler "*scheduler-vm3*" est injoignable.

Dans ce cas, on exécute le 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 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: v[illegible]
        INFO:    This daemon do not have any spare defined
```

Le démon a chargé une configuration enregistrée

Lorsque le module *last-configuration-recorder* est actif, pour les démons de type :

- Broker,
- Poller,
- Reactionner,
- Receiver et
- Scheduler,

Au démarrage, le démon charge la précédente configuration qu'il a reçu de l'Arbiter, en attendant que ce dernier le contacte.

- Il permet un redémarrage du démon même si l'Arbiter ne peut pas joindre le démon (*ex : coupure réseau*)
- Cela permet aussi de rendre le redémarrage du démon plus rapide (*dans le cas où l'Arbiter n'a pas changé de version de configuration*).

Un message " **AT RISK** " indique que le démon n'a pas encore été contacté par l'Arbiter

```
[ brokers ]
    [broker: broker-master]
        OK:      Configuration seems valid
        OK:      Connection to broker-master is OK at port 7772
        OK:      Daemon version is:
        INFO:    This daemon do not have any spare defined
        Arbiters connection:
            1.Architecture : Shinken-Production [loaded from retention] :
                AT RISK: Arbiter arbiter-master : loaded from retention 23s ago [waiting for Arbiter connection]
```

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-domtom]
    ERROR:   Daemon version is 02.04.01-003_BUILD04.fr and arbiter version is 02.04.01-003_BUILD05.fr. 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 de corriger immédiatement l'erreur de configuration.

```
ERROR:   Arbiters CONFLICT :
    1.Architecture : Production [                ] :
        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 [                ] :
        ERROR:   Arbiter arbiter-master : last connection 14s ago. Defined on the server with uuid 785e0227-455a-449d-b8f1-516897000e3b (/var/lib/shinken/server.uui
```

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 DEV-
        AT RISK: Arbiter arbiter-master: Missed connection from arbiter since 17m 54s ( > daemon check_interval * max_check_attempts
```

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

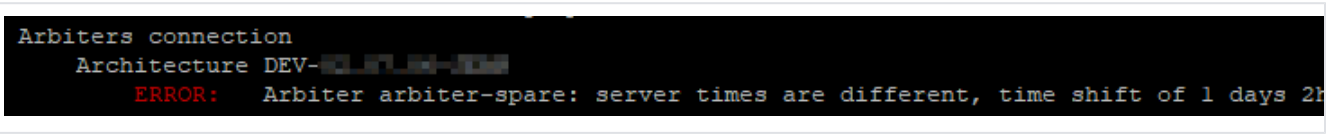
Ce message apparaît quand la commande healthcheck utilise l'option --locale ou -l et que la propriété *host_name* n'a été renseignée dans aucun fichier de configuration de ou des Arbiter(s).

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/),* cha nger 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
```

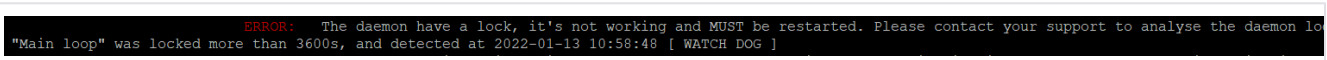
Les serveurs ne sont pas à la même heure

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.



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. L'erreur suivante est alors remontée:



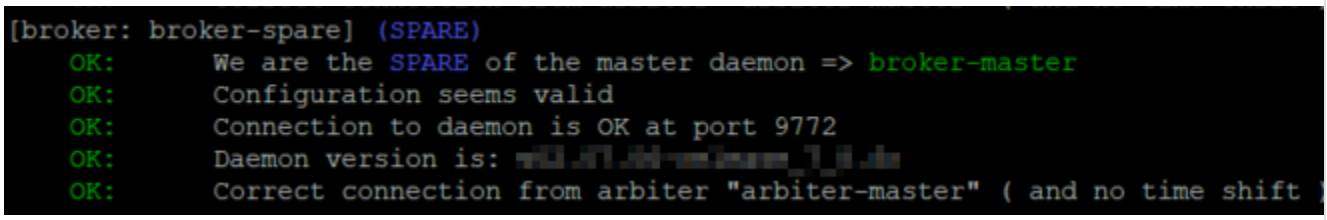
Un démon est configuré en tant que Spare

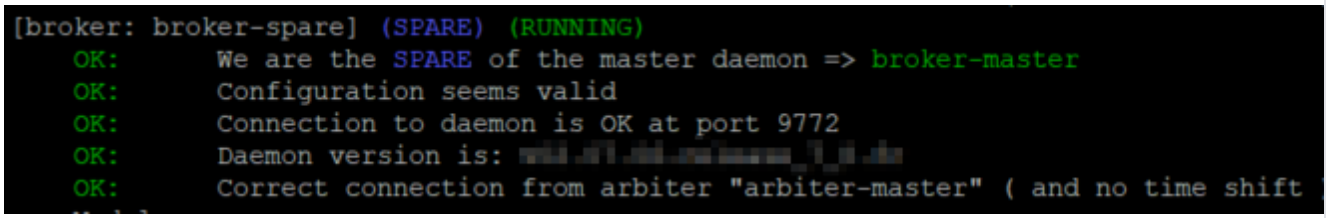
Un démon peut être configuré en tant que Spare. Un démon Spare est par défaut inactif, et devient actif lorsqu'un démon principal n'est plus disponible, afin d'assurer une continuité du service.

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 additionnelle "RUNNING" est affichée.

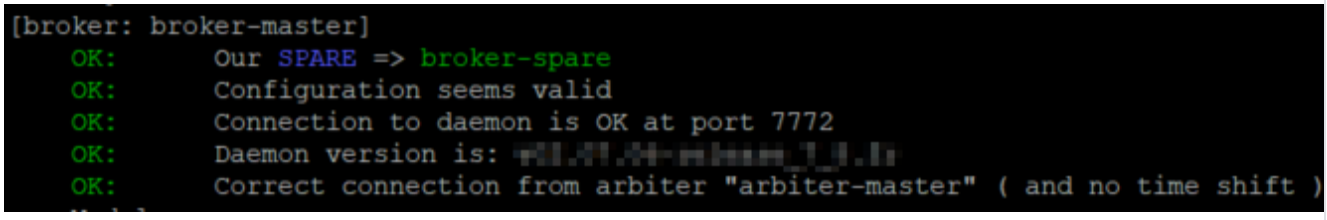
Dans le cas d'un démon Broker, une ligne supplémentaire va être affichée afin d'indiquer de quel démon master il est le spare.





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

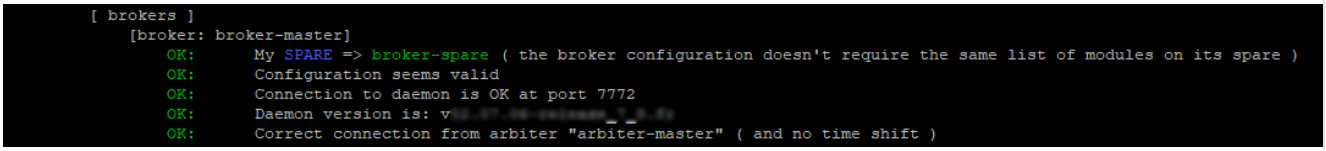
Quand un démon master a un démon spare de désigné (*via sa propriété* ***spare_daemon***) alors il sera listé.



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

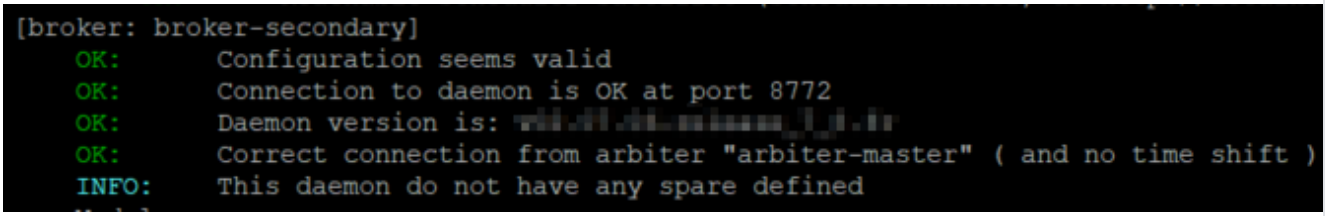
Quand un master autorise son spare à ne pas avoir la même liste de modules, via la propriété "broker__manage_spare__spare_must_have_the_same_list_of_module_type"

le texte entre parenthèses le précise (*voir la Le Broker*)



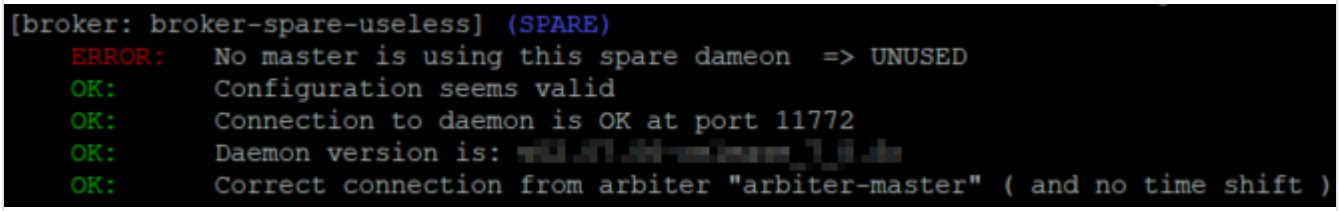
Un démon master n'a pas de démon spare de désigné (seulement pour les Brokers)

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



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

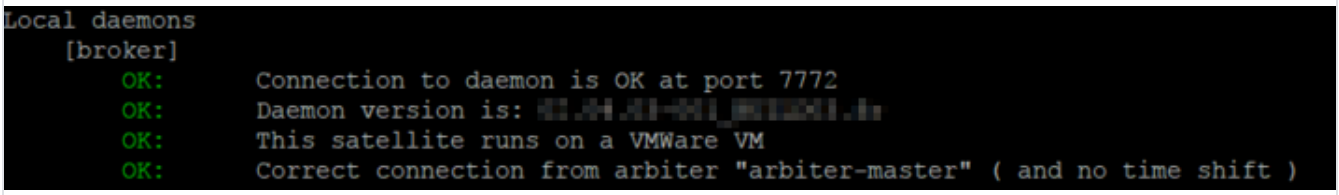
Quand un démon spare n'est le spare d'aucun autre démon, il est noté comme inutilisé.



Informations et détections dans le cadre de virtualisation Vmware

Il est possible de virtualiser l'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 le satellite fonctionne sur une architecture VMWare.



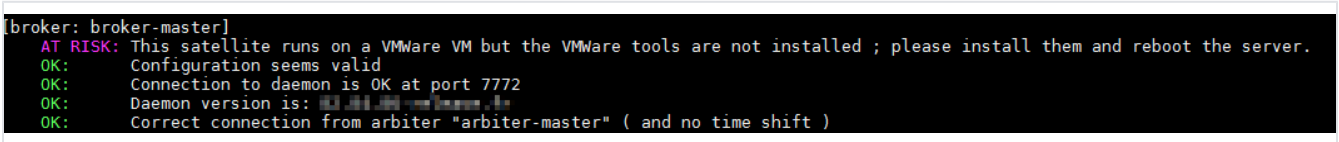
Détection de la présence des "VMWare tools"

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

Il est conseillé de toujours avoir les "VM tools" installés, et à jour, sur l'ensemble du parc de VM.

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

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



Détection d'un pourcentage élevé de "CPU Stolen"

Enfin, si la configuration de virtualisation ne permet pas à une machine virtuelle une utilisation correcte des CPU physiques de son hôte via les VCPU qui lui sont associées, alors une information sera affichée. Les 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.

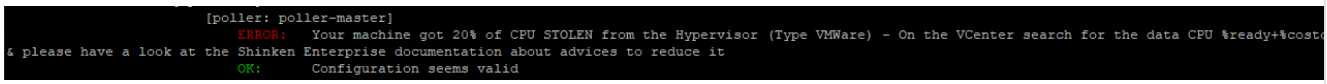
La valeur "CPU Stolen" en pourcentage est utilisée pour la détection. Si cette valeur :

- 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 un de ces messages survient, plusieurs options sont possibles :

- Réduire le nombre d'allocations de VCPU sur la VM
- Augmenter les ressources CPU physiques des ESXi
- Déplacer les VM sur un autre serveur physique moins sollicité en terme CPU
- Réduire le nombre de snapshots présents sur les VMs

Pour plus de détails 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).



Le démon Gatherer n'est pas disponible

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

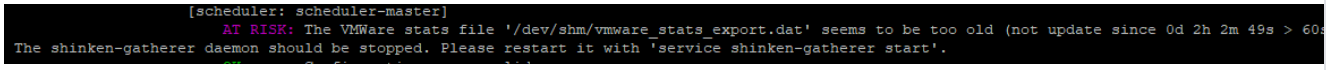
/dev/shm/vmware_stats_export.dat que chaque démon va lire pour répondre au Shinken-healthcheck.

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 healtcheck.

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 :

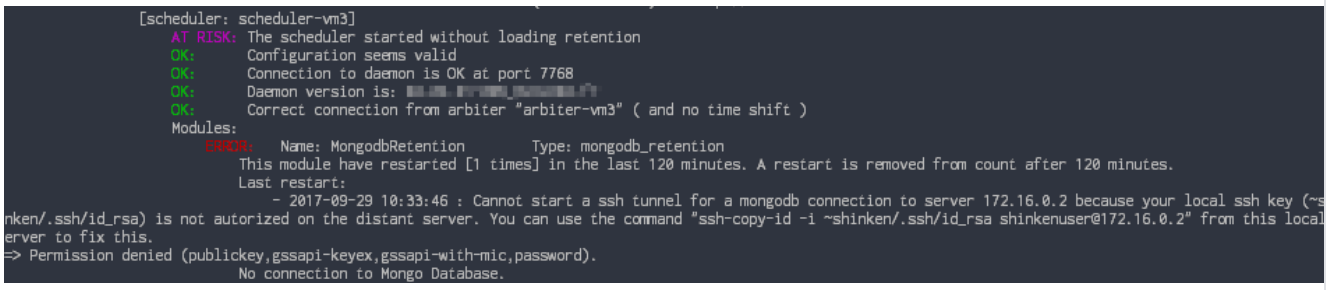
```
service shinken-gatherer restart
```



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.

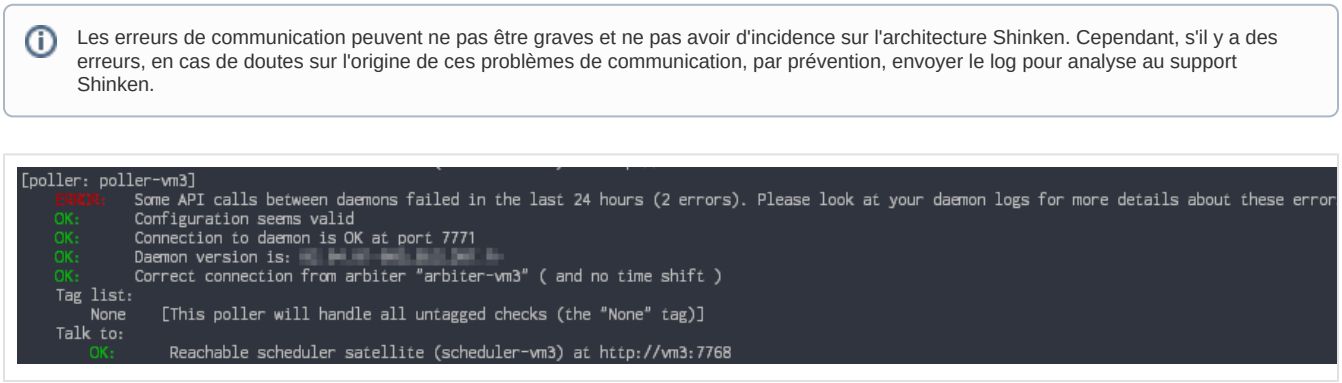
Lorsqu'un module a redémarré dans les 2 dernières heures, un avertissement est affiché pour le module en question, car il peut s'agir d'un problème récurrent et/ou potentiellement gênant. Une analyse des logs peut permettre d'avoir plus d'informations sur les erreurs du module.



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 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.



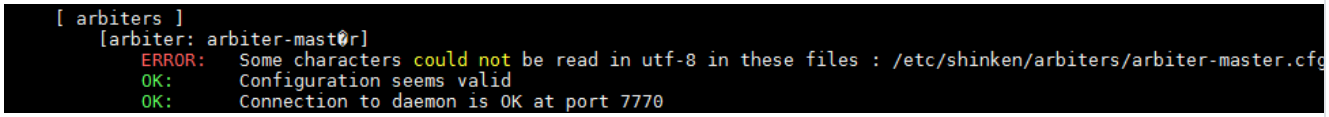
Les erreurs de communication peuvent ne pas être graves et ne pas avoir d'incidence sur l'architecture Shinken. Cependant, s'il y a des erreurs, en cas de doutes sur l'origine de ces problèmes de communication, par prévention, envoyer le log pour analyse au support Shinken.

Erreur d'encodage dans les fichiers de configuration d'un démon

Si les fichiers de configuration des démons situés dans **/etc/shinken** ne pas encodé en utf-8.

Dans l'exemple ci-contre le fichier arbiter-master.cfg ne respecte pas le bon encodage (*utf-8*) et apparaît en erreur dans le résultat de la commande shinken-healthcheck.

Correction : Il suffit de réencoder le fichier au format utf-8

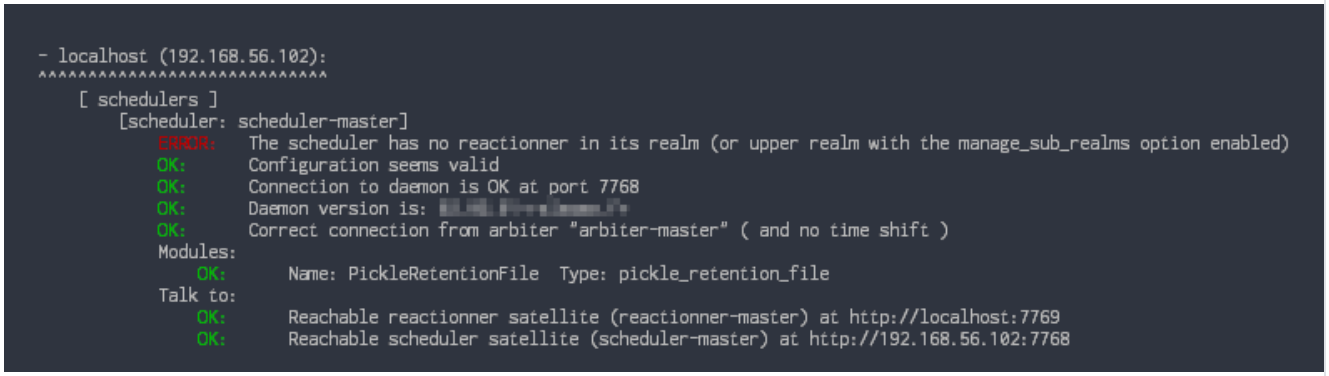


Un scheduler n'a pas de broker ou de poller ou de reactionner

Les Schedulers produisent des jobs à effectuer pour les Pollers/Reactionners, 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 /Reactionner/Broker*).

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



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, retentées 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

C'est le cas, si le message suivant apparaît :

```
[*] 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 anywh
 (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 brokers du même royaume écrivent dans une même base graphite

C'est le cas, si le message suivant apparaît :

```
[*] 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. Il faut donc :

- 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; consulter 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-
 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 : vérifier qu'il n'y a pas d'erreurs 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 unkno
```

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 vali
 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 foun
```

Les erreurs de lecture

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

C'est le cas, si le message suivant apparaît :

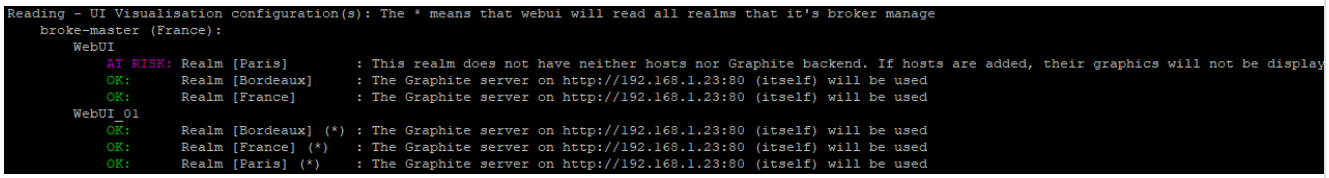
```
Realm [*] : This realm is an unknown realm. Know realms are : *
```

Dans notre 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
```



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

C'est le cas, si le message suivant apparaît :

```
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 : Reunic
            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

C'est le cas, si le message suivant apparaît :

```
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

C'est le cas, si le message suivant apparaît :

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

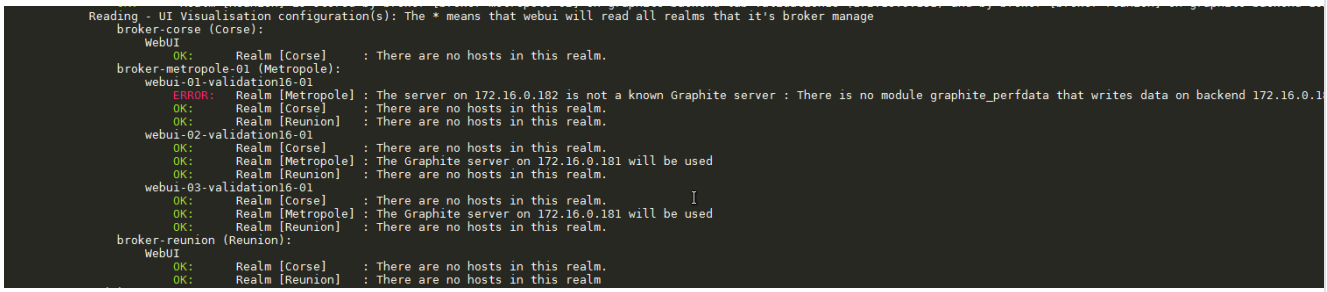
Exemple : La configuration de webui précise que les données du royaume Metropole sont sur 172.16.0.182 qui n'est pas un serveur graphite connu (aucun module graphite_perfdata n'est configuré pour écrire des données sur ce serveur)

```
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



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

C'est le cas, si le message suivant apparaît :

```
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
```

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" paramet
                        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 displa
                        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

C'est le cas, si le message suivant apparaît :

```
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 u serveur graphite)

```
    Server(s)
        192.168.1.59:
            Write connexion status
                OK:      success: broker-prod
            Read connection 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 hos
al 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 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
.168.1.23
                OK:       Realm [ Bordeaux ] is stored by the module [ Graphite-Perfdata-Bordeaux ] of the broker [ broker-bordeaux ] on graphite ba
nd 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 [ 127.0.0.1:invalid ] is incorrect : The port [ invalid ] is not val
 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 (192.168.1.2
            Reading - UI Visualisation configuration(s): The * means that webui 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.

Erreurs de format dans le paramètre graphite_backends de la WebUI

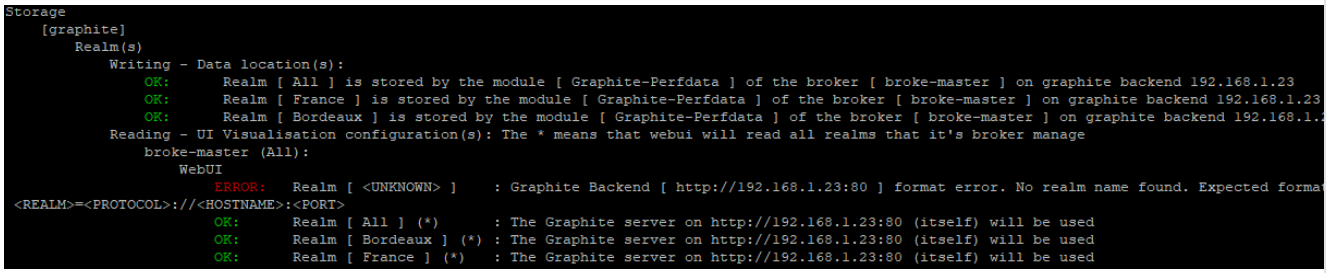
Cas 1 : Pas de royaume dans le backend

Configuration du graphite_backends sans nom de royaume

```
graphite_backends         http://localhost:80, France:http://192.168.1.68
```

Lorsque aucun royaume n'a été trouvé dans la définition d'un backend, ou bien que le caractère servant à séparer le nom du royaume de l'adresse n'est pas trouvé (*caractère =*), l'erreur suivante est remontée :

```
 Graphite Backend [ BACKEND ] format error. No realm name found. Expected format :
<REALM>=<PROTOCOL>://<HOSTNAME>:<PORT>
```



Correction : Rajouter un nom de royaume et faire en sorte que chaque backend suive le format : **<Realm>=<Protocol>://<Hostname>:<Port>**

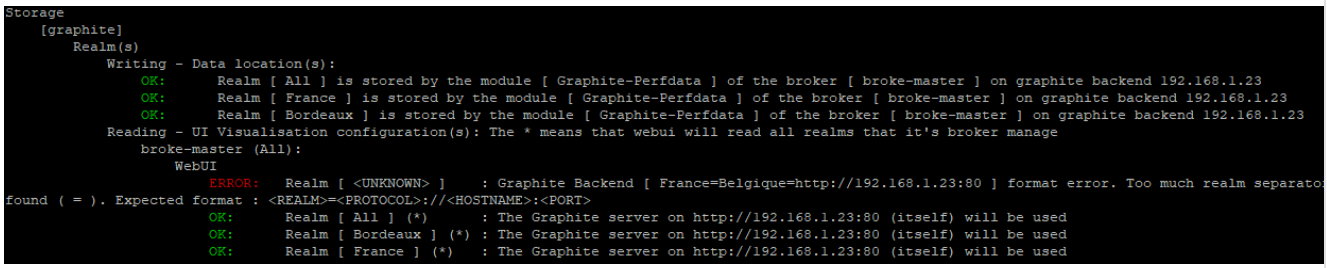
Cas 2 : Trop de séparateur de royaume

Configuration du graphite_backends avec trop de séparateur de royaume

```
graphite_backends          France=Belgique=:http://192.168.1.23:80
```

Lorsque le caractère servant à séparer le nom du royaume et l'adresse (*caractère =*) est présent plus d'une fois dans le backend, l'erreur suivante est remontée :

```
Graphite Backend [ BACKEND ] format error. Too much realm separators found ( = ). Expected format :
<REALM>=<PROTOCOL>://<HOSTNAME>:<PORT>
```



Correction : Enlever le caractère *"="* en trop pour que le backend respecte le format suivant : **<Realm>=<Protocol>://<Hostname>:<Port>**

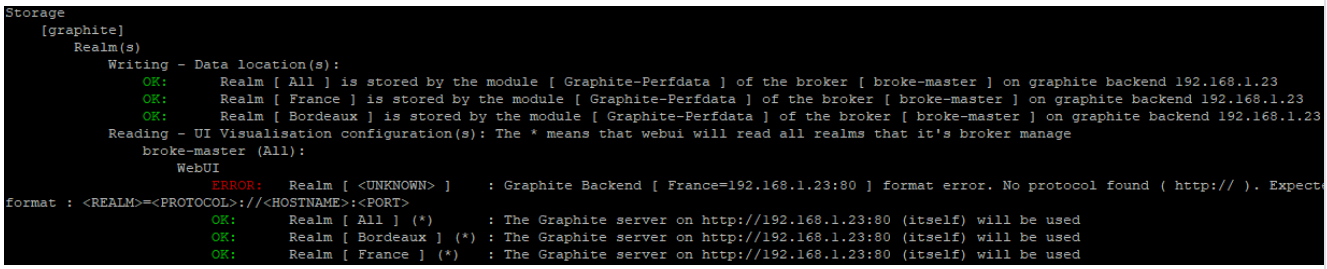
Cas 3 : Pas de protocole

Configuration du graphite_backends sans protocole

```
graphite_backends          France=192.168.1.23:80
```

Lorsque le protocole n'a pas été trouvé dans le backend (*ou que la chaîne "://" n'est pas trouvée*), l'erreur suivante est remontée :

```
Graphite Backend [ BACKEND ] format error. No protocol found ( http:// ). Expected format :
<REALM>=<PROTOCOL>://<HOSTNAME>:<PORT>
```



Correction : Rajouter la protocole manquant pour respecter le format suivant : **<Realm>=<Protocol>://<Hostname>:<Port>**

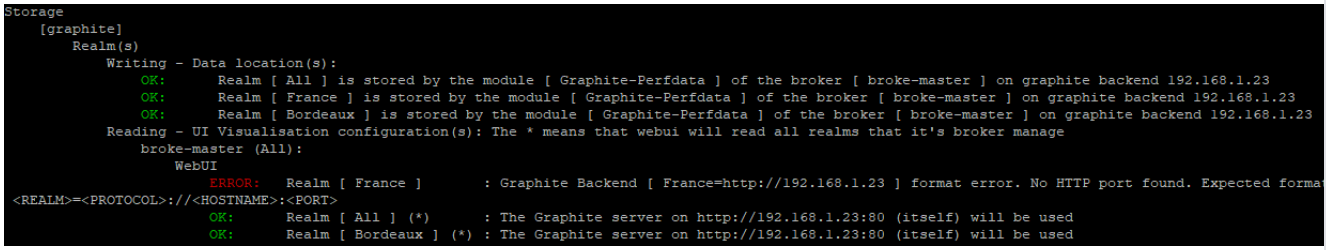
Cas 4 : Pas de port HTTP

Configuration du graphite_backends sans port HTTP

```
graphite_backends         France=http://192.168.1.23
```

Lorsque le port HTTP n'est pas précisé dans le backend, l'erreur suivante est remontée :

```
Graphite Backend [ BACKEND ] format error. No HTTP port found. Expected format :
<REALM>=<PROTOCOL>://<HOSTNAME>:<PORT>
```



Correction : Rajouter le port manquant pour respecter le format suivant : **<Realm>=<Protocol>://<Hostname>:<Port>**

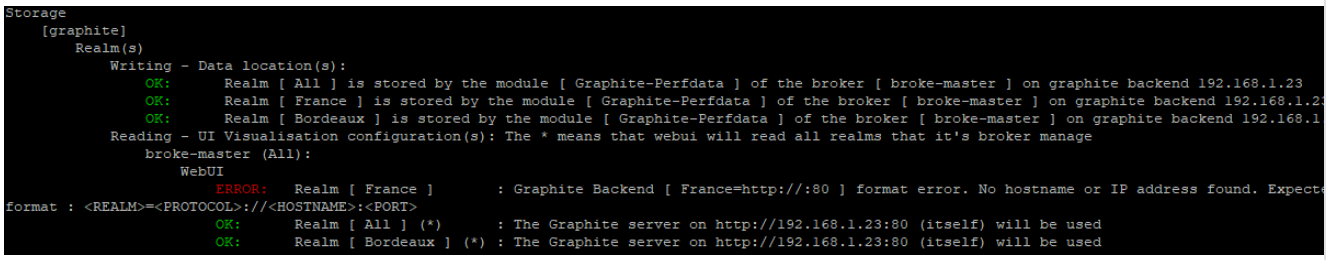
Cas 5 : Pas de nom d'hôte ou d'adresse IP

Configuration du graphite_backends sans nom d'hôte

```
graphite_backends         France=http://:80
```

Lorsque aucun nom d'hôte ou d'adresse IP n'est précisée dans l'adresse du backend, l'erreur suivante est remontée :

```
Graphite Backend [ BACKEND ] format error. No hostname or IP address found. Expected format :
<REALM>=<PROTOCOL>://<HOSTNAME>:<PORT>
```



Correction : Ajouter un nom d'hôte (*ou un adresse IP*) pour respecter le format suivant : **<Realm>=<Protocol>://<Hostname>:<Port>**

Erreurs dans le contenu de la valeur de la clé graphite_backend

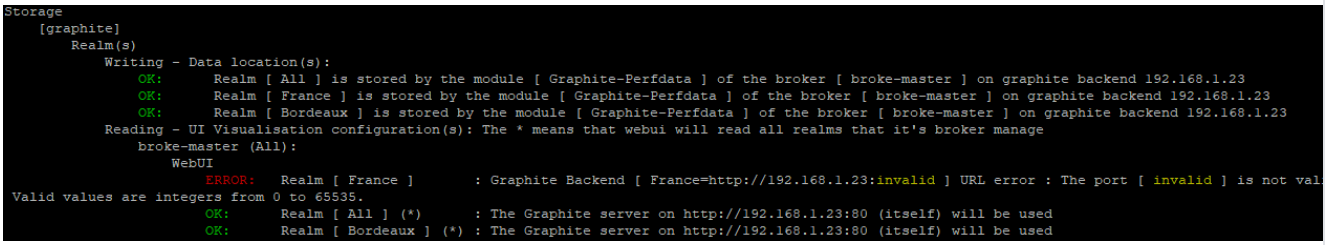
Cas 1 : Port invalide

Configuration du graphite_backends avec un port incorrect

```
graphite_backends         France=http://localhost:invalid
```

Lorsqu'un port non valide a été renseigné dans un backend, l'erreur suivante est remontée :

```
Graphite Backend [ BACKEND ] URL error : The port [ invalid ] is not valid. Valid values are integers from 0
to 65535.
```



Correction : Remplacer le port erroné par un entier entre 0 et 65535 (*inclus*).

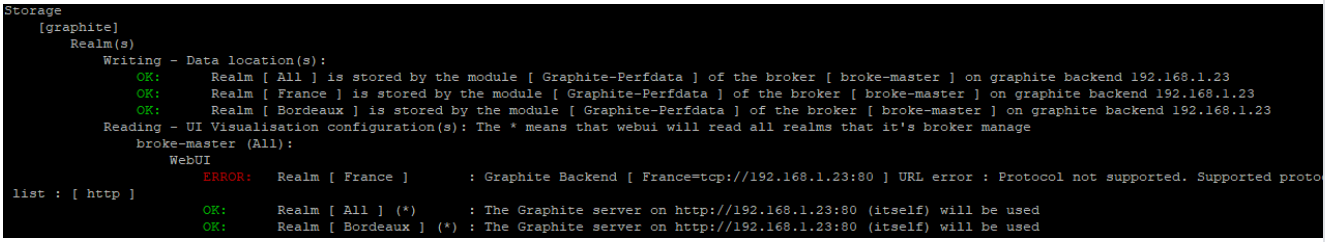
Cas 2 : Protocole non supporté

Configuration du graphite_backends avec un protocole non supporté

```
graphite_backends         France=tcp://localhost:80
```

Lorsque le protocole renseigné dans un backend n'est pas un protocole supporté, l'erreur suivante est remontée :

```
Graphite Backend [ BACKEND ] URL error : Protocol not supported. Supported protocol list : [ http ]
```



Correction : Changer le protocole par un protocole supporté. Pour l'instant seul le protocole HTTP et son extension HTTPS est supporté.

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.