# Configuration du serveur Linux supervisé via le pack linux-by-SNMP__shinken

**Sommaire**



## Contexte

Cette page a pour but de décrire la mise en place d'une configuration minimale SNMP pour un serveur Linux supervisé par le pack **linux-by-SNMP__shinken.**

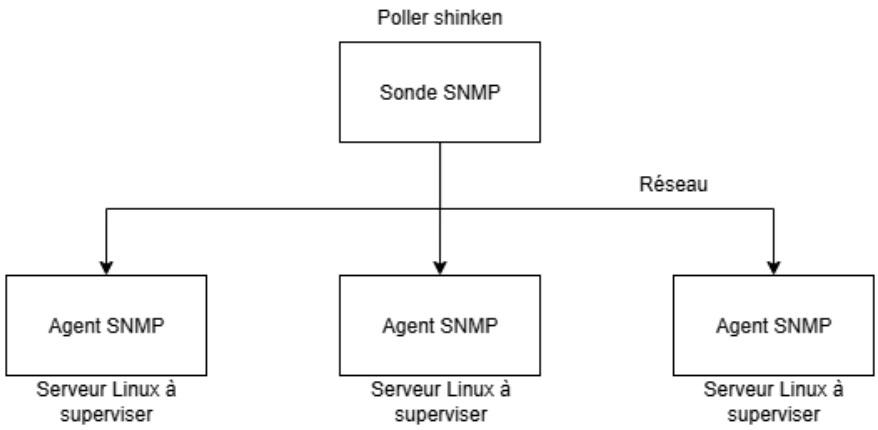


Afin de bénéficier des fonctionnalités des dernières mises à jour du pack linux-by-SNMP, il est possible que vous devez mettre à jour la configuration SNMP de vos hôtes.
Pour cela, suivez le chapitre suivant : Mise a jour des commandes SNMP sur les Linux supervisés

## Procédure de configuration

La supervision d'un serveur Linux supervisé par un **Poller Shinken** se fait par requête/réponse SNMP.

- Sur le **Poller shinken**, la sonde SNMP installé est responsable d'envoyer les requêtes et de traiter les réponses.
- Sur les Serveurs Linux à superviser, c'est l'**agent SNMP** qui est responsable de répondre aux requêtes.




Il est donc **essentiel** de correctement installer et configurer l'**agent SNMP**.

## Installation de l'agent SNMP

### Installation du démon snmpd

L'agent SNMP sous linux à installer provient du paquet **net-snmp** .

Il s'installe de la façon suivante :

```
# Ubuntu, Debian
apt-get install snmpd

# RHEL, Alma, Rocky, Centos, Fedora, OpenSUSE
yum -y install net-snmp

# Arch, Manjaro
pacman -Syy net-snmp
```

Des erreurs peuvent occurrer lors de cette étape avec Alma Linux. Se référer en cas d'erreur à la page Les erreurs lors de la configuration d'un serveur Linux à configurer pour le pack linux-by-SNMP__shinken.

### Installation des outils snmp

Vous pouvez également installer le paquet **net-snmp-utils** ( *Utilitaires de gestion de réseau utilisant SNMP, issus du projet NET-SNMP* ).
L'installation de ce paquet est optionnelle pour le fonctionnement du pack.

```
# Ubuntu, Debian
apt-get install snmp

# RHEL, Alma, Rocky, Centos, Fedora, OpenSUSE
yum -y install net-snmp-utils

# Arch, Manjaro
# Les utilitaires sont déja installés avec le packet net-snmp
```

Une fois installé, vous pouvez activer le service snmpd :

```
systemctl enable snmpd
systemctl start snmpd
```

### Configuration du pare-feu ( firewall )

#### Vérification des règles pour SNMP du par-feu

Sur certains systèmes, le firewall peut bloquer SNMP.  Il faut donc autoriser le trafic sur le port **161/UDP** utilisé par SNMP ( *ici le port par défaut est utilisé, si vous avez configuré votre propre port, remplacez 161 par le vôtre* ).

Ces commandes sont à effectuer avec un utilisateur ayant les droits root.

Sur les systèmes **Ubuntu** / **Debian** / **Fedora** ( *avec ufw* ) :

- Exécuter la commande suivante :

  ```
  ufw status verbose
  ```

- Vérifier qu'une règle autorise l'accès au port **161** en **UDP** aux **IP des polleurs shinken** :
  Exemple de retour valide :

  ```
  Status: active

  To                         Action      From
  --                         ------      ----
  161/udp                    ALLOW       Anywhere
  161/udp (v6)               ALLOW       Anywhere (v6)
  ```

- Sinon, passer à l'étape suivante et ajouter une règle à votre firewall.

Sur les systèmes **RHEL** / **Alma** / **Rocky** / **Centos** / **Fedora** / **OpenSUSE** ( *avec firewalld* ) :

- Exécuter la commande suivante :

  ```
  firewall-cmd --list-all
  ```

- Vérifier que "**snmp**" se trouve dans "**services**" **OU** que le port **161/UDP** est présents dans "**ports**"

  Exemple de retour valide :

```
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpv6-client ntp snmp ssh
  ports: 80/tcp 7765/tcp 7766/tcp 7767/tcp 7768/tcp 7769/tcp 7770/tcp 7771/tcp 7772/tcp 7773/tcp 7777
/tcp 7780/tcp 50000/tcp
  protocols:
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

- Sinon, passer à l'étape suivante et ajouter une règle à votre firewall.

### Ajout de règle pour SNMP au pare-feu

Il est possible de passer cette étape si votre firewall est déjà configuré pour autoriser les requêtes SNMP.

Sur les systèmes **Ubuntu** / **Debian** / **Fedora** ( *avec ufw* ):

```
ufw allow 161/udp
ufw reload
```

Sur les systèmes **RHEL** / **Alma** / **Rocky** / **Centos** / **Fedora** / **OpenSUSE** ( *avec firewalld* ) :

```
firewall-cmd --permanent --add-service=snmp
firewall-cmd --reload
```

## Configuration par script de l'agent snmp

Pour éviter de faire manuellement la configuration, le pack met à disposition **deux scripts distincts** afin de configurer un serveur Linux à superviser :

- `configure-host-SNMPv1v2.sh` : pour configurer une supervision minimale en **SNMP v1 et v2**.
- `configure-host-SNMPv3.sh` : pour configurer une supervision minimale en **SNMP v3**.

Ces scripts utilisent des **templates de configuration fournis avec le pack** :

- `snmpd-template-v1v2.conf`
- `snmpd-template-v3.conf`

> Il n'est pas possible d'appliquer les deux configurations en même temps : **vous devez choisir entre SNMP v1/v2 ou SNMP v3** selon vos besoins.

Les scripts permettent de **REMPLACER** la configuration existante par la configuration minimale que nous livrons en SNMP, v1, v2 ou v3, et de faire fonctionner les checks du pack.

- Ils  sont livrés dans le dossier " ***supervised-host*** " dans le pack.

Si vous avez déjà une configuration SNMP fonctionnel, **N'UTILISEZ PAS** le script de configuration, car il va **écraser** votre **configuration actuelle**.

Dans ce cas, il est plus **logique** de **suivre la documentation** de configuration **pas à pas** , pour vérifier que chaque autorisation nécessaire au pack est possible.

Les scripts sont à exécuter en local sur le serveur Linux à superviser.

- Il faut alors télécharger le dossier "**supervised-host**" sur le serveur Linux, puis **exécuter le script adapté (v1v2 ou v3)**.
- L'outil **ne permet donc pas de déployer** automatiquement sur un parc de machines.
  - Pour cela, vous devez intégrer notre dossier " ***supervised-host*** " dans vos solutions de déploiement ( *Virtual Box, Docker , Ansible, Terraform, AWS EC2, ...* ).

Pour **déployer** les scripts de configuration SNMP (v1/v2 et v3) ainsi que les templates associés **depuis le serveur Shinken vers l'hôte supervisé**, utilisez la commande "**rsync**". Si "**rsync**" n'est pas disponible sur le système, vous pouvez utiliser **scp** en alternative.

Commande avec **rsync** :

```
rsync -avz /chemin/vers/supervised-host/ utilisateur@linux_supervise:/chemin/destination/supervised-host/
```

- `-a` : mode archive (préserve les permissions, dates, liens, etc.)
- `-v` : mode verbeux (affiche le détail des fichiers transférés)
- `-z` : compression pendant le transfert

Commande alternative avec **scp** :

```
scp -r /chemin/vers/supervised-host/ utilisateur@linux_supervise:/chemin/destination/supervised-host/
```

- `-r` : copie récursivement un dossier ainsi que tous ses sous-dossiers et fichiers.

### Prérequis avant d'exécuter le script

Avant d'exécuter le script choisi, il est nécessaire de :

- installer l'agent SNMP et ouvrir les ports SNMP en suivant ce chapitre
- installer les dépendances sur les hôtes à superviser en suivant ce chapitre.

Avant de déployer le script de configuration sur plusieurs serveurs Linux à superviser, il est fortement conseillé de :

- Faire une installation manuelle sur UN serveur Linux, pour bien comprendre les modifications nécessaires,
- Puis tester le script de configuration sur UN serveur Linux, pour valider son comportement dans votre environnement.

### Exécuter le script

Chaque script va **ÉCRASER** le fichier ("**/etc/snmp/snmpd.conf**") et le remplacer par le template correspondant.

- Un fichier de sauvegarde ("**/etc/snmp/snmpd.conf.[DATE].bak**") sera généré avant d'écraser la configuration par défaut.

#### Exemple **SNMPv1v2 par défaut**:

Si vous souhaitez utiliser **uniquement les paramètres par défaut du pack**, exécutez :

```
./configure-host-SNMPv1v2.sh --override-default-conf
```

Cette commande configure immédiatement l'hôte avec la communauté `public` ( *les valeurs par défaut fournies dans le pack seront utilisées* ).

- Aucune configuration supplémentaire pour les paramètres d'authentification n'est nécessaire.

#### Exemple SNMPv1v2 personnalisé :

> Pour des enjeux de sécurité, il est vivement **recommandé de personnaliser** ces paramètres d'authentification.

Si vous voulez **adapter la configuration à votre environnement** , vous pouvez surcharger certains paramètres en ligne de commande.

Options disponibles :

**SNMPv1v2** :

| option | description | Pour version SNMP | Valeur par défaut | Variable de modèle d'hôte correspondant |
|---|---|---|---|---|
| -c | Communauté | V1 V2 | **public** | *LINUX-BY-SNMP__V1V2-COMMUNITY* |

Exemple avec une communauté personnalisée :

```
./configure-host-SNMPv1v2.sh --override-default-conf -c my_custom_community
```

#### Exemple **SNMPv3** par défaut :

De la même manière, pour appliquer directement la configuration SNMPv3 avec les **valeurs par défaut du pack**, exécutez :

```
./configure-host-SNMPv3.sh --override-default-conf
```

Les identifiants utilisés seront alors ( *les valeurs par défaut fournies dans le pack seront utilisées* ) :

- utilisateur SNMP "`shinken`" *( il ne s'agit pas d'un utilisateur Linux )*
- mot de passe "`shinkenpassword`"
- protocole d'authentification "`SHA`"
- protocole de confidentialité "`AES`"
- clé de chiffrement "`shinkenencryptionkey`"

Aucune configuration supplémentaire pour les paramètres d'authentification n'est nécessaire.

#### Exemple **SNMPv3 personnalisé :**

> Pour des enjeux de sécurité, il est vivement **recommandé de personnaliser** ces paramètres d'authentification.

Si vous voulez **adapter la configuration à votre environnement** , vous pouvez surcharger certains paramètres en ligne de commande.

Options disponibles :

**SNMPv3** :

| option | description | Pour version SNMP | Valeur par défaut | Variable de modèle d'hôte correspondant |
|---|---|---|---|---|
| -a | Protocole pour l'identification | V3 | **SHA** | *LINUX-BY-SNMP__V3-PROTOCOL-AUTH* |
| -A | Mot de passe | V3 | **shinkenpassword** | *LINUX-BY-SNMP__V3-PASSPHRASE-AUTH* |
| -x | Protocole de confidentialité | V3 | **AES** | *LINUX-BY-SNMP__V3-PROTOCOL-PRIV* |
| -X | Clef de chiffrement | V3 | **shinkenencryptionkey** | *LINUX-BY-SNMP__V3-PASSPHRASE-PRIV* |
| -u | Nom d'utilisateur | V3 | **shinken** | *LINUX-BY-SNMP__V3-LOGIN* |
| -s | Niveau de sécurité | V3 | **priv** | *N/A* |

L'argument "**-s**" permet de définir le **niveau de sécurité SNMPv3** associé à l'utilisateur créé. Trois valeurs sont possibles :

- **priv** : l'utilisateur est restreint au mode **authPriv** (authentification + chiffrement) . C'est le niveau de sécurité le plus élevé.
  - Il faudra obligatoirement utiliser le modèle d'hôte :
    - **linux-by-SNMP__authPriv** .
- **auth** : l'utilisateur peut se connecter en **authNoPriv** (authentification sans chiffrement) ou en **authPriv** . L'authentification est donc obligatoire, mais le chiffrement reste optionnel.
  - Ici, les modèles d'hôtes suivant peuvent être utilisés :
    - **linux-by-SNMP__authPriv**
    - **linux-by-SNMP__authNoPriv**
- **noauth** : l'utilisateur peut se connecter avec n'importe quel niveau de sécurité (**noAuthNoPriv** , **authNoPriv** ou **authPriv** ). C'est le niveau le plus permissif, car il autorise un accès non authentifié.
  - Ici, **TOUS** les modèles d'hôtes peuvent être utilisés :
    - **linux-by-SNMP__authPriv**
    - **linux-by-SNMP__authNoPriv**
    - **linux-by-SNMP__noAuthNoPriv**

**Important :** L'utilisateur configuré ici est un **utilisateur SNMP** ( ex. `-u shinken` en SNMPv3 ), il s'agit d'un compte défini **dans la configuration de l'agent SNMP**, et **non pas d'un utilisateur Linux du système**.

Dans la majorité des cas, vous devrez définir vos **propres identifiants SNMPv3** afin d'assurer la sécurité de votre environnement.
Voici un exemple complet avec des valeurs personnalisées :

```
./configure-host-SNMPv3.sh --override-default-conf -u my_new_user -a SHA -A my_new_password -x AES -X
my_new_encryption_key -s priv
```

## Test de connexion SNMPv1v2

Sur votre machine locale, vous pouvez exécuter les commandes suivantes pour tester votre configuration SNMP.

Tester la configuration v2 ( *les paramètres d'authentification peuvent être à changer* ) :

```
snmpwalk localhost -v2c -c public  1.3.6.1.4.1
```

Le résultat attendu est une longue liste d'OID et de leurs valeurs associés, comme ci-dessous :

```
UCD-SNMP-MIB::memIndex.0 = INTEGER: 0
UCD-SNMP-MIB::memErrorName.0 = STRING: swap
UCD-SNMP-MIB::memTotalSwap.0 = INTEGER: 5119996 kB
UCD-SNMP-MIB::memAvailSwap.0 = INTEGER: 4175664 kB
UCD-SNMP-MIB::memTotalReal.0 = INTEGER: 1820668 kB
UCD-SNMP-MIB::memAvailReal.0 = INTEGER: 106600 kB
...
```

### Test de connexion SNMPv3

Sur votre machine locale, vous pouvez exécuter les commandes suivantes pour tester votre configuration SNMP.

Tester la configuration v3 ( *les paramètres d'authentification peuvent être à changer* ) :

```
# Tester authPriv
snmpwalk localhost -v3 -l authPriv -u shinken -a SHA -A "shinkenpassword" -x AES -X "shinkenencryptionkey"
1.3.6.1.4.1

# Tester authNoPriv
snmpwalk localhost -v3 -l authNoPriv -u shinken -a SHA -A "shinkenpassword" 1.3.6.1.4.1

# Tester noAuthNoPriv
snmpwalk localhost -v3 -l noAuthNoPriv -u shinken 1.3.6.1.4.1
```

Le résultat attendu est une longue liste d'OID et de leurs valeurs associés, comme ci-dessous :

```
UCD-SNMP-MIB::memIndex.0 = INTEGER: 0
UCD-SNMP-MIB::memErrorName.0 = STRING: swap
UCD-SNMP-MIB::memTotalSwap.0 = INTEGER: 5119996 kB
UCD-SNMP-MIB::memAvailSwap.0 = INTEGER: 4175664 kB
UCD-SNMP-MIB::memTotalReal.0 = INTEGER: 1820668 kB
UCD-SNMP-MIB::memAvailReal.0 = INTEGER: 106600 kB
...
```

## Configuration manuelle de l'agent

Le fichier principal de configuration SNMP est : "**/etc/snmp/snmpd.conf**".

Par précaution, faîtes une copie puis éditez le fichier de configuration :

```
cp /etc/snmp/snmpd.conf /etc/snmp/snmpd.conf.bak
vim /etc/snmp/snmpd.conf
```

Si vous n'avez pas déjà de configuration SNMP,

- Vous pouvez utiliser la configuration d'exemple
- **Sinon**, suivez la configuration pas à pas.

### Configuration d'exemple

Si vous n'avez jamais configuré le serveur SNMP ( vous avez le fichier par défaut ), vous pouvez utiliser la configuration suivante.

Cette **configuration minimale** offre un accès en lecture à l'**agent SNMP** supportant le v1, v2 ( *v2c* ) et v3 ( v3 ).

```
#       sec.name        source        community
com2sec notConfigUser   default       public

#       groupName       securityModel  securityName
group   notConfigGroup  v1             notConfigUser
group   notConfigGroup  v2c            notConfigUser
group   notConfigGroup  usm            shinken

#       name          incl/excl      subtree         mask(optional)
view    shinken       included       .1.3.6.1

#        group           context  sec.model  sec.level  prefix  read        write  notif
access   notConfigGroup  ""       any        noauth     exact   shinken      none   none

rouser shinken

# Inclus la configuration necessaire pour le bon fonctionnement des checks.
includeDir /etc/snmp/shinken
```

Cette configuration définit :

- une communauté " ***public*** " à utiliser pour la connexion SNMP v1 et v2
- un utilisateur " ***shinken*** " à utiliser pour la connexion SNMP v3

Ensuite, il est NÉCESSAIRE de suivre :

- L'étape Connexion SNMP v3 pour finaliser la configuration V3.
- La configuration nécessaire au check pour le bon fonctionnement des checks.
- L'installation des dépendances nécessaires aux checks.

### Configuration pas à pas

Si vous avez déjà votre propre configuration SNMP personnalisée, ou que vous avez des besoins spécifiques, vous pouvez suivre la configuration pas à pas.

Vous trouverez également la documentation officielle de la configuration **snmpd** ici.

#### Connexion SNMPv1v2

Il est possible de passer cette étape si vous souhaitez uniquement configurer la connexion SNMPv3.

La ligne suivante permet de créer une communauté, ici " ***public*** " et de l'associer à un nouveau nom de sécurité.

```
####
# First, map the community name "public" into a "security name"

#       sec.name       source        community
com2sec notConfigUser  default       public
```

Par défaut, la communauté est définie à ***public*** dans **les modèles de supervisions du pack**.

- Si vous modifiez vos paramètres, il faudra donc les modifier dans l'interface de configuration.

Il est possible de changer le champ " ***source*** " pour restreindre l'accès à la communauté définit, pour une adresse ou une plage d'adresse.

```
#       sec.name       source                              community
com2sec notConfigUser  my.host.com                                 public    # Hostname
com2sec notConfigUser  192.0.0.123                                 public          # Address
com2sec notConfigUser  10.10.10.0/255.255.255.0            public    # IP/MASK
```

Ensuite, il est nécessaire de rattacher le nom de sécurité crée à un groupe, et à un modèle de sécurité.

```
#       groupName       securityModel  securityName
group   notConfigGroup  v1             notConfigUser
group   notConfigGroup  v2c            notConfigUser
```

### Connexion SNMP v3

Il est possible de passer cette étape si vous souhaitez uniquement configurer la connexion SNMPv1v2.

La connexion SNMP v3 nécessite la mise en place d'un utilisateur qui sera utilisé pour se connecter sur les hôtes supervisés.

Voici un exemple de création d'un utilisateur sur la machine supervisée qui sera interrogée par le pack en SNMP v3.

Arrêtez le service SNMP pour pouvoir lancer la commande de création d'un utilisateur :

```
service snmpd stop
```

Créez vote utilisateur avec ses informations d'identification :

```
net-snmp-create-v3-user -ro -A shinkenpassword -a SHA -X shinkenencryptionkey -x AES shinken
```

Redémarrez le service SNMP :

```
service snmpd start
```

À noter qu'ici, nous avons défini :

- **shinken**: nom de l'utilisateur côté serveur SNMPv3
- **shinkenpassword**: mot de passe de l'utilisateur. Attention : il ne peut pas être plus petit que 8 caractères.
- **shinkenencryptionkey**: clé de chiffrement pour cet utilisateur
- **AES**: protocole de chiffrement de l'utilisateur
- **SHA**: méthode de hashage des informations de l'utilisateur

Ces **paramètres sont par défaut dans le pack et seront utilisés dans les modèles de supervisions** pour interroger les équipements supervisés. Si vous créez vos propres paramètres, il faudra donc les modifier dans l'interface de configuration.

Ensuite, il est nécessaire de rattacher l'utilisateur crée à un groupe, et à un modèle de sécurité.

```
#     groupName      securityModel securityName
group notConfigGroup usm           shinken
```

usm signifie User Security Model et a été introduit et utilisé pour SNMP v3

### Autorisations d'accès aux données

Quelle que soit la version de SNMP configuré, V1, V2 ou V3, **il est essentiel de configurer l'accès aux données**.

Une fois les utilisateurs, noms de sécurités, et groupes créés, il faut ajouter une vue pour leur donner accès aux données.

```
#       name          incl/excl      subtree         mask(optional)
view   shinken        included       .1.3.6.1
```

Ici la vue " ***shinken*** " définit l'accès à l'arbre " ***.1.3.6.1*** ". L'ensemble des OIDs qui commencent par " ***.1.3.6.1*** " sont donc inclus.

Pour finir, il faut donner l'accès à la vue au groupe définit plus tôt.

```
#        group           context  sec.model  sec.level  prefix  read        write  notif
access   notConfigGroup  ""       any        noauth     exact   shinken     none   none
```

Ici seulement les droits de lectures ont été donnés au groupe " ***notConfigGroup*** "

### Test de connexion SNMPv1v2

Sur votre machine locale, vous pouvez exécuter les commandes suivantes pour tester votre configuration SNMP.

Tester la configuration v2 ( *les paramètres d'authentification peuvent être à changer* ) :

```
snmpwalk localhost -v2c -c public  1.3.6.1.4.1
```

Le résultat attendu est une longue liste d'OID et de leurs valeurs associés, comme ci-dessous :

```
UCD-SNMP-MIB::memIndex.0 = INTEGER: 0
UCD-SNMP-MIB::memErrorName.0 = STRING: swap
UCD-SNMP-MIB::memTotalSwap.0 = INTEGER: 5119996 kB
UCD-SNMP-MIB::memAvailSwap.0 = INTEGER: 4175664 kB
UCD-SNMP-MIB::memTotalReal.0 = INTEGER: 1820668 kB
UCD-SNMP-MIB::memAvailReal.0 = INTEGER: 106600 kB
...
```

### Test de connexion SNMPv3

Sur votre machine locale, vous pouvez exécuter les commandes suivantes pour tester votre configuration SNMP.

Tester la configuration v3 ( *les paramètres d'authentification peuvent être à changer* ) :

```
# Tester authPriv
snmpwalk localhost -v3 -l authPriv -u shinken -a SHA -A "shinkenpassword" -x AES -X "shinkenencryptionkey"
1.3.6.1.4.1

# Tester authNoPriv
snmpwalk localhost -v3 -l authNoPriv -u shinken -a SHA -A "shinkenpassword" 1.3.6.1.4.1

# Tester noAuthNoPriv
snmpwalk localhost -v3 -l noAuthNoPriv -u shinken 1.3.6.1.4.1
```

Le résultat attendu est une longue liste d'OID et de leurs valeurs associés, comme ci-dessous :

```
UCD-SNMP-MIB::memIndex.0 = INTEGER: 0
UCD-SNMP-MIB::memErrorName.0 = STRING: swap
UCD-SNMP-MIB::memTotalSwap.0 = INTEGER: 5119996 kB
UCD-SNMP-MIB::memAvailSwap.0 = INTEGER: 4175664 kB
UCD-SNMP-MIB::memTotalReal.0 = INTEGER: 1820668 kB
UCD-SNMP-MIB::memAvailReal.0 = INTEGER: 106600 kB
...
```

### Configuration SNMP nécessaire aux checks

Les checks du pack **linux-by-SNMP__shinken** nécessitent la configuration ci-dessous pour fonctionner.

```
# linux-by-SNMP__shinken snmpd configuration file
# This file is essential for checks to work
# The file must be installed on hosts supervised by Shinken
# It must be included from the main snmpd configuration file

# check : Disks Usage by SNMP
includeAllDisks 10%

# check : Ntp Sync by SNMP
extend shinken__linux-by-snmp__ntp-sync__ntpq   /bin/sh -c "export LC_LANG=C && unset LANG && ntpq -p ; date
+'%H:%M:%S.%3N'"

# check : Ntp Sync Chrony by SNMP
extend shinken__linux-by-snmp__ntp-sync-chrony__chronyc /bin/sh -c "export LC_LANG=C && unset LANG &&
chronyc tracking ; date +'%H:%M:%S.%3N'"

# check : Stats CPU by SNMP
extend shinken__linux-by-snmp__stats-cpu__processes-cpu-time /bin/sh -c "export LC_LANG=C && unset LANG &&
awk '{ut[\$1]=\$14; st[\$1]=\$15} END { system(\"sleep 1\"); for (p in ut) { getline < (\"/proc/\" p \"
/stat\"); split(\$0, d, \" \"); printf \"%s %d %d %d %d\\n\", p, ut[p], st[p], d[14], d[15] } }' /proc/[0-9]*
/stat"
extend shinken__linux-by-snmp__stats-cpu__processes-cpu /bin/sh -c "export LC_LANG=C && unset LANG && ps -eo
pcpu,pid,args --sort=-pcpu --no-headers"
extend shinken__linux-by-snmp__stats-cpu__frequency /bin/sh -c "export LC_LANG=C && unset LANG && cat /proc
/cpuinfo | grep 'cpu MHz' | uniq | cut -d ' ' -f 3"
extend shinken__linux-by-snmp__stats-cpu__mpstat /bin/sh -c "export LC_LANG=C && unset LANG && mpstat -P ALL
1 1"
```

Copiez cette configuration et enregistrez-la dans un fichier : " ***/etc/snmp/shinken/linux-by-SNMP__shinken.conf*** "

```
mkdir -p /etc/snmp/shinken/
vim /etc/snmp/shinken/linux-by-SNMP__shinken.conf
```

Il faut ensuite inclure ce fichier depuis la configuration principale snmpd.
Dans le fichier " ***/etc/snmp/snmpd.conf*** " :

```
vim /etc/snmp/snmpd.conf
```

Assurez vous d'avoir la ligne suivante, sinon ajoutez la :

```
includeDir /etc/snmp/shinken
```

### Mise à jour des commandes SNMP sur les Linux supervisés

Comme expliqué dans le chapitre précedent, le pack a besoin d'une configuration SNMP spécifique sur les Linux supervisés afin d'exécuter correctement ses checks.

Lors de mise à jours du packs, cette configuration peut évoluer. Le script "**update-SNMP-check-conf.sh**" dans le répertoire "**supervised-host**" permet de mettre à jour cette configuration.

Le script va **ÉCRASER** le fichier "**/etc/snmp/shinken/linux-by-SNMP__shinken.conf**" et le remplacer par la nouvelle configuration.

- Un fichier de sauvegarde "**/etc/snmp/shinken/log/linux-by-SNMP__shinken.conf.[DATE].bak**" sera généré avant d'écraser la configuration par défaut.

Exemple :

Si vous souhaitez utiliser le script, exécutez :

```
./update-SNMP-check-conf.sh --override
```

Cette commande configure immédiatement le fichier avec les commandes nécessaires au pack ( *les valeurs par défaut fournies dans le pack seront utilisées* ).

## Installations des dépendances

Certain checks nécessitent des dépendances sur le serveur Linux à superviser. Il faut les installer pour leur bon fonctionnement.

### Stats CPU by SNMPvX

Le check CPU Stats SNMPvX utilise le paquet **SysStat**, nécessaire au bon fonctionnement du check. Il faut l'installer avec la commande suivante :

```
# Debian, Ubuntu
apt-get install sysstat

# RHEL, Alma, Rocky, Centos, Fedora, OpenSUSE
yum install sysstat

# Arch, Manjaro
pacman -Syy sysstat
```

## Configuration du module de sécurité Linux

Certain checks du pack nécessitent l'accès en lecture à des fichiers du système Linux supervisé. Il est possible que le serveur Linux supervisé ait un module de sécurité Linux ( LSM ) qui va protéger l'accès à ces fichiers.

Alors, il faut vérifier quel module de sécurité est configuré sur votre Linux, et donner les permissions suffisantes pour le bon fonctionnement des checks.

Les checks concernés sont :

- Connection Failed by SNMPvXXX
- Stats NFS by SNMPvXXX

### Configuration pour SELinux

#### Vérification de SELinux

SELinux est installé par défaut sur les distributions **RHEL, Alma, Rocky, Centos, Fedora**.

Vous pouvez vérifier si SELinux est activé avec la commande :

```
sestatus
```

Vous devriez observer parmi le résultat les ligne suivante :

```
SELinux status:                 enabled
Current mode:                   enforcing
```

Si SELinux est bien activé et en mode 'enforcing' , vous pouvez alors rajouter des règles afin de permettre au service SNMP (snmpd) à accéder aux fichiers voulus.
Si un autre module de sécurité est installé sur votre hôte distante, il faudra le configurer de façon similaire.

### Configuration de SELinux

#### Résolution par script

Dans le script de configuration d'hôte livré dans le pack, une option permet de rajouter ces règles.
Déployez le dossier '*supervised-host*' sur votre hôte ( scp, ftp .. ).
Sur le serveur Linux supervisé, exécutez :

```
cd supervised-host
./configure-host.sh --configure-selinux
```

#### Résolution manuelle

Sur le serveur Linux supervisé, exécutez les commandes suivantes :

```
mkdir -p /etc/selinux/shinken
vim /etc/selinux/shinken/linux-by-SNMP__shinken.te
```

Dans le fichier, remplissez et sauvegardez :

```
module linux-by-SNMP__shinken 1.0;
require {
    type snmpd_t;
    type sysctl_rpc_t;
    type faillog_t;
    class file { read open getattr };
    class dir { search };
}
# Rules for check Stats NFS by SNMPvXXX
# Allow snmpd to read /proc/net/rpc/nfsd
allow snmpd_t sysctl_rpc_t:file { read open getattr };
# Autorisation pour accéder au dossier /proc/net/rpc
allow snmpd_t sysctl_rpc_t:dir { search };

# Rules for check Connection Failed by SNMPvXXX
# Allow snmpd to read /var/log/btmp
allow snmpd_t faillog_t:file { read open getattr };
```

Puis exécutez :

```
checkmodule -M -m -o "/etc/selinux/shinken/linux-by-SNMP__shinken.mod" "/etc/selinux/shinken/linux-by-
SNMP__shinken.te"
semodule_package -o "/etc/selinux/shinken/linux-by-SNMP__shinken.pp" -m "/etc/selinux/shinken/linux-by-
SNMP__shinken.mod"
semodule -i "/etc/selinux/shinken/linux-by-SNMP__shinken.pp"
```

Ces commandes vont compiler, empaqueter et installer le module SELinux créé.

## Erreurs lors de l'utilisation du pack

Pour toute erreur survenue lors de l'exécution des checks, voir la page Erreurs du pack linux-by-SNMP__shinken

## Erreurs lors de la configuration du serveur Linux à superviser

La page suivante répertorie certaines erreurs qui peuvent intervenir lors de la configuration : Les erreurs lors de la configuration d'un serveur Linux à configurer pour le pack linux-by-SNMP__shinken