Pack Linux

Sommaire

Introduction

Cette page décrit comment le pack Linux permet de superviser des serveurs Linux au travers d'une connexion SSH. Ce pack supervise les ressources principales d'un serveur Linux comme:

- Les ressources du noyau
- La mémoire
- Le réseau
- Les disques
- Les ressources CPU
- Le démon SSH
- Le démon NTP

Ce pack récupère les informations nécessaires à la supervision en se connectant via SSH à l'hôte distant. Pendant l'installation de Shinken, la procédure d'installation crée automatiquement un utilisateur "shinken" sur tous les serveurs Shinken (*et donc les Pollers également*).

- Le pack Linux utilise par défaut cet utilisateur pour exécuter les commandes.
- L'authentification sur l'hôte distant s'effectue à l'aide d'une clé SSH.
- L'utilisateur distant ne requiert pas de droits particuliers sur le système.

Configuration de la connexion SSH

Côté client (serveur supervisé)

Créer un utilisateur local "shinken" avec un dossier "home" et un mot de passe

```
adduser -m -r shinken
passwd shinken
```

Côté Shinken (serveur hébergeant le Poller)

SI la machine qui héberge le Poller est une machine en RHEL/Alma/Rocky 9 ou Debian 13 et que des machines à superviser ont une version de OpenSSH <= 5.3 (CentOS 6 par exemple) une configuration supplémentaire est nécessaire. (voir Cas particulier d'un serveur à superviser avec une version d'OpenSSH inférieure ou égale à 5.3)

Cas général

Se connecter sur le serveur en tant que l'utilisateur "shinken" et copier la clé SSH vers le serveur à superviser :

```
[root@shinken-poller ~]# su - shinken
[shinken@shinken-poller ~]#  ssh-copy-id  remote_host
The authenticity of host '192.168.1.19 (192.168.1.19)' can't be established.
RSA key fingerprint is 00:ff:ee:dd:cc:bb:aa:d6:d3:79:1d:f6:93:47:80:27.
Are you sure you want to continue connecting (yes/no)? yes
shinken@remote_host's password: XXXXXXXXXX
Now try logging into the machine, with "ssh '192.168.1.19'", and check in:
  .ssh/authorized_keys
to make sure we haven't added extra keys that you weren't expecting.

ssh shinken@remote_host -i .ssh/id_rsa
```

Cette copie de clé permet d'autoriser le Poller à se connecter sur la machine à superviser.

Sur une architecture Shinken qui contient plusieurs Pollers, il faudra alors effectuer cette opération sur chaque Poller pour permettre à chaque Poller de se connecter en SSH sur le serveur à superviser.

Cas particulier d'un serveur à superviser avec une version d'OpenSSH inférieure ou égale à 5.3

Si la machine qui héberge le Poller est une distribution RHEL/Alma/Rocky 9 ou Debian13 et que le serveur à superviser a une version ancienne de OpenSSH (*<= 5.3*) par exemple pour des machines CentOS 6 une configuration supplémentaire est nécessaire afin de pouvoir se connecter en SSH à ces serveurs.

Sur une architecture Shinken qui contient plusieurs Pollers, il faudra alors effectuer cette opération sur chaque Poller pour permettre à chaque Poller de se connecter en SSH sur le serveur à superviser.

Sur RHEL/Alma/Rocky 9

Pour se connecter à une machine avec une ancienne version d'OpenSSH, il est nécessaire de réactiver l'algorithme SHA1 au niveau système, nécessaire pour négocier les connexions SSH avec OpenSSH <= 5.3. Ce qui va ramener le niveau de sécurité à un niveau équivalent à celui d'une machine RHEL/Alma/Rocky 8 avec sa politique par défaut.

Les distributions RHEL/Alma/Rocky 9 n'autorisent par défaut aucun des algorithmes de chiffrement utilisables sur les versions d'OpenSSH inférieures à 5.3 il faut donc mettre à jour la politique par défaut pour pouvoir superviser des machines utilisant ces versions.

Pour ce faire, il faut lancer la commande suivante sur la machine du Poller :

```
[root@shinken-poller ~]# update-crypto-policies --set DEFAULT:SHA1
```

Ensuite, il suffit de suivre le cas général pour copier la clé SSH sur la machine à superviser. (*voir* Cas général)

Sur Debian 13

Sur Debian 13 par défaut, les clés SSH générées sont avec un algorithme qui n'est pas pris en charge par les anciennes versions d'OpenSSH il faut donc créer une nouvelle clé utilisable avec ces versions et configurer la machine du Poller pour l'utiliser.

Il faut se connecter avec l'utilisateur shinken de la machine qui contient le Poller :

```
[root@shinken-poller ~]# su - shinken
```

Généré la nouvelle clé, elle s'appelle id_rsa_debian ici :

```
[shinken@shinken-poller ~]# ssh-keygen -t rsa -b 2048 -f ~/.ssh/id_rsa_debian -N ""
```

Il faut ensuite rajouter les lignes suivantes dans la configuration de ssh de l'utilisateur shinken qui vont autoriser l'utilisation d'une sécurité réduite pour la connexion sur les machines ayant une version ancienne d'OpenSSH.

Il faut remplacer IP_DU_SERVEUR par l'adresse du/des serveurs à superviser qui ont une version ancienne d'OpenSSH.

La valeur possible est un mix d'adresses IP (*ex : 192.168.1.78*), de plages d'IP (*ex : 192.168.1.**), d'adresses (*ex : monserveur.local*) toutes séparées par un espace.

```
[shinken@shinken-poller ~]# vi ~/.ssh/config
```

Il faut également changer le nom de la clé si elle ne porte pas le même nom.

```
Host IP_DU_SERVEUR
    HostKeyAlgorithms +ssh-rsa
    PubkeyAcceptedKeyTypes +ssh-rsa
    KexAlgorithms +diffie-hellman-group14-sha1
    Ciphers +aes128-cbc
    MACs +hmac-sha1
    IdentityFile ~/.ssh/id_rsa_debian
```

Il ne reste plus qu'à copier la clé sur l'utilisateur shinken du serveur à superviser :

```
ssh-copy-id -i ~/.ssh/id_rsa_debian.pub -o PreferredAuthentications=password -o PubkeyAuthentication=no
shinken@IP_DU_SERVEUR
```

Il faut maintenant tester la connexion sur le serveur distant :

```
ssh shinken@remote_host -i .ssh/id_rsa_debian
```

Définition de surcharges locales aux connexions SSH sur un poller via le fichier /var/lib/shinken/.ssh/config

Il est possible sur un poller de définir un comportement spécifique pour les connexions SSH du pack Linux qui ne seront valide que sur ce poller.

Ceci peux être utile si par exemple le poller a besoin de passer par une machine proxy pour se connecter aux serveurs distants.

Cette définition spécifique se fait via le fichier **/var/lib/shinken/.ssh/config** (configuration standard du client openssh sur linux) qui sera lue par les commandes du pack Linux avant d'effectuer leurs connexions.

Plusieurs paramètres sont pris en compte:

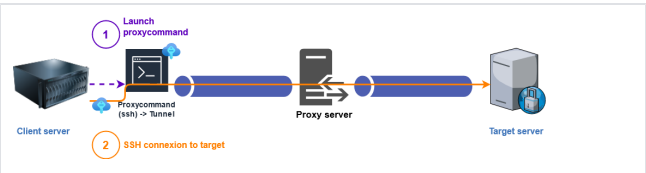
- **proxycommand**: permet de définir une commande qui sera appelée pour établir un tunnel, comme pour passer par une machine bastion par exemple
- **key_filename**: chemin vers la clé SSH publique utilisée pour la connexion
- **port**: port utilisé pour la connexion
- **username**: utilisateur distant utilisé pour la connexion

Comme mis dans la documentation de openssh, ces paramètres peuvent être mis dans un bloc au nom de la machine, ou un bloc "*" qui sera valable pour tous les hôtes.

Exemple de surcharge SSH afin de passer par un serveur bastion via un paramètre proxycommand

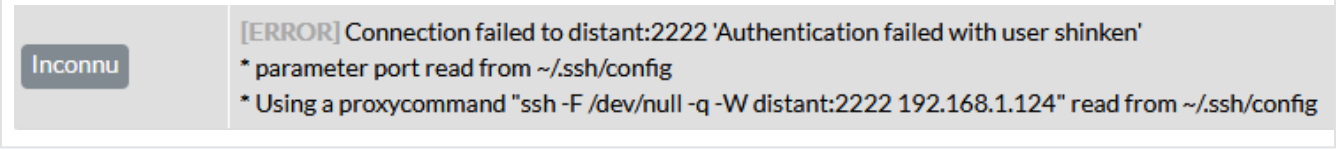
Par exemple, pour faire passer toutes ses connexions SSH par un serveur bastion, on peux définir le fichier **/var/lib/shinken/.ssh/config** (***avec les droits 600 pour l'utilisateur shinken***):

```
Host *
   ProxyCommand ssh -F /dev/null -q -W %h:%p IP-
BASTION
```



Messages d'erreurs additionnels quand une surcharge est présente sur le poller

En cas de problème de connexion quand un fichier config existe sur le poller et a fourni une surcharge, la ou les surcharges seront affichés dans le message d'erreur afin d'aider à comprendre d'où viennent les paramètres de connexions:



- le *port* a été surchargé en 2222
- une *proxycommand* a été utilisée pour effectuer la connexion

Comment utiliser le pack Linux

Le pack Linux peut être utilisé en appliquant le modèle d'hôte "linux" sur un hôte.

Cette opération peut être effectuée de 2 manières différentes :

Via l'interface de Configuration

Dans l'interface de Configuration, créer et Éditer un Hôte et ajouter le modèle d'hôte "linux" dans la propriété " **Modèles d'hôte hérités**" à l'aide du me nu déroulant.

Via un fichier de configuration d'un collecteur (cfg)

Dans un fichier de configuration .cfg, créer un hôte et définir la propriété "use" à "linux".

Ce fichier .cfg doit ensuite être importé dans Shinken Entreprise via une source (*plus d'informations sont disponibles dans la page de documentation sur la* Syntaxe des fichiers d'imports).

Problèmes usuels et connus

Le check NTP est en statut "Inconnu"

Le démon ntp n'est probablement pas installé.
Installer et configurer le démon ntp.

Le check "Reboot" est en "Critique".

La valeur par défaut de ce check est de 3600 secondes (*1h*).
En dessous de ce seuil, un redémarrage a pour effet de mettre le check en état critique. Modifier le seuil selon les besoins pour éviter les notifications ou fausses alertes.

Le check "CPU Stats" est en statut "Inconnu".

Vérifier que le paquet "sysstat" est bien installé sur le système.

Résumé des checks

	Nom du check	Description	Valeurs possibles	Seuil d'avertissement par défaut	Donnée seuil d'avertissement	Seuil Critique par défaut	Donnée seuil Critique
1	CPU Stats	Récupère les informations des CPU via la commande "mpstat"	0-100	> 80	CPU_WARN	> 90	CPU_CRIT
2	Disks	Récupère les informations sur l'utilisation des disques via la commande "df"	0-100	> 90	STORAGE_WARN	> 95	STORAGE_CRIT
3	Disks stats	Récupère les statistiques de lecture/écriture des disques depuis /proc	N/A	N/A	N/A	N/A	N/A
4	Kernel stats	Récupère des statistiques sur le noyau Linux via /proc /vmstat	N/A	N/A	N/A	N/A	N/A
5	Load Average	Récupère la charge (Load Average) du système via /proc /loadavg	0-n,0-n,0-n	> "1.5,1.5,1.5"	LOAD_WARN	> "3,3,3"	LOAD_CRIT
6	Memory	Récupère les informations sur la consommation de mémoire via la commande "free"	0-100	> 90	MEMORY_WARN	> 95	MEMORY_CRIT
7	NET Stats	Récupère les statistiques des interfaces réseau via /proc /net/dev	N/A	N/A	N/A	N/A	N/A
8	NFS Stats	Récupère les statistiques de lecture/écriture des points de montage NFS via /proc/net/rpc/nfsd	N/A	N/A	N/A	N/A	N/A
9	NTPSync	Récupère le décalage de temps avec le serveur NTP	0-n	> 40		> 60	

10	Read-only Filesystems	Vérifie si un système de fichiers est en lecture seule	N/A	N/A	N/A	N/A	N/A
11	Reboot	Récupère le temps d'uptime du serveur via la commande "uptime"	0-n	N/A	N/A	< 3600	LINUX_UPTIME_CRIT
12	SSH connexions	Vérifie qu'une connexion SSH vers le serveur est bien possible	N/A	N/A	N/A	N/A	N/A
13	TCP states	Récupère les statistiques d'usage TCP	N/A	N/A	N/A	N/A	N/A

Check NtpSync

Selon le démon utilisé pour NTP, le check peut obtenir des détails supplémentaires sur la synchronisation de l'horloge.

Si le démon NTP utilisé est chrony, l'utilisation du modèle d'hôte "chrony" en plus du modèle "linux" permet au check d'obtenir plus de détails sur l'état de la synchronisation NTP.

Personnaliser les seuils d'Avertissement et Critique

Le pack Linux définit des seuils par défaut sur les checks qu'il utilise qui sont modifiables.

Changer les seuils pour un seul hôte

Les modèles d'hôte livrés dans Shinken contiennent souvent des variables permettant de changer les seuils et options des checks.

Pour changer ces seuils sur un seul hôte en particulier, la manière la plus simple est de changer des variables via les données présentes sur l'hôte directement:

- Dans l'interface de Configuration, éditer l'hôte et aller dans l'onglet "**Données**"
  Par exemple, changer la donnée "**CPU_CRIT**" à 60 aura pour effet de changer le seuil Critique du check "CPU Stats" à 60
- Via un fichier de configuration (cfg).
  Pour changer le seuil Critique du check "**CPU Stats**", éditer l'hôte et changer la donnée "**_CPU_CRIT**"

Changer les seuils pour tous les hôtes qui utilisent le modèle d'hôte "linux"

Pour changer le seuil sur tous les hôtes qui utilisent le modèle d'hôte "linux", on pourrait être tenté de modifier directement le modèle d'hôte "linux" et changer les données personnalisées de cet hôte.

Mais, dans la prochaine mise à jour de Shinken Entreprise, ces modèles d'hôtes/checks peuvent être modifiés, ce qui occasionne un changement de comportement et causé des problèmes dans la configuration Shinken.

L'alternative conseillée est de cloner les éléments utilisés dans le pack Linux et de les renommer. On peut ensuite modifier sans aucun risque lié aux mises à jour de Shinken Entreprise.

Ce pack linux cloné peut ensuite être modifié et personnalisé selon les besoins.

Par exemple, le changement des seuils s'effectue de la même manière en changeant les données comme décrites dans la section précédente, mais dans le modèle d'hôte linux au lieu de faire cette modification directement dans les hôtes.

Pour information, détail sur le chiffrement de la connexion

La librairie utilisée pour se connecter en SSH est la librairie paramiko (https://pypi.org/project/paramiko/)

La version utilisée est la:

- 1.18.5 (de 2018) à partir de la version v02.08.02
  - les versions supérieures ne sont pas compatibles avec la RH6
  -
    - aes128-ctr
    - aes192-ctr
    - aes256-ctr
    - aes128-cbc
    - blowfish-cbc
    - aes192-cbc
    - aes256-cbc
    - 3des-cbc
    - arcfour128
    - arcfour256
  -
    - diffie-hellman-group1-sha1
    - diffie-hellman-group14-sha1
    - diffie-hellman-group-exchange-sha1
    - diffie-hellman-group-exchange-sha256
- 1.15.1 (de 2014) avant la v02.08.02
  -
    - aes128-ctr
    - aes256-ctr
    - aes128-cbc
    - blowfish-cbc

- aes256-cbc
- 3des-cbc
- arcfour128
- arcfour256
- diffie-hellman-group14-sha1
- diffie-hellman-group-exchange-sha1
- diffie-hellman-group1-sha1