

Paquetage SSHD - Secure Shell Daemon

Version 4.0.0-stable-x86_64-r60708

Frank Meyer
courriel: frank@fli41.de

L'équipe fli4l
courriel: team@fli41.de

17 août 2022

Table des matières

1	Documentation du paquetage SSHD	3
1.1	SSHD - Secure-Shell, Secure-Copy	3
1.1.1	Installation du service Secure-shell	3
1.1.2	Installation du dbclient	7
1.1.3	Installation du client plink	7
1.1.4	Installation d'un serveur sftp	8
1.1.5	Littérature	8
	Table des figures	9
	Liste des tableaux	10
	Index	11

1 Documentation du paquetage SSHD

1.1 SSHD - Secure-Shell, Secure-Copy

Secure Shell offre la possibilité d'ajouter une connexion codée sur votre routeur fli4l. De plus, avec la commande Secure Copy, vous pouvez transférer des fichiers cryptés sur le routeur fli4l. Si à la connexion [vous utilisez une clé publique](#) (Page 6), vous pourrez alors exécuter des commandes sur le routeur fli4l et transférer des fichiers script qui pourront être exécutés. A partir de la version le 2.1.7 il a été rajouté le serveur SSH2.

1.1.1 Installation du service Secure-shell

OPT_SSHD Installation par défaut : `OPT_SSHD='no'`

Si vous voulez accéder au routeur au moyen du ssh, il faut paramétrer la variable `OPT_SSHD` sur 'yes'. Cela installe un serveur-ssh dropbear sur le routeur fli4l. Cela permet également de copier les fichiers sur le routeur.

SSHD_ALLOWPASSWORDLOGIN Installation par défaut : `SSHD_ALLOWPASSWORDLOGIN='yes'`

Si vous paramétrez la variable `SSHD_ALLOWPASSWORDLOGIN` sur 'no', la connexion ssh, avec un mot de passe au routeur fli4l, ne sera plus possible. Alors, la connexion au routeur se fera seulement au moyen d'une clé privée et d'une clé publique (key private/public). Cela suppose qu'une [clé publique](#) (Page 6) soit installée sur le routeur.

SSHD_CREATEHOSTKEYS Installation par défaut : `SSHD_CREATEHOSTKEYS='no'`

Le serveur-ssh a besoin d'une hostKey (ou clé d'hôte) qui doit être exceptionnelle et unique pour que le serveur-ssh s'identifie clairement au client-ssh. Certes le paquetage `opt-sshd` est fourni avec une hostKey, qui permet de se connecter pour la première fois au routeur fli4l avec le client-ssh, mais cette hostKey qui a été livrée doit être remplacée le plus vite possible, la hostKey sera remplacée et connu que par vous sont même. Générer votre propre hostKey est important parce que c'est la seule manière possible de vous protéger contre les soi-disant Man-In-The-Middle-Attack. Votre client ssh peut remarquer, si un prétendu pirate est sur votre routeur fli4l, car le pirate ne connaît pas votre hostKey. Votre client ssh vous avertira par un message, si votre hostKey a été changé par le pirate.

La création de votre hostKey (ou clé d'hôte) est entièrement automatique, une fois que le paramètre la variable `SSHD_CREATEHOSTKEYS` est sur 'yes'. Ce processus est très gourmand et peut prolonger, le temps du boot de plusieurs minutes. Si vous avez démarré le routeur fli4l avec l'activation de la variable `SSHD_CREATEHOSTKEYS`, une (ou plusieurs) hostKey(s) sera produite dans le répertoire `/tmp/ssh`. Les fichiers produits à cet endroit, doivent être copiés dans le répertoire, `etc/ssh` de votre sous-répertoire config (sur l'ordinateur donc vous avez créé le média de boot fli4l). Dans mon cas, voici l'arborescence du répertoire fli4l et le répertoire `config.babel` :



FIGURE 1.1 – Structure des répertoires fi4l

Faite attention, au sous-répertoire `config`, vous devez avoir le répertoire `etc` et le répertoire `ssh`. C'est précisément à cet endroit que la ou les `hostKey(s)` produite est copiée. A partir de la version 2.1.5 de `fli4l`, les fichiers de votre sous-répertoire `config`, sont prioritaire par rapport au sous-répertoire `opt`. Ainsi à la prochaine mise à jour de la version du routeur `fli4l`, les fichiers qui se trouvent dans le répertoire `config/etc/ssh` seront intégrés et non les fichiers qui se trouvent dans le répertoire `opt/etc/ssh`. Ainsi, il est possible pour chaque routeur `fli4l` d'utiliser ces propre `hostKey`. Lors de la construction des fichiers du routeur `fli4l`, un message apparaîtra à la fin, «`appending config specific files to opt.img ...`». Alors, tous les fichiers venant du répertoire `config` seront listés et non les fichiers du répertoire `opt`.

```
#
# appending config specific files to opt.img ...
#
etc/ssh/dropbear_dss_host_key
etc/ssh/dropbear_rsa_host_key
```

Si vous avez produit une nouvelle clé d'hôte, remettez le paramètre de la variable `SSHD_CREATEHOSTKEYS` sur `'no'`, de sorte que le script ne génère plus de nouvelle clé d'hôte à chaque démarrage du routeur `fli4l`.

Après une mise à jour de votre routeur `fli4l`, si vous vous connectez au routeur, un message d'avertissement sera indiqué (différent selon le programme) par votre client `ssh`, qui attire l'attention sur un changement de votre `hostKey`. C'est normal, puisque vous venez justement de changer la `hostKey`, par celle fournie par `fli4l`. Suivez les instructions de votre client `ssh`, et modifié de façon permanente votre `hostKey`. Si vous recevez encore une fois ce message d'avertissement à une date ultérieure, vous devriez vérifier dans tous les cas, le pourquoi de cet avertissement, qui a été émis et non pas accepter aveuglément le changement de la `hostKey`.

```

#####
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!    @
#####
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)~!
It is also possible that the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
ca:a4:ab:e7:af:d8:68:05:d3:1f:e6:15:08:d6:ed:36.
Please contact your system administrator.
Add correct host key in /home/babel/.ssh/known_hosts to get rid of this message.
Offending key in /home/babel/.ssh/known_hosts:7
Password authentication is disabled to avoid man-in-the-middle attacks.
```

SSHD_PORT Installation par défaut : `SSHD_PORT='22'`

Avec la variable `SSHD_PORT` vous pouvez indiquer un port différent du port par défaut sur lequel le serveur `ssh` doit fonctionner.

Si vous voulez que l'on puisse accéder au `ssh` de l'extérieur, il faut paramétrer la variable `INPUT_ACCEPT_PORT_x` (Page ??)

Saisie des commandes pour utiliser le protocole `SSH` sur un ordinateur `Unix` ou `Linux` avec `fli4l` :

- ssh - Secure Shell
- scp - Secure Copy

Les programmes pour Windows sont aussi disponibles :

<http://www.chiark.greenend.org.uk/~sgtatham/putty/>

<http://winscp.net/eng/docs/lang:fr>

<http://www.tectia.com/en/en.iw3>

SSHD_PUBLIC_KEY_N Installation par défaut : `SSHD_PUBLIC_KEY_N='0'`

Vous indiquez dans la variable `SSHD_PUBLIC_KEY_N` le nombre de clés publiques qui doit être copié sur le routeur fli4l.

SSH permet l'authentification, à l'aide d'une procédures de cryptage asymétrique. Le contrôle d'authentification est utilisé avec une key Public/Privat à la place du nom d'utilisateur et du mot de passe. On s'épargne ainsi d'entrée d'un mot de passe. On produit la paire de clés à l'aide de `keygen-ssh` (ou `puttygen`, si on emploi `putty` sous Windows avec le client `ssh`). Optionnel une passphrase (ou phrase confidentielle) peut être indiqué pour une signature de la clé (on a besoin de ce mot de passe, si vous voulez utiliser la clé) cela augmente encore plus la sécurité. Si vous utilisez une passphrase vous devez réfléchir à l'utilisation d'un agent de clé (voir `ssh-agent` ou `pageant`).

Important: *il y a deux clés, une clé publique et une clé privée, la clé privée, doit est traitée avec soin comme un mot de passe, puisqu'il remplit la même fonction. La clé privée est installée dans le client `ssh`. La clé publique est installée sur le routeur fli4l. Nous avons mis à disposition les variables suivant `SSHD_PUBLIC_KEY_x` ou `SSHD_PUBLIC_KEYFILE_x` pour gérer la clé publique.*

Pour de plus amples informations, voir les pages du manuel `ssh` et ces composants, pour la documentation de `putty` voir (<http://www.chiark.greenend.org.uk/~sgtatham/putty/>).

SSHD_PUBLIC_KEY_x Dans cette variable vous pouvez indiquer la partie publique de la clé, pour l'utilisateur qui veut obtenir un accès `ssh` sur le routeur fli4l. Le plus simple pour récupérer la clé est d'utiliser le Cut-and-Paste (ou le couper-coller) à partir de la fenêtre du terminal. Cela pourrait par ex. ressembler à ceci :

```
SSHD_PUBLIC_KEY_1='1024 ... username@hostname'
```

Important: *la clé ne contient pas de saut de ligne. Vous pouvez insérer les clés, produit par `puttygen` en externe, avec Cut-and-Paste (ou couper-coller). Cependant, les sauts de ligne doivent être supprimés.*

Actuellement, les clés prises en charge pour les méthodes de chiffrement sont les suivantes :

- DSA
- RSA
- ECDSA

SSHD_PUBLIC_KEYFILE_N Installation par défaut : `SSHD_PUBLIC_KEYFILE_N='0'`

Dans cette variable vous pouvez indiquer le nombre de fichiers Key. Au lieu de copier le contenu de la clé publique dans le fichier `sshd.txt`, vous pouvez copier la clé publique directement dans l'archive-opt. Cela fonctionne comme la variable `SSH_CREATEHOSTKEYS` décrit plus haut. Copiez votre clé publique dans un fichier et placez-le dans le répertoire `<config>/etc/ssh`.

SSHD_PUBLIC_KEYFILE_x Dans cette variable vous pouvez indiquer le nom du fichier de la clé publique que vous avez enregistré dans répertoire `<config>/etc/ssh`.

```
SSHD_PUBLIC_KEYFILE_1='root@fli4l'
```

Actuellement, les clés prises en charge pour les méthodes de chiffrement sont les suivantes :

- DSA
- RSA
- ECDSA

SSH_CLIENT_PRIVATE_KEYFILE_N Installation par défaut : `SSH_CLIENT_PRIVATE_KEYFILE_N='0'`

Dans cette variable vous pouvez indiquer le nombre de fichiers Key. Si la clé privée est compatible avec le client ssh ou plink pour une connexion à un serveur SSH désiré, vous pouvez copier cette dernière dans le répertoire `<config>/etc/ssh`. Cela fonctionne comme la variable `SSH_CREATEHOSTKEYS` décrit plus haut. Si vous avez copié votre clé privée dans le répertoire `<config>/etc/ssh`. La clé privée au format OpenSSH sera convertie automatiquement à chaque processus de départ de fli4l au format dropbear.

SSH_CLIENT_PRIVATE_KEYFILE_x Dans cette variable vous pouvez indiquer le nom du fichier de la clé publique que vous avez enregistré dans le répertoire `<config>/etc/ssh`.

```
SSHD_PRIVATE_KEYFILE_1='babel@rootserver'
```

Actuellement, les clés prises en charge pour les méthodes de chiffrement sont les suivantes :

- DSA
- RSA
- ECDSA

1.1.2 Installation du dbclient

OPT_SSH_CLIENT Installation par défaut : `OPT_SSH_CLIENT='no'`

Si vous voulez utiliser un authentique client ssh2, vous pouvez activer dbclient dropbear en paramétrant la variable sur `OPT_SSH_CLIENT='yes'`. Ce client a l'avantage de partager de nombreux programmes codés, avec le serveur ssh dropbear. Vous pouvez ainsi épargner de la place dans l'archive-OPT. Le dbclient est compatible avec de nombreux client ssh/scp, de plus, les paramètres de commandes sont semblables. Il y a également un lien symbolique créé vers `/usr/bin/ssh`, alors cela fonctionne habituellement avec ssh `<host>` ou scp `<source> <target>`.

Si vous voulez sauvegarder la clé d'hôte du dbclient de façon permanente, vous devez copier le fichier `known_hosts` du répertoire `/.ssh` sur le routeur fli4l, dans le répertoire `config/etc/ssh`. Cela se passe comme si l'on produisait une clé d'hôte. Dans l'exemple suivant le dépaquetage fli4l se trouve dans le répertoire (dans le quel le support de boot fli4l est créée) `/home/babel/fli4l-4.0.0-stable-x86_64-r60708`. Tous les fichiers de configuration se trouvent dans le répertoire `config.babel`.

```
cd /home/babel/fli4l-4.0.0-stable-x86_64-r60708
mkdir -p config.babel/etc/ssh
scp fli4l:/.ssh/* config.babel/etc/ssh
```

1.1.3 Installation du client plink

OPT_PLINK_CLIENT Installation par défaut : `OPT_PLINK_CLIENT='no'`

Installer sur le routeur fli4l le client ssh1/ssh2/telnet. Le programme plink est la version Unix, du programme PuTTY connu sous Windows. Un appel de plink sur le routeur fli4l affiche la page d'aide, pour l'utilisation de plink.

Si vous voulez sauvegarder la hostKey (ou clé d'hôte) dans plink, de façon permanente, vous devez copier le fichier sshhostkeys à partir du répertoire /.putty sur le routeur fli4l, il doit être copier dans le répertoire config/etc/plink. Cela se passe comme si l'on produisait une hostKey. Dans l'exemple suivant le dépaquetage de fli4l se trouve dans le répertoire (dans le quel le support de boot fli4l est produite /home/babel/fli4l-4.0.0-stable-x86_64-r60708 . Tous les fichiers de configuration se trouvent dans le répertoire config.babel.

```
cd /home/babel/fli4l-4.0.0-stable-x86_64-r60708
mkdir -p config.babel/etc/plink
scp fli4l:/.putty/* config.babel/etc/plink
```

1.1.4 Installation d'un serveur sftp

OPT_SFTPSERVER Installation par défaut : OPT_SFTPSERVER='no'

Installe sur le routeur fli4l un serveur-sftp.

1.1.5 Littérature

Site Web de Dropbear SSH2 : <http://matt.ucc.asn.au/dropbear/dropbear.html>

Première version de la documentation par Claas Hilbrecht <babel@fli4l.de>, Avril 2004

Table des figures

1.1	Structure des répertoires fli4l	4
-----	---	---

Liste des tableaux

Index

OPT_PLINK_CLIENT, [7](#)
OPT_SFTPSERVER, [8](#)
OPT_SSH_CLIENT, [7](#)
OPT_SSHD, [3](#)

SSH_CLIENT_PRIVATE_KEYFILE_N, [7](#)
SSH_CLIENT_PRIVATE_KEYFILE_x, [7](#)
SSHD_ALLOWPASSWORDLOGIN, [3](#)
SSHD_CREATEHOSTKEYS, [3](#)
SSHD_PORT, [5](#)
SSHD_PUBLIC_KEY_N, [6](#)
SSHD_PUBLIC_KEY_x, [6](#)
SSHD_PUBLIC_KEYFILE_N, [6](#)
SSHD_PUBLIC_KEYFILE_x, [6](#)