

Paket SSHD - secure shell daemon

Version 4.0.0-stable-x86_64-r60708

Frank Meyer
E-Mail: frank@fli4l.de

Das fli4l-Team
E-Mail: team@fli4l.de

17. August 2022

Inhaltsverzeichnis

1	Dokumentation des Paketes SSHD	3
1.1	SSHD - Secure Shell, Secure Copy	3
1.1.1	Installation des Secure-Shell-Dienstes	3
1.1.2	Installation des dbclients	7
1.1.3	Installation des plink Clients	8
1.1.4	Installation des sftp-server	8
1.1.5	Literatur	8
	Abbildungsverzeichnis	9
	Tabellenverzeichnis	10
	Index	11

1 Dokumentation des Paketes SSHD

1.1 SSHD - Secure Shell, Secure Copy

Eine Secure-Shell bietet die Möglichkeit, eine verschlüsselte Verbindung mit dem fli4l-Router aufzunehmen. Außerdem können mit dem Secure-Copy-Befehl Dateien verschlüsselt auf den fli4l-Router übertragen werden. Wird zusätzlich eine [Public Key Anmeldung](#) (Seite 6) benutzt, können Befehle auf dem fli4l-Router und Dateiübertragungen auch scriptgesteuert ausgeführt werden. Ab der Version 2.1.7 gibt es nur noch einen SSH2 Server.

1.1.1 Installation des Secure-Shell-Dienstes

OPT_SSHD Standard-Einstellung: `OPT_SSHD='no'`

Soll der Zugriff auf den Router mittels ssh ermöglicht werden, bedarf es der Änderung auf von `OPT_SSHD` auf `'yes'`. Dies installiert den ssh-Server Dropbear auf dem fli4l-Router. Dies ermöglicht auch das Kopieren von Dateien auf den Router.

SSHD_ALLOWPASSWORDLOGIN Standard-Einstellung: `SSHD_ALLOWPASSWORDLOGIN='yes'`

Wird `SSHD_ALLOWPASSWORDLOGIN` auf `'no'` eingestellt, ist die Anmeldung mit ssh über ein Passwort auf dem fli4l-Router nicht mehr möglich. Die Anmeldung kann dann nur noch mittels privatem/öffentlichem Schlüsselpaar (private/public key) erfolgen. Dies setzt voraus, dass ein [öffentlicher Schlüssel](#) (Seite 6) auf dem Router hinterlegt ist.

SSHD_CREATEHOSTKEYS Standard-Einstellung: `SSHD_CREATEHOSTKEYS='no'`

Ein ssh-Server benötigt einen sogenannten Hostkey, der weltweit einmalig sein sollte, damit sich der ssh-Server eindeutig gegenüber einem ssh-Client identifizieren kann. Das `sshd opt`-Paket liefert zwar einen Hostkey mit, um das erste Einloggen auf dem fli4l-Router per ssh-Client zu erlauben, aber der mitgelieferte Hostkey sollte so schnell wie möglich durch einen selbst generierten, nur Ihnen bekannten Hostkey ersetzt werden. Die Generierung eines eigenen Hostkeys ist deshalb so wichtig, weil nur auf diese Weise Schutz gegen so genannte Man-in-the-Middle-Attacken möglich ist. Ihr ssh-Client bemerkt es, wenn ein Cracker vorgibt, Ihr fli4l-Router zu sein, da dem Cracker dessen Hostkey nicht bekannt ist. Ihr ssh-Client warnt Sie daraufhin mit einer Meldung, dass der Hostkey sich geändert hat.

Die Erzeugung Ihres eigenen Hostkeys geschieht vollkommen automatisch, sobald Sie die Einstellung `SSHD_CREATEHOSTKEYS` auf `'yes'` setzen. Dieser Vorgang ist sehr rechenintensiv und kann deshalb die Bootzeit um mehrere Minuten verlängern. Wenn der fli4l-Router mit aktiviertem `SSHD_CREATEHOSTKEYS` Eintrag startet, wird ein (oder mehrere) Hostkey(s) in dem Verzeichnis `/tmp/ssh` erzeugt. Die Dateien die dort stehen, kopieren Sie in das Verzeichnis `etc/ssh` unterhalb Ihres config Verzeichnisses (auf dem Rechner, auf dem sie fli4ls Bootmedium erzeugen). In meinem Fall sieht ein Directorylisting des config.babel Verzeichnisses so aus:



Abbildung 1.1: Verzeichnisstruktur fli4l

Beachten Sie, dass unterhalb des `config` Verzeichnis erst das Verzeichnis `etc` kommt und darunter dann das Verzeichnis `ssh`. Und genau dorthin muss der oder die eben erzeugte(n) Hostkey(s) kopiert werden. Ab der `fli4l` Version 2.1.5 werden Dateien die unterhalb Ihres `config` Verzeichnisses stehen vorrangig vor den Dateien aus dem `opt` Verzeichnis behandelt. Dadurch werden bei dem nächsten Update Ihres `fli4l`-Routers die Dateien aus dem Verzeichnis `config/etc/ssh` eingebunden und nicht die Dateien, die im Verzeichnis `opt/etc/ssh` stehen. So ist es möglich für jeden `fli4l`-Router, den Sie konfigurieren, einen eigenen Hostkey zu benutzen. Wenn Sie die `fli4l`-Routerdateien erzeugen, erscheint ziemlich zum Schluss die Meldung „appending config specific files to `opt.img` ...“. Dort werden dann alle Dateien aufgelistet, die aus Ihrem `config` Verzeichnis kommen und nicht aus dem `opt` Verzeichnis.

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

Wenn Sie einen neuen Hostkey erzeugt haben, setzen Sie danach den Wert `SSHD_CREATEHOSTKEYS` wieder auf `'no'`, damit die Startskripte des `fli4l`-Routers nicht ständig einen neuen Hostkey generieren.

Wenn Sie sich nach dem Update des Hostkey auf Ihrem `fli4l`-Router anmelden, wird eine (je nach Programm unterschiedliche) Warnmeldung von Ihrem `ssh`-Client ausgegeben, die Sie auf einen geänderten Hostkey hinweist. Das ist normal, da Sie ja gerade den von `fli4l` mitgelieferten Hostkey gegen den von Ihnen erzeugten Hostkey ausgetauscht haben. Befolgen Sie die Hinweise Ihres `ssh`-Client, wie Sie den geänderten Hostkey permanent übernehmen können. Sollten Sie diese Warnmeldung zu einem späteren Zeitpunkt noch einmal bekommen, sollten Sie in jedem Fall prüfen, warum diese Warnung ausgegeben wurde und nicht einfach blind den geänderten Hostkey akzeptieren.

```

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    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 Standard-Einstellung: `SSHD_PORT='22'`

Mit `SSHD_PORT` kann abweichend vom Standard ein Port angegeben werden, auf dem der `ssh`-Server laufen soll.

Möchte man den ssh-Zugang auch von außen erlauben, ist INPUT_ACCEPT_PORT_x (Seite ??) anzupassen.

Die Befehle, um von einem Unix-/Linux-Rechner über das SSH-Protokoll auf fli4l zuzugreifen, lauten:

- ssh - Secure Shell
- scp - Secure Copy

Entsprechende Programme für Windows sind ebenso verfügbar, s. auch:
<http://www.chiark.greenend.org.uk/~sgtatham/putty/>
<http://winscp.net/eng/docs/lang:de>
<http://www.tectia.com/de/de.iw3>

SSHD_PUBLIC_KEY_N Standard-Einstellung: SSHD_PUBLIC_KEY_N='0'

SSHD_PUBLIC_KEY_N beschreibt die Anzahl der öffentlichen Schlüssel, die auf den fli4l-Router kopiert werden sollen.

SSH gestattet die Authentifizierung mit Hilfe von asymmetrischen Verschlüsselungsverfahren. Dabei erfolgt die Authentifizierung anstatt über Nutzernamen und Passwort über Nutzernamen und einem Public-/Privatekey. Damit kann man sich die Eingabe eines Passwortes sparen. Das Schlüsselpaar erzeugt man mit Hilfe von ssh-keygen (oder puttygen, wenn putty unter Windows als ssh-Client eingesetzt wird). Optional kann beim Schlüsselerzeugen eine Passphrase (also ein Passwort, das man braucht, wenn man den Schlüssel benutzen will) vergeben werden, welche die Sicherheit noch zusätzlich erhöht. Benutzt man Passphrases sollte man über den Einsatz eines Schlüsselagenten nachdenken (siehe ssh-agent oder pageant).

Wichtig: *Der private Teil des Schlüsselpaares, ist so sorgfältig zu behandeln wie ein Passwort, da er die gleiche Funktion erfüllt. Der private Teil des Schlüssel wird bei dem ssh-Client hinterlegt. Der öffentliche Teil des Schlüssel wird für den fli4l-Router gebraucht und mit SSHD_PUBLIC_KEY_x oder SSHD_PUBLIC_KEYFILE_x zur Verfügung gestellt.*

Für weitere Informationen siehe die manual Pages von ssh und Konsorten bzw. die Dokumentation zu putty (<http://www.chiark.greenend.org.uk/~sgtatham/putty/>).

SSHD_PUBLIC_KEY_x Für jeden Nutzer, der über ssh Zugang zum fli4l-Router erlangen möchte, kann hier der öffentliche Teil des Schlüssel angegeben werden. Am einfachsten geht das per Cut-and-Paste aus einem Terminalfenster heraus. Das könnte z.B. in etwa wie folgt aussehen:

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

Wichtig: *Der Schlüssel enthält keine Zeilenumbrüche. Bei Cut-and-Paste aus puttygen heraus werden aber eventuell selbige eingefügt. Diese Zeilenumbrüche müssen wieder entfernt werden.*

Zur Zeit werden Schlüssel für die folgenden Verschlüsselungsverfahren unterstützt:

- DSA

- RSA
- ECDSA

SSHD_PUBLIC_KEYFILE_N Standard-Einstellung: `SSHD_PUBLIC_KEYFILE_N='0'`

Anstatt den Inhalt des öffentlichen Teil des Schlüssel in die `sshd.txt` Datei zu kopieren, können Sie den öffentlichen Teil des Schlüssel auch direkt in das `opt`-Archiv kopieren lassen. Das funktioniert genauso wie bei `SSH_CREATEHOSTKEYS` beschrieben wurde. Kopieren Sie Ihren öffentlichen Teil des Schlüssel in das Verzeichnis `<config>/etc/ssh`.

SSHD_PUBLIC_KEYFILE_x Der Dateiname des öffentlichen Teil des Schlüssels im `<config>/etc/ssh` Verzeichnis.

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

Zur Zeit werden Schlüssel für die folgenden Verschlüsselungsverfahren unterstützt:

- DSA
- RSA
- ECDSA

SSH_CLIENT_PRIVATE_KEYFILE_N Standard-Einstellung: `SSH_CLIENT_PRIVATE_KEYFILE_N='0'`

Wenn Sie mit dem `ssh` oder `plink` Client private Schlüssel zur Anmeldung an einen `ssh` Server benutzen wollen können Sie diese in das Verzeichnis `<config>/etc/ssh` kopieren. Das funktioniert genauso wie bei `SSH_CREATEHOSTKEYS` beschrieben wurde. Kopieren Sie Ihren privaten Teil des Schlüssel in das Verzeichnis `<config>/etc/ssh`. Private Schlüssel im OpenSSH-Format werden automatisch bei jedem Startvorgang von `fli4l` ins das `dropbear`-Format konvertiert.

SSH_CLIENT_PRIVATE_KEYFILE_x Der Dateiname des privaten Teil des Schlüssels im `<config>/etc/ssh` Verzeichnis.

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

Zur Zeit werden Schlüssel für die folgenden Verschlüsselungsverfahren unterstützt:

- DSA
- RSA
- ECDSA

1.1.2 Installation des dbclients

OPT_SSH_CLIENT Standard-Einstellung: `OPT_SSH_CLIENT='no'`

Wenn man einen reinen `ssh2/scp` Client benutzen möchte, kann man den `dbclient` von `dropbear` durch Setzen von `OPT_SSH_CLIENT='yes'` aktivieren. Dieser Client hat den Vorteil, dass er sich viel Programmcode mit dem `dropbear ssh` Server teilt. Dadurch wird sehr viel Platz im `OPT`-Archiv gespart. Der `dbclient` ist weitgehend kompatibel mit dem `ssh/scp` Client, die Befehlsparameter sind ähnlich. Es wird auch ein symbolischer Link

auf `/usr/bin/ssh` bzw. `/usr/bin/scp` angelegt, damit ein gewohntes `ssh <host>` bzw. `scp <source> <target>` funktioniert.

Wenn man die dbclient bekannten Hostkeys permanent speichern will muss man die Datei `known_hosts` auf dem Verzeichnis `/.ssh` auf dem fli4l-Router in das `config/etc/ssh` kopieren. Das geschieht ähnlich wie mit einem erzeugten Hostkey. In dem folgenden Beispiel ist das ausgepackte fli4l Verzeichnis (in der die fli4l-Bootmedium erzeugt wird) in `/home/babel/fli4l-4.0.0-stable-x86_64-r60708` zu finden. Die Konfigurationsdateien liegen alle im Verzeichnis `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 des plink Clients

OPT_PLINK_CLIENT Standard-Einstellung: `OPT_PLINK_CLIENT='no'`

Installiert auf dem fli4l-Router einen `ssh1/ssh2/telnet` Client. Das `plink` Programm ist die Unixversion des bekannten PuTTY Programms für Windows. Ein Aufruf von `plink` auf dem fli4l-Router gibt eine Hilfeseite für die Benutzung von `plink` aus.

Wenn man die `plink` bekannten Hostkeys permanent speichern will muss man die Datei `sshhostkeys` auf dem Verzeichnis `/.putty` auf dem fli4l-Router in das `config/etc/plink` kopieren. Das geschieht ähnlich wie mit einem erzeugten Hostkey. In dem folgenden Beispiel ist das ausgepackte fli4l Verzeichnis (in der das fli4l-Bootmedium erzeugt wird) in `/home/babel/fli4l-4.0.0-stable-x86_64-r60708` zu finden. Die Konfigurationsdateien liegen alle im Verzeichnis `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 des sftp-server

OPT_SFTPSERVER Standard-Einstellung: `OPT_SFTPSERVER='no'`

Installiert auf dem fli4l-Router einen `sftp-server`.

1.1.5 Literatur

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

Erste Version der Dokumentation von Claas Hilbrecht <babel@fli4l.de>, im April 2004

Abbildungsverzeichnis

1.1 Verzeichnisstruktur fli4l 4

Tabellenverzeichnis

Index

OPT_PLINK_CLIENT, [8](#)
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, [7](#)
SSHD_PUBLIC_KEYFILE_x, [7](#)