

# **Paket TOOLS - Zusätzliche Werkzeuge zum Debugging Version 3.10.4**

Frank Meyer                      Das fli4l-Team  
E-Mail: [frank@fli4l.de](mailto:frank@fli4l.de)      E-Mail: [team@fli4l.de](mailto:team@fli4l.de)

25. Oktober 2015

# Inhaltsverzeichnis

<b>1</b>	<b>Dokumentation des Paketes TOOLS</b>	<b>3</b>
1.1	TOOLS - Zusätzliche Werkzeuge zum Debugging . . . . .	3
1.1.1	Netzwerk-Tools . . . . .	3
1.1.2	Die Hardware-Erkennung . . . . .	6
1.1.3	Dateien-Tools . . . . .	7
1.1.4	Entwickler-Tools . . . . .	8
	<b>Abbildungsverzeichnis</b>	<b>9</b>
	<b>Tabellenverzeichnis</b>	<b>10</b>
	<b>Index</b>	<b>11</b>

# 1 Dokumentation des Paketes TOOLS

## 1.1 TOOLS - Zusätzliche Werkzeuge zum Debugging

Das Paket TOOLS liefert eine Reihe von Unix Programmen, die zumeist für Administrations- und Debugzwecke gedacht sind. Andere Programme wie wget werden z.B. dafür verwendet, die erste (Werbungs-)Seite einiger Provider abzufangen. Mit dem Wert 'yes' wird das jeweilige Programm mit auf den fli4l-Router kopiert. Die Standardeinstellung ist 'no'. Die Programme werden nur kurz vorgestellt, wie sie zu bedienen sind, entnehme man bitte den man Pages einer beliebigen Unix/ Linux Distribution oder online unter: <http://www.linuxmanpages.com>

### 1.1.1 Netzwerk-Tools

#### **OPT\_DIG** Schweizer Taschenmesser fürs DNS

Der Befehl dig erlaubt es, vielfältige DNS-Abfragen durchzuführen.

#### **OPT\_FTP** FTP-Client

Mit dem Programm ftp können eine FTP-Verbindung zu einem FTP-Server aufgebaut und Dateien zwischen Router und FTP-Server übertragen werden.

**FTP\_PF\_ENABLE\_ACTIVE** Die Einstellung `FTP_PF_ENABLE_ACTIVE='yes'` fügt dem Paketfilter eine Regel hinzu, die auf dem Router initiiertes aktives FTP ermöglicht. Bei `FTP_PF_ENABLE_ACTIVE='no'` muss eine solche Regel (falls gewünscht) manuell zum `PF_OUTPUT_%-Array` hinzugefügt werden, ein Beispiel ist in diesem [Abschnitt](#) (Seite ??) zu finden.

Passives FTP ist immer möglich, hierfür ist weder diese Variable noch eine explizite Paketfilter-Regel notwendig.

#### **OPT\_IFTOP** Netzwerküberwachung

Mit dem Programm iftop wird eine Auflistung alle aktiven Netzwerkverbindung und deren Durchsatz direkt auf dem fli4l angezeigt.

Das Programm iftop wird nach dem Anmelden auf dem fli4l-Router durch Eingabe von iftop gestartet.

#### **OPT\_IMONC** Textorientiertes Steuerprogramm für imond

Dieses Programm liefert ein textorientiertes Frontend für den Router, um den imond zu steuern.

#### **OPT\_IPERF** Performancemessung im Netzwerk

Mit dem Programm iperf kann eine Performancemessung des Netzwerks durchgeführt werden. Dazu wird das Programm auf den beiden beteiligten Testsystemen gestartet. Auf dem Server wird das Programm mit

```
fli4l-server 3.10.4~# iperf -s
-----
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
-----
```

gestartet. Der Server wartet dann auf eine Verbindung vom Client. Der Client wird durch

```
fli4l-client 3.10.4~# iperf -c 1.2.3.4
-----
Client connecting to 1.2.3.4, TCP port 5001
TCP window size: 16.0 KByte (default)
-----
[  3] local 1.2.3.5 port 50311 connected with 1.2.3.4 port 5001
[ ID] Interval      Transfer    Bandwidth
[  3]  0.0-10.0 sec   985 MBytes  826 Mbits/sec
```

gestartet. Sofort startet die Performancemessung und zeigt die ersten Ergebnisse an. iperf kennt noch eine Reihe weiterer Optionen, für Details schauen Sie sich bitte die Informationen auf der Homepage <http://iperf.sourceforge.net/> an.

**OPT\_NETCAT** Übertragen von Daten an TCP basierte Server

**OPT\_NGREP** Ein grep der direkt auf einem Netzwerkdevice arbeiten kann.

**OPT\_NTTCP** Netzwerktest

Mit dem Programm NTTCP kann man die Netzwerkgeschwindigkeit testen. Dazu wird auf einer Seite ein Server gestartet und auf einer anderen Seite ein entsprechende Client.

Den Server startet man durch Eingabe von `nttcp -i -v`. Der Server wartet dann auf eine Testanforderung des Clients. Um jetzt z.B die Geschwindigkeit zu testen gibt man auf dem Client `nttcp -t <IP Adresse des Servers>` ein.

So sieht ein gestarteter nttcp Server aus:

```
fli4l-server 3.10.4~# nttcp -i -v
nttcp-1: nttcp, version 1.47
nttcp-1: running in inetd mode on port 5037 - ignoring options beside -v and -p
```

So sieht ein Test mit einem nttcp Client aus:

```
fli4l-client 3.10.4~# nttcp -t 192.168.77.77
1~~8388608~~~~4.77~~~~0.06~~~~~14.0713~~~~1118.4811~~~~2048~~~~429.42~~~34133.3
1~~8388608~~~~4.81~~~~0.28~~~~~13.9417~~~~239.6745~~~~6971~~~1448.21~~~24896.4
```

Die Hilfeseite von nttcp zeigt alle weiteren Parameter:

```
Usage: nttcp [local options] host [remote options]
      local/remote options are:
      -t          transmit data (default for local side)
```

```

-r      receive data
-l#     length of bufs written to network (default 4k)
-m      use IP/multicasting for transmit (enforces -t -u)
-n#     number of source bufs written to network (default 2048)
-u      use UDP instead of TCP
-g#us   gap in micro seconds between UDP packets (default 0s)
-d      set SO_DEBUG in sockopt
-D      don't buffer TCP writes (sets TCP_NODELAY socket option)
-w#     set the send buffer space to #kilobytes, which is
        dependent on the system - default is 16k
-T      print title line (default no)
-f      give own format of what and how to print
-c      compares each received buffer with expected value
-s      force stream pattern for UDP transmission
-S      give another initialisation for pattern generator
-p#     specify another service port
-i      behave as if started via inetd
-R#     calculate the getpid()/s rate from # getpid() calls
-v      more verbose output
-V      print version number and exit
-?      print this help
-N      remote number (internal use only)
default format is: %9b%8.2rt%8.2ct%12.4rbr%12.4cbr%8c%10.2rcr%10.1ccr

```

**OPT\_RTMON** Installiert ein tool, dass Änderungen der Routingtabelle überwacht. Primäre Verwendung: Debugging

**OPT\_SOCAT** Das Programm “socat” ist quasi eine verbesserte und mit mehr Funktionen “vollgestopfte” Version des [“netcat”-Programms](#) (Seite 4). Mit “socat” können nicht nur diverse Netzwerk-Verbindungen aufgebaut bzw. entgegengenommen werden, sondern auch Daten an UNIX-Sockets, Geräte, FIFOs etc. gesandt bzw. von dort ausgelesen werden. Insbesondere können Quellen und Ziele *verschiedener* Typen miteinander verbunden werden: Ein Beispiel wäre etwa ein via TCP auf einem Port horchender Netzwerk-Server, der empfangene Daten in einen lokalen FIFO schreibt bzw. Daten aus dem FIFO ausliest und diese dann übers Netzwerk an den Client schickt. Siehe <http://www.dest-unreach.org/socat/doc/socat.html> für mehr Informationen sowie Anwendungsbeispiele.

**OPT\_TCPDUMP** debug

Mit dem Programm tcpdump kann Netzwerkverkehr beobachtet, ausgewertet mitgeschnitten werden. Mehr dazu unter z.B. Google mit den Suchworten “tcpdump man“  
tcpdump <parameter>

**OPT\_WGET** http/ftp Client

Mit dem Programm wget können Daten von einem Webserver im Batch abgerufen werden. Praktisch ist aber (und deswegen ist wget im fli4l-Paket dabei), dass man damit

Umlenkungen des Providers auf den eigenen Webserver nach einem Verbindungsaufbau auf einfache Weise abfangen kann, z.B. für Freenet. Wie das geht, hat Steffen Peiser in einem Mini-HowTo erklärt.

Siehe: <http://www.fli4l.de/hilfe/howtos/einsteiger/wget-und-freenet/>

### 1.1.2 Die Hardware-Erkennung

Oftmals weiß man nicht genau, welche Hardware im eigenen Rechner steckt bzw. welche Treiber man nun genau für seine Netzwerkkarte oder seinen USB-Chipsatz verwenden soll. Die Hardware kann an der Stelle helfen. Sie liefert eine Liste von Geräten im Rechner und wenn möglich den dazugehörigen Treiber. Man kann dabei auswählen, ob die Erkennung gleich beim Booten erfolgen soll (was sich vor einer Erstinstallation empfiehlt) oder später bei laufendem Rechner bequem über das Web-Interfaces getriggert werden soll. Die Ausgabe könnte dabei z.B. wie folgt aussehen:

```
fli4l 3.10.4 # cat /bootmsg.txt
#
# PCI Devices and drivers
#
Host bridge: Advanced Micro Devices [AMD] CS5536 [Geode companion] Host Bridge (rev 33)
Driver: 'unknown'
Entertainment encryption device: Advanced Micro Devices [AMD] Geode LX AES Security Block
Driver: 'geode_rng'
Ethernet controller: VIA Technologies, Inc. VT6105M [Rhine-III] (rev 96)
Driver: 'via_rhine'
Ethernet controller: VIA Technologies, Inc. VT6105M [Rhine-III] (rev 96)
Driver: 'via_rhine'
Ethernet controller: VIA Technologies, Inc. VT6105M [Rhine-III] (rev 96)
Driver: 'via_rhine'
Ethernet controller: Atheros Communications, Inc. AR5413 802.11abg NIC (rev 01)
Driver: 'unknown'
ISA bridge: Advanced Micro Devices [AMD] CS5536 [Geode companion] ISA (rev 03)
Driver: 'unknown'
IDE interface: Advanced Micro Devices [AMD] CS5536 [Geode companion] IDE (rev 01)
Driver: 'amd74xx'
USB Controller: Advanced Micro Devices [AMD] CS5536 [Geode companion] OHC (rev 02)
Driver: 'ohci_hcd'
USB Controller: Advanced Micro Devices [AMD] CS5536 [Geode companion] EHC (rev 02)
Driver: 'ehci_hcd'
```

Hier stecken also im wesentlichen 3 Netzwerkkarten drin, die vom 'via\_rhine'-Treiber verwaltet werden und eine Atheros-Wlan-Karte, die vom madwifi-Treiber verwaltet wird (der Name wird noch nicht korrekt aufgelöst).

**OPT\_HW\_DETECT** Diese Variable sorgt dafür, dass die für die Hardware-Erkennung Dateien auf dem Router landen. Man kann sich die Ergebnisse dann entweder nach dem Booten auf der Konsole ansehen, wenn man HW\_DETECT\_AT\_BOOTTIME auf 'yes' gesetzt hat oder im Web-Interface ansehen, wenn man [OPT\\_HTTPD](#) (Seite ??) auf 'yes' gesetzt hat. Im Web-Interface kann man sich natürlich auch den Inhalt von '/bootmsg.txt' ansehen, wenn man schon einen funktionierenden Netzzugang hat.

**HW\_DETECT\_AT\_BOOTTIME** Startet die Hardware-Erkennung beim Booten. Die Erkennung läuft im Hintergrund (sie dauert ein wenig) und schreibt dann ihre Ergebnisse auf die Konsole und nach `/bootmsg.txt`.

**OPT\_LSPCI** Auflisten aller PCI-Geräte

**OPT\_I2CTOOLS** Tools für I<sup>2</sup>C Zugriffe.

**OPT\_IWLEEPROM** Tool zum Zugriff auf das EEPROM von Intel und Atheros WLAN Karten.

Wird benötigt um z.B. bei ath9k Karten die Reg-Domain passend zu setzen (siehe <http://blog.asiantuntijakaveri.fi/2014/08/one-of-my-atheros-ar9280-minipcie-cards.html>).

**OPT\_ATH\_INFO** Tool zur Hardware diagnose von WLAN Karten mit Atheros Chipsatz.

Mithilfe dieses Tools können z.B. bei ath5k WLAN Karten detaillierte Informationen über die verwendete Hardware gewonnen werden. Dazu gehören z.B. der verwendete Chipsatz oder Angaben zur Kalibrierung.

### 1.1.3 Dateien-Tools

**OPT\_E3** Ein Editor für fli4l

Dies ist ein sehr kleiner, in Assembler geschriebener Editor. Er stellt verschiedene Editor-Modi zur Verfügung, die andere („große“) Editoren nachstellen. Um einen bestimmten Modus zu wählen, reicht es e3 mit dem richtigen Befehl zu starten. Eine Kurzübersicht der Tastenbelegung bekommt man, wenn man e3 ohne Parameter startet oder Alt+H drückt (außer im VI-Modus, dort muß man im CMD-Modus „:h“ eintippen). Zu beachten ist auch, dass das Caret-Zeichen (^) für die Ctrl-/Strg-Taste steht.

Befehl	Modus
e3 / e3ws	WordStar, JOE
e3vi	VI, VIM
e3em	Emacs
e3pi	Pico
e3ne	NEdit

**OPT\_MTOOLS** Die mtools stellen eine Reihe von DOS-ähnlichen Befehlen zum vereinfachten Umgang (Kopieren, Formatieren, etc.) mit DOS-Datenträgern bereit.

Die genaue Syntax der Befehle kann in der Dokumentation von mtools nachgeschlagen werden:

<http://www.gnu.org/software/mtools/manual/mtools.html>

**OPT\_SHRED** Installiert das Programm *shred* auf dem Router, ein Programm zum gründlichen Löschen von Blockgeräten.

**OPT\_YTREE** Datei-Manager

Installiert Datei-Manager Ytree auf dem Router.

#### 1.1.4 Entwickler-Tools

**OPT\_OPENSSL** Mit dem Programm openssl können z.B. Test der Cryptobeschleuniger durchgeführt werden.

```
openssl speed -evp des -elapsed openssl speed -evp des3 -elapsed openssl speed -evp aes128 -elapsed
```

**OPT\_STRACE** debug

Mit dem Programm strace können die Funktionsaufrufe, der Ablauf eines Programmes beobachtet werden

```
strace <programm>
```

**OPT\_REAVER** Brute force Angriff aus Wifi WPS PINs

Testet alle möglichen WPS PINs aus um das WPA Passwort zu ermitteln. Details für die Verwendung auf der Kommandozeile bitte unter <http://code.google.com/p/reaver-wps/> nachlesen.

**OPT\_VALGRIND** Installiert Valgrind auf dem Router.



# Abbildungsverzeichnis

# **Tabellenverzeichnis**

# Index

FTP\_PF\_ENABLE\_ACTIVE, [3](#)

HW\_DETECT\_AT\_BOOTTIME, [6](#)

OPT\_ATH\_INFO, [7](#)

OPT\_DIG, [3](#)

OPT\_E3, [7](#)

OPT\_FTP, [3](#)

OPT\_HW\_DETECT, [6](#)

OPT\_I2CTOOLS, [7](#)

OPT\_IFTOP, [3](#)

OPT\_IMONC, [3](#)

OPT\_IPERF, [3](#)

OPT\_IWLEEPROM, [7](#)

OPT\_LSPCI, [7](#)

OPT\_MTOOLS, [7](#)

OPT\_NETCAT, [4](#)

OPT\_NGREP, [4](#)

OPT\_NTTCP, [4](#)

OPT\_OPENSSL, [8](#)

OPT\_REAVER, [8](#)

OPT\_RTMON, [5](#)

OPT\_SHRED, [7](#)

OPT\_SOCAT, [5](#)

OPT\_STRACE, [8](#)

OPT\_TCPDUMP, [5](#)

OPT\_VALGRIND, [8](#)

OPT\_WGET, [5](#)

OPT\_YTREE, [7](#)