Paket OPENVPN
Version 3.10.19

Claas Hilbrecht
E-Mail: babel (+at+) fli4l dot de

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

2. Februar 2020

Inhaltsverzeichnis

1. Dokumentation des Paketes OPENVPN

1.1. OpenVPN - VPN-Support

Ab Version 2.1.5 ist das OpenVPN Paket fester Bestandteil von fli4l.

Wichtig: *Für die Nutzung von OpenVPN über das Internet, ist in jedem Fall eine Flatrate oder eine volumenbasierte Abrechnung notwendig! Wenn der fli4l-Router permanent eingeschaltet bleibt, wird die Verbindung nicht getrennt, da permanent Daten (wenn auch nur ein paar Bytes alle paar Sekunden) übertragen werden. Mit der Nutzung eines der VPN Pakete und der Verwendung eines VPN Tunnel über das Internet, legt der fli4l-Router nicht mehr auf und es entstehen hohe Kosten, wenn keine Flatrate oder ein volumenbasiertes Abrechnungsmodel benutzt wird! Gleiches gilt natürlich für eine ISDN Wählleitung.*

Neben OpenVPN gibt es in der opt-Datenbank http://www.fli4l.de/download/zusatzpakete/ noch das VPN Paket OPT_PoPToP.

Die Entscheidung für eine VPN Lösung hängt in erster Linie von der Sicherheit und der Funktion der eingesetzten Lösung ab. Aussagen zur Sicherheit der hier angebotenen VPN Lösungen gibt das fli4l Team bewusst nicht ab, verweist dafür auf folgende Webseiten und Berichte:

Linux-Magazin Ausgabe Januar 2004
http://diswww.mit.edu/bloom-picayune/crypto/14238
http://sites.inka.de/bigred/archive/cipe-l/2003-09/msg00263.html

Zur Funktionalität kann das fli4l Team aber eine klare Aussage treffen. In diesem Punkt ist OpenVPN der klare Gewinner gegenüber CIPE und poptop. OpenVPN unterstüzt neben einem Tunnel- und Bridgemodus auch Datenkompression und läuft im Gegensatz zu CIPE wesentlich stabiler auf dem fli4l-Router. Außerdem gibt es für OpenVPN auch eine Windows Version, die ab Windows 2000 eingesetzt werden kann. Einziger Nachteil von OpenVPN ist seine Größe im opt-Archiv gegenüber CIPE und die fehlende OpenVPN Unterstützung für fli4l Version 2.0.x.

1.1.1. OpenVPN - Einführendes Beispiel

Um den Einstieg in die Konfiguration zu erleichtern, vorab ein kleines Beispiel. Es sollen zwei Netze, die beide einen fli4l-Router einsetzen, über das Internet verbunden werden. Dazu wird von OpenVPN auf den zwei fli4l-Routern ein verschlüsselter Tunnel eingerichtet, durch den die Computer aus den enfernten Netzen miteinander kommunizieren können. Dabei spielen die im Bild 1.1 gezeigten Konfigurationsvariablen eine Rolle.

local net, remote net repräsentieren die beiden Netze, die über den Tunnel miteinander verbunden werden sollen. Die beiden zu verbindenden Netze müssen in unterschiedlichen TCP/IP Netzen sein und dürfen sich in ihren Netzmasken auch nicht überschneiden. Die Einstellungen von `IP_NET_x` (Seite **??**) in der jeweiligen base.txt Konfigurationsdateien dürfen also nicht gleich sein. Es ist also mit einem VPN Tunnel nicht möglich zwei Netze miteinander zu verbinden, die beide das IP Netz 192.168.6.0/24 benutzen.

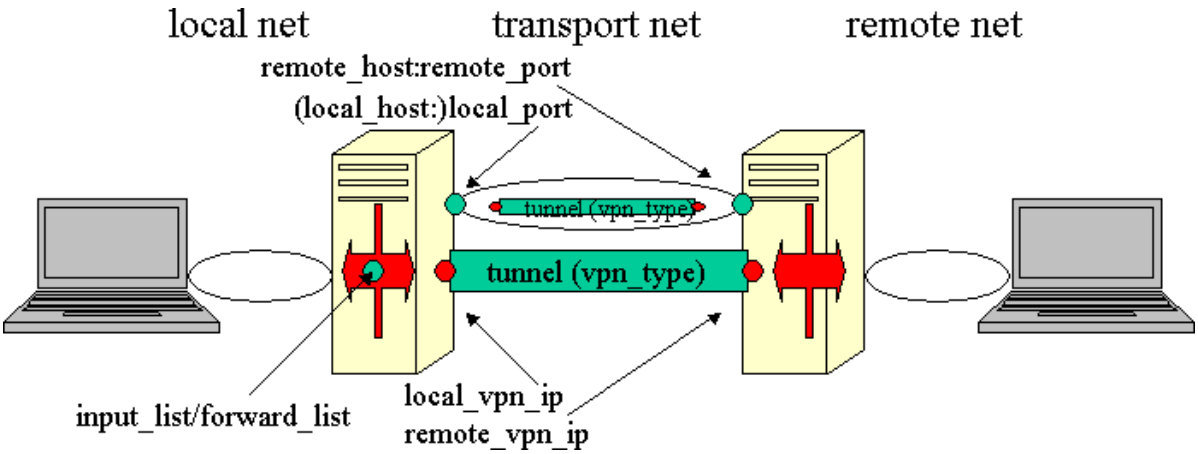

Abbildung 1.1.: VPN-Konfigurationsbeispiel — Tunnel zwischen zwei Routern

transport net das Transport Netzwerk besteht aus zwei Elementen:

- der Verbindung zwischen zwei OpenVPN-Daemons, beschrieben durch *remote_host:remote_port* und *(local_host:)local_port*. Das entspricht den OpenVPN Einstellungen `OPENVPN_x_REMOTE_HOST`, `OPENVPN_x_REMOTE_PORT`, `OPENVPN_x_LOCAL_HOST` und `OPENVPN_x_LOCAL_PORT`.
- und dem Tunnel, über den die Verbindung zwischen den OpenVPN-Daemons etabliert wird, beschrieben durch *local_vpn_ip/remote_vpn_ip*. Dies entspricht dann wieder `OPENVPN_x_LOCAL_VPN_IP` und `OPENVPN_x_REMOTE_VPN_IP`. Die beiden VPN IP-Adressen dürfen sich dabei in keinem anderen, den beiden Routern bekannten Netzen liegen.

input_list, **forward_list** Pakete, die über den Tunnel gehen sollen, müssen zuerst durch den Paketfilter. Dieser erlaubt standardmäßig nur ICMP-Nachrichten (z. B. ping), die man zum Testen des Tunnels verwenden kann. Alles andere muß erst explizit erlaubt werden, im einfachsten Falle durch

```
OPENVPN_x_PF_INPUT_POLICY='ACCEPT'
OPENVPN_x_PF_FORWARD_POLICY='ACCEPT'
```

Bitte denken Sie daran, dass das komplette „Freigeben“ einer VPN Verbindung sicherheitstechnisch sehr bedenklich ist. Benutzen Sie lieber die tmpl: Syntax des Paketfilters, um nur gezielt die Dienste freizugeben, die Sie auch benötigen.

Mehr Einstellungen sind für einen einfachen VPN Tunnel nicht notwendig. Alle weiteren Einstellmöglichkeiten behandeln erweiterte Funktionen oder sind für spezielle Anwendungsfälle gedacht. Sie sollten mit diesen erweiterten Einstellungen erst dann arbeiten, weil der VPN Tunnel mit den minimalen Einstellungen erfolgreich aufgebaut werden kann.

1.1.2. OpenVPN - Konfiguration

Da OpenVPN ziemlich komplex ist, beginnen wir mit der Erklärung der zwingend notwendigen Angaben für jede VPN Verbindung. Erst wenn der fli4l-Router mit diesen Einstellungen eine Verbindung aufgebaut hat, sollten Sie sich daran wagen die erweiterten Konfigurationsmöglichkeiten von OpenVPN zu nutzen.

OPT_OPENVPN Default: `OPT_OPENVPN='no'`

Mit `'yes'` wird das OpenVPN Paket aktiviert. Die Einstellung `'no'` deaktiviert das OpenVPN Paket komplett.

OPENVPN_N Default: `OPENVPN_N='0'`

Wieviele OpenVPN Konfigurationen sind in der Konfigurationsdatei aktiv?

OPENVPN_x_REMOTE_HOST Default: `OPENVPN_x_REMOTE_HOST="`

Die IP-Adresse oder eine DNS-Adresse der OpenVPN Gegenstelle. Bei einem Roadwarrior (Seite 27) muss diese Zeile komplett fehlen. Wird diese Einstellung weggelassen, wartet OpenVPN auf einen Verbindungsaufbau und versucht nicht selbstständig die Verbindung aufzubauen.

OPENVPN_x_REMOTE_HOST_N Default: `OPENVPN_x_REMOTE_HOST_N='0'`

Bei der Benutzung von dynamischen DNS Diensten passiert es leider ab und an, dass ein Dienst nicht 100% zuverlässig funktioniert. Daher macht es in diesen Fällen Sinn, einfach zwei oder mehr DynDNS Dienste zu benutzen und seine aktuelle IP-Adresse bei allen Diensten gleichzeitig zu registrieren. Damit OpenVPN in diesem Fall auch alle DynDNS Namen durchgehen kann, muss hier noch die Liste der *zusätzlichen* DNS Namen eingegeben werden. Zusammen mit `OPENVPN_x_REMOTE_HOST` ergibt sich dann die Liste der DynDNS Adressen, die OpenVPN in zufälliger Reihenfolge zu kontaktieren versucht. Der Eintrag `OPENVPN_x_REMOTE_HOST` muss also weiterhin vorhanden sein!

OPENVPN_x_REMOTE_HOST_x Default: `OPENVPN_x_REMOTE_HOST_x="`

Es gilt die gleiche Beschreibung wie unter `OPENVPN_x_REMOTE_HOST` (Seite 5).

OPENVPN_x_REMOTE_PORT Default: `OPENVPN_x_REMOTE_PORT="`

Jede OpenVPN Verbindung braucht eine auf dem fli4l-Router bisher nicht benutzte Portadresse. Es empfiehlt sich, einen Port oberhalb von 10000 zu nehmen, da dort normalerweise keine häufig benutzen Ports liegen. Wenn Sie eine Verbindung für eine Gegenstelle bereitstellen wollen, die eine wechselnde IP-Adresse hat und über keine DynDNS Adresse verfügt, lassen Sie diesen Eintrag genau wie `OPENVPN_x_REMOTE_HOST` komplett weg.

OPENVPN_x_LOCAL_HOST Default: `OPENVPN_x_LOCAL_HOST="`

Gibt an, an welche IP-Adresse OpenVPN gebunden werden soll. Bei Verbindungen über das Internet sollte dieser Eintrag leer bleiben oder komplett weggelassen werden. Wird hier eine IP-Adresse angegeben, horcht OpenVPN nur auf dieser IP-Adresse auf eingehende Verbindungsanfragen. Wenn Sie eine WLAN Verbindung absichern wollen, sollten Sie hier die IP-Adresse der WLAN Karte vom fli4l-Router eintragen.

OPENVPN_x_LOCAL_PORT Default: `OPENVPN_x_LOCAL_PORT="`

Gibt die Portnummer an, auf der der lokale OpenVPN Daemon horcht. Für jede OpenVPN Einstellung benötigen Sie einen dafür reservierten Port, d.h. dieser Port kann nur von dieser einer OpenVPN Verbindung benutzt werden und darf auch von keiner anderen Software auf dem fli4l-Router benutzt werden. Die Einstellungen `OPENVPN_x_REMOTE_PORT` und `OPENVPN_x_LOCAL_PORT` jeder OpenVPN Verbindung müssen zusammenpassen! Wenn Sie auf einer Seite des Tunnel `OPENVPN_x_REMOTE_PORT='10111'` setzen *muss* die andere zwingend auf `OPENVPN_x_LOCAL_PORT='10111'` gesetzt werden.

Nochmal: Es ist sehr wichtig, diese Einstellungen auf die jeweilige OpenVPN Gegenstelle anzupassen, sonst ist eine Verbindung zwischen den OpenVPN Partnern nicht möglich.

Damit OpenVPN auf eingehende Verbindungen horchen kann, öffnet OpenVPN selbstständig die Ports im Paketfilter, die unter `OPENVPN_x_LOCAL_PORT` angegeben werden. Wenn dies nicht gewünscht wird, können Sie dies unter `OPENVPN_DEFAULT_OPEN_OVPNPORT` (Seite 12) anpassen. Es ist *nicht* notwendig den Eintrag `OPENVPN_DEFAULT_OPEN_OVPNPORT='yes'` zu setzen, da das die Standardeinstellung ist!

Es ist nicht möglich OpenVPN auf Ports kleiner als 1025 horchen zu lassen. Wenn Sie z.B. einen OpenVPN als tcp-server auf Port 443 (der https Port) konfigurieren wollen, müssen Sie den Port 443 per Paketfilter an einen Port über 1024 weiterleiten. Wenn Sie z.B. den OpenVPN auf Port 5555 horchen lassen und den Port 443 weiterleiten wollen, muss folgendes in die `PF_PREROUTING` eingetragen werden.

```
PF_PREROUTING_5='tmpl:https dynamic REDIRECT:5555'
```

OPENVPN_x_SECRET Default: `OPENVPN_x_SECRET="`

OpenVPN benötigt zum Verschlüsseln der OpenVPN Verbindung ein sogenanntes Keyfile. Dieses Keyfile kann unter Windows oder Linux direkt mit OpenVPN erzeugt werden. Für Anfänger bietet es sich an, die Windows Software von OpenVPN zu installieren oder die OpenVPN WebGUI zu benutzen. Wenn Sie OpenVPN unter Windows nicht einsetzen wollen, sondern nur die für OpenVPN benötigen Keyfiles erstellen wollen, reicht es, die Punkte *OpenVPN User-Space Components*, *OpenSSL DDLs*, *OpenSSL Utilities*, *Add OpenVPN to PATH* und *Add Shortcuts to OpenVPN* zu installieren. Mit dem Menüpunkt *Generate a static OpenVPN key*, den Sie im Startmenü unter OpenVPN finden, können dann die benötigten Keydateien erzeugt werden. Nach dem Aufruf des Menüpunktes kommt die Meldung „Randomly generated 2048 bit key written to `C:/Programme/OpenVPN/config/key.txt`“. Die erstellte `key.txt` Datei ist das benötigte Keyfile. Kopieren Sie diese Datei einfach in das Verzeichnis `<config>/etc/openvpn` und bennenen Sie die `key.txt` entsprechend um, so dass der Dateiname aussagekräftig wird. Sie können ein Keyfile auch automatisch vom fli4l-Router erstellen lassen, wenn Sie `OPENVPN_CREATE_SECRET` auf `'yes'` stellen und den fli4l-Router neu starten. Wenn Sie also das erste Mal OpenVPN konfigurieren, tragen Sie alle Daten in die Konfigdatei ein und setzen entweder `OPENVPN_DEFAULT_CREATE_SECRET` (Seite 12) auf `'yes'`, wenn Sie gleich für alle OpenVPN Verbindungen neue Keyfiles erzeugen wollen oder nur für die entsprechende OpenVPN Verbindung `OPENVPN_x_CREATE_SECRET` auf `'yes'`. Nach dem Start des fli4l-Routers werden dann die oder das Keyfile(s) automatisch erzeugt und in `/etc/openvpn` mit dem hier angegebenen Namen abgelegt. Das oder die Keyfile(s) kann

dann per scp kopiert oder mit einer Diskette übertragen werden. Sie müssen nach dem Erstellen der Schlüsseldateien die Einstellung wieder auf 'no' setzen, das fli4l Bootmedium neu erzeugen und die neu erzeugte Konfiguration starten. Bleibt die Einstellung auf 'yes' werden bei jedem Start des fli4l-Routers neue Schlüsseldateien erzeugt aber kein OpenVPN Daemon gestartet. Es also kann kein Tunnel aufgebaut werden. Sie können `OPENVPN_x_CREATE_SECRET` auch auf `'webgui'` setzen, wenn Sie die WebGUI verwenden möchten um Keyfile(s) zu generieren. Dazu müssen Sie in der WebGUI in die Detailansicht der Verbindung(en) gehen und den Punkt Keymanagement auswählen. Genaueres dazu im Abschnitt 1.1.6

Tipp: Mit dem Kommando

```
openvpn --genkey --secret <dateiname>
```

können Sie ein Keyfile auf dem fli4l-Router auch von Hand erstellen.

Die Keyfiles müssen in das Verzeichnis `<config>/etc/openvpn` kopiert werden, wie in folgendem Bild zu sehen ist. Der Dateiname des Keyfiles ohne den Pfad muss anschließend in `OPENVPN_x_SECRET` hinterlegt werden. Dann werden die Keyfiles beim Erstellen des opt-Archives mit eingepackt.

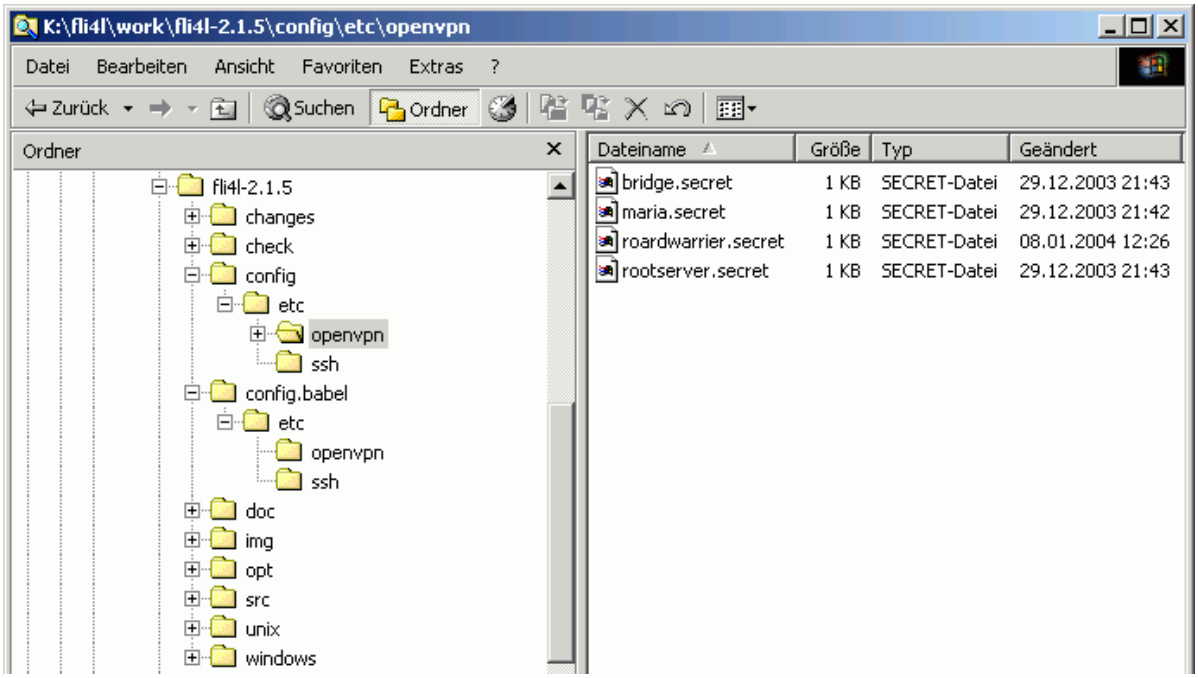

Abbildung 1.2.: fli4l config Directory mit OpenVPN *.secret Dateien

OPENVPN_x_TYPE Default: `OPENVPN_x_TYPE="`

Eine OpenVPN Verbindung kann entweder als Tunnel oder als Bridge benutzt werden. Über einen OpenVPN Tunnel kann ausschliesslich IP-Traffic geroutet werden. Über eine Bridge werden Ethernetframes übertragen, also nicht nur IP-Traffic, sondern z.B. auch

IPX oder NetBEUI. Wenn OpenVPN als Transport für Ethernetframes benutzt werden soll, wird in jedem Fall noch das advanced_networking Paket benötigt. Bitte bedenken Sie, dass die Benutzung von Bridging über eine DSL Leitung sehr langsam werden kann!

1.1.3. OpenVPN - Bridgekonfiguration

Wenn Sie OpenVPN für eine Bridge benutzen wollen, sind folgende Einträge gültig. Bitte denken Sie daran, dass bei der Benutzung einer Bridge über das Internet der entstehende Broadcasttraffic unter Umständen schon eine relativ hohe Bandbreite benötigt.

Denken Sie daran, dass die folgenden Einstellungen nur gültig sind, wenn der OPENVPN_x_TYPE (Seite 7) für diese OpenVPN Verbindung auf 'bridge' eingestellt wurde! Ausserdem wird eine konfigurierte Bridge aus dem advanced_networking Paket benötigt, an die sich die VPN Verbindung hängen kann.

OPENVPN_x_BRIDGE Default: OPENVPN_x_BRIDGE="

Hier wird der Name der Bridge angegeben, an die sich diese OpenVPN Verbindung hängen soll. Wenn also in BRIDGE_DEV_x_NAME='cuj-br' steht und sich diese OpenVPN Verbindung an diese Bridge hängen soll, muss hier ebenfalls 'cuj-br' angegeben werden.

OPENVPN_x_BRIDGE_COST Default: OPENVPN_x_BRIDGE_COST="

Wenn Sie STP (siehe http://de.wikipedia.org/wiki/Spanning_Tree oder die Dokumentation im advanced_networking Paket) benutzen, können Sie hier die Kosten der Verbindung angeben.

OPENVPN_x_BRIDGE_PRIORITY Default: OPENVPN_x_BRIDGE_PRIORITY="

Wenn Sie STP (siehe http://de.wikipedia.org/wiki/Spanning_Tree oder die Dokumentation im advanced_networking Paket) benutzen, können Sie hier die Priorität der Verbindung angeben.

1.1.4. OpenVPN - Tunnelkonfiguration

OPENVPN_x_REMOTE_VPN_IP Default: OPENVPN_x_REMOTE_VPN_IP="

Diese Einstellung ist nur gültig, wenn der OPENVPN_x_TYPE (Seite 7) für diese OpenVPN Verbindung auf 'tunnel' einstellt wurde!

Die VPN IP-Adresse der OpenVPN Gegenstelle. Die VPN IP-Adressen werden nur zum Routen benötigt und können fast frei gewählt werden. Es gelten dabei folgende Einschränkungen für die Auswahl der VPN IP-Adressen:

- Die IP-Adresse darf an keiner Stelle im lokalen Netz benutzt werden. Sie darf also nicht im Subnetz des fli4l-Routers liegen.
- Die IP-Adresse darf für kein lokales Netzwerkdevice benutzt werden.
- Die IP-Adresse darf nicht zu einem Netzwerk gehören, das mit IP_ROUTE_x geroutet wird.
- Die IP-Adresse darf nicht zu einem Netzwerk gehören, das mit ISDN_CIRC_ROUTE_x geroutet wird.

- Die IP-Adresse darf nicht zu einem Netzwerk gehören, das mit CIPE_ROUTE_x geroutet wird.
- Die IP-Adresse darf nicht zu einem Netzwerk gehören, das mit OPENVPN_ROUTE_x geroutet wird.
- Die IP-Adresse darf nicht zu einem Netzwerk gehören, das auf irgendeine andere Weise zum fli4l Netzwerk gehört oder vom fli4l-Router geroutet wird.

Wie Sie sehen darf die VPN IP-Adresse nirgends sonst benutzt werden. Bevor Sie mit der Konfiguration von OpenVPN beginnen, sollten Sie sich ein Netz suchen, was von keinem Netzwerk benutzt wird, in das Sie eine VPN Verbindung aufbauen wollen. Das Netzwerk sollte auch unbedingt zu einem der privaten Netze gehören (siehe http://ftp.univie.ac.at/netinfo/rfc/rfc1597.txt).

OPENVPN_x_LOCAL_VPN_IP Default: OPENVPN_x_LOCAL_VPN_IP="

Diese Einstellung ist nur gültig, wenn der OPENVPN_x_TYPE (Seite 7) für diese OpenVPN Verbindung auf 'tunnel' einstellt wurde.

Die IP-Adresse, die das lokale OpenVPN tunX Device bekommt. Für die Auswahl der IP-Adresse gelten die gleichen Einschränkungen wie bei OPENVPN_x_REMOTE_VPN_IP (Seite 8).

Es ist übrigens möglich, für alle lokalen OpenVPN Verbindungen die gleiche IP-Adresse bei OPENVPN_x_LOCAL_VPN_IP zu benutzen. So ist es problemlos möglich, dass ein Host in einem VPN immer die gleiche IP-Adresse benutzt. Das vereinfacht die Paketfilterregeln unter Umständen drastisch.

OPENVPN_x_IPV6 Default: OPENVPN_x_IPV6='no'

Hiermit kann der native IPv6-Support von OpenVPN eingeschaltet werden. Da dieser Code noch recht neu ist, ist der Support als experimentell zu bezeichnen. Damit das Ganze einen Effekt hat, muss OPT_IPV6 aktiviert und konfiguriert sein. Bei OPENVPN_x_IPV6='no' und/oder OPT_IPV6='no' werden die IPv6 relevanten Variablen ignoriert.

ACHTUNG!!! Zur Zeit gibt es hier keine Überprüfung ob sich die Angaben mit anderen Teilen der Konfiguration überschneiden! Dies gilt für OPENVPN_x_LOCAL_VPN_IPV6, OPENVPN_x_REMOTE_VPN_IPV6 und OPENVPN_x_ROUTE_x.

OPENVPN_x_REMOTE_VPN_IPV6 Default: OPENVPN_x_REMOTE_VPN_IPV6="

Für die IPv6 gilt das Gleiche wie für die OPENVPN_x_REMOTE_VPN_IP (Seite 8).

```
OPENVPN_X_REMOTE_IPV6='FD00::1'
```

OPENVPN_x_LOCAL_VPN_IPV6 Default: OPENVPN_x_LOCAL_VPN_IPV6="

Für die IPv6 gilt das Gleiche wie für die OPENVPN_x_LOCAL_VPN_IP (Seite 9). Wird hier kein Subnetz gesetzt wird automatisch /64 als Subnetz genutzt.

```
OPENVPN_X_LOCAL_IPV6='FD00::2/112'
```

OPENVPN_x_ROUTE_N Default: `OPENVPN_x_ROUTE_N=''`

Diese Einstellung ist nur gültig, wenn der OPENVPN_x_TYPE (Seite 7) für diese OpenVPN Verbindung auf `'tunnel'` einstellt wurde.

Die angegebenen Routen werden automatisch von OpenVPN gesetzt, sobald OpenVPN gestartet wird. Es können bis zu 50 Netze über eine OpenVPN Verbindung geroutet werden. Sie müssen aber für jedes zu routende Netzwerk einen `OPENVPN_x_ROUTE_x` Eintrag erzeugen.

Bitte beachten Sie, dass Sie notwendige Paketfilterregeln in der `OPENVPN_PF_FORWARD_x` `OPENVPN_PF_INPUT_x` bzw `OPENVPN_PF6_FORWARD_x` `OPENVPN_PF6_INPUT_x` bzw selber setzen müssen. OpenVPN erlaubt nur ICMP über die VPN Verbindungen und verbietet allen anderen Datenverkehr. Details finden Sie unter OPENVPN_x_PF_INPUT_N (Seite 19) und OPENVPN_x_PF_FORWARD_N (Seite 19) bzw. unter OPENVPN_x_PF6_INPUT_N (Seite 20) und OPENVPN_x_PF6_FORWARD_N (Seite 20) bzw.

OPENVPN_x_ROUTE_x Default: `OPENVPN_x_ROUTE_x=''`

Sie müssen hier die Netze angeben, die Sie über die OpenVPN Gegenstelle erreichen wollen. Sind hinter der OpenVPN Gegenstelle z.B. die Netzwerke 192.168.33.0/24 und 172.18.0.0/16 erreichbar und sollen diese über den OpenVPN Tunnel erreicht werden, müssen Sie diese beiden Netze jeweils einzeln unter `OPENVPN_x_ROUTE_x` eintragen. Es können hier auch Hostrouten (`/32`) eingetragen werden.

Wenn die Defaultroute über einen OpenVPN Tunnel gesetzt werden soll, geben sie bitte 0.0.0.0/0 bzw. ::/0 für IPv6 und ein optionales Flag als Route an. Auch hier gilt, das für IPv6 Routen OPT_IPv6 aktiv sein muss, die lokale und remote IPv6-Adresse für den Tunnel müssen gesetzt sein und OPENVPN_x_IPV6 auf yes stehen. OpenVPN kennt verschiedene Möglichkeiten die Defaultroute einzurichten, die sie mit dem Flag auswählen können. Jede Methode, die Defaultroute einzurichten, hat ihre Vor- und Nachteile. Im Moment unterstützt OpenVPN folgende Flags:

local Das *local* Flag sollten sie wählen, wenn die OpenVPN Gegenstelle innerhalb eines direkt von ihrem fli4l-Router erreichbaren Subnetz liegt. Das ist z.B. oft bei einer Defaultroute mit OpenVPN über WLAN der Fall.

def1 Mit diesem Flag werden zusätzliche zu einer Hostroute an die OpenVPN Gegenstelle zwei neue Routen, 0.0.0.0/1 und 128.0.0.0/1, eingetragen. Diese Routen übernehmen die Funktion einer Defaultroute und über diese Routen wird dann der komplette (verschlüsselte) Traffic an die OpenVPN Gegenstelle (die noch über die Hostroute erreichbar ist) geroutet.

Lassen Sie das optionale Flag weg wählt OpenVPN die Methode aus, wie die Defaultroute umgesetzt wird. Die Auswahl der Methode erfolgt dabei über die OpenVPN Version, im Moment wird als Standardeinstellung *local* benutzt.

```
OPENVPN_1_ROUTE_N='3'
OPENVPN_1_ROUTE_1='192.168.33.0/24'
OPENVPN_1_ROUTE_2='172.18.0.0/16'
OPENVPN_1_ROUTE_3='2001:db8:/32'
```

OpenVPN - Delegation von DNS und Reverse-DNS

OPENVPN_x_DOMAIN Default: `OPENVPN_x_DOMAIN="`

Über diesen Parameter gibt man die Remote Domain an. Diese Variable kann mehrere Domains enthalten die durch Leerzeichen getrennt angegeben werden müssen. Wenn nur dieser Parameter gesetzt ist (ohne Angabe eines zusätzlichen DNS-Servers) wird davon ausgegangen, dass der DNS Server auf der IP des gegenüberliegenden Tunnelendes lauscht (siehe `OPENVPN_x_REMOTE_VPN_IP` (Seite 8)). Dazu müssen auf dem Remote Router natürlich eingehende DNS-Anfragen erlaubt werden. (z.B. via `OPENVPN_x_INPUT_y='tmpl:dns ACCEPT'`)

OPENVPN_x_ROUTE_x_DOMAIN Default: `OPENVPN_x_ROUTE_x_DOMAIN="`

Den verschiedenen Subnetzen können auch verschiedene Domains zugeordnet sein. Hier kann man pro `OPENVPN_x_ROUTE_y` eine entsprechende Domain konfigurieren. Sollte ein dazugehöriges `OPENVPN_x_ROUTE_y_DNSIP` existieren, so wird dieser Server benutzt, andernfalls der unter `OPENVPN_x_DNSIP` angegebene. Die Wirkung ist dann die selbe wie mit `OPENVPN_x_DOMAIN`, allerdings eignet sich diese Methode auch zur Dokumentation.

OPENVPN_x_DNSIP Default: `OPENVPN_x_DNSIP="`

Ist der Tunnelendpunkt nicht der zuständige DNS-Server, so kann hier die IP des zuständigen DNS-Servers angegeben werden. Ist nichts angegeben, wird die unter `OPENVPN_x_REMOTE_VPN_IP` (Seite 8) angegebene IP benutzt.

OPENVPN_x_ROUTE_x_DNSIP Default: `OPENVPN_x_ROUTE_x_DNSIP="`

Verschiedene geroutete Subnetze können auch durch verschiedene DNS-Server abgedeckt sein - hier kann man pro `OPENVPN_x_ROUTE_x` (Seite 10) einen eigenen zuständigen Server definieren.

1.1.5. Experteneinstellungen

Die in diesem Kapitel beschriebenen Einstellungen sind alle optional und sollten nur verändert werden, wenn die OpenVPN Verbindung mit den Standardeinstellungen funktioniert und Optimierungen (beispielsweie ein anderer Verschlüsselungsalgorithmus) vorgenommen werden sollen.

Alle nachfolgend beschriebenen `OPENVPN_DEFAULT_` Einstellungen sind optional, d.h. diese Optionen brauchen nicht in die openvpn.txt Konfigurationsdatei geschrieben werden. Fehlt der entsprechende Eintrag in der openvpn.txt Datei, wird vom OpenVPN Startskript der hier angegebene Defaultwert benutzt. Wenn Sie nicht vorhaben die Standardwerte der Vorgaben zu änden, schreiben Sie diese auch nicht in die openvpn.txt Konfigurationsdatei!

allgemeine Einstellungen

OPENVPN_DEFAULT_CIPHER Default: `OPENVPN_DEFAULT_CIPHER='BF-CBC'`

Eine der verfügbaren Verschlüsselungsmethoden. Die Verschlüsselungsmethode `'BF-CBC'` wird von allen OpenVPN Versionen (auch nicht fli4l spezifischen Versionen) als Standardeinstellung benutzt.

OPENVPN_DEFAULT_COMPRESS Default: `OPENVPN_DEFAULT_COMPRESS='yes'`

OpenVPN benutzt eine adaptive LZO Datenkomprimierung, um die Bandbreite einer Verbindung zu erhöhen. Adaptiv bedeutet, dass OpenVPN selbstständig erkennt, wenn z.B. bereits gepackte ZIP Dateien über eine OpenVPN Verbindung geschickt werden. In einem solchen Fall wird die Datenkomprimierung solange abgeschaltet, bis wieder Daten übertragen werden, die auch von einer Datenkomprimierung profitieren. Es gibt fast nie einen Grund die Datenkomprimierung zu deaktiveren, da dadurch die Bandbreite quasi kostenlos erhöht wird. Einziger Nachteil der Datenkomprimierung ist eine geringe Erhöhung der Latenzzeit von wenigen Millisekunden. Für Online-Games via VPN, bei denen die Reaktionszeit ("guter" ping) entscheidend ist, wäre es also unter Umständen sinnvoll die Datenkomprimierung abzuschalten.

OPENVPN_DEFAULT_CREATE_SECRET Default: `OPENVPN_DEFAULT_CREATE_SECRET='no'`

Mit dieser Einstellung erstellt OpenVPN automatisch Keyfiles beim Start des fli4l-Routers. Die entsprechende OpenVPN Verbindung wird allerdings nicht gestartet. Für Details lesen Sie bitte den Punkt `OPENVPN_x_SECRET` (Seite 6) nach.

OPENVPN_DEFAULT_DIGEST Default: `OPENVPN_DEFAULT_DIGEST='SHA1'`

Hier wird eine der verfügbaren Prüfsummen eingetragen. OpenVPN nutzt als Standardeinstellung die Prüfsummenmethode `'SHA1'`.

OPENVPN_DEFAULT_FLOAT Default: `OPENVPN_DEFAULT_FLOAT='yes'`

Bei OpenVPN Verbindungen mit Gegenstellen, die eine DynDNS Adresse benutzen, ist es jederzeit möglich, dass sich die IP-Adresse der OpenVPN Gegenstelle ändert. Damit OpenVPN auch die geänderte IP-Adresse akzeptiert, muss `OPENVPN_DEFAULT_FLOAT` auf `'yes'` eingestellt werden. Mit der Einstellung `'no'` wird die Änderung der IP-Adresse nicht erlaubt. Das ist in der Regel nur bei WLAN Verbindungen oder OpenVPN Verbindungen mit Gegenstellen, die eine statische IP-Adresse (wie z.B. die rootserver diverser Anbieter) haben, sinnvoll. Sie können diese Einstellung auch wie alle anderen `OPENVPN_DEFAULT_` Einstellungen für eine bestimmte OpenVPN Verbindung überschreiben.

OPENVPN_DEFAULT_KEYSIZE Default: `OPENVPN_DEFAULT_KEYSIZE="`

Die Schlüssellänge (=KEYSIZE) hängt von der verwendeten Verschlüsselungsmethode ab. Ändern Sie diese Einstellung nur, wenn Sie mit einer OpenVPN Gegenstelle zusammenarbeiten müssen, die nicht den Standardwert benutzt oder deren Einstellung Sie nicht beeinflussen können. Können Sie die Schlüssellänge selbst bestimmen, sollte dieser Wert immer leer bleiben. OpenVPN wendet dann eine optimale Schlüssellänge für die jeweilige Verschlüsselungmethode an.

OPENVPN_DEFAULT_OPEN_OVPNPORT Default: `OPENVPN_DEFAULT_OPEN_OVPNPORT='yes'`

Damit eine OpenVPN Gegenstelle mit Ihnen Kontakt aufnehmen kann, müssen Sie den Paketfilter Ihres fli4l-Routers entsprechend anpassen. In der Regel müssen Sie also auf allen TCP oder UDP Ports, je nach `OPENVPN_x_PROTOCOL` Einstellung, auf denen ein OpenVPN horcht, die `PF_INPUT_x` (Seite **??**) in der base.txt anpassen. Mit der Einstellung `'yes'` werden diese Paketfilterregeln automatisch generiert. Bei einzelnen Verbindungen

kann es aber durchaus Sinn machen, diese Einstellung auf 'no' zu setzen und selber entsprechende Paketfilterregeln zu definieren.

OPENVPN_DEFAULT_ALLOW_ICMPPING Default: OPENVPN_DEFAULT_ALLOW_ICMPPING='yes'

Mit 'yes' wird der Paketfilter für die entsprechende Verbindung so konfiguriert, dass ping Datenpakete den Filter passieren dürfen. Wenn es keinen sehr wichtigen Grund gibt, sollte der ICMP Ping immer zugelassen werden. Diese Einstellung hat *nichts* mit der Pingoption von OpenVPN zu tun!

OPENVPN_DEFAULT_PF_INPUT_LOG Default: OPENVPN_DEFAULT_PF_INPUT_LOG='BASE'

Mit 'yes' oder 'no' wird eingestellt, ob der Paketfilter eingehende Datenpakete in der INPUT Liste, die für die entsprechende VPN Verbindung gedacht waren mitschreiben soll, wenn das Datenpaket abgelehnt wird. Mit der Einstellung 'BASE' wird die Einstellung von 'PF_INPUT_LOG=' aus der base.txt übernommen.

OPENVPN_DEFAULT_PF_INPUT_POLICY Default: OPENVPN_DEFAULT_PF_INPUT_POLICY='REJECT'

Diese Einstellung entspricht 'PF_INPUT_POLICY=' (Seite **??**) aus der base.txt. Zusätzlich zu den dort möglichen Einstellungen, kann mit 'BASE' die Einstellung von 'PF_INPUT_POLICY=' aus der base.txt übernommen werden.

OPENVPN_DEFAULT_PF_FORWARD_LOG Default: OPENVPN_DEFAULT_PF_FORWARD_LOG='BASE'

Mit 'yes' oder 'no' wird eingestellt, ob der Paketfilter eingehende Datenpakete in der FORWARD Liste, die für die entsprechende VPN Verbindung gedacht waren mitschreiben soll, wenn das Datenpaket abgelehnt wird. Mit der Einstellung 'BASE' wird die Einstellung von 'PF_FORWARD_LOG=' aus der base.txt übernommen.

OPENVPN_DEFAULT_PF_FORWARD_POLICY Default: OPENVPN_DEFAULT_PF_FORWARD_POLICY='REJECT'

Diese Einstellung entspricht 'PF_FORWARD_POLICY=' (Seite **??**) aus der base.txt. Zusätzlich zu den dort möglichen Einstellungen kann mit 'BASE' die Einstellung von 'PF_FORWARD_POLICY=' aus der base.txt übernommen werden.

OPENVPN_DEFAULT_PING Default: OPENVPN_DEFAULT_PING='60'

Um einen einmal aufgebauten Tunnel offen zu halten und zu erkennen, ob die OpenVPN Gegenstelle noch erreichbar ist, wird in dem angegebenen Intervall in Sekunden ein kryptografisch gesicherter ping über die Leitung geschickt. Mit der Einstellung 'off' schickt OpenVPN kein ping über die Leitung. Mit der Einstellung 'off' werden nur dann Daten über den VPN Tunnel geschickt wenn auch tatsächlich Nutzdaten über die VPN Tunnel übertragen werden.

OPENVPN_DEFAULT_RENEG_SEC Default: OPENVPN_DEFAULT_RENEG_SEC='3600'

Hiermit kann die Variable RENEG-SEC aus OpenVPN angepasst werden, so dass gerade bei langsamen DSL-Verbindungen (Oder ISDN) es nicht zu einem Timeout und Abbruch der Verbindung kommt.

OPENVPN_DEFAULT_PING_RESTART Default: OPENVPN_DEFAULT_PING_RESTART='180'

Wird in dem angegebenen Intervall in Sekunden kein OpenVPN ping erfolgreich über die VPN Verbindung geschickt oder werden keine anderen Daten übertragen, wird die entsprechende OpenVPN Verbindung neu gestartet. Der Wert

von OPENVPN_DEFAULT_PING_RESTART muss immer größer sein als der Wert von OPENVPN_DEFAULT_PING. Die Einstellung 'off' unterbindet den automatischen Neustart einer OpenVPN Verbindung.

OPENVPN_DEFAULT_RESOLV_RETRY Default: OPENVPN_DEFAULT_RESOLV_RETRY='infinite'

Sind in OPENVPN_x_REMOTE_HOST oder OPENVPN_x_LOCAL_HOST DNS Namen statt IP-Adressen hinterlegt, dann müssen die Namen beim Start einer OpenVPN Verbindung erst zu IP-Adressen aufgelöst werden. Sollte dies fehlschlagen, versucht OpenVPN für die angegebene Zeit in Sekunden die DNS Adresse erneut aufzulösen. Gelingt dies schlußendlich nicht in der vorgegebenen Zeit, kommt keine OpenVPN Verbindung zustande. Mit der Angabe 'infinite' (= unendlich) wird unendlich lange versucht, die DNS Auflösung vorzunehmen. Diese Einstellung sollte bis auf besondere Fälle nicht verändert werden!

OPENVPN_DEFAULT_RESTART Default: OPENVPN_DEFAULT_RESTART='ip-up'

Nach einer Trennung der Verbindung ist es sinnvoll, dass der entsprechende OpenVPN Tunnel sofort neu gestartet wird, damit die Unterbrechung des Tunnels möglichst kurz ist. Für alle OpenVPN Verbindungen, die über eine Wählverbindung wie DSL oder ISDN laufen, sollte hier 'ip-up' eingetragen werden. OpenVPN Verbindungen über eine WLAN Strecke dagegen sollten hier 'never' eintragen. In diesem Fall wird die Verbindung nicht nach einer Neueinwahl wieder gestartet, da ja die WLAN Verbindung unabhängig von einer Einwahl ist. Wenn ein OpenVPN Tunnel über eine ISDN Wählverbindung mit ISDN_CIRC_x_TYPE='raw' aufgebaut werden soll, muss hier 'raw-up' eingetragen werden.

OPENVPN_DEFAULT_PROTOCOL Default: OPENVPN_DEFAULT_PROTOCOL='udp'

Mit dieser Variablen wird festgelegt, welche Protokollvariante als Default benutzt werden soll. Normalerweise ist UDP eine sehr gute Wahl, allerdings gibt es manchmal nur die Möglichkeit mit TCP zu arbeiten. Der dadurch entstehende Overhead ist allerdings beachtlich. Mögliche Einstellungen sind 'udp', 'udp6' 'tcp-server', 'tcp-server6', 'tcp-client' oder 'tcp-client6'. Die 'tcp-server' oder 'tcp-client' Einstellungen sind in der Regel nur dann sinnvoll, wenn ein VPN Tunnel durch div. andere Paketfilter oder andere Tunnel aufgebaut werden soll. Wenn Sie keine Spezialfälle behandeln müssen, sollten Sie *immer* den Standardwert 'udp' benutzen. Mit angehängtem '6' ist der Tunnel IPv6 fähig (WAN) und kann über ein IPv6-Internet erreicht werden.

OPENVPN_DEFAULT_START Default: OPENVPN_DEFAULT_START='always'

Eine OpenVPN Verbindung kann entweder immer (='always') oder später von Hand (='on-demand') gestartet werden. Sie können also bestimmte OpenVPN Verbindungen erst bei Bedarf über die OpenVPN WebGUI (siehe 1.1.6) starten. Alternativ ist der Start aber auch jederzeit über die fli4l Konsole möglich. Melden Sie sich dazu auf der fli4l Konsole an und führen Sie folgende Befehle direkt auf der Konsole aus:

```
cd /etc/openvpn
openvpn --config name.conf --daemon openvpn-name
```

Damit wird ein OpenVPN Tunnel gestartet, der ab sofort im Hintergrund läuft. Anstelle name.conf nehmen Sie natürlich den Namen Ihrer Konfigurationsdatei in dem /etc/openvpn Verzeichnis.

OPENVPN_DEFAULT_VERBOSE Default: `OPENVPN_DEFAULT_VERBOSE='2'`

Diese Variable gibt an, wie gesprächig OpenVPN sein soll. Wenn eine VPN Verbindung einwandfrei läuft, ist es möglich diesen Wert auf `'0'` zu setzen, um alle Meldungen zu unterbinden. Für die ersten Tests ist ein Wert von `'3'` sinnvoll. Noch größere Werte erhöhen die Debugmeldungen und helfen unter Umständen Fehler zu finden. Der Maximalwert beträgt `'11'`.

OPENVPN_DEFAULT_MANAGEMENT_LOG_CACHE Default:
`OPENVPN_DEFAULT_MANAGEMENT_LOG_CACHE='100'`

Diese Variable gibt an, wie viele Log Zeilen gespeichert werden sollen. Dieses Log kann dann über die WebGUI (Seite 21) abgefragt werden.

OPENVPN_DEFAULT_MUTE_REPLAY_WARNINGS Default:
`OPENVPN_DEFAULT_MUTE_REPLAY_WARNINGS='no'`

Mit dieser Variablen wird eingestellt, ob beim Empfang doppelter Pakete eine Warnung im Log ausgegeben werden soll, da dies Hinweise auf ein Sicherheitsproblem im Netzwerk sein können. Insbesondere bei schwachen WLAN-Verbindungen, kann es aber häufig passieren, dass Pakete doppelt gesendet werden. Dann ist es sinnvoll die Warnungen auszustellen, damit diese nicht das Log füllen. Die Einstellung dieser Variablen hat *keinen* Einfluss auf die Sicherheit der OpenVPN Verbindung.

OPENVPN_DEFAULT_MSSFIX Default: `OPENVPN_DEFAULT_MSSFIX="`

Mit der MSSFIX Einstellung wird die Größe der TCP Datenpakete über die VPN Verbindung vorgegeben. Mit `OPENVPN_DEFAULT_MSSFIX='0'` wird diese Option ausgeschaltet. Wird eine Fragmentgröße angegeben und der MSSFIX Eintrag leer gelassen, wird automatisch die Fragmentgröße benutzt. Diese Einstellung funktioniert nur, wenn `OPENVPN_x_PROTOCOL='udp'` gesetzt wird.

OPENVPN_DEFAULT_FRAGMENT Default: `OPENVPN_DEFAULT_FRAGMENT='1300'`

Aktiviert die interne Fragmentierung von OpenVPN mit einer Paketgröße von x Bytes. Diese Einstellung funktioniert nur, wenn `OPENVPN_x_PROTOCOL='udp'` gesetzt wird.

Mit `OPENVPN_DEFAULT_FRAGMENT='0'` wird die Fragmentierung komplett deaktiviert.

OPENVPN_DEFAULT_TUN_MTU Default: `OPENVPN_DEFAULT_TUN_MTU='1500'`

Stellt die MTU des virtuellen OpenVPN Adapters auf x Bytes ein. Diese Option sollte nur geändert werden, wenn man weiss was man macht. Es ist meistens sinnvoller, erst mit den Fragment oder MSSFIX Optionen zu arbeiten.

OPENVPN_DEFAULT_TUN_MTU_EXTRA Default: `OPENVPN_DEFAULT_TUN_MTU_EXTRA="`

Wenn bei `OPENVPN_x_PROTOCOL='bridge'` eingestellt wird, werden 32 Bytes als extra Speicher für die Verwaltung der Puffers für das tap Gerät reserviert. Bei `OPENVPN_x_PROTOCOL='tunnel'` wird kein extra Speicher reserviert. Diese Einstellung wirkt sich nur auf den Speicherbedarf im Router aus und hat keinen Einfluss auf die Datenmenge die über den Tunnel verschickt wird.

OPENVPN_DEFAULT_LINK_MTU Default: `OPENVPN_DEFAULT_LINK_MTU="`

Stellt die MTU der OpenVPN Verbindung auf x Bytes ein. Diese Option sollte nur benutzt werden, wenn man weiss was man macht. Es ist meistens sinnvoller erst mit den Fragment oder MSSFIX Optionen zu arbeiten.

OPENVPN_DEFAULT_SHAPER Default: `OPENVPN_DEFAULT_SHAPER="`

Begrenzt die *ausgehende* Bandbreite des Tunnel auf die angegebene Anzahl von Bytes pro Sekunde. Möglich sind Werte von 100 Bytes bis zu 100000000 Bytes. Bei Werten bis 1000 Bytes pro Sekunde sollten Sie die MTU der Verbindung reduzieren, sonst steigen die ping Zeiten stark an. Wenn Sie einen Tunnel auf eine Bandbreite in beide Richtungen begrenzen wollen, müssen Sie dies auf beiden Seiten getrennt einstellen.

In der aktuellen OpenVPN Version funktioniert das Shaping nicht korrekt, d.h. die Übertragungsrate durch einen mittels Shapping konfigurierten Tunnel ist unter Umständen extrem schwankend bzw. der Durchsatz bricht total ein. Das Problem kann je nach eingesetzer Hardware auftreten zu komplett unterschiedlichen Verhalten führen. Im Moment sollte die Shappingfunktion mit Vorsicht behandelt werden und im Zweifel bei jeder Änderung ausgiebig getestet werden.

OPENVPN_EXPERT Default: `OPENVPN_EXPERT='no'`

Der Expert-Mode erlaubt Ihnen, native Openvpn Konfigurationsdateien zu nutzen. Diese müssen im Konfigurationsordner unter etc/openvpn sowie etc/openvpn/scripts abgelegt werden. Alle dort liegenden Dateien werden auf den Router übertragen.

Der Expert-Mode ignoriert die restlichen Konfigurationseinstellungen. Deshalb muss `OPENVPN_N='0'` eingestellt werden.

Der Expert-Mode richtet keinerlei Firewall-Regeln ein. Diese müssen Sie manuell in die base.txt eintragen.

verbindungsspezifische Einstellungen

Die folgenden OpenVPN Optionen gelten nur für die jeweilige OpenVPN Verbindung. Auch hier gibt es nur wenige Angaben, die zwingend sind. Die meisten Optionen können einfach weggelassen werden. Für alle Defaultwerte gilt, dass diese von der jeweils gleichlautenden `OPENVPN_DEFAULT_x` Einstellung übernommen werden. Wenn Sie also den entsprechenden `OPENVPN_DEFAULT_` Wert ändern, gilt dieser Defaultwert für alle OpenVPN Verbindungen, die den allgemeinen Defaultwert nicht überschreiben.

OPENVPN_x_NAME Default: `OPENVPN_x_NAME="`

Legt einen bis zu 16 Zeichen langen Namen für die jeweilige OpenVPN Verbindung fest. Unter diesem Namen wird die Konfigurationsdatei im Verzeichnis /etc/openvpn (mit dem Anhang .conf) angelegt. Außerdem erscheint der Name im syslog. Wenn Sie beispielsweise den Namen `'peter'` eintragen, taucht später im syslog der Eintrag 'openvpn-peter' auf. Damit können Sie die verschiedenen OpenVPN Verbindungen gut auseinanderhalten. Der Name darf Buchstaben, Zahlen und das '-' Zeichen enthalten.

OPENVPN_x_ACTIV Default: `OPENVPN_x_ACTIV='yes'`

Wenn man eine OpenVPN Verbindung deaktivieren, aber die Konfiguration nicht löschen will, kann diese OpenVPN Verbindung mit der Einstellung 'no' deaktiviert werden. Die

Konfigurationsdaten werden dann zwar in der rc.cfg aufgenommen, aber es wird keine entsprechende OpenVPN Verbindung erzeugt.

OPENVPN_x_CHECK_CONFIG Default: `OPENVPN_x_CHECK_CONFIG='yes'`

Die erweiterten Prüfungen von OpenVPN sind in seltenen Fällen zu streng. Wenn beispielsweise eine ISDN Backupverbindung die gleichen Routingeinträge benutzt wie eine Verbindung, die über das Internet läuft, wird die erweiterte Prüfung diese Verbindungen mit einer Fehlermeldung bedenken. In diesem Fall sollte bei der Backupverbindung die erweitere Prüfung deaktiviert werden. Dazu setzen Sie `OPENVPN_x_CHECK_CONFIG='no'`, um die Prüfungen für diese Verbindung auszuschalten.

OPENVPN_x_CIPHER Default siehe: `OPENVPN_DEFAULT_CIPHER`

Siehe `OPENVPN_DEFAULT_CIPHER` (Seite 11). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_COMPRESS Default siehe: `OPENVPN_DEFAULT_COMPRESS`

Siehe `OPENVPN_DEFAULT_COMPRESS` (Seite 12). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_CREATE_SECRET Default siehe: `OPENVPN_DEFAULT_CREATE_SECRET='no'`

Siehe `OPENVPN_x_SECRET` (Seite 6). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_DIGEST Default siehe: `OPENVPN_DEFAULT_DIGEST`

Siehe `OPENVPN_DEFAULT_DIGEST` (Seite 12). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_FLOAT Default siehe: `OPENVPN_DEFAULT_FLOAT`

Siehe `OPENVPN_DEFAULT_FLOAT` (Seite 12). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_KEYSIZE Default siehe: `OPENVPN_DEFAULT_KEYSIZE`

Siehe `OPENVPN_DEFAULT_KEYSIZE` (Seite 12). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_ISDN_CIRC_NAME Default `OPENVPN_x_ISDN_CIRC_NAME=”`

Gibt an über welchen ISDN Circuit diese OpenVPN Verbindung aufgebaut wird. Hier wird der Name des entsprechenden ISDN Circuits eingetragen, der mit `ISDN_CIRC_x_NAME="` (Seite **??**) definiert wird. Der entsprechende ISDN Circuit muss dabei vom Typ `'raw'` sein.

OPENVPN_x_PING Default siehe: `OPENVPN_DEFAULT_PING`

Siehe `OPENVPN_DEFAULT_PING` (Seite 13). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_PROTOCOL Default: `OPENVPN_x_PROTOCOL='udp'`

Gibt an mit welchem Protokoll der OpenVPN Tunnel aufgebaut werden soll. Mögliche Einstellungen sind `'udp'`, `'tcp-server'` oder `'tcp-client'`. Die `'tcp-server'` oder `'tcp-client'` Einstellungen sind in der Regel nur dann sinnvoll, wenn ein VPN Tunnel durch diverse andere Paketfilter oder andere Tunnel aufgebaut werden soll. Wenn Sie keine Spezialfälle behandeln müssen, sollten Sie *immer* den Standardwert `'udp'` benutzen.

OPENVPN_x_RESOLV_RETRY Default siehe: `OPENVPN_DEFAULT_RESOLV_RETRY`

Siehe `OPENVPN_DEFAULT_RESOLV_RETRY` (Seite 14). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_PING_RESTART Default siehe: `OPENVPN_DEFAULT_PING_RESTART`

Siehe `OPENVPN_DEFAULT_PING_RESTART` (Seite 13). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_START Default siehe: `OPENVPN_DEFAULT_START`

Siehe `OPENVPN_DEFAULT_START` (Seite 14). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_VERBOSE Default siehe: `OPENVPN_DEFAULT_VERBOSE`

Siehe `OPENVPN_DEFAULT_VERBOSE` (Seite 15). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_MANAGEMENT_LOG_CACHE Default siehe: `OPENVPN_DEFAULT_MANAGEMENT_LOG_CACHE`

Siehe `OPENVPN_DEFAULT_MANAGEMENT_LOG_CACHE` (Seite 15). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_MUTE_REPLAY_WARNINGS Default siehe: `OPENVPN_DEFAULT_MUTE_REPLAY_WARNINGS`

Siehe `OPENVPN_DEFAULT_MUTE_REPLAY_WARNINGS` (Seite 15). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_RESTART Default siehe: `OPENVPN_DEFAULT_RESTART`

Siehe `OPENVPN_DEFAULT_RESTART` (Seite 14). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_ALLOW_ICMPPING Default siehe: `OPENVPN_DEFAULT_ALLOW_ICMPPING`

Siehe `OPENVPN_DEFAULT_ALLOW_ICMPPING` (Seite 13). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_OPEN_OVPNPORT Default siehe: `OPENVPN_DEFAULT_OPEN_OVPNPORT`

Siehe `OPENVPN_DEFAULT_OPEN_OVPNPORT` (Seite 12). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_PF_INPUT_LOG Default siehe: `OPENVPN_DEFAULT_PF_INPUT_LOG`

Siehe `OPENVPN_DEFAULT_PF_INPUT_LOG` (Seite 13). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_PF_INPUT_POLICY Default siehe: OPENVPN_DEFAULT_PF_INPUT_POLICY

Siehe OPENVPN_DEFAULT_PF_INPUT_POLICY (Seite 13). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_PF_INPUT_N Default: OPENVPN_x_PF_INPUT_N='0'

Gibt die Anzahl der folgenden OPENVPN_x_PF_INPUT_x= Einträge an.

OPENVPN_x_PF_INPUT_x Default: OPENVPN_x_PF_INPUT_x="

Wie im base Paket stehen hier die Anweisungen für den Paketfilter. Es wird genau die gleiche Syntax wie in base.txt benutzt. Auch tmpl: und Hostaliase sind möglich. Zusätzlich gibt es noch die Möglichkeit, einige spezielle symbolische Namen zu benutzen. Es werden folgende symbolische Namen unterstützt:

VPNDEV Entspricht dem aktuellen tun Device der jeweiligen OpenVPN Verbindung.

LOCAL-VPN-IP Setzt die IP-Adresse von OPENVPN_x_LOCAL_VPN_IP ein.

REMOTE-VPN-IP Setzt die IP-Adresse von OPENVPN_x_REMOTE_VPN_IP ein.

REMOTE-NET Setzt die IP-Adresse von OPENVPN_x_REMOTE_VPN_IP ein und zusätzlich noch alle Netze die mit OPENVPN_x_ROUTE_x angegeben wurden.

OPENVPN_x_PF_FORWARD_LOG Default siehe: OPENVPN_DEFAULT_PF_FORWARD_LOG

Siehe OPENVPN_DEFAULT_PF_FORWARD_LOG (Seite 13). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_PF_FORWARD_POLICY Default siehe: OPENVPN_DEFAULT_PF_FORWARD_POLICY

Siehe OPENVPN_DEFAULT_PF_FORWARD_POLICY (Seite 13). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_PF_FORWARD_N Default: OPENVPN_x_PF_FORWARD_N='0'

Gibt die Anzahl der folgenden OPENVPN_x_PF_FORWARD_x= Einträge an.

OPENVPN_x_PF_FORWARD_x Default: OPENVPN_x_PF_FORWARD_x="

Siehe OPENVPN_x_PF_INPUT_x (Seite 19).

OPENVPN_x_PF_PREROUTING_N Default: OPENVPN_x_PF_PREROUTING_N='0'

Gibt die Anzahl der folgenden OPENVPN_x_PF_PREROUTING_x= Einträge an.

OPENVPN_x_PF_PREROUTING_x Default: OPENVPN_x_PF_PREROUTING_x="

Siehe OPENVPN_x_PF_INPUT_x (Seite 19).

OPENVPN_x_PF_POSTROUTING_N Default: OPENVPN_x_PF_POSTROUTING_N='0'

Gibt die Anzahl der folgenden OPENVPN_x_PF_POSTROUTING_x= Einträge an.

OPENVPN_x_PF_POSTROUTING_x Default: OPENVPN_x_PF_POSTROUTING_x="

Ab fli4l Revision 3.5.0 (oder 3.5.0-rev18133 für tarball Benutzer) ergibt sich eine Änderung im Verhalten. War es früher möglich Einträge in der Form von

OPENVPN_1_PF_POSTROUTING_1='MASQUERADE'

anzugeben wird ab jetzt die Angabe einer Quell und Zieladresse erzwungen. Dies ist notwendig geworden, weil die POSTROUTING Regeln sonst nicht im vollem Umfang genutzt werden konnten. In den meisten Fällen reicht es einfach die Regeln und IP_NET_x (Seite **??**) und REMOTE-NET zu ergänzen.

Siehe OPENVPN_x_PF_INPUT_x (Seite 19).

OPENVPN_x_PF6_INPUT_N Default: OPENVPN_x_PF6_INPUT_N='0'

Gibt die Anzahl der folgenden OPENVPN_x_PF6_INPUT_x= Einträge an.

OPENVPN_x_PF6_INPUT_x Default: OPENVPN_x_PF6_INPUT_x="

Wie im IPv6 Paket stehen hier die Anweisungen für den Paketfilter. Es wird genau die gleiche Syntax wie in ipv6.txt benutzt. Auch tmpl: und Hostaliase sind möglich. Zusätzlich gibt es noch die Möglichkeit, einige spezielle symbolische Namen zu benutzen. Siehe dazu OPENVPN_x_PF_INPUT_x (Seite 19)

OPENVPN_x_PF6_FORWARD_N Default: OPENVPN_x_PF6_FORWARD_N='0'

Gibt die Anzahl der folgenden OPENVPN_x_PF6_FORWARD_x= Einträge an.

OPENVPN_x_PF6_FORWARD_x Default: OPENVPN_x_PF6_FORWARD_x="

Siehe OPENVPN_x_PF6_INPUT_x (Seite 20).

OPENVPN_x_MSSFIX Default siehe: OPENVPN_DEFAULT_MSSFIX

Siehe OPENVPN_DEFAULT_MSSFIX (Seite 15). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_FRAGMENT Default siehe: OPENVPN_DEFAULT_FRAGMENT

Siehe OPENVPN_DEFAULT_FRAGMENT (Seite 15). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_TUN_MTU Default siehe: OPENVPN_DEFAULT_TUN_MTU

Siehe OPENVPN_DEFAULT_TUN_MTU (Seite 15). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_TUN_MTU_EXTRA Default siehe: OPENVPN_DEFAULT_TUN_MTU_EXTRA

Siehe OPENVPN_DEFAULT_TUN_MTU_EXTRA (Seite 15). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_LINK_MTU Default siehe: OPENVPN_DEFAULT_LINK_MTU

Siehe OPENVPN_DEFAULT_LINK_MTU (Seite 15). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_SHAPER Default siehe: OPENVPN_DEFAULT_SHAPER="

Siehe OPENVPN_DEFAULT_SHAPER (Seite 16). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

1.1.6. OpenVPN - WebGUI

Seit der Version 2.1.10 ist es möglich über eine WebGUI, die konfigurierten OpenVPN Verbindungen zu starten, stoppen und andere grundlegende Funktionen auszuführen. Dazu wird das mini_httpd Paket benötigt. Zudem muss die Variable `OPENVPN_WEBGUI` in der openvpn.txt auf 'yes' gesetzt werden. Dann wird der fli4l Weboberfläche der Menüpunkt OpenVPN hinzugefügt. Wählt man diesen Menüpunkt an, erscheint eine Übersicht über die konfigurierten OpenVPN Verbindungen, deren Status und die jeweils zur Verfügung stehenden Aktionen (siehe Abbildung 1.3).

OpenVPN - WebGUI - Verbindungsübersicht

Abbildung 1.3.: Verbindungsübersicht

Status: Der Status einer Verbindung wird mit Ampelmännchen symbolisiert. Ein rotes Männchen bedeutet, dass der OpenVPN-Prozess nicht läuft, ein gelbes, dass der Prozess zwar läuft aber (noch) keine Verbindung zur Gegenstelle aufgebaut werden konnte und ein grünes, dass die Verbindung mit der Gegenstelle „steht". Genauere Informationen über den Status erhält man als Tooltip über dem Ampelmännchen. Das kann insbesondere bei „gelbem" Status aufschlussreich sein.

Name: In dieser Spalte steht der Name der OpenVPN-Verbindung wie er in der Konfiguration

angegeben wurde. Ein Klick auf den Namen führt in eine Übersicht, in der genauere Informationen zu dieser Verbindung angezeigt werden. Dazu später mehr.

Aktion: Hier sind die zur Verfügung stehenden Aktionen als Buttons symbolisiert. Was sie jeweils bedeuten, erfährt man über Tooltips. Folgende Buttons gibt es:

Symbol	Erläuterung
	Der OpenVPN-Prozess wird gestartet und es wird versucht eine Verbindung aufzubauen.
	Der OpenVPN-Prozess wird beendet.
	Die Verbindung wird zurückgesetzt.
	Die Verbindung wird zurückgesetz und auf 'hold' geschaltet. Dann gehen keine Daten mehr über die Verbindung.
	Die Verbindung wird wieder freigegeben. Daten können wieder über die Verbindung fließen.

Tabelle 1.1.: Aktionen der OpenVPN-Webgui

OpenVPN - WebGUI - Detailansicht einer Verbindung

Statistik: Hier werden ein paar interessante Statistiken angezeigt. Die Statistik kann nur angezeigt werden, wenn die Verbindung gestartet und nicht angehalten ist.

Log: Zeigt die letzten 20 Zeilen des Verbindungslogs. Wenn Sie mehr Zeilen sehen möchten, können Sie auch deren Anzahl angeben und auf 'Anzeigen' klicken. Wenn Sie als Anzahl 'all' einngeben, wird das gesamte Log angezeigt. Dieser Reiter wird nur angezeigt, wenn die Verbindung gestartet ist.

Debug-Log: Zeigt die Ausgabe eines Startvorgangs. Die OpenVPN-Verbindung wird gestartet und die Ausgaben dabei angezeigt. Das ist dann nützlich, wenn die Verbindung über den Start Button nicht starten will und man dadurch kein normales Log zu sehen bekommt. Dieser Reiter wird nur angezeigt, wenn die Verbindung nicht gestartet ist.

Paketfilter: Zeigt die Paketfilterkonfiguration an, die für diese Verbindung gültig ist. Der Paketfilter ist nur konfiguriert, wenn die Verbindung gestartet und als Tunnel eingerichtet ist.

Bridge: Zeigt die Konfiguration der Bridges auf dem Router an. Dieser Punkt nur angezeigt, wenn die Verbindung als Bridge eingerichtet ist.

Konfiguration: Über diesen Punkt kann die beim Booten generierte Konfiguration der Verbindung angeschaut werden.

Abbildung 1.4.: Detailansicht einer Verbindung (Keymanagement)

Keymanagement: Über diesen Punkt kann für diese Verbindung ein Schlüssel erzeugt und auch heruntergeladen werden. (siehe Abbildung 1.4) Ist kein Schlüssel vorhanden (beim ersten Start) wird automatisch einer generiert und angezeigt. Er kann über das Download-Symbol direkt heruntergeladen, oder mit copy/paste in eine Textdatei übertragen werden. Um einen neu generierten Schlüssel auf dem Router zu speichern, ist auf das Disketten-Symbol zu klicken. Ein solcher Speichervorgang kann mit einem Klick auf das Wiederherstellen Symbol rückgängig gemacht werden.

Supportinformationen: Hier werden alle Dinge angezeigt, die bei Problemen relevant sein könnten. Sie können diese Informationen dann beispielsweise auf Anfrage per copy&paste in einen Artikel der Newsgroup übertragen.

1.1.7. OpenVPN - Zusammenarbeit unterschiedlicher OpenVPN Versionen

Bei der Zusammenarbeit unterschiedlicher OpenVPN Versionen muss darauf geachtet werden, dass diese unterschiedliche Standardwerte für die Parameter einer Verbindung benutzen. Das betrifft insbesondere die MTU, Fragment und MSSFIX Einstellungen. Wenn die entsprechenden Werte nicht „zusammenpassen“, ist ein Verbindungsaufbau nicht möglich oder die Verbindung funktioniert zwar mit einem Pingbefehl, bricht aber beispielsweise bei der Benutzung von ssh zusammen. Typische Fehlermeldungen in einem solchen Fall sind z. B.:

```
FRAG_IN error flags=0xfa2a187b: FRAG_TEST not implemented
FRAG_IN error flags=0xfa287f34: spurrious FRAG_WHOLE flags
```

Die entscheidenden Parameter für das Zustandekommen einer Verbindung sind folgende Einstellungen:

OPENVPN_x_TUN_MTU Der MTU Wert des TUN Geräte war bei OpenVPN 1.x auf 1300 eingestellt. Ab OpenVPN 2.0 wird 1500 als Standardwert angenommen.

OPENVPN_x_LINK_MTU Die Bytegröße der Verbindung der beiden OpenVPN Daemonen. Dieser Standardwert ist abhängig von der verwendeten OpenVPN Version und des Betriebssystems.

OPENVPN_x_FRAGMENT Datenpakete (egal ob UDP oder TCP), deren Größe über der Fragmentgrenze liegt, werden auf Datenpakete aufgeteilt, die nicht größer sind als die unter `OPENVPN_x_FRAGMENT` angegeben Bytegröße.

OPENVPN_x_MSSFIX Damit TCP Verbindungen, die über das VPN Daten austauschen, nach Möglichkeit die Datenpakete nicht fragmentieren müssen, kann hier eine gewünschte maximale Größe der TCP Datenpakete vorgegeben werden. An diese Vorgabe halten sich dann aktuelle Betriebssysteme und ein aufwendiges fragmentieren der Datenpakete ist nicht notwendig.

Die unterschiedlichen OpenVPN Versionen benutzen folgende Werte als Standardwerte. Diese Werte müssen Sie beachten, wenn Sie mit OpenVPN Versionen Kontakt aufnehmen wollen, die nicht auf einem fli4l-Router laufen. Die Standardwerte auf dem fli4l-Router sind in der zweiten Tabelle aufgeführt.

OpenVPN Version/Option	1.xx	2.00
OPENVPN_x_TUN_MTU	1300	1500
OPENVPN_x_TUN_MTU_EXTRA	unbekannt	32
OPENVPN_x_FRAGMENT	unbekannt	nicht konfiguriert
OPENVPN_x_MSSFIX	nicht konfiguriert	1450

Tabelle 1.2.: Unterschiedliche MTU Parameter der unterschiedlichen OpenVPN Versionen.

fli4l Version/Option	bis einschließlich 2.1.8	ab 2.1.9
OPENVPN_x_TUN_MTU	1300	1500
OPENVPN_x_TUN_MTU_EXTRA	64	32
OPENVPN_x_FRAGMENT	nicht konfiguriert	1300
OPENVPN_x_MSSFIX	nicht konfiguriert	1300

Tabelle 1.3.: Unterschiedliche MTU Parameter der fli4l-Router Versionen.

Aufgrund dieser unterschiedlichen Einstellungen, sollten Sie die für Ihr Netzwerk passenden Standardwerte ermitteln und diese dann explizit in die config/openvpn.txt schreiben. Folgende Werte sind in den meisten Fällen gute Startwerte für die ersten Tests.

```
OPENVPN_DEFAULT_TUN_MTU='1500'
OPENVPN_DEFAULT_MSSFIX='1300'
OPENVPN_DEFAULT_FRAGMENT='1300'
```

Leider gibt es für fli4l Versionen vor 2.1.9 keine Möglichkeit den „tun-mtu“ Parameter direkt zu setzen. Allerdings läßt sich dieser Parameter indirekt über den `OPENVPN_x_LINK_MTU` beeinflussen. Der tun-mtu Wert ist ca. 45 Bytes kleiner als der bei `OPENVPN_x_LINK_MTU` angegebene Wert. Um den genauen Wert zu ermitteln, hilft nur ausprobieren.

1.1.8. OpenVPN - Beispiele

Einige Beispiele verdeutlichen die Konfiguration des OpenVPN Paketes.

Beispiel - Zwei Netze mit fli4l-Routern verbinden

Im ersten Beispiel werden zwei fli4l-Router miteinander verbunden. Die Netzwerke hinter den fli4l-Routern sollen dabei Zugriff auf das jeweils andere Netzwerk erhalten. In diesem Beispiel wollen Peter und Maria ihre Netzwerke über ihre fli4l-Router miteinander verbinden. Peter benutzt als privates Netz 192.168.145.0/24 und als DynDNS Adresse 'peter.eisfair.net'. Bei Maria sieht es ähnlich aus, nur benutzt sie das Netzwerk 10.23.17.0/24 und als DynDNS Adresse 'maria.eisfair.net'. Das sich beide unbegrenzt vertrauen, erlauben sie sich gegenseitig den kompletten Zugriff auf ihre jeweiligen Netze.

OpenVPN Option	Peter	Maria
OPENVPN_1_NAME=	'maria'	'peter'
OPENVPN_1_REMOTE_HOST=	'maria.eisfair.net'	'peter.eisfair.net'
OPENVPN_1_REMOTE_PORT=	'10000'	'10001'
OPENVPN_1_LOCAL_PORT=	'10001'	'10000'
OPENVPN_1_SECRET=	'pema.secret'	'pema.secret'
OPENVPN_1_TYPE=	'tunnel'	'tunnel'
OPENVPN_1_REMOTE_VPN_IP=	'192.168.200.202'	'192.168.200.193'
OPENVPN_1_LOCAL_VPN_IP=	'192.168.200.193'	'192.168.200.202'
OPENVPN_1_ROUTE_N=	'1'	'1'
OPENVPN_1_ROUTE_1=	'10.23.17.0/24'	'192.168.145.0/24'
OPENVPN_1_PF_INPUT_N=	'1'	'1'
OPENVPN_1_PF_INPUT_1=	'ACCEPT'	'ACCEPT'
OPENVPN_1_PF_FORWARD_N=	'1'	'1'
OPENVPN_1_PF_FORWARD_1=	'ACCEPT'	'ACCEPT'

Tabelle 1.4.: OpenVPN Konfiguration mit 2 fli4l-Routern

Beispiel - Zwei Netze mit einer Bridge verbinden

Im nächsten Beispiel wird eine Bridge über eine Funkverbindung aufgebaut. Bei einer Bridge kann der Paketfilter nicht sinnvoll konfiguriert werden, da dort nur Ethernetframes weitergeleitet werden, aber nicht unbedingt IP-Pakete. Bitte immer daran denken, dass bei einer Bridgekonfiguration ein gemeinsames Netz benutzt werden muss. Und es dürfen keine IP-Adressen doppelt vergeben werden.

Zusätzlich zu den Angaben für OpenVPN muss natürlich noch eine Bridge in advanced_networking konfiguriert werden und die base.txt so angepaßt werden, dass dort die Bridge und nicht eth0 als Netzwerkdevice für das interne Netzwerk benutzt wird. Hier nochmal die relevanten Auszüge aus der Konfiguration von advanced_networking und base.

OpenVPN Option	Peter	Maria
OPENVPN_2_NAME	'bridge'	'bridge'
OPENVPN_2_REMOTE_HOST	'10.1.0.1'	'10.2.0.1'
OPENVPN_2_REMOTE_PORT	'10005'	'10006'
OPENVPN_2_LOCAL_HOST	'10.2.0.1'	'10.1.0.1'
OPENVPN_2_LOCAL_PORT	'10006'	'10005'
OPENVPN_2_FLOAT	'no'	'no'
OPENVPN_2_RESTART	'never'	'never'
OPENVPN_2_SECRET	'bridge.secret'	'bridge.secret'
OPENVPN_2_TYPE	'bridge'	'bridge'
OPENVPN_2_BRIDGE	'pema-br'	'pema-br'

Tabelle 1.5.: OpenVPN Konfiguration mit 2 fli4l-Routern deren Netzwerk über eine Funkverbindung gebridgt wird.

advanced_networking Option	Peter	Maria
OPT_BRIDGE_DEV	'yes'	'yes'
BRIDGE_DEV_BOOTDELAY	'no'	'no'
BRIDGE_DEV_N	'1'	'1'
BRIDGE_DEV_1_NAME	'pema-br'	'pema-br'
BRIDGE_DEV_1_DEVNAME	'br0'	'br0'
BRIDGE_DEV_1_DEV_N	'1'	'1'
BRIDGE_DEV_1_DEV_1_DEV	'eth0'	'eth0'

Tabelle 1.6.: OpenVPN Konfiguration mit 2 fli4l-Routern deren Netzwerk über eine Funkverbindung gebridgt wird. Die Konfiguration der Bridge in advanced_networking.

base Option	Peter	Maria
IP_NET_N	'1'	'1'
IP_NET_1	'192.168.193.254/24'	'192.168.193.1/24'
IP_NET_1_DEV	'br0'	'br0'

Tabelle 1.7.: OpenVPN Konfiguration mit 2 fli4l-Routern deren Netzwerk über eine Funkverbindung gebridgt wird. Die Konfiguration der Bridge in der Basiskonfiguration (base.txt).

Beispiel - Zugriff für einen Roadwarrior konfigurieren

Bei diesem Beispiel (Roadwarrior) wird über ein Notebook mit Windows XP und einem GPRS Zugang der Zugang zu dem LAN hinter dem fli4l-Router ermöglicht. Dazu wird auf dem Windows XP Notebook OpenVPN installiert und die entsprechende *.ovpn Datei angepaßt. Leider ist der tun/tap Treiber unter Windows nicht ganz so flexibel wie sein Unix Gegenstück. Daher müssen die Point-to-Pointadressen für die VPN IP-Adressen in einem 255.255.255.252 (oder /30) Netz liegen. Wenn der Roadwarrior nur auf Dienste im LAN hinter und auf dem fli4l-Router zugreifen soll und nicht selber angesprochen werden muss, ist die Angabe einer Route auf der fli4l Seite nicht notwendig. Der Roadwarrior kann bei Bedarf über seine virtuelle IP-Adresse (`OPENVPN_3_REMOTE_VPN_IP`) angesprochen werden. Wenn der Roadwarrior über eine feste IP-Adresse verfügt, könnte man auch alternativ eine Hostroute eintragen. Wenn der Roadwarrior z.B. die feste IP-Adresse 192.168.33.33 hat, könnte man folgendes noch in die fli4l openvpn.txt Konfigurationsdatei einfügen:

```
OPENVPN_3_ROUTE_N='1'
OPENVPN_3_ROUTE_1='192.168.33.33/32'
```

Mit der Paketfilterkonfiguration, die hier im Beispiel gezeigt wird erlauben wir wieder die komplette Kommunikation in beide Richtungen. Nur auf den fli4l-Router direkt kann der Roadwarrior nicht zugreifen. Das wäre z.B. notwendig, wenn der Roadwarrior den DNS Server auf dem fli4l-Router benutzen soll.

```
OPENVPN_3_PF_FORWARD_N='1'
OPENVPN_3_PF_FORWARD_1='ACCEPT'
```

Soll der Zugriff vom Roadwarrior auf den internen DNS Server auf dem fli4l-Router erlaubt werden, muss noch folgendes zur fli4l Konfiguration dazugeschrieben werden:

```
OPENVPN_3_PF_INPUT_N='1'
OPENVPN_3_PF_INPUT_1='if:VPNDEV:any tmpl:dns ACCEPT'
```

Beispiel - WLAN Verbindung absichern

In diesem Beispiel wird eine WLAN Verbindung mit Hilfe von OpenVPN abgesichert. Es wird davon ausgegangen, dass im fli4l-Router sowohl eine LAN als auch eine WLAN Karte verwendet wird, oder ein Accesspoint an einer zusätzlichen Netzwerkkarte im fli4l angeschlossen ist. Ziel soll es sein, dass ein WLAN Client ohne VPN-Verbindung lediglich Zugriff auf den VPN Port des fli4l-Routers hat. Erst nach dem erfolgreichen Verbinden mit OpenVPN, soll uneingeschränkter Austausch mit dem Kabelgebundenen LAN möglich sein. Es müssen dafür auch Änderungen am DNSMASQ DHCP Server durchgeführt werden. Außerdem wird das advanced_networking Paket benötigt. Einstellungen in base.txt: `IP_NET_1` ist dabei das kabelgebundene LAN und `IP_NET_2` das WLAN.

```
IP_NET_N='2'
IP_NET_1='192.168.3.254/24'
IP_NET_1_DEV='br0'
IP_NET_2='192.168.4.254/24'
IP_NET_2_DEV='eth2'
```

OpenVPN Option fli4l-Router	roadwarrior
OPENVPN_3_NAME='roadwarrior'	remote peter.eisfair.net
OPENVPN_3_LOCAL_PORT='10011'	rport 10011
OPENVPN_3_SECRET='roadwarrior.secret'	secret roadwarrior.secret
OPENVPN_3_TYPE='tunnel'	dev tun
OPENVPN_3_REMOTE_VPN_IP='192.168.200.238'	
OPENVPN_3_LOCAL_VPN_IP='192.168.200.237'	ifconfig 192.168.200.238 192.168.200.237
OPENVPN_3_ROUTE_N='0'	
OPENVPN_3_PF_FORWARD_N='1'	
OPENVPN_3_PF_FORWARD_1='ACCEPT'	
	route 192.168.145.0 255.255.255.0
	comp-lzo
	persist-tun
	persist-key
	ping-timer-rem
	ping-restart 60
	proto udp
	tun-mtu 1500
	fragment 1300
	mssfix

Tabelle 1.8.: OpenVPN Konfiguration mit einem Windowsrechner über GPRS.

Die DHCP-Range ist nach Belieben einzustellen. Für `IP_NET_2` sind aber unbedingt folgende Einstellungen hinzuzufügen:

```
DHCP_RANGE_2_DNS_SERVER1='none'
DHCP_RANGE_2_NTP_SERVER='none'
DHCP_RANGE_2_GATEWAY='none'
```

Einstellung in advanced_networking.txt:

```
OPT_BRIDGE_DEV='yes'
BRIDGE_DEV_BOOTDELAY='yes'
BRIDGE_DEV_N='1'
BRIDGE_DEV_1_NAME='br'
BRIDGE_DEV_1_DEVNAME='br0'
BRIDGE_DEV_1_DEV_N='1'
BRIDGE_DEV_1_DEV_1_DEV='eth0'
```

1.1.9. Weiterführende Links zum Thema OpenVPN

Abschliessend noch einige Links, die sich mit der Konfiguration von OpenVPN beschäftigen:
http://openvpn.net
http://de.wikipedia.org/wiki/OpenVPN
http://openvpn.se/
http://arnowelzel.de/wiki/de/fli4l/openvpn
http://wiki.freifunk.net/OpenVPN
http://w3.linux-magazine.com/issue/24/Charly.pdf
http://w3.linux-magazine.com/issue/25/WirelessLAN_Intro.pdf
http://w3.linux-magazine.com/issue/25/OpenVPN.pdf

OpenVPN Option Router	WLAN-Client
OPENVPN_4_NAME='wlan1'	
OPENVPN_4_LOCAL_HOST='192.168.4.254'	remote 192.168.4.254
OPENVPN_4_LOCAL_PORT='20001'	rport 20001
OPENVPN_4_SECRET='wlan1.secret'	secret wlan1.secret
OPENVPN_4_TYPE='bridge'	dev tap
OPENVPN_4_BRIDGE='br'	
OPENVPN_4_RESTART='never'	
OPENVPN_4_MUTE_REPLAY_WARNINGS='yes'	
	comp-lzo
	persist-tun
	persist-key
	ping-timer-rem
	ping-restart 60
	proto udp
	tun-mtu 1500
	fragment 1300
	mssfix

Tabelle 1.9.: OpenVPN Absicherung eines WLAN.

A. Anhang zum Paket OPENVPN

Abbildungsverzeichnis

Tabellenverzeichnis

Index