OpenVPN Debian 7

Aus Wiki

Achtung: wir noch Fertiggestellt!

Dies ist eine kleine Schritt für Schritt Anleitung, wie man einen OpenVPN[1] Server auf Debian 7[2] bei einem Hoster[3] installiert.

Inhaltsverzeichnis

System auf dem Aktuellen Stand bringen

Zum Updaten vom System verwende ich *Aptitude*[4]

```
aptitude update
aptitude safe-upgrade
aptitude full-upgrade
```

Überprüfen ob TUN bei Kernel unterstützt wird

Mit diesem Einzeiler kann man als *Root* in der Shell[5] kurz Prüfen ob der *TUN*[6] Support aktiviert wurde.

Falls man einen VPS oder OpenVZ[7] Server hat, könnte es sein dass es vom Hostsystem der *TUN Support* aktiviert werden muss.

```
test ! -c /dev/net/tun && echo openvpn requires tun support || echo tun is available
```

OpenVPN installieren

OpenVPN installation per *Aptitude*[4]

```
aptitude install openvpn
```

RSA erstellen mit easy-rsa

Erstellen von *easy-rsa*[8] unter */root/easy-rsa*.

Dazu kopieren wir von */usr/share/doc/openvpn/examples/easy-rsa/2.0* nach */root/easy-rsa*.

```
cp -prv /usr/share/doc/openvpn/examples/easy-rsa/2.0 /root/easy-rsa
```

```
cd /root/easy-rsa
cp vars{,.orig}
```

Einstellen von Standart werten bei easy-rsa

Einstellen von Standart werten mit diesem Skript.

```
nano ./vars

KEY_SIZE=2048
KEY_COUNTRY="AT"
KEY_PROVINCE="Tirol"
KEY_CITY="Innsbruck"
KEY_ORG="example Street"
KEY_EMAIL="webmaster@pratznschutz.com"
```

Exportiere es

```
source ./vars
```

Lösche zuvor erstellte Zertifikate

```
./clean-all
```

Erstelle das CA Zertifikat

Erstelle das CA Zertifikat mit dem *build-ca* Skript.

```
./build-ca
```

Erstelle das Server Zertifikat

Erstellen von VPN Server Zertifikat mit dem Tool *build-key-server*

```
./build-key-server server
```

Diese 2 Fragen mit *Yes* beantworten

- Signiere das Zertifikat [y/n]: y
- 1 out of 1 certificate requests certified, commit? [y/n]: y

Erstelle ein Diffi-Hellman PEM Zertifikat

Zum erstellen der Diffi-Hellman verwendet wir das Skript *build-dh*

```
./build-dh
```

Client Zertifikat erstellen

Client Zertifikat mit *build-key* erstellen.

Dabei könnte man auch den Namen des Clients verwenden, anstatt *client*.

```
./build-key client
```

Die beiden Fragen wieder mit *Yes* Beantworten.

- Das Zertifikat Signieren [y/n]: y
- 1 out of 1 certificate requests certified, commit? [y/n]: y

HMAC Schlüssel erstellen

Erstelle den HMAC Schlüssel mit dem folgendem Befehl

```
openvpn --genkey --secret /root/easy-rsa/keys/ta.key
```

Zertifikate Verteilen / Verwalten

Kopiere die benötigten Zeritifikate auf die jeweiligen Maschinen (Client/Server).

- Der Public **ca.crt** wird auf allen Servern oder Client gebraucht.
- Der Privat Schlüssel **ca.key** wird nur auf den Maschinen gebraucht die Schlüssel und Zertifikate erstellen müssen.
- Ein Server braucht **server.crt**,**dh2048.pem** (public) und **ca.key** (privat)
- Ein Client braucht **client.crt** (public), **client.key** und **ca.key** (privat)

Konfigurieren der Zertifikate auf dem Server

Die Zertifikate und Schlüssel werden auf dem Server in das Verzeichnis */etc/openvpn/certs* abgelegt.

```
mkdir -p /etc/openvpn/certs
cp -pv /root/easy-rsa/keys/{ca.{crt,key},server.{crt,key},ta.key,dh2048.pem} /etc/openvpn/certs/
```

Konfiguriere den OpenVPN Server

Unter */etc/openvpn/server.conf* konfigurieren wir den Server

```
cat > /etc/openvpn/server.conf

port 1194
proto udp
dev tun

ca /etc/openvpn/certs/ca.crt
cert /etc/openvpn/certs/server.crt
key /etc/openvpn/certs/server.key
dh /etc/openvpn/certs/dh2048.pem
tls-auth /etc/openvpn/certs/ta.key 0

server 192.168.88.0 255.255.255.0
ifconfig-pool-persist ipp.txt
push "redirect-gateway def1 bypass-dhcp"
push "dhcp-option DNS 8.8.8.8"
push "dhcp-option DNS 8.8.4.4"

client-to-client
keepalive 1800 4000

cipher DES-EDE3-CBC # Triple-DES
comp-lzo

max-clients 10

user nobody
group nogroup

persist-key
persist-tun

#log openvpn.log
#status openvpn-status.log
verb 5
mute 20
```

OpenVPN Server starten

Mit folgendem Befehl startet man den OpenVPN Server und aktiviert ihn für den Automatischen Start.

```
service openvpn restart
update-rc.d -f openvpn defaults
```

Forwarding aktivieren und die IP-Tables konfigurieren

Netzwerk Forwarding Auskommentieren/hinzufügen *net.ipv4.ip_forward = 1* in der *sysctl.con* um es zu aktivieren

```
vim /etc/sysctl.conf
:%s/net.ipv4.ip_forward = 0/net.ipv4.ip_forward = 1/
```

```
sysctl -p
```

IPTables Regeln einstellen

```
iptables -A INPUT -p udp -m state --state NEW -m udp --dport 1194 -j ACCEPT
iptables -A FORWARD -s 192.168.88.0/24 -j ACCEPT
iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -t nat -A POSTROUTING -s 192.168.88.0/24 -o eth0 -j MASQUERADE
# Bei OpenVZ/vServer muss man dies eingeben
# iptables -t nat -A POSTROUTING -s 192.168.88.0/24 -j SNAT --to-source <PUBLIC_VPN_IP>

iptables-save > /etc/iptables.rules
```

Damit die IPTables Konfiguration auch einen Neustart überlebt, kann man dies mit dem Packet *iptables-persistent* oder mit einem simplen Skript erledigen.

Das Skript wird in das Verzeichnis */etc/network/if-pre-up.d/* hinterlegt.

Dies ladet das Skript */etc/iptables.rules* mit *iptables-restore*.

Als Beispiel:

```
cat /etc/network/if-pre-up.d/iptables

#!/bin/bash
test -e /etc/iptables.rules && iptables-restore -c /etc/iptables.rules
```

Quellen

1. ↑ *OpenVPN.net (http://openvpn.net/)*
2. ↑ *Debian (https://www.debian.org/index.de.html) 7 (Wheezy)*
3. ↑ *Hetzner AG: Root Server Webseite (http://www.hetzner.de/)*
4. ↑ [4,0] [4,1] *Debian: Erklärung zu Aptitude (https://wiki.debian.org/Aptitude)*
5. ↑ *Wikipedia (http://de.wikipedia.org/wiki/Unix-Shell) Shell erklärung*
6. ↑ *OpenVPN Wiki: Vergleich TUN/TAP (http://wiki.openvpn.eu/index.php/Vergleich_TUN/TAP)*
7. ↑ *Wikipedia Beitrag zu OpenVZ (http://de.wikipedia.org/wiki/OpenVZ)*
8. ↑ *Debian Package: easy-rsa (https://packages.debian.org/de/source/sid/easy-rsa)*