

# Ubiquiti

- [EdgeMAX](#)
  - [DualStack IPv6 mit 1&1 auf dem EdgeRouter PoE 5-Port](#)
  - [EdgeMAX Tipps und Tricks](#)
  - [EdgeRouter PoE 5-Port](#)
  - [Eigenes SSL-Zertifikat für die Web-GUI hinterlegen](#)
  - [IPv6 Tunnelbroker \(Hurricane Electric\)](#)
  - [OpenVPN ohne Konfigurationsänderung neu starten](#)
  - [tcpdump zu lokalem Wireshark umleiten](#)
  - [USB-Flashspeicher im EdgeRouter ersetzen](#)

EdgeMAX

# DualStack IPv6 mit 1&1 auf dem EdgeRouter PoE 5-Port

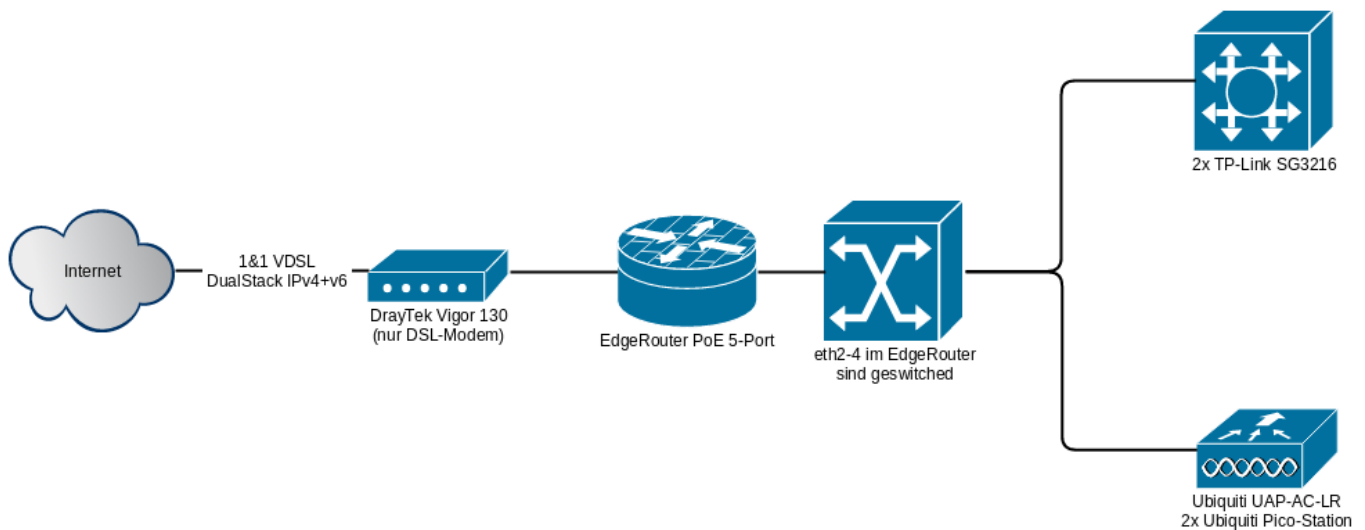
Zuletzt getestet am 08.07.2016

Schon seit einiger Zeit kann man mit Telekom, bzw. den Reseller-Anschlüssen von 1&1 DualStack [IPv6](#) fahren (laut verschiedenen Quellen aber nur bei VDSL). In diesem Artikel will ich meine Konfiguration vorstellen.

## meine aktuelle Hardware

- DSL-Modem DrayTek Vigor 130 (wie man das mit 1&1 VDSL50 einrichtet kann man [hier](#) nachlesen)
- Ubiquiti EdgeRouter PoE 5-Port
- 2x TP-Link TL-SG3216 Managed Switches
- Ubiquiti UAP-AC-LR
- 2x Ubiquiti Pico-Station M2
- TP-Link WR1043ND (für Freifunk)

## mein aktuelles Netzwerksetup



## VLANS

- Management VLAN untagged
- VLAN 10 - mein eigenes Netz
- VLAN 11 - eingeschränktes Gästernetzwerk
- VLAN 20 - Freifunk Client-Netz
- VLAN 21 - Freifunk BATMAN
- VLAN 22 - Freifunk WAN (Uplink für einen Router, hätte ich auch ins Gästernetz stecken können)

IPv6 benötige ich für die VLANs 10 und 11. In den Übrigen ist es aktuell nicht notwendig oder nicht erwünscht.

# Ports am EdgeRouter

- eth0 - geht zum LAN-Port des DrayTek DSL-Modems
- eth1 - unbelegt
- eth2-4 - Zusammengefasst zu switch0
- switch0
  - Uplink mit PoE zum WLAN AP (Ubiquiti UAP-AC-LR)
  - Uplink zu den beiden TP-Link Managed Switchen

## Konfiguration des Routers

Ich zeige hier nur die relevanten Teile als „show configuration commands“ (also zum einfachen Übernehmen per Copy&Paste). Meine komplette Konfiguration (ohne Passwörter und private Teile natürlich) kann man sich hier anschauen.

Vor dem EdgeRouter hängt ein DrayTek Vigor 130 (als reines DSL-Modem konfiguriert). Die PPPoE Einwahl konfiguriere ich also im EdgeRouter.

Zuerst die IPv6 Firewall für das WAN-Interface pppoe0:

```
set firewall ipv6-name WAN_IN_v6 default-action drop
set firewall ipv6-name WAN_IN_v6 description 'incoming IPv6 traffic to local networks'
set firewall ipv6-name WAN_IN_v6 enable-default-log
set firewall ipv6-name WAN_IN_v6 rule 1 action accept
set firewall ipv6-name WAN_IN_v6 rule 1 description 'Allow established/related'
set firewall ipv6-name WAN_IN_v6 rule 1 log disable
set firewall ipv6-name WAN_IN_v6 rule 1 protocol all
set firewall ipv6-name WAN_IN_v6 rule 1 state established enable
set firewall ipv6-name WAN_IN_v6 rule 1 state invalid disable
set firewall ipv6-name WAN_IN_v6 rule 1 state new disable
set firewall ipv6-name WAN_IN_v6 rule 1 state related enable
set firewall ipv6-name WAN_IN_v6 rule 2 action drop
set firewall ipv6-name WAN_IN_v6 rule 2 description 'Drop invalid state'
set firewall ipv6-name WAN_IN_v6 rule 2 log disable
set firewall ipv6-name WAN_IN_v6 rule 2 protocol all
set firewall ipv6-name WAN_IN_v6 rule 2 state established disable
set firewall ipv6-name WAN_IN_v6 rule 2 state invalid enable
set firewall ipv6-name WAN_IN_v6 rule 2 state new disable
set firewall ipv6-name WAN_IN_v6 rule 2 state related disable
set firewall ipv6-name WAN_IN_v6 rule 3 action accept
set firewall ipv6-name WAN_IN_v6 rule 3 description 'allow icmpv6'
set firewall ipv6-name WAN_IN_v6 rule 3 log disable
set firewall ipv6-name WAN_IN_v6 rule 3 protocol icmpv6
set firewall ipv6-name WAN_LOCAL_v6 default-action drop
set firewall ipv6-name WAN_LOCAL_v6 description 'incoming IPv6 traffic to EdgeRouter'
set firewall ipv6-name WAN_LOCAL_v6 enable-default-log
set firewall ipv6-name WAN_LOCAL_v6 rule 1 action accept
set firewall ipv6-name WAN_LOCAL_v6 rule 1 description 'Allow established/related'
set firewall ipv6-name WAN_LOCAL_v6 rule 1 log disable
set firewall ipv6-name WAN_LOCAL_v6 rule 1 protocol all
set firewall ipv6-name WAN_LOCAL_v6 rule 1 state established enable
set firewall ipv6-name WAN_LOCAL_v6 rule 1 state invalid disable
set firewall ipv6-name WAN_LOCAL_v6 rule 1 state new disable
set firewall ipv6-name WAN_LOCAL_v6 rule 1 state related enable
set firewall ipv6-name WAN_LOCAL_v6 rule 2 action drop
set firewall ipv6-name WAN_LOCAL_v6 rule 2 description 'Drop invalid state'
set firewall ipv6-name WAN_LOCAL_v6 rule 2 log disable
set firewall ipv6-name WAN_LOCAL_v6 rule 2 protocol all
set firewall ipv6-name WAN_LOCAL_v6 rule 2 state established disable
set firewall ipv6-name WAN_LOCAL_v6 rule 2 state invalid enable
set firewall ipv6-name WAN_LOCAL_v6 rule 2 state new disable
set firewall ipv6-name WAN_LOCAL_v6 rule 2 state related disable
set firewall ipv6-name WAN_LOCAL_v6 rule 3 action accept
set firewall ipv6-name WAN_LOCAL_v6 rule 3 description 'allow icmpv6'
set firewall ipv6-name WAN_LOCAL_v6 rule 3 log disable
set firewall ipv6-name WAN_LOCAL_v6 rule 3 protocol icmpv6
set firewall ipv6-name WAN_LOCAL_v6 rule 4 action accept
set firewall ipv6-name WAN_LOCAL_v6 rule 4 description 'allow dhcpv6'
set firewall ipv6-name WAN_LOCAL_v6 rule 4 destination port 546
set firewall ipv6-name WAN_LOCAL_v6 rule 4 protocol udp
set firewall ipv6-name WAN_LOCAL_v6 rule 4 source port 547
```

```

set firewall ipv6-name WAN_OUT_v6 default-action accept
set firewall ipv6-name WAN_OUT_v6 description 'outgoing IPv6 traffic'
set firewall ipv6-receive-redirects disable
set firewall ipv6-src-route disable
set firewall options mss-clamp interface-type all
set firewall options mss-clamp mss 1452
set firewall options mss-clamp6 interface-type all
set firewall options mss-clamp6 mss 1412
set firewall receive-redirects disable
set firewall send-redirects enable
set firewall source-validation disable
set firewall syn-cookies enable
set firewall ip-src-route disable
set firewall log-martians enable

```

Damit werden DHCPv6 und ICMPv6 eingehend auf den Router erlaubt. established und related Pakete dürfen auch in die internen Netze und ausgehend ist über v6 alles erlaubt.

Der PPPoE-Teil sieht so aus:

```

set interfaces ethernet eth0 address 192.168.xxx.2/24
set interfaces ethernet eth0 description 'Internet (PPPoE)'
set interfaces ethernet eth0 duplex auto
set interfaces ethernet eth0 speed auto
set interfaces ethernet eth0 pppoe 0 default-route auto
set interfaces ethernet eth0 pppoe 0 description 'l&l VDSL-50'
set interfaces ethernet eth0 pppoe 0 dhcpv6-pd pd 0 interface switch0 host-address '::1d1e:f001'
set interfaces ethernet eth0 pppoe 0 dhcpv6-pd pd 0 interface switch0 no-dns
set interfaces ethernet eth0 pppoe 0 dhcpv6-pd pd 0 interface switch0 prefix-id 0
set interfaces ethernet eth0 pppoe 0 dhcpv6-pd pd 0 interface switch0 service slaac
set interfaces ethernet eth0 pppoe 0 dhcpv6-pd pd 0 interface switch0.10 host-address '::dead:be
set interfaces ethernet eth0 pppoe 0 dhcpv6-pd pd 0 interface switch0.10 no-dns
set interfaces ethernet eth0 pppoe 0 dhcpv6-pd pd 0 interface switch0.10 prefix-id 10
set interfaces ethernet eth0 pppoe 0 dhcpv6-pd pd 0 interface switch0.10 service slaac
set interfaces ethernet eth0 pppoe 0 dhcpv6-pd pd 0 interface switch0.11 host-address '::b00b:ba
set interfaces ethernet eth0 pppoe 0 dhcpv6-pd pd 0 interface switch0.11 no-dns
set interfaces ethernet eth0 pppoe 0 dhcpv6-pd pd 0 interface switch0.11 prefix-id 11
set interfaces ethernet eth0 pppoe 0 dhcpv6-pd pd 0 interface switch0.11 service slaac
set interfaces ethernet eth0 pppoe 0 dhcpv6-pd pd 0 prefix-length 56
set interfaces ethernet eth0 pppoe 0 dhcpv6-pd prefix-only
set interfaces ethernet eth0 pppoe 0 dhcpv6-pd rapid-commit enable
set interfaces ethernet eth0 pppoe 0 firewall in ipv6-name WAN_IN_v6
set interfaces ethernet eth0 pppoe 0 firewall in name WAN_IN
set interfaces ethernet eth0 pppoe 0 firewall local ipv6-name WAN_LOCAL_v6
set interfaces ethernet eth0 pppoe 0 firewall local name WAN_LOCAL
set interfaces ethernet eth0 pppoe 0 firewall out ipv6-name WAN_OUT_v6
set interfaces ethernet eth0 pppoe 0 firewall out name WAN_OUT
set interfaces ethernet eth0 pppoe 0 ipv6 address autoconf
set interfaces ethernet eth0 pppoe 0 ipv6 dup-addr-detect-transmits 1
set interfaces ethernet eth0 pppoe 0 ipv6 enable
set interfaces ethernet eth0 pppoe 0 mtu 1492
set interfaces ethernet eth0 pppoe 0 name-server auto
set interfaces ethernet eth0 pppoe 0 password XXXXXXXXXXXXXXXXXXXXXXXX
set interfaces ethernet eth0 pppoe 0 user-id Hlundl/pt1234-567@online.de

```

Weitere Konfiguration ist nicht mehr nötig. Die internen Netze werden per DHCP Prefix-Delegation und SLAAC (Clients erzeugen sich ihre Adresse selbst mit dem entsprechenden Prefix) mit IPv6-Adressen versorgt. Dabei habe ich den Interfaces auf dem Gateway „sprechende IPs“ verpasst ;)

## IPv6 Privacy Extensions

Es empfiehlt sich die IPv6 Privacy Extensions zu aktivieren, sofern diese nicht bereits standardmäßig aktiv sind. Wenn diese nicht aktiviert sind, läßt sich der eigene Client ziemlich leicht anhand der in der IPv6 Adresse vorhandenen MAC-Adresse verfolgen bzw. wieder erkennen. Wie sich die Privacy Extensions aktivieren lassen, kann man z.B. in diesem [Artikel auf heise.de](#) nachlesen.

# EdgeMAX Tipps und Tricks

## Logging per Syslog auf einen entfernten Server

Auf dem entfernten System muss das remote-Logging aktiviert und der Empfang von Paketen auf Port 514 (UDP normalerweise) möglich sein.

```
set system syslog host <server fqdn> facility all level notice
```

## Logging von DHCPREQUEST und DHCPACK

```
set system syslog file dhcpd facility local2 level debug
set system syslog file dhcpd archive files 5
set system syslog file dhcpd archive size 5000
set service dhcp-server global-parameters 'log-facility local2;'
```

## DHCP-Bootoptionen für das Deployment von SNOM-Telefonen

Diese DHCP-Optionen lassen sich z.B. mit der Telefonanlage Telpho einsetzen, um SNOM-Telefone mit aktueller Firmware und der SIP-Konfiguration zu versorgen (Auto Provisioning).

```
set service dhcp-server global-parameters "option boot-server code 66 = string;"
set service dhcp-server global-parameters "option boot-server &quot;http://<asterisk-server>/sno
```

## dnsmasq als DNS Recursor

```
set service dhcp-server use-dnsmasq enable
set service dns forwarding cache-size 1024
set service dns forwarding listen-on eth0
set service dns forwarding listen-on switch0
set service dns forwarding listen-on switch0.10
set service dns forwarding listen-on eth1
set service dns forwarding listen-on switch0.22
set service dns forwarding listen-on switch0.11
set service dns forwarding listen-on vtun0
set service dns forwarding name-server 1.1.1.1
set service dns forwarding name-server 1.0.0.1
set service dns forwarding name-server '2606:4700:4700::1111'
set service dns forwarding name-server '2606:4700:4700::1001'
set service dns forwarding options domain-needed
set service dns forwarding options 'dhcp-option=tag:brotnetz,option:domain-search,fue.ovtec.it,o
```

## Logging von DNS Queries (mit dnsmasq)

```
set service dns forwarding options log-queries
```

## DHCP Client speziellen Nameserver zuweisen

Die MAC-Adresse des Clients wird für die Zuweisung verwendet. Die Eintragung erfolgt im globalen options-Bereich.

```
set service dns forwarding options 'dhcp-mac=set:special,ac:5f:3e:1c:42:02'
set service dns forwarding options 'dhcp-option=tag:special,option:dns-server,172.16.66.26'
```

## Konfiguration bei commit auf entferntem Server speichern

Damit lässt sich die geänderte Konfiguration automatisch auf einem externen Server ablegen. Mögliche Übertragungarten sind scp, ftp und tftp.

```
admin@gate# set system config-management commit-archive location
Possible completions:
  <uri>  Uniform Resource Identifier

Detailed information:

  "scp://<user>:<passwd>@<host>/<dir>"
  "ftp://<user>:<passwd>@<host>/<dir>"
  "tftp://<host>/<dir>"

# Beispiel:
# set system config-management commit-archive location scp://user:passwd@fqdn.de/mysettings
```

## upnp2 aktivieren

```
service {
    upnp2 {
        listen-on switch0
        nat-pmp enable
        secure-mode enable
        wan pppoe0
    }
}
```

## externe Links

- [Firmware-Images verwalten](#)
- [Eigene Änderungen über das Firmwareupdate hinaus behalten](#)
- [Adblocker über dnsmasq](#)

EdgeMAX

# EdgeRouter PoE 5-Port

Dieser kleine Router ist als SOHO-Router konzipiert. Die Software basiert auf dem Vyatta Router-OS und ist bequem über CLI oder auch über eine Weboberfläche konfigurierbar. Die Web-GUI stellt zudem einige praktische Graphen zum aktuellen Traffic bereit.



Der Router kann z.B. bei [Amazon](#) bestellt werden. Der Preis liegt bei ca. 180€.

Die Daten sind wie folgt (neue Features werden regelmäßig durch Firmwareupdates implementiert, zudem lassen sich auch Debian-Pakete installieren):

<b>Performance</b>	1 Million packets per second (64 byte size)
<b>Ports</b>	5 GBit Port 24V/48V Passive PoE (3 für Switching)
<b>Interface/Encapsulation Ethernet</b>	802.1q VLAN PPPoE GRE IP in IP Bridging Bonding (802.3ad)
<b>Addressing</b>	Static IPv4/IPv6 Addressing DHCP/DHCPv6
<b>Routing Static Routes</b>	OSPF/OSPFv3 RIP/RIPng BGP (with IPv6 Support) IGMP Proxy
<b>Security</b>	ACL-Based Firewall Zone-Based Firewall NAT
<b>VPN</b>	IPSec Site-to-Site and Remote Access OpenVPN Site-to-Site and Remote Access PPTP Remote Access L2TP Remote Access PPTP Client
<b>Services</b>	DHCP/DHCPv6 Server DHCP/DHCPv6 Relay Dynamic DNS DNS Forwarding VRRP RADIUS Client Web Caching PPPoE Server
<b>QoS</b>	FIFO Stochastic Fairness Queueing Random Early Detection Token Bucket Filter Deficit Round Robin Hierarchical Token Bucket Ingress Policing



Management	Web UI CLI (Console, SSH, Telnet) SNMP NetFlow LLDP NTP UBNT Discovery Protocol Logging
------------	--

# Eigenes SSL-Zertifikat für die Web-GUI hinterlegen

Die Web-GUI der EdgeMAX Geräte läuft auf einem Lighttpd-Webserver. Will man diese per SSL/TLS absichern, muss man dies direkt über die Shell erledigen. Der configure-Modus bietet dafür keine Option.

Ich gehe davon aus, dass man ein vollständiges (evtl. auch selbstsigniertes) Zertifikat vorliegen hat (weitere Informationen dazu in meinem [SSL-Artikel](#)).

Für die Installation des Zertifikats loggt man sich per SSH auf dem Router ein. Auf dem Prompt dann per `sudo -i` zum Shell-User root wechseln.

Das Zertifikat muss jetzt direkt im Konfigurationsordner von lighttpd hinterlegt werden:

```
cd /etc/lighttpd
mv server.pem server.pem_old # falls bereits ein Zertifikat vorhanden ist

# Key + Zertifikat (ggf. Zwischenzertifikat der CA) hier reinkopieren:
vi /etc/lighttpd/server.pem

chmod 400 /etc/lighttpd/server.pem
chown root:root /etc/lighttpd/server.pem

# lighttpd neu starten (er droppt dann seine Privilegien und läuft als www-data)
pgrep lighttpd | xargs kill
/usr/sbin/lighttpd -f /etc/lighttpd/lighttpd.conf
```

# IPv6 Tunnelbroker (Hurricane Electric)

Diese Anleitung soll die Einrichtung eines 6to4 IPv6 Tunnels beschreiben. Bitte die Anleitung nicht einfach per Copy&Paste durcharbeiten, sondern auch verstehen. Meine persönlichen Angaben zum Tunnel oder Zugangsdaten wurden anonymisiert oder geschwärzt.

Mit dem [IPv6 Tunnelbroker](#) bietet [Hurricane Electric](#) (HE) einen kostenlosen 6to4 Tunnelservice an (ähnlich wie Sixxs, nur ist HE nicht so [assig](#) und [unfreundlich](#)). Man bekommt direkt ein /64er Netz zugewiesen und verfügt damit über 18446744073709551616 IP-Adressen. Das sollte eine Weile ausreichen. Man kann sich auch ein /48er Netz geben lassen. In ein /48 passen nochmal 65536 /64er Netze. Das entspricht dann 1208925819614629174706176 IPv6-Adressen.

Um nun den Tunnelbroker Service mit [Ubiquiti EdgeMAX Routern](#) verwenden zu können sind ein paar vorbereitende Schritte notwendig. Verfügt man über eine statische IP für den Internetzugang, kann der DynDNS-Teil weg gelassen werden. Die Anleitung geht im Folgenden davon aus, dass man seine WAN-IP dynamisch zugeteilt bekommt.

1. Anmeldung bei [Tunnelbroker](#)
2. einen IPv6 Tunnel registrieren
3. Anmeldung bei [DNS-O-MATIC](#). Dieser kostenlose Service kann verschiedenste andere Services über eine neu zugeteilte WAN-IP informieren. Ich nutze das z.B. mit [afraid.org](#) (dynDNS Provider) und um die clientseitige Tunnel-IP bei HE zu setzen, da sonst der Tunnel nicht aufgebaut werden kann

## DNS-O-MATIC

Beginnen wir mit der Konfiguration von DNS-O-MATIC. Auf deren Webseite habe ich die beiden Services [afraid.org](#) und Tunnelbroker angelegt.

Bei [afraid.org](#) kann man sich eine Subdomain unter verschiedensten Domains aussuchen und diese bei einer Änderung der WAN-IP durch seinen Router oder andere Mechanismen automatisch mit der neuen IP updaten lassen.

Um das durch DNS-O-MATIC erledigen zu lassen, besorgt man sich auf der afraid-Seite <http://freedns.afraid.org/dynamic/> seinen Update-Key. Dieser versteckt sich bei der registrierten Subdomain hinter dem „Direct URL“ Link. Einfach den String nach „<http://freedns.afraid.org/dynamic/update.php?>“ rauskopieren und in das Key-Feld bei DNS-O-MATIC eintragen und speichern.

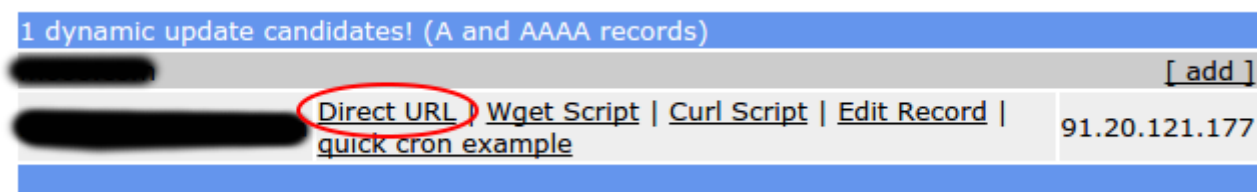


Abb.1

Um die IP bei Tunnelbroker auch updaten zu können braucht man folgende Infos von deren Seite:

- Username (nicht die lange User-ID, die im Inhaltsteil nach dem Login angezeigt wird)

- Passwort ist der Update Key auf der Detailseite des Tunnels (Advanced), alternativ das Tunnelbroker Passwort, wenn man keinen Update Key hat
- den DNS-Namen des Tunnel. Dieser wird z.B. auf der Übersichtsseite angezeigt und hat folgendes Format: <user>-<index>.tunnel.<tunnel-server>.<datacenter>.ipv6.he.net



Hurricane Electric Free IPv6 Tunnel Broker									
<b>Account Menu</b> <a href="#">Main Page</a> <a href="#">Account Info</a> <a href="#">Logout</a>	Name: [REDACTED] User ID: [REDACTED] <b>Tunnel Broker News:</b> ±Update - 6 January 2015 [January 06, 2015]  ±Update - 25 November 2014 [November 25, 2014]  ±Additional API utilities [February 14, 2014]  ±Authentication updates [January 31, 2014]  ±Server Maintenance - 16 Nov 2013 [UPDATEx2] [November 14, 2013]								
<b>User Functions</b> <a href="#">Create Regular Tunnel</a> <a href="#">Create BGP Tunnel</a> <a href="#">IPv6 Portscan</a>	<table border="1"> <thead> <tr> <th>Tunnel [ 1 / 5 ]</th> <th>Routed /64</th> <th>Routed /48</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[REDACTED]</td> <td>[REDACTED]</td> <td>[REDACTED]</td> <td>None</td> </tr> </tbody> </table>	Tunnel [ 1 / 5 ]	Routed /64	Routed /48	Description	[REDACTED]	[REDACTED]	[REDACTED]	None
Tunnel [ 1 / 5 ]	Routed /64	Routed /48	Description						
[REDACTED]	[REDACTED]	[REDACTED]	None						



Abb.2

**Account Menu**

Main Page  
Account Info  
Logout

**User Functions**

Create Regular Tunnel  
Create BGP Tunnel  
IPv6 Portscan

**Tunnel Details**

IPv6 Tunnel
Example Configurations
Advanced

Tunnel ID: [REDACTED]

Creation Date: [REDACTED]

Description: [REDACTED]

[Delete Tunnel](#)

May 13, 2015

---

**IPv6 Tunnel Endpoints**

Server IPv4 Address:

Server IPv6 Address:

Client IPv4 Address:

Client IPv6 Address:

2. [REDACTED]

3. [REDACTED]

87.158.137.201

4. [REDACTED]

---

**Routed IPv6 Prefixes**

Routed /64:

Routed /48:

1. [REDACTED]

[Assign /48](#)

---

**Available DNS Resolvers**

Anycasted IPv6 Caching Nameserver:

Anycasted IPv4 Caching Nameserver:

5. [REDACTED]

2001:470:20::2

74.82.42.42

---

**rDNS Delegations** [Edit](#)

rDNS Delegated NS1:

rDNS Delegated NS2:

rDNS Delegated NS3:

rDNS Delegated NS4:

rDNS Delegated NS5:

Abb.3

Damit wäre der Dyndns-Service vorbereitet und kann auf dem Router konfiguriert werden (pppoe0 ist mein WAN-Interface):

```
admin@gate:~$ configure

set service dns dynamic interface pppoe0 service dyndns host-name all.dnsomatic.com
set service dns dynamic interface pppoe0 service dyndns login <dns-o-matic login>
set service dns dynamic interface pppoe0 service dyndns password <dns-o-matic password>
set service dns dynamic interface pppoe0 service dyndns protocol dyndns2
set service dns dynamic interface pppoe0 service dyndns server updates.dnsomatic.com

commit
save
exit
```

Will man gleich testen, ob das klappt, erzwingt man am einfachsten einen Reconnect:

```
admin@gate:~$ disconnect interface pppoe0
admin@gate:~$ connect interface pppoe0
```

Auf der DNS-O-MATIC-Seite sollten nun beide Services mit einem grünen Daumen-hoch markiert und die aktuelle IP hinterlegt sein.

Bevor der Tunnel konfiguriert wird, empfiehlt es sich eine passende Firewall zu konfigurieren. Mit LOCAL\_TUN0 ist dabei ausgehender Datenverkehr (LAN ? WAN), TUN0\_LOCAL eingehend zum Router, TUN0\_IN eingehend ins lokale Netz (FORWARD) gemeint. Das kann dann z.B. so aussehen:

```
admin@gate:~$ configure

set firewall all-ping enable
set firewall broadcast-ping disable
set firewall ipv6-name LOCAL_TUN0 default-action accept
set firewall ipv6-name LOCAL_TUN0 description 'outgoing to IPv6 Tunnel'
set firewall ipv6-name TUN0_IN default-action drop
set firewall ipv6-name TUN0_IN description 'IPv6 Tunnel to local net'
set firewall ipv6-name TUN0_IN enable-default-log
set firewall ipv6-name TUN0_IN rule 1 action accept
set firewall ipv6-name TUN0_IN rule 1 description 'Must be allowed or MTU discovery will break'
set firewall ipv6-name TUN0_IN rule 1 icmpv6 type packet-too-big
set firewall ipv6-name TUN0_IN rule 1 protocol icmpv6
set firewall ipv6-name TUN0_IN rule 2 action accept
set firewall ipv6-name TUN0_IN rule 2 description 'Allow ICMP echo reply'
set firewall ipv6-name TUN0_IN rule 2 icmpv6 type pong
set firewall ipv6-name TUN0_IN rule 2 limit burst 1
set firewall ipv6-name TUN0_IN rule 2 limit rate 50/minute
set firewall ipv6-name TUN0_IN rule 2 protocol icmpv6
set firewall ipv6-name TUN0_IN rule 3 action accept
set firewall ipv6-name TUN0_IN rule 3 description 'May cause fragmentation issues otherwise'
set firewall ipv6-name TUN0_IN rule 3 icmpv6 type time-exceeded
set firewall ipv6-name TUN0_IN rule 3 protocol icmpv6
set firewall ipv6-name TUN0_IN rule 4 action accept
set firewall ipv6-name TUN0_IN rule 4 description 'Allow established/related'
set firewall ipv6-name TUN0_IN rule 4 protocol all
set firewall ipv6-name TUN0_IN rule 4 state established enable
set firewall ipv6-name TUN0_IN rule 4 state invalid disable
set firewall ipv6-name TUN0_IN rule 4 state new disable
set firewall ipv6-name TUN0_IN rule 4 state related enable
set firewall ipv6-name TUN0_IN rule 5 action drop
set firewall ipv6-name TUN0_IN rule 5 description 'Drop invalid state'
set firewall ipv6-name TUN0_IN rule 5 log disable
set firewall ipv6-name TUN0_IN rule 5 protocol all
set firewall ipv6-name TUN0_IN rule 5 state established disable
set firewall ipv6-name TUN0_IN rule 5 state invalid enable
set firewall ipv6-name TUN0_IN rule 5 state new disable
set firewall ipv6-name TUN0_IN rule 5 state related disable
set firewall ipv6-name TUN0_LOCAL default-action drop
set firewall ipv6-name TUN0_LOCAL description 'IPv6 Tunnel to EdgeRouter'
set firewall ipv6-name TUN0_LOCAL enable-default-log
set firewall ipv6-name TUN0_LOCAL rule 1 action accept
set firewall ipv6-name TUN0_LOCAL rule 1 description 'Must be allowed or MTU discovery will break'
set firewall ipv6-name TUN0_LOCAL rule 1 icmpv6 type packet-too-big
set firewall ipv6-name TUN0_LOCAL rule 1 protocol icmpv6
set firewall ipv6-name TUN0_LOCAL rule 2 action accept
set firewall ipv6-name TUN0_LOCAL rule 2 description 'Allow ICMP echo reply'
set firewall ipv6-name TUN0_LOCAL rule 2 icmpv6 type pong
set firewall ipv6-name TUN0_LOCAL rule 2 limit burst 1
set firewall ipv6-name TUN0_LOCAL rule 2 limit rate 50/minute
set firewall ipv6-name TUN0_LOCAL rule 2 protocol icmpv6
set firewall ipv6-name TUN0_LOCAL rule 3 action accept
set firewall ipv6-name TUN0_LOCAL rule 3 description 'May cause fragmentation issues otherwise'
set firewall ipv6-name TUN0_LOCAL rule 3 icmpv6 type time-exceeded
set firewall ipv6-name TUN0_LOCAL rule 3 protocol icmpv6
set firewall ipv6-name TUN0_LOCAL rule 4 action accept
set firewall ipv6-name TUN0_LOCAL rule 4 description 'Allow established/related'
set firewall ipv6-name TUN0_LOCAL rule 4 protocol all
set firewall ipv6-name TUN0_LOCAL rule 4 state established enable
set firewall ipv6-name TUN0_LOCAL rule 4 state invalid disable
set firewall ipv6-name TUN0_LOCAL rule 4 state new disable
set firewall ipv6-name TUN0_LOCAL rule 4 state related enable
set firewall ipv6-name TUN0_LOCAL rule 5 action drop
set firewall ipv6-name TUN0_LOCAL rule 5 description 'Drop invalid state'
set firewall ipv6-name TUN0_LOCAL rule 5 log disable
set firewall ipv6-name TUN0_LOCAL rule 5 protocol all
set firewall ipv6-name TUN0_LOCAL rule 5 state established disable
set firewall ipv6-name TUN0_LOCAL rule 5 state invalid enable
set firewall ipv6-name TUN0_LOCAL rule 5 state new disable
set firewall ipv6-name TUN0_LOCAL rule 5 state related disable
```

```

set firewall ipv6-receive-redirects disable
set firewall ipv6-src-route disable
set firewall ip-src-route disable
set firewall log-martians enable
set firewall receive-redirects disable
set firewall send-redirects enable
set firewall source-validation disable
set firewall syn-cookies enable

commit
save
exit

```

Für die Konfiguration des Tunnels werden noch folgende Daten von der Tunnelbroker Detailseite benötigt. HE gibt auch hier ein komplettes /64er als Transfernetz heraus:

- Client IPv6 Address (endet auf ::2/64, Abb.4.4)
- Server IPv4 Address (Abb.4.2)
- Routed IPv6 Prefixes (Abb.4.1)
- die eigene interne LAN-IP

Achtung: Die IPv6 Tunnel Endpunkte und der geroutete IPv6-Prefix sind unterschiedlich! Auf der Tunnelbroker-Seite sind diese deswegen auch dick hervorgehoben. Das muss bei der weiteren Konfiguration beachtet werden.



Account Menu	Tunnel Details
<b>Main Page</b> <b>Account Info</b> <b>Logout</b>	<div> IPv6 Tunnel Example Configurations Advanced </div> <div> <b>Tunnel Options</b> <p>These settings will not need changing under normal circumstances.</p> <p>MTU: (Default) 1480 <input type="text"/> <input type="button" value="Update"/></p> <p> Update Key: <input type="text"/></p> </div> <div> <b>HE Dynamic DNS Settings</b> <p>These settings will be used to automatically update the hostname below whenever you update your tunnel's IPv4 endpoint using the DynDNS-compatible mechanism at <a href="https://ipv4.tunnelbroker.net/nic/update">https://ipv4.tunnelbroker.net/nic/update</a>. This is only for hostnames set dynamic with dns.he.net.</p> <p>Last Status: N/A @ N/A</p> <p>Hostname: <input type="text"/></p> <p>API Key: <input type="text"/></p> <p><input type="button" value="Clear"/> <input type="button" value="Refresh"/></p> <p><input type="button" value="Save"/></p> </div>

Abb.4

Die Konfiguration des Tunnels sieht dann so aus:

```

admin@gate:~$ configure

set interfaces tunnel tun0 address '<Client IPv6 Address>'
set interfaces tunnel tun0 description 'he.net IPv6 Tunnel'

```

```

set interfaces tunnel tun0 encapsulation sit
set interfaces tunnel tun0 firewall in ipv6-name TUN0_IN
set interfaces tunnel tun0 firewall local ipv6-name TUN0_LOCAL
set interfaces tunnel tun0 firewall out ipv6-name LOCAL_TUN0
set interfaces tunnel tun0 local-ip <interne LAN-IP>
set interfaces tunnel tun0 multicast disable
set interfaces tunnel tun0 remote-ip <Server IPv4 Address>
set interfaces tunnel tun0 ttl 255

commit
save
exit

```

Der Tunnel sollte nun schon aufgebaut sein. Die Firewall ist aktiviert.

```

admin@gate:~$ show interfaces tunnel tun0 brief
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface      IP Address      S/L  Description
-----
tun0           <ipv6 network>/64  u/u  he.net IPv6 Tunnel

```

„u/u“ bedeutet State und Link sind up, der Tunnel ist also funktional.

Damit das IPv6 Netz auch im LAN verwendbar wird, fehlen noch ein paar Dinge.

Erstmal wird eine IPv6-IP (die erste freie IP (::1 ist HE, ::2 deine Seite des Tunnels) aus deinem IPv6-Netz (::3) auf das LAN-Interface des Routers gelegt (bei mir ist das switch0). Danach werden noch die Default-Route für IPv6, DHCPv6 (soll nur für Nameserver, Searchdomain und NTP verwendet werden) und die Router-Adverts (d.h. die Clients konfigurieren sich ihre IPv6-Adresse anhand ihrer MAC-Adresse selbst) konfiguriert. Der Name-Server kommt von der Tunnelbroker Webseite (Abb.4.5).

```

admin@gate:~$ configure
set protocols static interface-route6 '::/0' next-hop-interface tun0

set service dhcpv6-server shared-network-name LAN1-IPv6 subnet '<ipv6 routed network>' domain-se
set service dhcpv6-server shared-network-name LAN1-IPv6 subnet '<ipv6 routed network>' name-serv
set service dhcpv6-server shared-network-name LAN1-IPv6 subnet '<ipv6 routed network>' snntp-serv

set interfaces switch switch0 address '<ipv6 routed network>::1/64'
set interfaces switch switch0 description LAN1
set interfaces switch switch0 dhcpv6-options parameters-only
set interfaces switch switch0 ipv6 dup-addr-detect-transmits 1
set interfaces switch switch0 ipv6 router-advert cur-hop-limit 64
set interfaces switch switch0 ipv6 router-advert default-preference high
set interfaces switch switch0 ipv6 router-advert link-mtu 1280
set interfaces switch switch0 ipv6 router-advert managed-flag false
set interfaces switch switch0 ipv6 router-advert max-interval 600
set interfaces switch switch0 ipv6 router-advert other-config-flag true
set interfaces switch switch0 ipv6 router-advert prefix '<ipv6 routed network>' autonomous-flag
set interfaces switch switch0 ipv6 router-advert prefix '<ipv6 routed network>' on-link-flag tru
set interfaces switch switch0 ipv6 router-advert prefix '<ipv6 routed network>' valid-lifetime 2
set interfaces switch switch0 ipv6 router-advert reachable-time 0
set interfaces switch switch0 ipv6 router-advert retrans-timer 0
set interfaces switch switch0 ipv6 router-advert send-advert true

commit
save
exit

```

Damit sollten sich nun die IPv6-fähigen Clients in eurem LAN bereits ihre Adresse geholt/erzeugt haben und über den Tunnel nach draußen gehen können.

Hier ein paar Möglichkeiten den IPv6-Verkehr zu beobachten / debuggen:

- Informationen zum Tunnel: show interfaces tunnel tun0 brief
- Packetdump: show interfaces tunnel tun0 capture
- Bytes, Packets und Fehlerraten zeigen: show interfaces tunnel detail



- Traceroute: traceroute6 [www.heise.de](http://www.heise.de)
- Ping: ping6 [www.heise.de](http://www.heise.de)

# OpenVPN ohne Konfigurationsänderung neu starten

Standardmäßig sorgt ein `reset openvpn interface vtun1` für eine Neuaushandlung der Verbindung, Konfigurationsdateien werden dabei nicht neu eingelesen. Das mag ganz hilfreich sein, wenn sich z.B. die eigene WAN-IP geändert hat und man die Verbindung schnell wieder online haben will.

Ein kompletter Neustart lässt sich mit Boardmitteln nicht ausführen. Mit einer kleinen Änderung in der VPN-Konfiguration lässt sich aber das reset-Kommando entsprechend umbiegen:

```
configure
set interfaces openvpn vtun0 openvpn-option '--remap-usrl SIGHUP'
commit
save
```

Aus der openvpn Hilfe dazu: `--remap-usrl s : On SIGUSR1 signals, remap signal (s='SIGHUP' or 'SIGTERM')`

EdgeMAX

# tcpdump zu lokalem Wireshark umleiten

tcpdump muss auf dem EdgeRouter installiert sein. tcpdump wird per SSH von remote gestartet und das Capture auf die Standardausgabe gelenkt. Gleichzeitig wird auf dem lokalen Host ein wireshark gestartet, das darüber seinen Input bezieht.

```
$ ssh admin@edgerouter 'sudo tcpdump -f -i switch0 -w -' | wireshark-gtk -k -i -
```

Je nach Desktop muss wireshark-gtk oder wireshark-qt verwendet werden.

Mittels plink aus dem Putty-Paket funktioniert das auch für Windows (pagent sollte laufen und der passende Key geladen sein):

```
plink -batch -l admin -P 22 edgerouter sudo /usr/bin/tshark -i switch0 -w - | "c:\Program Files
```

# USB-Flashspeicher im EdgeRouter ersetzen

Diese Anleitung wurde getestet mit einem Ubiquiti EdgeMax - EdgeRouter PoE - ERPoe-5

Vorbereitung und Voraussetzungen:

- kompatibler USB Stick (ich habe den Kingston DataTraveler DTSE9H 8GB, ca. 8€, verwendet)
  - muss recht klein sein
  - Liste von möglichen Alternativen: <https://community.ubnt.com/t5/EdgeRouter/List-of-Compatible-USB-drives/m-p/1185171/highlight/true#M57699>
- Runterladen EdgeMax Rescue Kit: <http://packages.vyos.net/tools/emrk/0.9c/emrk-0.9c.bin>
- Runterladen aktuelle EdgeRouter Firmware: <https://www.ui.com/download/edgemax> (ich habe es mit Version ER-e100.v2.0.0.5155284 getestet)
- Webserver, z.B. über das SimpleHTTP Python Modul
- Minicom für die serielle Kommunikation
- Serielles Kabel RJ-45 für den Konsolenport (das hier sollte funktionieren: <https://www.amazon.de/Unilike-Cisco-Konsolenkabel-Ftdi-Chip-RJ45-Kabel-Windows/dp/B01NGYKRCS/>)

Schritte:

1. USB Stick mit FAT32 formatieren und mounten
2. Die Datei emrk-0.9c.bin auf den Stick kopieren, dann unmounten
3. Die 3 kleinen Schrauben des EdgeRouter Gehäuses lösen und den Deckel runterschieben
4. Den alten USB-Stick mit dem Neuen ersetzen
5. Serielle Konsole anschließen und minicom starten
  - Port /dev/ttyUSB0
  - Baud 115200
  - 8 Bits
  - Parität aus (N)
  - Stopbits 1
  - Flusskontrolle aus (rtscts + xonxoff)
6. EdgeRouter anstecken und Booten lassen, es sollte folgender Prompt zu sehen sein „Oxteon  
ubnt\_e100#“
7. Bei den Bootmeldungen den USB Teil kontrollieren, sollten dort irgendwelche Warnungen auftauchen könnte der Stick nicht ordentlich funktionieren
8. Mit „fatload usb 0 \$loadaddr emrk-0.9c.bin“ das Rescue Kit laden
9. und mit „bootoctlinux \$loadaddr“ booten
10. Die Warnung mit „yes“ bestätigen und das Netzwerk konfigurieren (entweder DHCP oder statisch möglich, ich nutzte die statische Variante, da der EdgeRouter bei mir DHCP macht)
11. Den EdgeRouter via LAN mit dem Computer verbinden, auf dem die Downloads liegen, und via Ping die konfigurierte IP testen
12. Die heruntergeladene Firmware via HTTP verfügbar machen, am einfachsten geht das mit Python. Auf dem PC eine Shell öffnen und in den Ordner mit der heruntergeladenen Firmware wechseln. Über „python -m SimpleHTTPServer“ den Webserver starten (default ist Port 8000)
13. Das Script „emrk-reinstall“ starten, die Abfrage mit „yes“ bestätigen. Das Script bereitet nun den USB-Stick für die EdgeOS Firmware vor
14. Als image URL nehmen wir unseren Webserver mit der Firmware, z.B. „<http://192.168.1.10:8000/ER-e100.v2.0.0.5155284.tar>“

15. Die Firmware wird nun auf den Stick kopiert und entpackt. Am Ende sollte „Installation finished, Please reboot your router“ erscheinen. Via reboot neu starten und den Bootvorgang auf der Konsole beobachten
16. Wenn das System vollständig hochgefahren ist, kann man sich via serieller Konsole oder SSH (192.168.1.1) einloggen. Standardbenutzer und -passwort ist: ubnt
17. Von der alten Konfiguration hat man hoffentlich immer ein aktuelles Backup. Dieses kann mit SCP auf den Router kopiert werden („`scp config.boot ubnt@192.168.1.1:`“). Auf dem Router dann über „`configure`“ in den Konfigurationsmodus wechseln und die Datei laden: „`load /home/ubnt/config.boot`“. Via „`commit`“ kann die Konfiguration angewendet werden. Dabei sieht man dann auch gleich eventuelle Fehler, z.B. fehlende SSL-Zertifikate für VPN-Verbindungen. Eventuell vorhandene Konfigurationen und Skripte in `/config/user-data` oder `/config/scripts` müssen ebenfalls aus einem Backup wiederhergestellt werden

Hier sind noch zwei weitere Varianten wie man die Firmware auf einen neuen USB-Stick bekommen kann:

- via USB: <https://austinrobertson.com/blog/2016/04/07/how-to-restore-an-edgerouter-from-usb/>
- via TFTP: <https://community.ubnt.com/t5/EdgeRouter/EdgeMax-rescue-kit-now-you-can-reinstall-EdgeOS-from-scratch/m-p/514857/highlight/true#M12098>