Kurzfassung

Migration von IPv4 nach IPv6 und deren Auswirkungen auf unixoide Systeme in einem Unternehmen.

Bei einer Umstellung der IT Infrastruktur von IPv4 auf die Version IPv6 müssen im Vorhinein viele Faktoren beachtet werden um ungeplante und unberücksichtigte negative Folgen zu minimieren.

Um alle Probleme und Auswirkungen solch einer Umstellung aufzuzeigen wurde eine KMU IPv4 IT Infrastruktur mit Hilfe von verschiedenen Virtualisierungstechnologien geschaffen und auf IPv6 umgestellt.

Diese Bachelorarbeit befasst sich mit den Auswirkungen auf unixoide Systeme. Anhand der Umstellung eines Linux Rootservers von IPv4 auf einen Dual-Stack Betrieb wird gezeigt zu welchen Problemen es kommen kann und was für eine reibungslose Umstellung zu beachten ist. Die einzelnen Schritte werden beschrieben und es werden weiterführende Quellen für eigene Recherchen angegeben.

Die Auswirkungen auf Microsoft Windows Systeme werden in der Bachelorarbeit von Herrn Tarik Basalic mit dem Titel „Migration von IPv4 nach IPv6 und deren Auswirkungen auf Microsoft Windows Systeme in einem Unternehmen“ beschrieben.

Inhaltsverzeichnis

1 Einleitung

1.1 Problemstellung / Aufgabenstellung

Da die Nachfrage nach öffentlichen IPv4 Adressen immer größer und größer wird und fast alle IPv4 Adressen vergeben sind, wurde IPv6 bereits vor einiger Zeit entwickelt.
Der IPv6 Adressraum ist wesentlich größer und so sollte es in nächster Zukunft keine Engpässe mehr bei der Vergabe von öffentlichen IP Adressen geben.

Jedoch wird die Einführung von IPv6 von allen Beteiligten möglichst lange hinaus gezögert, da die Umstellung von vorhandener Infrastruktur auf das neue Internet Protokoll mit Kosten verbunden ist. Ein weiterer Grund für die Verzögerungen bei der Einführung von IPv6 ist, das sich die Umstellung für Serviceanbieter erst lohnt wenn es Kunden gibt die IPv6 verwenden. Auf der anderen Seite haben ISP keinen Grund ihren Kunden IPv6 anzubieten, da es noch kaum Services im Internet gibt die IPv6 unterstützen oder gar voraussetzen.

Auch wenn IPv6 mit dem Gedanken entwickelt wurde einen reibungslosen Übergang von IPv4 auf IPv6 zu ermöglichen bringt die neue Technologie einige Veränderungen mit sich. Bereits verwendete Hard- und Software muss IPv6 tauglich sein und ganze Systeme müssen mit dem neuen Internet Protocol v6 kommunizieren können.

Aufgabe ist es, die Hard- und Software einer KMU IT Infrastruktur auf IPv6 umzustellen, wobei diese Bachelorarbeit nur die softwareseitigen Umstellungen behandelt.

1.2 Zielsetzung

Ziel dieser Bachelorarbeit ist es, wichtige Erkenntnisse bei einer Umstellung des Internet Protokolls zu gewinnen. Diese Bachelorarbeit bezieht sich dabei auf die Umstellung von Unix ähnlichen Systemen wie Linux. Die Umstellung von Windows Systemen in einem KMU Unternehmen wird in der bereits erwähnten Bachelorarbeit von Tarik Basalic behandelt.

Im Rahmen dieser Bachelorarbeit werden alle gewonnen Erkenntnisse zusammengefasst und abschließend analysiert. Ziel ist es, anhand dieser Bachelorarbeit, anderen IT Technikern und IT Entscheidungspersonen Erfahrungen einer solchen Umstellung näherzubringen und Sie somit bei Ihrer eigenen Umstellung von IPv4 zu IPv6 zu unterstützen.

2 Das neue Internet Protokoll[1]

Das Internet Protocol ist das wichtigste Netzwerkprotokoll. In der Praxis kommt fast ausschließlich das Internet Protocol in der Version 4 zum Einsatz. Wegen der zu hohen Nachfrage nach IP Adressen wurde das Internet Protocol in der Version 6 entwickelt.

IP ist ein verbindungsloses Protokoll, das erst durch die Verwendung von TCP (Transportschicht) den Endgeräten Zustände ermöglicht. Im Internet Protocol wird bei der Adressvergabe zwischen öffentlich und privaten IP Adressen unterschieden. Jedes Endgerät oder Client bekommt in einem Netzwerk eine eindeutige IP Adresse, um mit anderen Clients im Netzwerk kommunizieren zu können. Die Internet Assigned Numbers Authority (IANA) regelt weltweit die Vergabe der öffentlichen IP Adressen und vergibt den Internetprovidern sogenannte Subnetze, die wiederum vom Internetprovider aufgeteilt werden und deren Kunden können dann IP Adressen oder IP Subnetze anfordern.

2.1 Vergleich der Internet Protokolle

Der für die meisten wichtigste Unterschied zwischen IPv4 und IPv6 ist der Adressraum. Mit IPv6 sind unvorstellbar viel mehr Adressen möglich wie mit IPv4. Auch wenn in der Praxis viel Adressen ungenutzt bleiben werden, da fast alle Netze eine minimale Größe von 64bit haben werden.

IPv4:	2^{32} (4,3 Milliarden) unterschiedliche Adressen möglich
IPv6:	2^{128} (340 Sextillionen) unterschiedliche Adressen möglich

Die Adresslänge und Schreibweise ist auch ein Merkmal das vielen Technikern Sorgen bereitet. Die IPv4 Adresse ist mit ihrer Schreibweise, als Dezimalzahl die durch Punkte getrennt ist, noch relativ einfach zu merken. Die IPv6 Adresse wird hingegen als Hexadezimalzahl mit 8 Blöcken die durch Doppeltpunkte getrennt sind, geschrieben und lässt sich nicht mehr so einfach merken.

IPv4:	207.142.131.235	32 Bit Adressen
IPv6:	2001:db8::1428:57ab	128 Bit Adressen

[1] IPv6 wird im RFC2460 beschrieben, welches unter http://tools.ietf.org/html/rfc2460 zu finden ist.

Außerdem lassen sich Adressen mit einfachen Regeln abkürzen und somit kann auch eine IPv6 Adresse so einfach zu lesen und so kurz sein wie eine IPv4 Adresse.

IPv4:	4 Blöcke mit 8Bit Dezimalzahlen durch einen Punkt getrennt
IPv6:	8 Blöcke mit 16Bit Hexzahlen durch einen : Punkt getrennt

Ein Problem bei der Schreibweise einer IP Adresse mit Doppelpunkten ist das bei der Verwendung in einer URL der Doppelpunkt für den Port nicht mehr einfach von der IP Adresse unterschieden werden kann. Deshalb muss die IPv6 Adresse, wenn ihr ein Port beigefügt wird mit Eckigen Klammern umgeben werden[2].

[2001:db8::1428:57ab]:80

2.2 Vorteile von IPv6

Trotz einiger Nachteile und Bedenken über die Sicherheit mit IPv6 überwiegen die Vorteile bei weiten, abgesehen davon dass es keine Alternativen gibt.

- Anzahl der verfügbaren Adressen
- IPv6 lässt sich dank seines einfacheren Headers effektiver routen.
- IPv6 Geräte unterstützen standardmäßig IPSec
- NAT und PAT wird nicht mehr benötigt
- Automatische Adresszuweisung ohne DHCP Server
- Broadcast werden durch Multicasts ersetzt

[2] Der Umgang mit IPv6 Adressen in URLs wird im RFC2732 beschrieben und ist unter http://www.ietf.org/rfc/rfc2732.txt zu finden.

2.3 Adressen und Netze

2.3.1 Adressbereiche[3]

Die IPv6 Adressen lassen sich grob in

- Unicast Adressen,
- Multicast Adressen und
- Anycast Adressen

einteilen. Über Unicast Adressen werden einzelne Host bzw Interfaces angesprochen. Multicast Adressen sprechen mehrere Hosts oder Klassen von Geräten, wie Router oder DHCP Server an. Anycast Adressen sprechen immer den nächsten Host mit dieser Adresse an. Broadcast Adressen wie noch in IPv4 gibt es nichtmehr, stattdessen werden Multicast Adressen verwendet um die gewünschte Gruppe von Hosts anzusprechen.

Die wichtigste Gruppe der Adressen sind die Global-Unicast-Adressen. Diese Adressen sind vergleichbar mit den globalen IPv4 Adressen. Die Adressen sind weltweit eindeutig, jeder Host mit einer solchen Adresse kann über das Internet kommunizieren. Alle Globale-Unicast-Adressen kommen aus dem Bereich *2000::/3* und werden von der IANA vergeben.

Eine weitere wichtige Gruppe an Adressen sind die Link-Local-Adressen, die innerhalb eines Links gültig sind und nicht geroutet werden. Praktisch jeder Host erzeugt sich zuerst automatisch eine Link-Local-Adresse und kommuniziert mit dieser mit den anderen IPv6 Geräten in seinem Netzwerk. Über diese Adresse können anschließend auch die Global-Unicast-Adresse vom Router oder vom DHCPv6 Server abgefragt werden. Die Adressen kommen aus dem Bereich *FE80:: 10*.

IPv6 besitzt auch schon wie IPv4 private Adressen die nicht im Internet geroutet werden, diese Adressen sind Unique-Local-Adressen und kommen aus dem Bereich FD00:: /8.

Eine sehr wichtige Adresse ist die Loopback-Adresse, die bei IPv6 aus 127 binären Nullen und einer Eins besteht: *:: 1*

[3] Eine Übersicht über die Adressbereiche bietet die IANA auf ihrer Homepage: http://www.iana.org/assignments/ipv6-address-space/ipv6-address-space.xml

2.3.2 Konfiguration der Hostadressen

Adressen können auf den Host auf verschiedene Arten konfiguriert werden. IPv6 bietet die Möglichkeit der Autokonfiguration[4], bei der aus der MAC Adresse der Hostteil der Adresse erzeugt wird.

Eine weitere Möglichkeit ist die Zuteilung der Adresse per DHCPv6 Server, die Funktioniert Großteils wie schon bei IPv4 mit dem Unterschied das wenn der DHCPv6 Server IPv6 Adressen vergibt er sich im statefull Mode befindet. Der zweite Modus eines DHCPv6 Server wäre der stateless Mode bei dem nur Informationen über das Netzwerk wie DNS Server oder Domainnamen vergeben werden, jedoch keine IPv6 Adressen.

Die einfachste Möglichkeit ist die statische Konfiguration auf dem Hostinterface. Dabei kann entweder die gesamt IPv6 Adresse statisch eingestellt werden oder nur der es wird nur der Netzwerkteil eingestellt und der Rest der Adresse wird automatisch aus der MAC- Adresse generiert.

Für Server kommt meist nur die statische Konfiguration in Frage, da die Zahl der Server meist überschaubar ist und die Adressen sich nicht ändern sollen. Auch können durch die statische Konfiguration den Servern einfache und leicht zu merkende Adressen vergeben werden.
IPv6 bietet den Vorteil dass ein Interface mehrere verschiedene IP Adressen besitzen kann. Deshalb muss nicht, wie bei IPv4, für jede Adresse ein virtuelles Interface erzeugt werden. Dies und die große Menge an Adressen bietet zum Beispiel bei Webserver einen Vorteil, da jetzt jeder Virtuelle Host (jede Domain) auf dem Server ihre eigene IPv6-Adresse erhalten kann.

2.3.3 IPv6 Präfixe / Netze

Mit Netze und Subnetze kann in IPv6 genau wie in IPv4 gearbeitet werden. Ein Unterschied ist die Schreibweise, wo bei IPv4 meist die Größe des Netzes mit der Netzmaske angegeben wird, wird unter IPv6 nur mehr die Präfixlänge nach der Netz-Adresse mit einem / getrennt angegeben. Außerdem sind in IPv6 alle Netze Klassenlos.

2001:0:0:1::/64
2001:0db8:85a3:08d3:1319:8a2e:0370:7347/64

[4] Die IPv6 Autokonfiguration wird im RFC4863 beschrieben: http://tools.ietf.org/html/rfc4862

Für Geschäftskunden und auch vielleicht für Privatkunden werden von den ISP Netze in der Größe von /48 oder /56 vergeben. Dieses Netz kann dann von den Unternehmen noch in weitere Subnetzte zerteilt werden. Jedoch sollte der Hostanteil der IPv6 Adresse genau 64bit betragen, da damit das Autokonfigurationsfeature von IPv6 genutzt werden kann.

2.4 Migrationsstrategien

Für die Migration von IPv4 auf IPv6 gibt es mehrere Strategien die einen Reibungslosen Umstieg ermöglichen sollen. Bei der Umstellung eines Services wie eines Webservers müssen dabei jedoch andere Dinge beachtet werden, wie bei der Umstellung eines internen Unternehmensnetzwerks.

Bei einem internen Netzwerk kommt es darauf an alle vorhandenen Geräte langsam auf IPv6 zu migrieren, dabei kann der Administrator überschauen welche Geräte in Netzwerk vorhanden sind und welche mit welchen kommunizieren müssen.

Bei der Umstellung eines Internet Services können die Clients nicht kontrolliert und auf den Server abgestimmt werden. Der Service muss von Clients die nur IPv4 sprechen, von Clients die sowohl IPv6 als auch IPv4 sprechen und auch von Client die nur IPv6 sprechen erreichbar sein. Dabei sind aber nicht nur die direkte Kommunikation zwischen dem Client und dem Server über beide Protokolle zu gewährleisten sondern auch die Namensauflösung und andere Dienste.

Dabei ist es am Einfachsten wenn alle beteiligten Server und Services über Dual-Stack sowohl über IPv4 als auch über IPv6 erreichbar sind. Der Client kann dann sein bevorzugtes Protokoll wählen.

Eine andere Möglichkeiten wäre das vorschalten eines Proxys der alle IPv6 Anfragen in IPv4 Anfragen umwandelt und an den vorhanden Server weiterleitet. Dies erspart größtenteils die Umstellung des Servers wird aber bei steigender Zahl von IPv6 Zugriffen problematischer, da der Proxy-Server mehr und mehr Rechenleistung dafür aufwenden muss.

Man sollte sich nicht die Gelegenheit entgehen lassen möglichst früh über eine Nachhaltige Umstellung auf IPv6 nachzudenken um spätere Probleme und Zeitdruck zu vermeiden. Empfehlenswert ist es die Umstellung mit einer Aktualisierung der Serverhardware oder Serversoftware zu koppeln um die Kosten zu senken und die Downtime zu minimieren.

3 KMU IT Infrastruktur

Anhand einer KMU IT Infrastruktur wird gezeigt, wie man ein IPv4 Netzwerk in ein IPv6 Netzwerk migriert. Diese KMU IT Infrastruktur besteht aus einem gehosteten Teil und einem lokal selbst verwalteten Teil. In dieser Bachelorarbeit wird gezeigt, wie der gehostete Teil um konfiguriert werden muss, um einen Dualbetrieb ermöglichen zu können. Dualbetrieb bedeutet, dass alle Komponenten sowohl über ein IPv4 Adresse als auch eine IPv6 Adresse besitzen und somit über beide IP Protokolle kommunizieren können. Es gibt mehrere Strategien bzw. Ansätze um ein Netzwerk auf IPv6 zu migrieren. Hier ist auf Dualbetrieb entscheiden worden, da die Services auf den Webservern weiter IPv4 und zusätzlich IPv6 kompatibel sein müssen. Andere Migrationsstrategien werden in der bereits erwähnten Bachelorarbeit von Tarik Basalic behandelt.

Dienste, die später im Dualbetrieb lauffähig sein müssen:

- HTTP / HTTPS
- FTP / SFTP
- SSH
- IMAP / POP3
- SMTP

3.1 Ausgangskonfiguration

Der gehostete Teil besteht aus einem angemieteten Rootserver von dem Hosting Betreiber Hetzner[5]. Da der Betrieb und die Wartung der darauf betriebenen Services kostengünstiger ist, hat man sich auf eine gehostete Variante der Webserver entschieden.

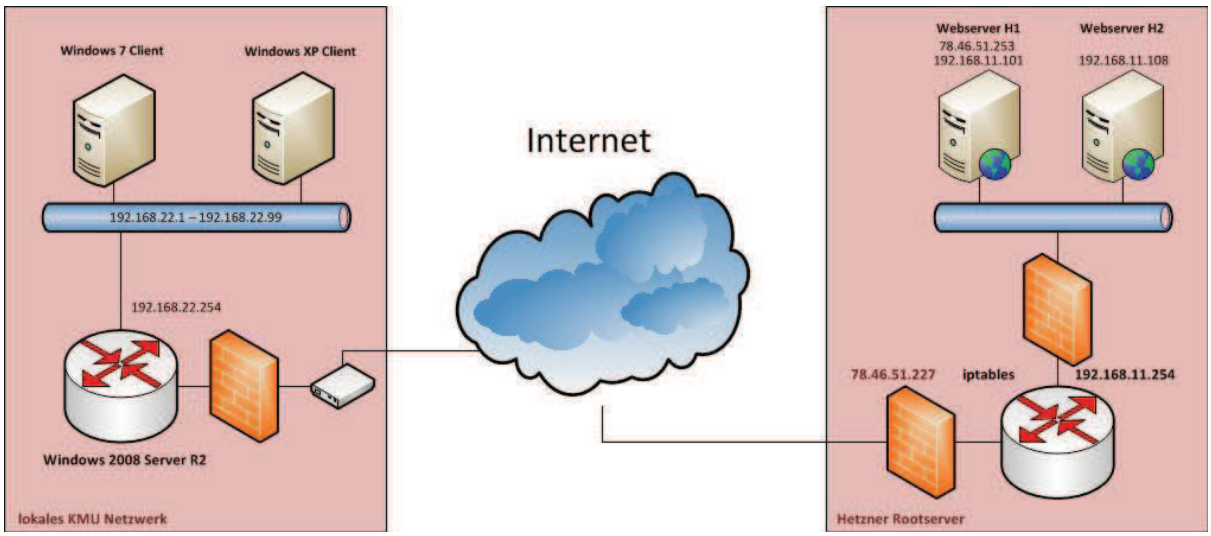

Abb. 1: Übersicht der Netzwerkstruktur

[5] Hetzner ist ein deutsches Hosting Unternehmen: http://www.hetzner.de

3.1.1 Ausgangskonfiguration des Hostsystems

Auf dem Hostsystem des Rootservers wird eine CentOS 5 Distribution[6] (5.4 Final) mit einer openVZ[7] Virtualisierungslösung verwendet. Auf dem Hostsystem laufen zwei virtuelle Maschinen auf denen Webserver betrieben werden, die nun zusätzlich IPv6 kompatibel gemacht werden müssen. Auf andere virtuelle Maschinen, die auf dem Host betrieben werden, wird nicht weiter eingegangen. Die virtuellen Maschinen, welche als Webserver verwendet werden, verwenden ebenfalls eine CentOS 5 Distribution (5.4 Final) als Betriebssystem.

Das Hostsystem dient zusätzlich als Router, der zwischen dem physikalischen Netz und den virtuellen Maschinen vermittelt, und als Firewall, welche mit netfilter (iptables) realisiert wurde.

- CentOS 5 Hostsystem (CentOS release 5.4 (Final) / 2.6.18)
- openVZ (Virtualisierung) (ovzkernel-2.6.18-164.10.1.el5.028stab067.4)
- iptables/netfilter (Firewall) (iptables-ipv6-1.3.5-5.3.el5_4.1 / iptables-1.3.5)

Das Hostsystem sowie jeder der virtuellen Server besitzen zwei virtuelle Netzwerkinterfaces. Eines davon für ein internes Netz und ein zweites für den Zugriff über eine öffentliche IP Adresse. Da die öffentlichen IPv4 Adressen begrenzt und kostenpflichtig sind, sind einige der virtuellen Maschinen nicht direkt über eine öffentliche IPv4 Adresse erreichbar. Um die Dienste dennoch auf unabhängige virtuelle Maschinen zu verteilen und über das Internet erreichbar zu machen wird PAT / NAT verwendet. Damit können die begrenzten IPv4 Adressen effektiv und kostengünstig genutzt werden.

- Hostsystem: 78.46.51.227 192.168.11.254
- Webserver 1: 78.46.51.253 192.168.11.101
- Webserver 2: 192.168.11.108

Die Firewall regelt den Datenverkehr und kann außerdem dazu genutzt werden die virtuellen Maschinen sowie die einzelnen Dienste zu protokollieren und somit den Traffic-Verbrauch zuzuordnen.

Das Hostsystem wird über die physikalische Schnittstelle eth0 mit dem Internet verbunden. Das Netzwerk an dem der Host angeschlossen ist 78.46.51.227/32. Somit leitet der Host alle Pakete über das default Gateway der mit einem eigenen Routing Eintrag angegeben ist.

[6] CentOS ist ein freies Server Betriebssystem welches zu RHEL kompatibel ist: http://www.centos.org
[7] openVZ ist eine freie Virtualisierungstechnologie für Linux: http://wiki.openvz.org/Main_Page

Dies entspricht nicht der realen Netzaufteilung wird aber aus Sicherheitsgründen vom Provider so gehandhabt.
Die virtuelle Schnittstelle venet0 kann von allen openVZ Guests für ein internes Netzwerk verwendet werden.[8]

```
eth0    Link encap:Ethernet  HWaddr 40:61:86:2B:8E:9D
        inet addr:78.46.51.227  Bcast:78.46.51.255  Mask:255.255.255.255
        inet6 addr: fe80::4261:86ff:fe2b:8e9d/64 Scope:Link
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:91175301 errors:0 dropped:0 overruns:0 frame:0
        TX packets:185541519 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:7489267004 (6.9 GiB)  TX bytes:248550131931 (231.4 GiB)
        Interrupt:82 Base address:0xa000

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:16436  Metric:1
        RX packets:45686 errors:0 dropped:0 overruns:0 frame:0
        TX packets:45686 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:2467044 (2.3 MiB)  TX bytes:2467044 (2.3 MiB)

venet0  Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
        UP BROADCAST POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
        RX packets:70209216 errors:0 dropped:0 overruns:0 frame:0
        TX packets:34495776 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:73661207464 (68.6 GiB)  TX bytes:3346114110 (3.1 GiB)
```

Abb. 2: Netzwerkkonfiguration(ifconfig) des Rootservers

Den Systemen, die mit privaten IPv4 Adressen ausgestattet sind, wird der Zugriff auf das Internet mittels Adresstausch (SNAT) ermöglicht.

```
-A POSTROUTING -s 10.10.8.0/255.255.255.0 -o eth0 -j SNAT --to-source 78.46.51.227
-A POSTROUTING -s 192.168.11.0/255.255.255.0 -o eth0 -j SNAT --to-source 78.46.51.227
```

Abb. 3: SNAT Konfiguration bei Firewall (/etc/sysconfig/iptables)

Um Zugriff auf Dienste ohne öffentliche IPv4 Adresse zu ermöglichen, wird die Ziel Adresse von einzelnen Ports auf dem Hostsystem auf IPv4 Adressen der Gastsysteme umgeschrieben (Port Forwarding).

[8] Das openVZ Netzwerkmodell wird unter folgendem Link näher erklärt: http://pve.proxmox.com/wiki/Network_Model

```
...
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 25 -j DNAT --to-
destination 192.168.11.108:25
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 80 -j DNAT --to-
destination 192.168.11.108:80
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 143 -j DNAT --to-
destination 192.168.11.108:143
...
```

Abb. 4: Port Forwarding / NAT Konfiguration bei Firewall (/etc/sysconfig/iptables)

Dienste von virtuellen Servern werden über die Forward Chain freigegeben.

```
...
-A LEGA-WEBSPACE-INPUT -p udp -m udp --dport 20 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p udp -m udp --dport 21 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p udp -m udp --dport 53 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 20000 -j ACCEPT
...
```

Abb. 5: Forward Chain bei Firewall (/etc/sysconfig/iptables)

Dienste auf dem Hostsystem werden über die INPUT / OUTPUT Chain freigegeben.

```
....
-A RH-Firewall-1-INPUT -p ah -j ACCEPT
-A RH-Firewall-1-INPUT -d 224.0.0.251 -p udp -m udp --dport 5353 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
...
```

Abb. 6: Forward Chain bei Firewall (/etc/sysconfig/iptables)

Die komplette Firewall Konfiguration ist unter 7.1 im Anhang zu finden.

3.1.2 Ausgangskonfiguration des Webserver H1 & H2

Um den Webserver zu verwalten wird die kostenlose Software Virtualmin[9] verwendet. Mit dieser ist es möglich die verschiedenen Serverdienste (Apache, MySQL, proFTPd, Postfix, uvm.) zentral über eine Weboberfläche zu verwalten. Die verwendete Version bietet bereits Unterstützung für IPv6 welche jedoch noch mit einigen Bugs behaftet ist. Deshalb werden die nötigen Konfigurationen für IPv6 direkt in den Konfigurationsfiles der einzelnen Dienste vorgenommen. Somit können die Konfigurationen auch an Webservern angewandt werden, die nicht Virtualmin verwenden.

[9] Virtualmin ist ein Verwaltungswerkzeug für Webhosting: http://www.virtualmin.com

Webserver H1:

- CentOS 5 (CentOS release 5.4 (Final))
- Apache Webserver (httpd-2.2.3-22.el5.1vm)
- MySQL Datenbankserver (mysql-server-5.0.77-4.el5_4.1)
- proFTPd FTP Server (proftpd-1.3.0a-3.el4)
- Postfix Mail Server (postfix-2.3.3-2.1.el5_2)
- Dovecot IMAP/POP3 Server (dovecot-1.0.7-7.el5)
- SSH Server (openssh-server-4.3p2-36.el5_4.3)
- Virtualmin (virtualmin-base-1.0-61.rh)

Dieser Webserver ist über die öffentliche IPv4 Adresse 78.46.51.253 und das Interface venet0:0 erreichbar. Die private IPv4 Adresse 192.158.11.101 wird über das Interface venet0:1 angesprochen.

```
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:293264 errors:0 dropped:0 overruns:0 frame:0
          TX packets:293264 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:51119875 (48.7 MiB)  TX bytes:51119875 (48.7 MiB)

venet0    Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:127.0.0.1  P-t-P:127.0.0.1  Bcast:0.0.0.0  Mask:255.255.255.255
          UP BROADCAST POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
          RX packets:193901 errors:0 dropped:0 overruns:0 frame:0
          TX packets:241121 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:37172963 (35.4 MiB)  TX bytes:19081898 (18.1 MiB)

venet0:0  Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:78.46.51.253  P-t-P:78.46.51.253  Bcast:78.46.51.253
Mask:255.255.255.255
          UP BROADCAST POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1

venet0:1  Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:192.168.11.101  P-t-P:192.168.11.101  Bcast:192.168.11.101
Mask:255.255.255.255
          UP BROADCAST POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
```

Abb. 7: Netzwerkkonfiguration(ifconfig) des Webservers H1

Webserver H2:

- CentOS 5 (CentOS release 5.4 (Final))
- Apache Webserver (httpd-2.2.3-53.el5.vm)
- MySQL Datenbankserver (mysql-server-5.0.84-2.el5.centos)
- proFTPd FTP Server (proftpd-1.3.3c-1.el5.vm)
- Postfix Mail Server (postfix-2.3.3-2.3.el5_6)
- Dovecot IMAP/POP3 Server (dovecot-1.0.7-7.el5_7.1)
- SSH Server (openssh-server-4.3p2-72.el5_6.3)
- Virtualmin (virtualmin-base-1.0-61.rh)

Dieser Webserver wird nur über die private IP-Adresse 192.168.11.108 und das Interface venet0:0 angesprochen.

```
Lo            Link encap:Local Loopback
              inet addr:127.0.0.1  Mask:255.0.0.0
              inet6 addr: ::1/128 Scope:Host
              UP LOOPBACK RUNNING  MTU:16436  Metric:1
              RX packets:0 errors:0 dropped:0 overruns:0 frame:0
              TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 txqueuelen:0
              RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

venet0        Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
              inet addr:127.0.0.1  P-t-P:127.0.0.1  Bcast:0.0.0.0  Mask:255.255.255.255
              UP BROADCAST POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
              RX packets:2913 errors:0 dropped:0 overruns:0 frame:0
              TX packets:3252 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 txqueuelen:0
              RX bytes:424837 (414.8 KiB)  TX bytes:244867 (239.1 KiB)

venet0:0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
              inet addr:192.168.11.108  P-t-P:192.168.11.108  Bcast:192.168.11.108
Mask:255.255.255.255
              UP BROADCAST POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
```

Abb. 8: Netzwerkkonfiguration(ifconfig) des Webservers H2

Eine Besonderheit an der Konfiguration der Server ist, dass das Hostsystem für den reibungslosen Betrieb, für das Netzwerk Discovery Protocol[10] (NDP) konfiguriert werden muss.

[10] Das NDP Protokoll ist im RFC4861 definiert und kann unter folgenden Link nachgelesen werden: http://tools.ietf.org/html/rfc4861

3.2 Netzwerkkonfiguration des gehosteten Teils

Die Migration auf IPv6 oder besser die Migration auf einen Dualbetrieb von IPv4 und IPv6 soll möglichst ohne Downtime erfolgen. Bei der Umstellung sollte es zu keinem Ausfall der Netzwerkverbindung kommen, eine Umstellung über eine Remoteverbindung (z.B.: SSH) ist so möglich. Zu beachten ist das nach einem Neustart der Netzwerkverbindungen des Hostsystems die virtuellen Maschinen neu gestartet werden sollten, da es sonst zu Problemen mit den Netzwerkschnittstellen kommen kann.

Außerdem sollte man beachten, dass die Standard Firewall Einstellungen von centOS 5 so gesetzt sind das nur wenige Ports am Host, sowie in der Forwarding Chain freigegeben sind. Es ist somit eine Verbindung per SSH über die neue IPv6 IP möglich aber andere Dienste werden meist blockiert.

3.2.1 Netzwerkkonfiguration des Hostsystems

3.2.1.1 IPv6 Adressvergabe

Der Hostingbetreiber Hetzner versorgt bereits seit einigen Monaten Kunden mit IPv6 Adressen. Über die Administrationswebsite des Hetzner Rootservers kann man dann IPv6 Adressen anfordern. Da IPv6 Netze immer mindestens (außer in wenigen Ausnahmefällen) als /64 Netze vergeben werden, sind genügend Adressen für die Server verfügbar. Man sollte sich ein Konzept überlegen bei dem einem Server ein eigenes Subnetz aus diesem Netz zugeordnet wird. Somit lassen sich IPv6 Adressen einfach den Hosts zuordnen und Fehler leichter erkennen. Mehrere IP Adressen auf einem Host sind durchaus sinnvoll, besonders bei Webserver die mit gültigen SSL Zertifikaten arbeiten wollen.

Folgendes IPv6 Netz ist dem Rootserver zugeteilt worden:

Netzwerk:	2a01:4f8:101:1145:: /64
Gateway:	2a01:4f8:101:1140::1 /59
Verwendbare IP-Adressen:	2a01:4f8:101:1145::2 bis 2a01:4f8:101:1145:ffff:ffff:ffff:ffff

Wie man sieht befindet sich der Default Gateway nicht im selben Netz, es muss also ein eigener Routing Eintrag dafür anlegen werden.[11]

[11] Hetzner bietet für die Umstellung auf IPv6 eigene Hilfeseiten in seinem Wiki an : http://wiki.hetzner.de/index.php/Zusaetzliche_IP-Adressen

3.2.1.2 IPv6 Netzwerkkonfiguration

Standardmäßig ist IPv6 bei centOS 5 deaktiviert und muss erst in den Netzwerk Konfigurationsfiles aktiviert werden[12]. Anschließend können die Netzwerkinterfaces konfiguriert und Hosting Einträge angelegt werden. Als erstes muss die Netzwerkkonfiguration geändert werden. Es muss die Unterstützung für IPv6 aktiviert werden.
Die Einstellung „`NETWORKING_IPV6=yes`“ aktiviert die systemweite IPv6 Initialisation und die Einstellung „`IPV6INIT=yes`“ aktiviert IPv6 für alle Netzwerk Interfaces.

```
NETWORKING=yes
NETWORKING_IPV6=yes
HOSTNAME=CentOS-54-64-minimal
IPV6INIT=yes
```

Abb. 9: IPv6 Aktivierung in /etc/sysconfig/network

Unsere Interface Einstellungen, hier das Interface eth0, sind statische Einträge wie es bei den meisten Servern der Fall sein wird.

Den Einstellungen für das Netzwerk Interface muss lediglich eine oder mehrere IPv6 Adressen hinzugefügt werden. Der Eintrag „`IPV6ADDR`“ fügt die neue IPv6 Adresse hinzu und der Eintrag „`IPV6ADDR_SECONDARIES`“ fügt weitere Adressen hinzu.
Würde sich der Default Gateway im selben Netz befinden, wie es normalerweise der Fall ist, würde man hier noch den Eintrag „`IPV6_DEFAULTGW`“ hinzufügen.

```
DEVICE=eth0
BOOTPROTO=static
BROADCAST=78.46.51.255
HWADDR=40:61:86:2b:8e:9d
IPADDR=78.46.51.227
NETMASK=255.255.255.255
SCOPE="peer 78.46.51.225"
IPV6ADDR="2a01:4f8:101:1145::0002/64"
IPV6ADDR_SECONDARIES="2a01:4f8:101:1145::0003/64 2a01:4f8:101:1145::0004/64"
```

Abb. 10: IPv6 Adressvergabe am Int. eth0 (/etc/sysconfig/network-scripts/ifcfg-eth0)

Da sich der Default Gateway außerhalb des Netzes befindet muss ein eigener Routing Eintrag anlegen werden, um eine Verbindung mit dem Internet herstellen zu können. Für IPv6 Routingeinträge gibt es in centOS ein eigenes Konfigurationsfile mit dem Namen `route6-eth0`, IPv4 Routingeintrage werden in `route-eth0` eingetragen (`eth0` beschreibt dabei immer das Interface für das der Routingeintrag bestimmt ist).

[12] Nähere Informationen zu IPv6 und CentOS kann unter folgendem Link nachlgelesen werden: http://www.cyberciti.biz/faq/rhel-redhat-fedora-centos-ipv6-network-configuration/

```
2a01:4f8:101:1140::1 dev eth0
default via 2a01:4f8:101:1140::1
```

Abb. 11: Default Gateway in /etc/sysconfig/network-scripts/route6-eth0 einstellen

3.2.1.3 Überprüfung der gesamten Netzwerkkonfiguration

Um die Einstellungen in den verschiedenen Konfigurationsfiles nun zu übernehmen muss der Server oder wenigstens der Netzwerk Service neu gestartet werden. Mit laufenden virtuellen Maschinen ist es zu empfehlen das gesamte System mit `reboot` neu zu starten. Sollte man nur den Netzwerk Service neu starten wollen, ist es zu empfehlen die virtuellen Maschinen anschließend neu zu starten.

```
# /etc/init.d/network restart
Shutting down interface eth0:                              [  OK  ]
Shutting down interface venet0:  Shutting down interface venet0:
                                                           [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Disabling IPv4 packet forwarding:  net.ipv4.ip_forward = 0
                                                           [  OK  ]
Bringing up loopback interface:  Global IPv6 forwarding is disabled in configuration, but
not currently disabled in kernel
Please restart network with '/sbin/service network restart'
                                                           [  OK  ]
Bringing up interface eth0:  Global IPv6 forwarding is disabled in configuration, but not
currently disabled in kernel
Please restart network with '/sbin/service network restart'
                                                           [  OK  ]
Bringing up interface venet0:  Bringing up interface venet0:
Configuring interface venet0:
net.ipv4.conf.venet0.send_redirects = 0
                                                           [  OK  ]
```

Abb. 12: Neustart der Netzwerkservices

Nach dem Neustart kann die Netzwerkkonfiguration ausgelesen werden. Es ist zu sehen dass die IPv6 Adressen auf dem Interface übernommen wurden. Es muss jetzt noch getestet werden ob sie auch funktionieren.

```
eth0      Link encap:Ethernet  HWaddr 40:61:86:2B:8E:9D
          inet addr:78.46.51.227  Bcast:78.46.51.255  Mask:255.255.255.255
          inet6 addr: 2a01:4f8:101:1145::2/64 Scope:Global
          inet6 addr: 2a01:4f8:101:1145::3/64 Scope:Global
          inet6 addr: 2a01:4f8:101:1145::4/64 Scope:Global
          inet6 addr: fe80::4261:86ff:fe2b:8e9d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:97469502 errors:0 dropped:0 overruns:0 frame:0
          TX packets:198430076 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7926360440 (7.3 GiB)  TX bytes:267494087686 (249.1 GiB)
          Interrupt:82 Base address:0xa000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:48374 errors:0 dropped:0 overruns:0 frame:0
          TX packets:48374 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:2612196 (2.4 MiB)  TX bytes:2612196 (2.4 MiB)

venet0    Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          UP BROADCAST POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
          RX packets:70452399 errors:0 dropped:0 overruns:0 frame:0
          TX packets:34726590 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:73683575237 (68.6 GiB)  TX bytes:3392891562 (3.1 GiB)
```

Abb. 13: Netzwerkkonfiguration(ifconfig) des Rootservers nach dem Neustart

Bei der Ausgabe der IPv6 Routingtabelle kann man sicherstellen, dass der Routingeintrag zu dem Default Gateway korrekt übernommen wurde.

```
# route -n -A inet6
Kernel IPv6 routing table
Destination                    Next Hop                   Flags Metric Ref    Use Iface
2a01:4f8:101:1140::1/128       ::                         U     1024   1        0 eth0
2a01:4f8:101:1145::/64         ::                         U     256    0        0 eth0
fe80::/64                      ::                         U     256    0        0 eth0
::/0                           2a01:4f8:101:1140::1       UG    1024   0        0 eth0
::1/128                        ::                         U     0      2        1 lo
2a01:4f8:101:1145::/128        ::                         U     0      0        1 lo
2a01:4f8:101:1145::2/128       ::                         U     0      0        1 lo
2a01:4f8:101:1145::3/128       ::                         U     0      0        1 lo
2a01:4f8:101:1145::4/128       ::                         U     0      0        1 lo
fe80::/128                     ::                         U     0      0        1 lo
fe80::4261:86ff:fe2b:8e9d/128  ::                         U     0      0        1 lo
ff00::/8                       ::                         U     256    0        0 eth0
```

Abb. 14: IPv6 Routingtabelle auf dem Rootserver

3.2.1.4 Testen der Netzwerkumstellung auf IPv6

Nach dem Neustart kann die IPv6 Konfiguration getestet werden. Das Programm ping6 führt einen Ping an einen IPv6 Host durch, damit geht man sicher das die Verbindung nicht doch über IPv4 zustande kommt. Ein mögliches IPv6 Ziel ist der IPv6 Server von Google mit dem Namen `ipv6.google.com`.

```
# ping6 ipv6.google.com
PING ipv6.google.com(muc03s01-in-x12.1e100.net) 56 data bytes
64 bytes from muc03s01-in-x12.1e100.net: icmp_seq=1 ttl=55 time=15.1 ms
64 bytes from muc03s01-in-x12.1e100.net: icmp_seq=2 ttl=55 time=15.2 ms
64 bytes from muc03s01-in-x12.1e100.net: icmp_seq=3 ttl=55 time=14.7 ms
64 bytes from muc03s01-in-x12.1e100.net: icmp_seq=4 ttl=55 time=15.0 ms
```

Abb. 15: Erfolgreicher Erreichbarkeitstest mittels PING Befehl

3.2.2 Umstellung der openVZ Virtualisierung[13]

Um unsere Virtualisierungslösung openVZ IPv6 fähig zu machen sind noch einige Einträge in den System-, Netzwerk- und openVZ-Konfigurationsfiles am Hostsystem nötig. Die IPv6 Pakte müssen vom Hostsystem richtig an die Gastsysteme weitergereicht werden und umgekehrt. Außerdem sind Besonderheiten bei der Netzwerkkonfiguration für einen gehosteten Server bei Hetzner zu beachten.[14]

3.2.2.1 Netzwerkkonfiguration an openVZ anpassen

Die Netzwerkkonfiguration muss noch zwei Einträge erweitert werden. Um die Weiterleitung von IPv6 Paketen zu aktivieren wird der Eintrag „`IPV6FORWARDING=yes`“ gesetzt. Anschließend wird mit „`IPV6_AUTOCONF=no`“ das IPv6 Autoconf Feature abgeschalten.

```
NETWORKING=yes
NETWORKING_IPV6=yes
HOSTNAME=CentOS-54-64-minimal
IPV6INIT=yes
IPV6FORWARDING=yes
IPV6_AUTOCONF=no
```

Abb. 16: IPv6 Weiterleitung in der Netzwerkkonfiguration (/etc/sysconfig/network)

[13] Das Wiki von openVZ bietet weitere Informationen über den Umstieg auf IPv6 mit openVZ als Virtualisierungslösung: http://wiki.openvz.org/IPv6

[14] Hetzner bietet dazu weitere Informationen: http://wiki.hetzner.de/index.php/Proxmox_VE

Um den virtuellen Maschinen Webserver H1 & H2 zu ermöglichen auf das Internet über das Hostsystem zuzugreifen müssen einige Einträge in die `sysctl.conf` Datei eingetragen werden. Der Eintrag `net.ipv6.conf.all.forwarding=1` aktiviert die Weiterleitung von Paketen zu anderen Systemen und anderen Netzwerkinterfaces (routing).

Um die virtuellen Maschinen Webserver H1 & H2 über das Network Discovery Protocol sichtbar zu machen muss am Host zusätzlich die Weiterleitung für NDP Pakete, mit dem Eintrag `net.ipv6.conf.all.proxy_ndp=1`, aktiviert werden.

```
# sysctl config
net.ipv4.ip_forward = 1
net.ipv6.conf.default.forwarding = 1
net.ipv6.conf.all.forwarding = 1
net.ipv4.conf.default.proxy_arp = 0

# Enables source route verification
net.ipv4.conf.all.rp_filter = 1

# Enables the magic-sysrq key
kernel.sysrq = 1

# We do not want all our interfaces to send redirects
net.ipv4.conf.default.send_redirects = 1
net.ipv4.conf.all.send_redirects = 0

net.ipv4.icmp_echo_ignore_broadcasts=1

net.ipv6.conf.all.proxy_ndp=1
```

Abb. 17: IPv6 Weiterleitung für die Webserver H1 & H2 (/etc/sysctl.conf)

In die Konfigurationsdatei `/etc/vz/vz.conf` für openVZ muss lediglich der Eintrag `IPV6="yes"` hinzugefügt werden. Damit wird die IPv6 Unterstützung von openVZ aktiviert und kann nach einem Neustart des Dienstes verwendet werden. Das gesamte Konfigurationsfile befindet sich unter 7.2 im Anhang.

3.2.2.2 Neustart der openVZ Umgebung und des Netzwerkservices

Die Änderungen in den Einstellungen für das Netzwerk und die Virtualiserung werden erst übernommen wenn die dazugehörigen Dienste neu gestartet werden. Optional kann auch das gesamte Hostsystem neu gestartet werden.

```
# /etc/init.d/network restart
Shutting down interface eth0:                              [  OK  ]
Shutting down interface venet0:  Shutting down interface venet0:
                                                           [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Disabling IPv4 packet forwarding:  net.ipv4.ip_forward = 0
                                                           [  OK  ]
Bringing up loopback interface:  Global IPv6 forwarding is disabled in configuration, but
not currently disabled in kernel
Please restart network with '/sbin/service network restart'
                                                           [  OK  ]
Bringing up interface eth0:  Global IPv6 forwarding is disabled in configuration, but not
currently disabled in kernel
Please restart network with '/sbin/service network restart'
                                                           [  OK  ]
Bringing up interface venet0:  Bringing up interface venet0:
Configuring interface venet0:
net.ipv4.conf.venet0.send_redirects = 0
                                                           [  OK  ]
```

Abb. 18: Neustart des Netzwerkdienstes (/etc/init.d/network restart)

```
# /etc/init.d/vz restart
Shutting down CT 101
Shutting down CT 102
Shutting down CT 103
...
Bringing down interface venet0:                            [  OK  ]
Stopping OpenVZ:                                           [  OK  ]
Starting OpenVZ:                                           [  OK  ]
Bringing up interface venet0:                              [  OK  ]
Configuring interface venet0:                              [  OK  ]
Configuring ipv6 venet0:                                   [  OK  ]
Starting CT 101:                                           [  OK  ]
Starting CT 102:                                           [  OK  ]
Starting CT 103:                                           [  OK  ]
...
```

Abb. 19: Neustart des openVZ Dienstes (/etc/init.d/vz restart)

3.3 Hinzufügen der Adresse zu einer virtuellen Maschine:

Um nun eine IPv6 Adresse zu einer VM hinzuzufügen gibt es den Befehl:[15]

```
vzctl set <id> --ipadd <ipv6_addr> --save
```

Dieser für eine IPv6 Adresse `<ipv6_addr>` permanent zu einer virtuellen Maschine mit der ID `<id>` hinzu. Nach dem erfolgreichen hinzufügen wird dies mit einer Meldung bestätigt.

[15] vzctl ist das zentralve Verwaltungswekzeug von openVZ: http://pve.proxmox.com/wiki/Vzctl_manual

```
# vzctl set 101 --ipadd 2a01:4f8:101:1145::1:1 --save
Adding IP address(es): 2a01:4f8:101:1145::1:1
Saved parameters for CT 101

# vzctl set 101 --ipadd 2a01:4f8:101:1145::1:2 --save
Adding IP address(es): 2a01:4f8:101:1145::1:2
Saved parameters for CT 101

# vzctl set 101 --ipadd 2a01:4f8:101:1145::1:3 --save
Adding IP address(es): 2a01:4f8:101:1145::1:3
Saved parameters for CT 101
```

Abb. 20: Konfiguration der IPv6 Adressen mit vzctl

Der virtuellen Maschine wurde damit drei IPv6 Adressen hinzugefügt. Der hier verwendet Interfacetyp von openVZ lässt eine Zuteilung der IP Adressen nur von dem Hostsystem zu. Es kann jedoch von Vorteil sein, wenn die virtuellen Maschinen sich ihre IP Adressen selbst vergeben können. Dies hat den Vorteil, dass bei vielen Verwaltungstools wie bei Virtualmin, das auf den Webservern dieses Hosts verwendet wird, IPv6 Adressen aus einem Pool automatisch hinzugefügt werden können. Somit kann jeder neu erstellen Webseite am Server eine eigene IP Adresse zugeteilt werden, was das Arbeiten mit SSL Zertifikaten erheblich vereinfacht.

3.3.1 Neustart der VM

Das hinzufügen der IP sollte von den VM sofort übernommen werden. Sollte es jedoch zu Problemen kommen, können die virtuellen Maschinen neu gestartet werden.

```
# vzctl restart 101
Restarting container
Stopping container ...
Container was stopped
Container is unmounted
Starting container ...
Container is mounted
Adding IP address(es): 78.46.51.253 192.168.11.101 2a01:4f8:101:1145::1:1
2a01:4f8:101:1145::2:1 2a01:4f8:101:1145::1:2 2a01:4f8:101:1145::1:3
SIOCSIFADDR: Network is down
SIOCSIFADDR: Network is down
SIOCSIFADDR: Network is down
SIOCSIFADDR: Network is down
Setting CPU units: 1000
Configure meminfo: 1560576
Set hostname: webspace.die-lega.org
File resolv.conf was modified
Container start in progress...
```

Abb. 21: Neustart einer virtuellen Maschine mit vzctl

3.3.2 Testen der IPv6 Verbindung

Wenn man sich auf der soeben Konfigurierten virtuellen Maschine einloggt, kann die IPv6 Netzwerkverbindung direkt kontrolliert und getestet werden. Dies läuft gleich wie am Hostsystem mit den Befehlen `ifconfig` und `ping6`.

```
...

venet0    Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:127.0.0.1  P-t-P:127.0.0.1  Bcast:0.0.0.0  Mask:255.255.255.255
          inet6 addr: 2a01:4f8:101:1145::1:1/128 Scope:Global
          inet6 addr: 2a01:4f8:101:1145::1:3/128 Scope:Global
          inet6 addr: 2a01:4f8:101:1145::1:2/128 Scope:Global
          inet6 addr: 2a01:4f8:101:1145::2:1/128 Scope:Global
          UP BROADCAST POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
          RX packets:122519 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160255 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:24331756 (23.2 MiB)  TX bytes:14798377 (14.1 MiB)

venet0:0  Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:78.46.51.253  P-t-P:78.46.51.253  Bcast:78.46.51.253
Mask:255.255.255.255
          UP BROADCAST POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1

venet0:1  Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:192.168.11.101  P-t-P:192.168.11.101  Bcast:192.168.11.101
Mask:255.255.255.255
          UP BROADCAST POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
```

Abb. 22: Auszug der Ausgabe von ifconfig auf einer virtuelllen Maschine

Wie man an der Ausgabe von `ifconfig` sieht wird nicht mehr für jede IPv6 Adresse ein eigenes virtuelles Interface erzeugt sondern alle IPv6 Adressen können einem Interface zugeordnet werden.

```
# ping6 ipv6.google.com
PING ipv6.google.com(muc03s01-in-x11.1e100.net) 56 data bytes
64 bytes from muc03s01-in-x11.1e100.net: icmp_seq=1 ttl=55 time=14.9 ms
64 bytes from muc03s01-in-x11.1e100.net: icmp_seq=2 ttl=55 time=14.9 ms
```

Abb. 23: Ausgabe von ping6 von einer virtuellen Maschine

3.4 Anlegen von DNS Einträgen für IPv6

Um nun die Dienste bzw. die Server wie gewohnt über DNS Namen ansprechen zu können, müssen für IPv6 eigene DNS Einträge gesetzt werden. Der Unterschied zwischen den IPv4 und IPv6 Einträgen ist nur marginal, ein A Record Eintrag wird zu einem AAAA Record.
Zum Testen der Verbindung und der DNS Auflösung sowie später des Webserver werden verschiedene DNS Einträge für die Domain `die-lega.org` angelegt. Einen Eintrag `ipv6.die-lega.org` der nur über einen AAAA Eintrag auf eine IPv6 Adresse verweist. Somit kann sichergestellt werden, dass der Server über die IPv6 Adresse erreicht werden kann. Außerdem wird ein AAAA Eintrag für `dual.die-lega.org` sowie ein A Eintrag für denselben Domain Namen angelegt. Damit kann getestet werden wie sich der Server, der Client und die Namensauflösung verhalten, wenn ein DNS Eintrag auf eine IPv6 und eine IPv4 Adresse aufgelöst werden kann. Bei einem produktiven Einsatz des Servers, der über IPv4 und IPv6 erreicht werden können soll, müssen die DNS Einträge ebenfalls auf IPv4 und IPv6 Adressen verweisen. Der Client sollte dann jene Adresse auswählen über die er den Server erreicht.

IPv4-only Eintrag für den Hostserver:

```
rootserver              IN A        78.46.51.227
```

IPv6-only Eintrag für den Hostserver:

```
ripv6                   IN AAAA     2a01:4f8:101:1145::2
```

IPv4-only Eintrag für den Webserver:

```
webspace                IN A        78.46.51.253
```

IPv6-only Eintrag für den Webserver:

```
ipv6                    IN AAAA     2a01:4f8:101:1145::1:1
```

Dualstack Einträge für den Webserver:

```
dual                    IN A        78.46.51.253
dual                    IN AAAA     2a01:4f8:101:1145::1:1
lansuite                IN A        78.46.51.253
lansuite                IN AAAA     2a01:4f8:101:1145::1:2
```

Die DNS Records liegen nach dem bearbeiten vollständig unter 7.3 im Anhang.

3.5 Windows Client IPv6 fähig machen

Um den Server nun korrekt testen zu können ist am besten ein Client PC IPv6 fähig zu machen. Sollte der Internet Provieder dies noch nicht anbieten kann man eine IPv6 Verbindung ziemlich einfach über einen Tunnelprovieder herstellen. Diese Provider bieten die Möglichkeit einen IPv6 Tunnel durch ein IPv4 Netzwerk (Internet) aufzubauen.

Einer diese Tunnelprovieder ist gogo6 welcher einen IPv6 Tunnel kostenlos anbietet. Um eine öffentliche IPv6 Adresse zu bekommen muss lediglich der Freenet6 Client von http://gogo6.com heruntergeladen und installiert werden. Eine Anmeldung ist nur nötig wenn man eine fixe IPv6 Adresse oder gar ein /56 Subnet von gogo6 bekommen will.
Freenet6 eignet sich perfekt um erste Erfahrungen mit IPv6 zu sammeln oder in unserem Fall den Zugriff auf den IPv6 Server zu testen.

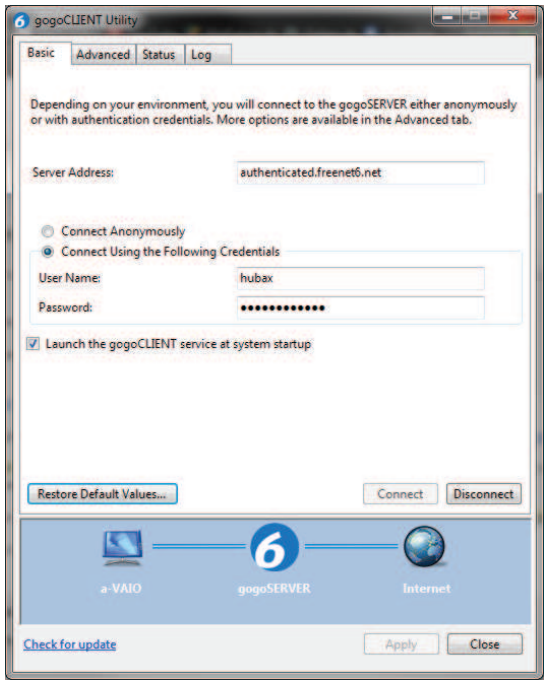

Abb. 24: Konfiguration des gogoCLIENTS auf einem Windows Rechenr

Nach dem Öffnen des gogoCLIENT und dem drücke auf Connect kann die Erreichbarkeit des Server mit einem einfachen `ping` Befehl getestet werden.

```
C:\Users\a>ping -6 ipv6.google.com

Ping wird ausgeführt für ipv6.l.google.com [2001:4860:800a::6a] mit 32 Bytes Daten:
Antwort von 2001:4860:800a::6a: Zeit=154ms
Antwort von 2001:4860:800a::6a: Zeit=152ms
Antwort von 2001:4860:800a::6a: Zeit=157ms
Antwort von 2001:4860:800a::6a: Zeit=155ms

C:\Users\a>ping webspace.die-lega.org

Ping wird ausgeführt für webspace.die-lega.org [78.46.51.252] mit 32 Bytes Daten:
Antwort von 78.46.51.252: Bytes=32 Zeit=56ms TTL=57
Antwort von 78.46.51.252: Bytes=32 Zeit=56ms TTL=57
Antwort von 78.46.51.252: Bytes=32 Zeit=58ms TTL=57
Antwort von 78.46.51.252: Bytes=32 Zeit=59ms TTL=57

C:\Users\a>ping dual.die-lega.org

Ping wird ausgeführt für dual.die-lega.org [2a01:4f8:101:1145::1:1] mit 32 Bytes Daten:
Zeitüberschreitung der Anforderung.
Antwort von 2a01:4f8:101:1145::1:1: Zeit=72ms
Antwort von 2a01:4f8:101:1145::1:1: Zeit=68ms
Antwort von 2a01:4f8:101:1145::1:1: Zeit=69ms

C:\Users\a>ping -4 dual.die-lega.org

Ping wird ausgeführt für dual.die-lega.org [78.46.51.252] mit 32 Bytes Daten:
Antwort von 78.46.51.252: Bytes=32 Zeit=61ms TTL=57
Antwort von 78.46.51.252: Bytes=32 Zeit=57ms TTL=57
Antwort von 78.46.51.252: Bytes=32 Zeit=59ms TTL=57
Antwort von 78.46.51.252: Bytes=32 Zeit=57ms TTL=57

C:\Users\a>ping ipv6.die-lega.org

Ping wird ausgeführt für ipv6.die-lega.org [2a01:4f8:101:1145::1:1] mit 32 Bytes Daten:
Antwort von 2a01:4f8:101:1145::1:1: Zeit=68ms
Antwort von 2a01:4f8:101:1145::1:1: Zeit=72ms
Antwort von 2a01:4f8:101:1145::1:1: Zeit=67ms
Antwort von 2a01:4f8:101:1145::1:1: Zeit=68ms
```

Abb. 25: Test der Erreichbarkeit des Servers mittels ping

Wie man an der Ausgabe von ping sieht werden die DNS Adressen korrekt aufgelöst. Adressen bei denen nur ein A-Record hinterlegt ist werden auf die IPv4 Adresse aufgelöst, Einträge bei denen nur ein AAAA-Record hinterlegt ist werden auf die IPv6 Adresse aufgelöst und Einträge bei denen sowohl ein A-Record als auch ein AAAA-Record vorhanden ist werden vorzugsweise auf die IPv6 Adresse aufgelöst.

Alle IP Adressen die vom mit Ping getestet werden sind wie gewünscht erreichbar.

3.6 Firewall Konfiguration

Die Firewall Einstellungen für den IPv6 Verkehr sind bei netfilter unabhängig von den Einstellungen für IPv4 Pakete. Konfiguriert können diese über das Tool ip6config werden oder bei centOS direkt über die Konfigurationsdatei `/etc/sysconfig/ip6tables`.[16]
Da die Standardeinstellung der Firewall fast alle eingehenden Verbindungen bis auf SSH blockieren, müssen die Firewall-Regeln für IPv6 erzeugt werden. Die meisten Regeln können aus der Konfiguration von netfilter für IPv4 übernommen werden, in manchen Regeln sind kleine Änderungen oder zusätzliche Optionen nötig.

Nachfolgend ist ein kurzes Konfigurationsfile für ip6tables zu sehen welches die wichtigsten Ports für einen Webserver freigibt und Pakete wie icmpv6 erlaubt.[17]

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD DROP  [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
:LEGA-WEBSPACE-INPUT - [0:0]

-A FORWARD -s 2a01:4f8:101:1145::1:1 -j ACCEPT
-A FORWARD -d 2a01:4f8:101:1145::1:1 -j LEGA-WEBSPACE-INPUT

-A INPUT -j RH-Firewall-1-INPUT
#-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmpv6 -j ACCEPT
-A RH-Firewall-1-INPUT -p 50 -j ACCEPT
-A RH-Firewall-1-INPUT -p 51 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp --dport 5353 -d ff02::fb -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 32768:61000 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 32768:61000 ! --syn -j ACCEPT
-A RH-Firewall-1-INPUT -m tcp -p tcp --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp6-adm-prohibited

-A LEGA-WEBSPACE-INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p icmpv6 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p udp -m udp --dport 20 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p udp -m udp --dport 21 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p udp -m udp --dport 53 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 443 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 80 -j ACCEPT
...
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 25 -j ACCEPT

COMMIT
```

Abb. 26: Auszug aus der Firewall Konfiguration des Hostsystems

[16] Informationen zu ip6tables unter centOS bietet die Dokumentation von centOS unter http://www.centos.org/docs/5/html/5.2/Deployment_Guide/s1-ip6tables.html
[17] Weiter Informationen über IPv6 und netfilter bietet das Wiki von sixxs.net: http://www.sixxs.net/wiki/IPv6_Firewalling

In der am Hostserver verwendeten Version gibt es keine NAT Unterstützung für IPv6. Obwohl dies nicht in den Spezifikationen für IPv6 vorgesehen ist, wurde es doch in neueren Versionen von netfilter implementiert. Dies kann durchaus sinnvoll sein, wenn man zum Beispiel Dienste die auf verschiedenen virtuellen Server laufen über dieselbe IP ansprechen will.
Die NAT Chain muss/kann weggelassen werden. Da jedem virtuellen Server jetzt eine eigene IP zugeteilt werden kann, ist diese nicht mehr nötig.

Werden die Regeln über das Konfigurationsfile `/etc/sysconfig/ip6tables` geändert und nicht direkt über das Programm ip6tables muss die Firewall anschließend mit `/etc/init.d/ip6tables restart` neugestartet werden.

```
# /etc/init.d/ip6tables restart
```

Abb. 27: Neustart der Firewall am Hostsystem(/etc/init.d/ip6tables restart)

3.7 HTTP(S) Server:

Bei IPv4 war es üblich und nötig vielen Webseiten die auf einem Host laufen dieselbe IP Adresse zu geben. Das minimierte die Anzahl der benötigten IP Adressen erheblich, so konnten sich hunderte von Seiten eine Adresse Teilen. Der Webserver entschied dabei über Virtuelle Server anhand des Domainnamens die richtige Seite und lieferte Diese bei Anfrage aus.
Das hat aber den Nachteil, dass SSL gesicherte Verbindung nur schwer zu realisieren sind. Die SSL Verbindung wird aufgebaut bevor dem Webserver der Domainname der gewünschten Website bekannt ist. Da aber SSL Zertifikate aber an eine Domain gebunden ist, weiß der Client nicht welches Zertifikat für die jeweilige Anfrage ausgeliefert werden soll.
Abhilfe schafft dabei nur für jedes Zertifikat eine eigene IP Adresse zu verwenden. Eine Lösung über „Server Name Indication“[18], bei der der Domainname schon vorzeitig dem Server bekannt gegeben wird, funktioniert nur mit modernen Browsern, Webserver und Betriebssystemen.

Durch die unglaubliche Anzahl an verfügbaren IPv6 Adressen kann nun jedem Server eine Adressrange zugeteilt werden. Somit kann jede Webseite über ihre eigene IPv6 Adresse angesprochen werden und es gibt keine Konflikte mehr bei SSL verschlüsselten Verbindungen.

[18] Weiter Informationen über SNI bietet die Dokumentation von Apache unter http://wiki.apache.org/httpd/NameBasedSSLVHostsWithSNI

Verwaltungstools für Webserver können dabei automatisch dem Server eine neue IPv6 Adresse aus einer Range zuweisen. Somit ist kein weiterer Aufwand damit verbunden jeder Website eine eigene IPv6 Adresse zu geben.

3.7.1 Konfiguration des Apache Webservers

Der Apache Webserver unterstützt seit der Version 2.0 IPv6.[19] Um einen vorhandenen IPv4 Apache Webserver IPv6 Fähig zu machen, müssen nur die VirtualHost Einträge kopiert und die IPv4 Adressen durch IPv6 Adressen ersetzt werden. Dabei ist zu beachten, dass IPv6 Adressen von [...] umgeben werden müssen um die Angabe von Portnummern zu ermöglichen.[20]

```
<VirtualHost [2001:db8::a00:20ff:fea7:ccea]:80>
ServerAdmin webmaster@host.example.com
DocumentRoot /www/docs/host.example.com
ServerName host.example.com
ErrorLog logs/host.example.com-error_log
TransferLog logs/host.example.com-access_log
</VirtualHost>
```

Abb. 28: Beispielkonfiguration eines VirtualHost Eintrags von Apache

Im Bedarfsfall muss zusätzlich noch der Listen Eintrag angepasst werden um auf alle gewünschten Adressen zu lauschen oder mit `Listen *:80` auf allen vorhanden Adressen zu lauschen.[21]

Da zum Testen der IPv6 Verbindung ein eigener Domainnamen verwenden wird, muss ein zusätzlicher VirtualHost Eintrag für die Domain `ipv6.die-lega.org` angelegt werden. Wird eine Webseite ohne einen passenden Domainnamen aufgerufen wird der Standardeintrag des Webservers angezeigt.

Bevor der Apache Webserver für IPv6 konfiguriert wird, wird getestet ob der Webserver noch per IPv4 erreichbar ist und was passiert wenn die neue IPv6 Adresse aufgerufen wird.

[19] Die Unterstützung für IPv6 kann dem Apache Changelog entnommen werden: http://httpd.apache.org/docs/2.2/en/new_features_2_0.html

[20] Weitere Informationen zum Virtualhost Eintrag kann der Apache Dokumentation entnommen werden: http://httpd.apache.org/docs/2.2/en/mod/core.html#virtualhost

[21] Weitere Informationen zum Listen Eintrag kann der Apache Dokumentation entnommen werden: http://httpd.apache.org/docs/2.2/en/mod/mpm_common.html#listen

Abb. 29: Ausgabe des Browsers bei einer Anfrage nach webspache.die-lega.org

Der Webserver ist wie gewohnt über den Domainnamen `webspace.die-lega.org` per IPv4 erreichbar und zeigt die gewünschte Seite.

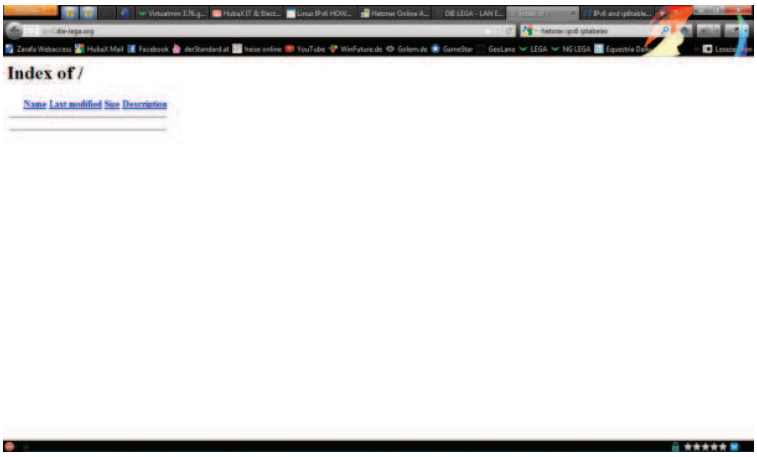

Abb. 30: Ausgabe des Browsers bei einer Anfrage nach ipv6.die-lega.org

Die neue IPv6 Domainadresse `ipv6.die-lega.org` kann vom Webserver nicht zugeordnet werden, somit Zeigt dieser einen default Eintrag.

Um den Apache Webserver nun über die neue Domain erreichbar zu machen, wird ein neuer VirtualHost Eintrag für die IPv6 Adresse angelegt und darauf geachtet, dass der Server auf alle (gewünschten) Adressen horcht.

```
Listen *:80

...

<VirtualHost [2a01:4f8:101:1145::1:1]:80>
SuexecUserGroup "#500" "#501"
ServerName die-lega.org
ServerAlias www.die-lega.org
ServerAlias dual.die-lega.org
ServerAlias ipv6.die-lega.org
ServerAlias webspace.die-lega.org

...

</VirtualHost>
```

Abb. 31: Auszug aus der Apache Konfiguration für den VirtualHost Eintrag mit IPv6

Mit dem „`Listen *:80`“ Eintrag antwortet der Apache Server auf alle Anfragen von allen Interfaces bzw. Adressen des Servers. Der VirtualHost Eintrag wird speziell für die IPv6 Adresse `2a01:4f8:101:1145::1:1` und den Port 80 angelegt und erkennt dort Anfragen für die Domainnamen `www.die-lega.org`, `dual.die-lega.org`, `ipv6.die-lega.org` und `webspace.die-lega.org`.

Der gesamte Virtualhost Eintrag befindet sich im unter 7.4 im Anhang.

3.7.2 Testen des Apache Webservers

Nach einem Neustart des Webserver wird der neu erstellte VirtualHost Eintrag von einem beliebigen Webbrowser aus getestet.

Abb. 32: Ausgabe des Browsers nach der Konfiguration für die Domain ipv6.die-lega.org

Jetzt wird vom Webserver die neu angelegte Domain zum richtigen VirtualHosts Eintrag zugeordnet.

3.7.3 Verhalten des Browsers bei verschiedenen DNS-Records

Um das Verhalten von Browsern mit verschiedenen DNS Records zu zeigen werden mit dem Firefox Webbrowser und dem Plugin Firebug die Zugriffe auf eine Website unter unterschiedlichen Domainnamen gezeigt.

Wenn eine globale IPv6 Adresse dem Host zugeordnet ist, wird die IPv6 Adresse verwendet. Aus den Netzwerkrequest des Browser ist außerdem zu sehen, dass Teile der Website nicht über IPv6 abgerufen werden können, da diese von einem anderen Domain Namen kommen, der keinen AAAA-Record besitzt.

Abb. 33: Adressauflösung des Browser für dual.die-lega.org mit globaler IPv6 Adresse

Ist dem Client keine globale IPv6 Adresse zugeordnet wird auf die Website mittels IPv4 zugegriffen. Auch wenn am DNS ein AAAA-Record vorhanden ist.

Abb. 34: Adressauflösung des Browsers für dual.die-lega.org ohne globale IPv6 Adresse

Wird die Website über einen Domainnamen abgerufen zu dem es nur einen IPv6 Eintrag gibt und keinen IPv4 Eintrag ist die Seite, wie schon bei IPv4, vom Client nur erreichbar wenn er eine gültige IPv6 Adresse hat.

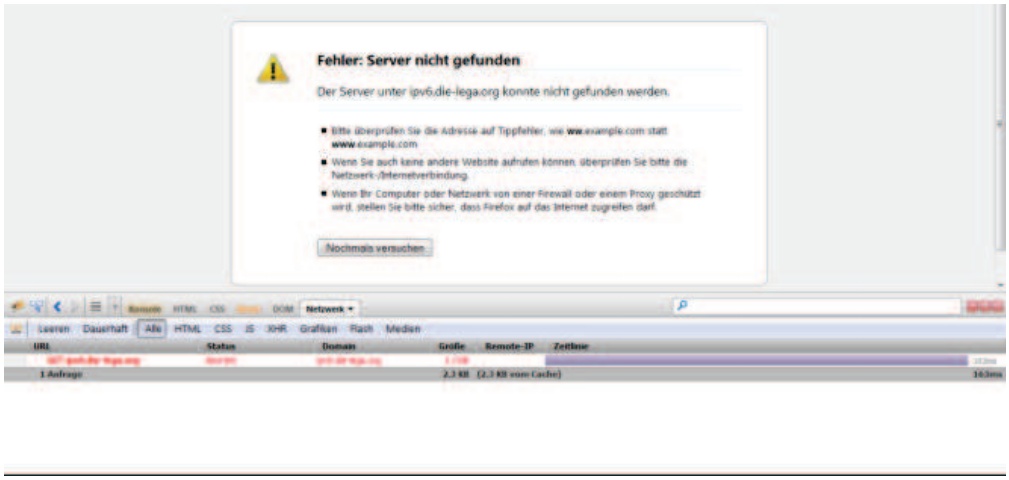

Abb. 35: Adressauflösung für ipv6.die-lega.og ohne globale IPv6 Adresse

Auch der Zugriff über HTTPS ist ohne Probleme möglich wenn der nötige Virtualhost Eintrag angelegt wurde.

Abb. 36: Adressauflösung und Zugriff über IPv6 mittels HTTPS

Selbst wenn der Client Rechner über eine globale IPv6 Adresse verfügt und mit dieser auf das Internet zugreifen kann, fällt er, wenn die Webseite über die IPv6 Adresse nicht erreichbar ist (z.B.: wegen einer fehlerhaften Firewall Konfiguration am Server) auf die IPv4 Adresse zurück und versucht über diese die Webseite zu erreichen.

3.7.4 Verschiede Virtuelle Hosts auf verschiedenen IP Adressen

Soll nun jeder virtuelle Server bzw jede Domain eine eigene IPv6 Adresse erhalten, müssen dem Netzwerkinterface des Webservers die neuen IPv6 Adresse hinzugefügt werden. Außerdem müssen neue VirtualHost Einträge in der Apache Konfiguration angelegt werden. Über die VirtualHost Einträge wird bestimmt, welche Domain(s) welchen IPv6 Adresse zugeteilt werden.

Dem Webserver kann jetzt zum Beispiel der Bereich `2a01:4f8:101:1145::1:1 - 2a01:4f8:101:1145::1:FFFF` zugeteilt werden, damit hat er 65536 Adressen zur Verfügung.

```
Listen *:80

...

<VirtualHost [2a01:4f8:101:1145::1:1]:80>
SuexecUserGroup "#500" "#501"
ServerName die-lega.org
ServerAlias www.die-lega.org
ServerAlias dual.die-lega.org
ServerAlias ipv6.die-lega.org
ServerAlias webspace.die-lega.org

...

</VirtualHost>

<VirtualHost [2a01:4f8:101:1145::1:2]:80>
SuexecUserGroup "#500" "#501"
ServerName lansuite.die-lega.org

...

</VirtualHost>
```

Abb. 37: Auszug aus der Apache Konfiguration mit verschiedenen IPv6 Adressen

Um den Zugriff auf die Webseite mit anderen IPv6 Adressen zu ermöglichen muss die Firewall konfiguriert werden. Dazu kann wenn nötig jede IP Adresse einzeln konfiguriert werden oder man verwendet dieselbe Konfiguration für das gesamte Subnetz (alle ihm zugeteilten Adressen) des Servers.

```
-A FORWARD -s 2a01:4f8:101:1145::1:1/112 -j ACCEPT
-A FORWARD -d 2a01:4f8:101:1145::1:1/112 -j LEGA-WEBSPACE-INPUT
```

Abb. 38: Auszug aus der Firewall Konfiguration des Hostsystems für ein Subnet

Um über die IPv6 Adresse auf eben konfigurierte Domain zugreifen zu können muss noch der AAAA Eintrag in das DNS Konfigurationsfile geschrieben werden. Anschließend kann mit einem Webbrowser der Zugriff auf die Domain getestet werden.

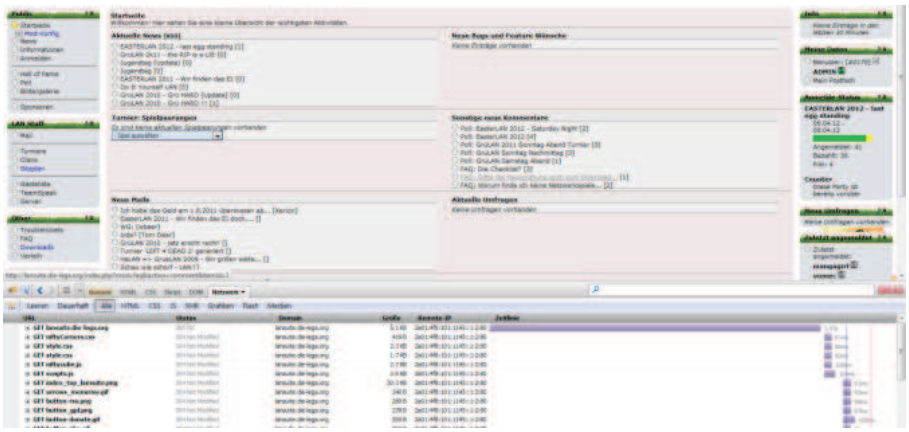

Abb. 39: Adressauflösung und Zugriff auf die Domain lansuite.die-lega.org

Wie zu sehen ist, wird für die Domain `lansuite.die-lega.org` die IPv6 Adresse `2a01:4f8:101:1145::1:2` verwendet statt der von uns für die erste Domain konfigurierte `2a01:4f8:101:1145::1:1`.

3.8 IMAP / POP3 / SMTP:

Die wichtigsten Dienste für den Versand und das Empfangen von E-Mails auf diesen Webserver sind:

- Postfix – zum entgegenehmen und weiterleiten der E-Mails
- Dovecot – Zugriff auf die Mailboxen per IMAP und POP3

Um die E-Mail Services IPv6 fähig zu machen sind nur kleine Änderungen in dem Hauptkonfigrationsfile von Postfix nötig. [22]
Dovecot ist bereits mit der default Konfiguration IPv6 fähig.[23]

```
...
inet_protocols = all
...
```

Abb. 40: Auszug aus der Postfix Konfiguration (/etc/postfix/main.cf)

3.8.1 Test des Zugriffs über IMAP und SMTP mit Outlook

Um den Zugriff auf die Mailkonten per IMAP sowie den Versand und Empfang von E-Mails zu testen wird das Programm „Microsoft Outlook 2007“ verwendet. Um den Verbindungsaufbau über IPv6 zu forcieren werden die IPv6-only Domainnamen des Servers verwendet und die Verbindung mit diesen getestet.

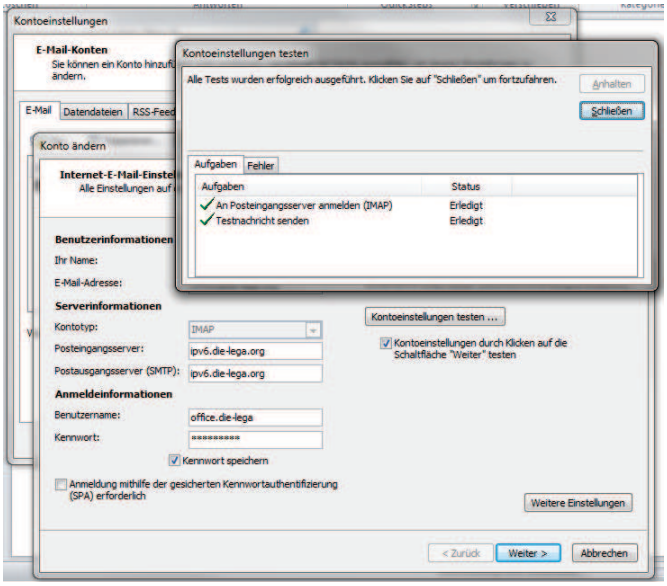

Abb. 41: Ausgabe des Verbindungstest von Outlook über eine IPv6 Adresse

[22] Weitere Informationen zu Postfix mit IPv6 sind auf der Website verfügbar: http://www.postfix.org/IPV6_README.html
[23] Weiter Informationen zu Dovecot mit IPv6 sind in deren Wiki zu finden: http://wiki.dovecot.org/MainConfig

Der Verbindungsaufbau über den Domainnamen `ipv6.die-lega.org` und somit über die IPv6 Adresse des Servers war per IMAP erfolgreich und auch das Senden einer Test E-Mail war erfolgreich, somit können E-Mail gesendet und empfangen werden.

3.8.2 Senden einer E-Mail an einen externen Empfänger

Das senden der Mails von einem IPv6 fähigen Client an einen IPv6 fähigen Server wird über IPv6 abgewickelt. Dies ist auch im Header der E-Mail zu sehen, nachdem sie das Ziel erreicht hat. Da der nächste Server (mx.google.com) nicht IPv6 fähig ist, wird die E-Mail über IPv4 an diesen Server weitergesendet.

```
Delivered-To: robert.huber@hubax.at
Received: by 10.182.58.133 with SMTP id r5csp317615obq;
        Wed, 25 Apr 2012 12:07:08 -0700 (PDT)
Received: by 10.216.132.75 with SMTP id n53mr2335235wei.51.1335380828407;
        Wed, 25 Apr 2012 12:07:08 -0700 (PDT)
Return-Path: <office@die-lega.org>
Received: from webspace.die-lega.org (webspace.die-lega.org. [78.46.51.253])
        by mx.google.com with ESMTP id d19si19359239wiw.0.2012.04.25.12.07.07;
        Wed, 25 Apr 2012 12:07:08 -0700 (PDT)
Received-SPF: pass (google.com: best guess record for domain of office@die-lega.org
designates 78.46.51.253 as permitted sender) client-ip=78.46.51.253;
Authentication-Results: mx.google.com; spf=pass (google.com: best guess record for domain
of office@die-lega.org designates 78.46.51.253 as permitted sender) smtp.mail=office@die-
lega.org
Received: from aVAIO (hubax.broker.freenet6.net [IPv6:2001:5c0:1400:b::c5b3])
       by webspace.die-lega.org (Postfix) with ESMTP id 3CBC51D9C007
       for <robert.huber@hubax.at>; Wed, 25 Apr 2012 21:07:48 +0200 (CEST)
From: "LAN Entertainment Group Austria" <office@die-lega.org>
To: "Robert Huber" <robert.huber@hubax.at>
Subject: test
Date: Wed, 25 Apr 2012 21:07:06 +0200
Message-ID: <007c01cd2316$9bd71830$d3854890$@die-lega.org>
MIME-Version: 1.0
Content-Type: multipart/alternative;
       boundary="----=_NextPart_000_007D_01CD2327.5F60AB80"
X-Mailer: Microsoft Outlook 14.0
Thread-Index: Ac0jFpgcFsfxzdHwSEyY0i30YFNj2w==
Content-Language: de-at

This is a multipart message in MIME format.

------=_NextPart_000_007D_01CD2327.5F60AB80
Content-Type: text/plain;
       charset="us-ascii"
Content-Transfer-Encoding: 7bit

test


...
```

Abb. 42: Auszug aus dem Header einer E-Mail über einen IPv6 fähigen SMTP Server

3.9 FTP Zugriff über IPv6

Bei dem verwendeten FTP Server „proFTPd“ sind keine Änderungen in der Konfiguration nötig, die Unterstützung für IPv6 ist standardmäßig aktiviert.[24] Am Client muss lediglich der Domainname oder die IP Adresse ersetzt werden.

Der Zugriff auf den FTP Server wurde mit FileZilla für Windows getestet. Als Serveradresse kann entweder ein DNS-Name oder eine IPv6 Adresse eingegeben werden.

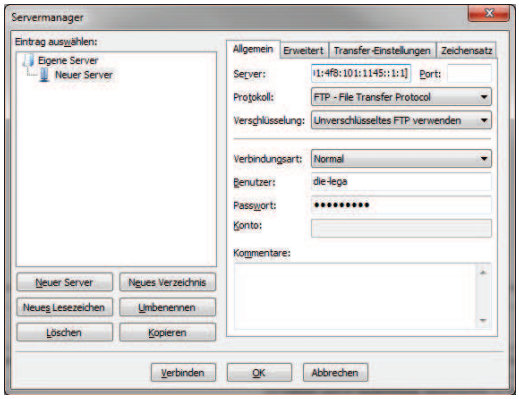

Abb. 43: Servermanager des FileZilla FTP Clients

```
Status: Auflösen der IP-Adresse für ipv6.die-lega.org
Status: Verbinde mit [2a01:4f8:101:1145::1:1]:21...
Status: Verbindung hergestellt, warte auf Willkommensnachricht...
Antwort:       220 FTPServer
Befehl: USER *****

...
```

Abb. 44: Verbindungslog des FileZilla FTP Clients beim Verbinden zu einem IPv6 Server

Der Log der Verbindung zeigt die Korrekte Auflösung des DNS Namens auf eine IPv6 Adresse sowie den erfolgreiche Verbindungsaufbau mit dem Server.

[24] Weitere Informationen zur Aktivierung von IPv6 in proftpd können der online Dokumentation entnommen werden: http://www.proftpd.org/docs/directives/linked/config_ref_UseIPv6.html

3.10 SSH über IPv6

Der openSSH Server ist standardmäßig per IPv6 erreichbar. Um ihn auf einzelne IPv6 Adressen zu konfigurieren kann der `ListenAddress` Eintrag geändert werden.

```
ListenAddress host
ListenAddress IPv4_addr:port
ListenAddress [IPv6_addr]:port
```

Abb. 45: Beispiele für einen ListeAddress Eintrag in einer sshd Konfiguration

Um auf jede IPv6 Adresse zu horchen kann auch `::` als Adresse eingetragen werden.

```
ListenAddress 0.0.0.0
ListenAddress ::

...
```

Abb. 46: Auszug aus dem sshd Konfigurationsfile (/etc/ssh/sshd_config)

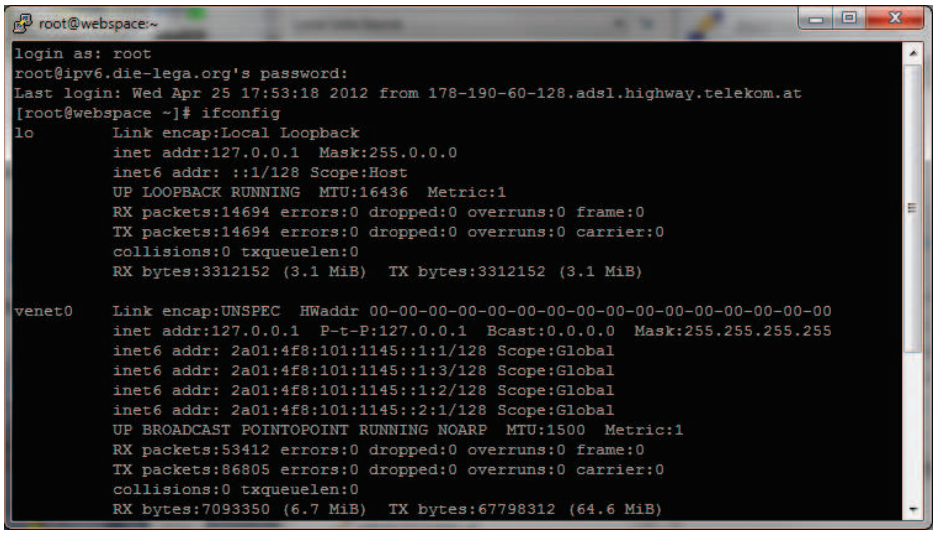

Abb. 47: Verbindungsaufbau zu einem SSH Server über ipv6.die-lega.org

3.11 Private IPv4 Adresse und öffentliche IPv6 Adresse

Um öffentliche IPv4 Adresse zu sparen wurde ein virtueller Webserver lediglich mit einer privaten IPv4 Adresse ausgestattet. Die Zugriffe von außen werden dabei über Port Forwarding ermöglicht und der Zugriff des Servers auf das Internet per NAT.

Da die Knappheit der Adressen bei IPv6 nicht mehr besteht können diese Server jetzt mit öffentlichen IPv6 Adressen ausgestattet werden.

Damit wird die Konfiguration der Server einfacher und ist praktisch identisch wie die Konfiguration des H1 Servers. Auszugsweise wird nur die Konfiguration der IPv6 Adressen und der Firewall Konfiguration für den Webserver H2 gezeigt.

3.11.1 Vergabe der IP Adressen

Dem virtuellen Server H2 wird die IPv6 Adresse `2a01:4f8:101:1145::8:1` zugewiesen. Außerdem werden dem Server direkt weiter IPv6 Adresse zugewiesem um wie schon beim Webserver H1 verschiedenen Domains verschiede IPv6 Adresse zuzuweisen.

```
# vzctl set 108 --ipadd 2a01:4f8:101:1145::8:1 --save
Adding IP address(es): 2a01:4f8:101:1145::8:1
Saved parameters for CT 108

# vzctl set 108 --ipadd 2a01:4f8:101:1145::8:2 --save
Adding IP address(es): 2a01:4f8:101:1145::8:2
Saved parameters for CT 108

# vzctl set 108 --ipadd 2a01:4f8:101:1145::8:3 --save
Adding IP address(es): 2a01:4f8:101:1145::8:3
Saved parameters for CT 108

# vzctl restart 108
Restarting container
Stopping container ...
Container was stopped
Container is unmounted
Starting container ...
Container is mounted
Adding IP address(es): 192.168.11.108 2a01:4f8:101:1145::8:1 2a01:4f8:101:1145::8:2
2a01:4f8:101:1145::8:3
SIOCSIFADDR: Network is down
SIOCSIFADDR: Network is down
SIOCSIFADDR: Network is down
Setting CPU units: 1000
Configure meminfo: 1560576
Set hostname: ng.die-lega.org
File resolv.conf was modified
Container start in progress...
```

Abb. 48: Adresszuweisung von IPv6 Adressen und Neustart des virtuellen Servers

3.11.2 Konfiguration der Firewall:

Da es sich bei dem virtuallen Server ebenfalls um einen Webserver handelt kann die Firewall Konfigruation des Server H1 für diesen Server übernommen werden und so werden alle Adressen aus dem Netz `2a01:4f8:101:1145::8:1/112` für diesen Server und für die verscheidenen Dienste erlaubt.

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
:LEGA-WEBSPACE-INPUT - [0:0]

-A FORWARD -s 2a01:4f8:101:1145::1:1/112 -j ACCEPT
-A FORWARD -d 2a01:4f8:101:1145::1:1/112 -j LEGA-WEBSPACE-INPUT

-A FORWARD -s 2a01:4f8:101:1145::8:1/112 -j ACCEPT
-A FORWARD -d 2a01:4f8:101:1145::8:1/112 -j LEGA-WEBSPACE-INPUT

...

-A LEGA-WEBSPACE-INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p icmpv6 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p udp -m udp --dport 20 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p udp -m udp --dport 21 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p udp -m udp --dport 53 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 20000 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 10000 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 443 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 993 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 143 -j ACCEPT
...
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 1935 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 22 -j ACCEPT

COMMIT
```

Abb. 49: Firewall Konfiguration für mehrere virtuelle Webserver (/etc/sysconfig/ip6tables)

Nach einem Neustart der Firewall kann der virtuelle Webserver H2 von außen erreicht werden.

Die weiter Konfiguration ist identisch mit der des Webserver H1.

4 Ergebnisse / Diskussion

Die Umstellung des gehosteten Servers ließ sich ohne Problem durchführen. Das Linux Betriebssystem centOS unterstützt schon lange IPv6. Auch die verwendete Virtualisierungstechnologie opvenVZ ist zu IPv6 kompatibel, auch wenn es in der verwendeten Version noch kleine Einschränkungen bei der Netzwerkkonfiguration gibt, welche sich aber nicht negativ auf diese Migration ausgewirkt haben.

Obwohl eine relativ aktuelle Version der Verwaltungssoftware Virtualmin verwendet wurde, waren deren IPv6 Feature noch nicht völlig Implementiert und auch nicht Fehlerfrei. Aus diesem Grund wurden die für den Webserver benötigten Services manuell und einzeln auf den Betrieb über IPv6 konfiguriert.
Die dabei verwendeten Serverprogramme wie Apache, oder proFTPd waren alle voll kompatibel zu IPv6 und die Unterstützung war bereits Standardmäßig aktiviert oder konnte durch kleine Konfigurationsänderungen aktiviert werden.

Die zuvor noch mühsam konfigurierten virtuellen Server, die aus Kostengründen keine eigene öffentlich IPv4 Adresse erhalten haben können mit IPv6 alle eine öffentliche Adresse erhalten und somit kann das arbeiten und konfigurieren von NAT und Portforwading entfallen.

5 Zusammenfassung

Die Migration eines vorhanden IPv4 Servers, im speziellen eines Webservers, ist nicht so aufwendig wie man vermuten würde. Jedoch kann der Aufwand sehr schnell steigen wenn veraltete oder inkompatible Software verwendet wird. Auch wird der Umstieg, abhängig von der Größe des Servers oder der Infrastruktur Aufwendiger.

Man sollte daher die Migration auf IPv6 möglich früh planen und diese mit anderen Softwareupdates oder Hardwareupgrades verbinden.

Da das IPv6 Protokoll schon mit dem Gedanken eines möglichst unkomplizierten Umstiegs entwickelt wurde gibt es von Seiten des Protokolls kaum Probleme.

Probleme bereitet am häufigsten die verwendete Software, da diese entweder nicht für IPv6 entwickelt wurde oder noch nicht auf IPv6 angepasst wurde. Viele Softwarehersteller sind sich diesem Problem jedoch bewusst und somit wurde schon früh der Support für IPv6 integriert und Möglichkeiten geschaffen einen unkomplizierten Umstieg zu ermöglichen.

6 Verzeichnisse

6.1 Literaturverzeichnis

Hagen, Silvia. *Plannung for IPv6.* O'Reilly Media, 2011.

Jarzyna, Dirk. *IPv6 Praxishandbuch.* mitp/bhv, 2011.

Hetzner Online AG. *Hetzner - DokuWiki*.
http://wiki.hetzner.de (Zugriff im Juni 2012).

Bieringer, Peter. *Linux IPv6 HOWTO (en).*
http://tldp.org/HOWTO/Linux+IPv6-HOWTO/ (Zugriff im Juli 2012).

Network Working Group. „Internet Protocol, Version 6 (IPv6) – Specification.“
http://tools.ietf.org/html/rfc2460.

OpenVZ Linux Containers Wiki.
http://wiki.openvz.org (Zugriff im Mai 2012).

Red Hat Enterprise Linux - Deployment Guide.
http://www.centos.org/docs/5/html/5.2/Deployment_Guide/ (Zugriff im Juni 2012).

SixXS - IPv6 Deployment & Tunnel Broker. *SixxS Frequently Asked Questions.*
http://www.sixxs.net/faq/ (Zugriff im Juli 2012).

SixxS Wiki.
http://www.sixxs.net/wiki/SixXS_Wiki (Zugriff in Juli 2012).

The Apache Software Foundation. *Apache HTTP Server Version 2.2 Documentation.*
http://httpd.apache.org/docs/2.2/en (Zugriff im Juni 2012).

6.2 Abkürzungsverzeichnis

IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
KMU	Kleine und mittlere Unternehmen
IP	Internet Protocol
ISP	Internet Service Provider
IT	Informationstechnik
TCP	Transmission Control Protocol
IANA	Internet Assigned Numbers Authority
URL	Uniform Resource Locator
NAT	Network Address Translation
PAT	Port Address Translation
IPSec	Internet Protocol Security
DHCP	Dynamic Host Configuration Protocol
MAC	Media-Access-Control
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
FTP	File Transfer Protocol
FTPS	File Transfer Protocol Secure
SSH	Secure Shell
IMAP	Internet Message Access Protocol
POP3	Post Office Protocol
SMTP	Simple Mail Transfer Protocol
SNAT	Source Network Address Translation
NDP	Neighbor Discovery Protocol
VM	Virtuelle Maschine
SSL	Secure Sockets Layer
SNI	Server Name Indication

6.3 Abbildungen

7 Anhang

7.1 IPv4 Firewall Konfiguration

```
# Generated by iptables-save v1.3.5 on Tue Apr 13 19:23:52 2010
*nat
:PREROUTING ACCEPT [1089548:70917337]
:POSTROUTING ACCEPT [1276153:82689913]
:OUTPUT ACCEPT [299322:19589025]
:LEGA-PORTFORWARD - [0:0]
-A PREROUTING -j LEGA-PORTFORWARD
-A POSTROUTING -s 10.10.8.0/255.255.255.0 -o eth0 -j SNAT --to-source 78.46.51.227
-A POSTROUTING -s 192.168.11.0/255.255.255.0 -o eth0 -j SNAT --to-source 78.46.51.227
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 20 -j DNAT --to-destination
192.168.11.108:20
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 21 -j DNAT --to-destination
192.168.11.108:21
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p udp -m udp --dport 20 -j DNAT --to-destination
192.168.11.108:20
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p udp -m udp --dport 21 -j DNAT --to-destination
192.168.11.108:21
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 25 -j DNAT --to-destination
192.168.11.108:25
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 80 -j DNAT --to-destination
192.168.11.108:80
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 143 -j DNAT --to-destination
192.168.11.108:143
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 110 -j DNAT --to-destination
192.168.11.108:110
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 993 -j DNAT --to-destination
192.168.11.108:993
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 995 -j DNAT --to-destination
192.168.11.108:995
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 443 -j DNAT --to-destination
192.168.11.108:443
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 10305 -j DNAT --to-
destination 192.168.11.108:10305
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p udp -m udp --dport 10305 -j DNAT --to-
destination 192.168.11.108:10305
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 10315 -j DNAT --to-
destination 192.168.11.108:10315
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p udp -m udp --dport 10325 -j DNAT --to-
destination 192.168.11.108:10325
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 10400:10499 -j DNAT --to-
destination 192.168.11.108:10400-10499
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 10500:10599 -j DNAT --to-
destination 192.168.11.108:10500-10599
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 10000 -j DNAT --to-
destination 192.168.11.108:10000
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 20000 -j DNAT --to-
destination 192.168.11.108:20000

-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 236 -j DNAT --to-destination
192.168.11.104:236
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 8081 -j DNAT --to-
destination 192.168.11.104:80

-A LEGA-PORTFORWARD -d 78.46.51.253 -i eth0 -p tcp -m tcp --dport 3979 -j DNAT --to-
destination 192.168.11.109:3979
-A LEGA-PORTFORWARD -d 78.46.51.253 -i eth0 -p udp -m udp --dport 3979 -j DNAT --to-
destination 192.168.11.109:3979
-A LEGA-PORTFORWARD -d 78.46.51.253 -i eth0 -p tcp -m tcp --dport 25565 -j DNAT --to-
destination 192.168.11.109:25565

-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 22101 -j DNAT --to-
destination 192.168.11.101:22
```

```
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 22102 -j DNAT --to-
destination 192.168.11.102:22
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 22103 -j DNAT --to-
destination 192.168.11.103:22
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 22104 -j DNAT --to-
destination 192.168.11.104:22
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 22105 -j DNAT --to-
destination 192.168.11.105:22
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 22106 -j DNAT --to-
destination 192.168.11.106:22
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 22108 -j DNAT --to-
destination 192.168.11.108:22
-A LEGA-PORTFORWARD -d 78.46.51.227 -i eth0 -p tcp -m tcp --dport 22109 -j DNAT --to-
destination 192.168.11.109:22
COMMIT


# Completed on Tue Apr 13 19:23:52 2010
# Generated by iptables-save v1.3.5 on Tue Apr 13 19:23:52 2010
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [11:638]
:OUTPUT ACCEPT [26618377:4260125111]

:LEGA-LOCAL-FORWARD - [0:0]
:LEGA-ROOTSERVER-INPUT - [0:0]
:LEGA-SERVER-FORWARD - [0:0]
:LEGA-WEBSPACE-INPUT - [0:0]
:RH-Firewall-1-INPUT - [0:0]

-A INPUT -j LEGA-ROOTSERVER-INPUT
-A INPUT -j RH-Firewall-1-INPUT

-A FORWARD -j LEGA-OPENVPN-FORWARD
-A FORWARD -j LEGA-LOCAL-FORWARD
-A FORWARD -j LEGA-SERVER-FORWARD

-A LEGA-LOCAL-FORWARD -d 192.168.11.0/255.255.255.0 -j ACCEPT
-A LEGA-LOCAL-FORWARD -s 192.168.11.0/255.255.255.0 -j ACCEPT
-A LEGA-OPENVPN-FORWARD -s 10.10.8.0/255.255.255.0 -j ACCEPT
-A LEGA-OPENVPN-FORWARD -d 10.10.8.0/255.255.255.0 -j ACCEPT
-A LEGA-ROOTSERVER-INPUT -p udp -m udp --dport 1194 -j ACCEPT
-A LEGA-ROOTSERVER-INPUT -p tcp -m tcp --dport 1194 -j ACCEPT
-A LEGA-ROOTSERVER-INPUT -s 78.46.51.253 -p tcp -m tcp --dport 22 -m state --state NEW -j
ACCEPT
-A LEGA-ROOTSERVER-INPUT -p tcp -m tcp --dport 22 -m state --state NEW -m limit --limit 1/min
--limit-burst 3 -j ACCEPT
-A LEGA-ROOTSERVER-INPUT -p tcp -m tcp --dport 22 -m state --state NEW -j DROP
-A LEGA-ROOTSERVER-INPUT -p tcp -m tcp --dport 4949 -j ACCEPT
-A LEGA-SERVER-FORWARD -s 78.46.51.253 -j ACCEPT
-A LEGA-SERVER-FORWARD -d 78.46.51.253 -j LEGA-WEBSPACE-INPUT
-A LEGA-SERVER-FORWARD -s 78.46.51.252 -j ACCEPT
-A LEGA-SERVER-FORWARD -d 78.46.51.252 -j LEGA-WEBSPACE-INPUT
-A LEGA-WEBSPACE-INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p icmp -m icmp --icmp-type any -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p udp -m udp --dport 20 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p udp -m udp --dport 21 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p udp -m udp --dport 53 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 20000 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 10000 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 443 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 993 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 143 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 995 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p udp -m udp --dport 9987 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 10011 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 30033 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 110 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 20 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 21 -j ACCEPT
```

```
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 53 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 25 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 3690 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 1935 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p tcp -m tcp --dport 25565 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -p udp -m udp --dport 25565 -j ACCEPT
-A LEGA-WEBSPACE-INPUT -j REJECT --reject-with icmp-port-unreachable

-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp -m icmp --icmp-type any -j ACCEPT
-A RH-Firewall-1-INPUT -p esp -j ACCEPT
-A RH-Firewall-1-INPUT -p ah -j ACCEPT
-A RH-Firewall-1-INPUT -d 224.0.0.251 -p udp -m udp --dport 5353 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT

# Completed on Tue Apr 13 19:23:52 2010
# Generated by iptables-save v1.3.5 on Tue Apr 13 19:23:52 2010
*mangle
:PREROUTING ACCEPT [220350101:278255638471]
:INPUT ACCEPT [168536507:243164502744]
:FORWARD ACCEPT [51813594:35091135727]
:OUTPUT ACCEPT [26618377:4260125111]
:POSTROUTING ACCEPT [78434191:39351543401]
COMMIT
# Completed on Tue Apr 13 19:23:52 2010
```

7.2 openVZ Konfiguration

```
## Global parameters
VIRTUOZZO=yes
LOCKDIR=/vz/lock
DUMPDIR=/vz/dump
VE0CPUUNITS=1000

## Logging parameters
LOGGING=yes
LOGFILE=/var/log/vzctl.log
LOG_LEVEL=0
VERBOSE=0

## Disk quota parameters
DISK_QUOTA=yes
VZFASTBOOT=no

# Disable module loading. If set, vz initscript do not load any modules.
#MODULES_DISABLED=yes

# The name of the device whose IP address will be used as source IP for CT.
# By default automatically assigned.
#VE_ROUTE_SRC_DEV="eth0"

# Controls which interfaces to send ARP requests and modify APR tables on.
NEIGHBOUR_DEVS=detect

## Template parameters
TEMPLATE=/vz/template

## Defaults for containers
VE_ROOT=/vz/root/$VEID
VE_PRIVATE=/vz/private/$VEID
CONFIGFILE="vps.basic"
DEF_OSTEMPLATE="fedora-core-4"

## Load vzwdog module
VZWDOG="no"

## IPv4 iptables kernel modules
IPTABLES="ipt_REJECT ipt_tos ipt_limit ipt_multiport iptable_filter iptable_mangle ipt_TCPMSS
ipt_tcpmss ipt_ttl ipt_length"

## Enable IPv6
IPV6="yes"

## IPv6 ip6tables kernel modules
IP6TABLES="ip6_tables ip6table_filter ip6table_mangle ip6t_REJECT"
```

7.3 DNS Zone Konfiguration

```
$TTL 86400
@   IN SOA ns1.first-ns.de. postmaster.robot.first-ns.de. (
    2012042502   ; serial
    14400        ; refresh
    1800         ; retry
    604800       ; expire
    86400 )      ; minimum

@                       IN NS      robotns3.second-ns.com.
@                       IN NS      robotns2.second-ns.de.
@                       IN NS      ns1.first-ns.de.

@                       IN A       62.99.149.131
dual                    IN A       78.46.51.253
lansuite                IN A       62.99.149.131
localhost               IN A       127.0.0.1
mail                    IN A       62.99.149.131
minecraft               IN A       91.118.87.120
ng                      IN A       78.46.51.227
rootserver              IN A       78.46.51.227
teamspeak               IN A       91.118.87.120
tf2                     IN A       91.118.87.120
webspace                IN A       78.46.51.253
www                     IN A       62.99.149.131
dual                    IN AAAA    2a01:4f8:101:1145::1:1
ipv6                    IN AAAA    2a01:4f8:101:1145::1:1
lansuite                IN AAAA    2a01:4f8:101:1145::1:2
ripv6                   IN AAAA    2a01:4f8:101:1145::2
*                       IN CNAME   www
ftp                     IN CNAME   www
imap                    IN CNAME   mail
loopback                IN CNAME   localhost
pop                     IN CNAME   mail
relay                   IN CNAME   mail
smtp                    IN CNAME   mail
verein                  IN CNAME   www
virtual                 IN CNAME   rootserver
vpn                     IN CNAME   rootserver
wiki                    IN CNAME   www
@                       IN MX 10   mail
```

7.4 Apache Server Konfiguration

```
Listen *:80

<VirtualHost [2a01:4f8:101:1145::1:1]:80>
SuexecUserGroup "#500" "#501"
ServerName die-lega.org
ServerAlias www.die-lega.org
ServerAlias webmail.die-lega.org
ServerAlias admin.die-lega.org
ServerAlias dual.die-lega.org
ServerAlias www.dual.die-lega.org
ServerAlias ipv6.die-lega.org
ServerAlias www.ipv6.die-lega.org
ServerAlias webspace.die-lega.org
ServerAlias www.webspace.die-lega.org
DocumentRoot /home/die-lega/public_html
ErrorLog /var/log/virtualmin/die-lega.org_error_log
CustomLog /var/log/virtualmin/die-lega.org_access_log combined
ScriptAlias /cgi-bin/ /home/die-lega/cgi-bin/
ScriptAlias /awstats /home/die-lega/cgi-bin
DirectoryIndex index.html index.htm index.php index.php4 index.php5
<Directory /home/die-lega/public_html>
Options -Indexes +IncludesNOEXEC +FollowSymLinks
allow from all
AllowOverride All
</Directory>
</VirtualHost>

Verschiede Virtuelle Hosts auf verschiedenen IP Adressen:

<VirtualHost 78.46.51.253:80>
SuexecUserGroup "#500" "#501"
ServerName die-lega.org
ServerAlias www.die-lega.org
ServerAlias webmail.die-lega.org
ServerAlias admin.die-lega.org
ServerAlias dual.die-lega.org
ServerAlias www.dual.die-lega.org
ServerAlias ipv6.die-lega.org
ServerAlias www.ipv6.die-lega.org
ServerAlias webspace.die-lega.org
ServerAlias www.webspace.die-lega.org
DocumentRoot /home/die-lega/public_html
ErrorLog /var/log/virtualmin/die-lega.org_error_log
CustomLog /var/log/virtualmin/die-lega.org_access_log combined
ScriptAlias /cgi-bin/ /home/die-lega/cgi-bin/
ScriptAlias /awstats /home/die-lega/cgi-bin
DirectoryIndex index.html index.htm index.php index.php4 index.php5
<Directory /home/die-lega/public_html>
Options -Indexes +IncludesNOEXEC +FollowSymLinks
allow from all
AllowOverride All
</Directory>
</VirtualHost>

<VirtualHost [2a01:4f8:101:1145::1:1]:80>
SuexecUserGroup "#500" "#501"
ServerName die-lega.org
ServerAlias www.die-lega.org
ServerAlias webmail.die-lega.org
ServerAlias admin.die-lega.org
ServerAlias dual.die-lega.org
ServerAlias www.dual.die-lega.org
ServerAlias ipv6.die-lega.org
ServerAlias www.ipv6.die-lega.org
ServerAlias webspace.die-lega.org
ServerAlias www.webspace.die-lega.org
```

```
DocumentRoot /home/die-lega/public_html
ErrorLog /var/log/virtualmin/die-lega.org_error_log
CustomLog /var/log/virtualmin/die-lega.org_access_log combined
ScriptAlias /cgi-bin/ /home/die-lega/cgi-bin/
ScriptAlias /awstats /home/die-lega/cgi-bin
DirectoryIndex index.html index.htm index.php index.php4 index.php5
<Directory /home/die-lega/public_html>
Options -Indexes +IncludesNOEXEC +FollowSymLinks
allow from all
AllowOverride All
</Directory>
</VirtualHost>

<VirtualHost 78.46.51.253:80>
SuexecUserGroup "#500" "#501"
ServerName lansuite.die-lega.org
ServerAlias www.lansuite.die-lega.org
ServerAlias webmail.lansuite.die-lega.org
ServerAlias admin.lansuite.die-lega.org
DocumentRoot /home/die-lega/domains/lansuite.die-lega.org/public_html
ErrorLog /var/log/virtualmin/lansuite.die-lega.org_error_log
CustomLog /var/log/virtualmin/lansuite.die-lega.org_access_log combined
ScriptAlias /cgi-bin/ /home/die-lega/domains/lansuite.die-lega.org/cgi-bin/
ScriptAlias /awstats /home/die-lega/domains/lansuite.die-lega.org/cgi-bin
DirectoryIndex index.html index.htm index.php index.php4 index.php5
<Directory /home/die-lega/domains/lansuite.die-lega.org/public_html>
Options -Indexes +IncludesNOEXEC +FollowSymLinks
allow from all
AllowOverride All
</Directory>
</VirtualHost>

<VirtualHost [2a01:4f8:101:1145::1:2]:80>
SuexecUserGroup "#500" "#501"
ServerName lansuite.die-lega.org
ServerAlias www.lansuite.die-lega.org
ServerAlias webmail.lansuite.die-lega.org
ServerAlias admin.lansuite.die-lega.org
DocumentRoot /home/die-lega/domains/lansuite.die-lega.org/public_html
ErrorLog /var/log/virtualmin/lansuite.die-lega.org_error_log
CustomLog /var/log/virtualmin/lansuite.die-lega.org_access_log combined
ScriptAlias /cgi-bin/ /home/die-lega/domains/lansuite.die-lega.org/cgi-bin/
ScriptAlias /awstats /home/die-lega/domains/lansuite.die-lega.org/cgi-bin
DirectoryIndex index.html index.htm index.php index.php4 index.php5
<Directory /home/die-lega/domains/lansuite.die-lega.org/public_html>
Options -Indexes +IncludesNOEXEC +FollowSymLinks
allow from all
AllowOverride All
</Directory>
</VirtualHost>

<VirtualHost 78.46.51.253:443>
SuexecUserGroup "#500" "#501"
ServerName die-lega.org
ServerAlias www.die-lega.org
ServerAlias webmail.die-lega.org
ServerAlias admin.die-lega.org
DocumentRoot /home/die-lega/public_html
ErrorLog /var/log/virtualmin/die-lega.org_error_log
CustomLog /var/log/virtualmin/die-lega.org_access_log combined
ScriptAlias /cgi-bin/ /home/die-lega/cgi-bin/
ScriptAlias /awstats /home/die-lega/cgi-bin
DirectoryIndex index.html index.htm index.php index.php4 index.php5
<Directory /home/die-lega/public_html>
Options -Indexes +IncludesNOEXEC +FollowSymLinks
allow from all
AllowOverride All
</Directory>
SSLEngine on
SSLCertificateFile /home/die-lega/ssl.cert
SSLCertificateKeyFile /home/die-lega/ssl.key
```

```
</VirtualHost>

<VirtualHost 78.46.51.253:443>
SuexecUserGroup "#500" "#501"
ServerName lansuite.die-lega.org
ServerAlias www.lansuite.die-lega.org
ServerAlias webmail.lansuite.die-lega.org
ServerAlias admin.lansuite.die-lega.org
DocumentRoot /home/die-lega/domains/lansuite.die-lega.org/public_html
ErrorLog /var/log/virtualmin/lansuite.die-lega.org_error_log
CustomLog /var/log/virtualmin/lansuite.die-lega.org_access_log combined
ScriptAlias /cgi-bin/ /home/die-lega/domains/lansuite.die-lega.org/cgi-bin/
ScriptAlias /awstats /home/die-lega/domains/lansuite.die-lega.org/cgi-bin
DirectoryIndex index.html index.htm index.php index.php4 index.php5
<Directory /home/die-lega/domains/lansuite.die-lega.org/public_html>
Options -Indexes +IncludesNOEXEC +FollowSymLinks
allow from all
AllowOverride All
</Directory>
SSLEngine on
SSLCertificateFile /home/die-lega/domains/lansuite.die-lega.org/ssl.cert
SSLCertificateKeyFile /home/die-lega/domains/lansuite.die-lega.org/ssl.key
</VirtualHost>

<VirtualHost [2a01:4f8:101:1145::1:1]:443>
SuexecUserGroup "#500" "#501"
ServerName die-lega.org
ServerAlias www.die-lega.org
ServerAlias webmail.die-lega.org
ServerAlias admin.die-lega.org
DocumentRoot /home/die-lega/public_html
ErrorLog /var/log/virtualmin/die-lega.org_error_log
CustomLog /var/log/virtualmin/die-lega.org_access_log combined
ScriptAlias /cgi-bin/ /home/die-lega/cgi-bin/
ScriptAlias /awstats /home/die-lega/cgi-bin
DirectoryIndex index.html index.htm index.php index.php4 index.php5
<Directory /home/die-lega/public_html>
Options -Indexes +IncludesNOEXEC +FollowSymLinks
allow from all
AllowOverride All
</Directory>
SSLEngine on
SSLCertificateFile /home/die-lega/ssl.cert
SSLCertificateKeyFile /home/die-lega/ssl.key
</VirtualHost>

<VirtualHost [2a01:4f8:101:1145::1:2]:443>
SuexecUserGroup "#500" "#501"
ServerName lansuite.die-lega.org
ServerAlias www.lansuite.die-lega.org
ServerAlias webmail.lansuite.die-lega.org
ServerAlias admin.lansuite.die-lega.org
DocumentRoot /home/die-lega/domains/lansuite.die-lega.org/public_html
ErrorLog /var/log/virtualmin/lansuite.die-lega.org_error_log
CustomLog /var/log/virtualmin/lansuite.die-lega.org_access_log combined
ScriptAlias /cgi-bin/ /home/die-lega/domains/lansuite.die-lega.org/cgi-bin/
ScriptAlias /awstats /home/die-lega/domains/lansuite.die-lega.org/cgi-bin
DirectoryIndex index.html index.htm index.php index.php4 index.php5
<Directory /home/die-lega/domains/lansuite.die-lega.org/public_html>
Options -Indexes +IncludesNOEXEC +FollowSymLinks
allow from all
AllowOverride All
</Directory>
SSLEngine on
SSLCertificateFile /home/die-lega/domains/lansuite.die-lega.org/ssl.cert
SSLCertificateKeyFile /home/die-lega/domains/lansuite.die-lega.org/ssl.key
</VirtualHost>
```