

IPv6-Internetzugang

Lokale Netze über IPv6 mit dem Internet verbinden

Reiko Kaps - 08.07.08



Das Internet-Protokoll in Version 6 überbrückt dank seines riesigen Adressbereichs die Grenzen zwischen lokalen Netzen und dem globalen Internet. Im Test zeigen IPv6-Internetzugänge per Tunnel oder DSL, wie sie LANs ans Internet anbinden.

Der Vorrat an IPv4-Adressen **erschöpft**[1] sich laut verschiedener Studien in den kommenden fünf Jahren. Techniken wie Network Address Translation (**NAT**[2]) und **Port Forwarding**[3] verlängerten bislang die Galgenfrist etwas, allerdings erfordern sie einige Arbeit beim Einrichten und sie funktionieren längst nicht mit jedem Dienst reibungslos. Unter IPv6 sind Adressen rein rechnerisch in Hülle und Fülle vorhanden, was derartige Hilfen überflüssig macht. Im Augenblick steht gut ein Achtel aller IPv6-Adressen für Rechner im Internet bereit, die verbleibenden 297,75 Sextillionen Adressen werden für spezielle Zwecke wie Multicast oder linklokale Adressen genutzt oder sind bislang noch durch die Internet Engineering Task Force (**IETF**[4]) reserviert.

Obwohl sich IPv4 und IPv6 untereinander nicht verstehen, können sie jedoch nebeneinander arbeiten. Der Umstieg zu IPv6 kann also langsam und schleichend stattfinden. Soweit die Theorie, in der Praxis hat sich das neue Protokoll jedoch nur in einigen Regionen der Erde verbreitet. Europa und Nordamerika zeigen **dank üppiger IPv4-Adressbereiche**[5] nur wenig Engagement für IPv6. Die Backbone-Betreiber nutzen es zwar, auch stellt beispielsweise T-Online für Geschäftskunden IPv6-Zugänge bereit, doch gibt es nur von wenigen, zumeist kleinen Anbietern IPv6-fähige Internet-Zugänge für Privatkunden.

Einen anderen Weg ins Internet der nächsten Generation ebnen 6in4-Tunnel, die das Protokoll über bestehende IPv4-Leitungen transportieren. Die beiden Endpunkte des Tunnels verpacken IPv6-Pakete in IPv4-Pakete und senden sie sich gegenseitig zu, sodass man darüber eine Brücke ins IPv6-Internet erhält.

In einem Test haben wir beide Möglichkeiten unter Windows XP, Vista und Linux auf ihre Praxistauglichkeit überprüft. Die Rechner arbeiteten dabei als **Router**[6], die den IPv6-Zugang für andere LAN-Teilnehmer bereitstellen.

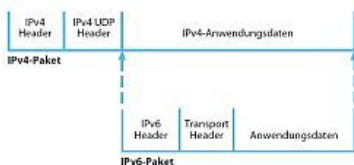
Die grundlegende IPv6-Einrichtung auf Windows und Linux beschreiben die Beiträge **IPv6:Das Mega-Netz**[7] und **IPv6 für kleine Netze**[8]. Damit der Zugang anderer LAN-Rechner ins IPv6-Internet klappt, beleuchtet dieser Praxistest besonders das Routing bei IPv6 und zeigt Möglichkeiten, wie man Internet-Dienste aufruft, die ausschließlich IPv4 beherrschen. Die IPv6-Adressen in den Beispielen entstammen **dem reservierten Präfix 2001:db8::/32**[9], das zur Dokumentation gedacht ist. Für eigene Versuche tauscht man ihn gegen das eigene, vom DSL- oder Tunnel-Provider zugewiesene IPv6-Präfix aus.

Beim Internet-Zugang per 6in4-Tunnel setzten wir auf den **Tunnelbroker sixxs.net**[10], der in Deutschland Tunnelendpunkte bereitstellt. Für den Tunnelaufbau haben die beiden Sixxs-Betreiber Pim van Pelt und Jeroen Massar die **Client-Software aiccu**[11] entwickelt, die unter Windows, zahlreichen Linux- und BSD-Distributionen läuft. Für andere Betriebssysteme steht der Quelltext bereit.

In einem weiteren Test-Netz brachten wir mittels eines DSL-Zugangs von Titan-DSL Linux und Vista-Rechner ins IPv6-Internet. Windows XP beherrscht die IPv6-Einwahl über **PPP**[12] leider nicht, sodass es hier nicht antreten konnte. Erst nach Abschluss des Tests kam der Einwahltreiber **cFos IPv6 Link**[13] auf den Markt, sodass wir ihn nicht berücksichtigen konnten.

Die IPv6-Router-Rolle übernimmt im Testbetrieb ein Notebook oder ein herkömmlicher PC. Viele Firmen-Router wie die von Cisco beherrschen IPv6 von Hause aus, bei Heim-Routern vermisst man es jedoch meist. Anders ist es bei Modellen, die mit freien Firmware-Varianten wie OpenWRT oder DD-WRT laufen. Sie sprechen IPv6 und bringen alle nötigen Hilfsmittel mit. Da die WRT-Firmware-Versionen auf Linux aufsetzen, **weicht die IPv6-Einrichtung**[14] kaum von der eines Standard-Linux ab. Zusätzliche Hilfen bieten die Projektseiten von **OpenWRT**[15] und **DD-WRT**[16].

IPv6-Schleuser



Ein 6in4-Tunnel verpackt IPv6-Pakete ins IPv4-Paketformat und sendet sie an entsprechende Tunnel-Gegenstellen, die sie extrahieren und weiterleiten.

Spricht der eigene DSL-Zugang kein IPv6, schleust man die IPv6-Pakete über das bestehende IPv4-Netz zu einem Tunnelserver, der sie ins IPv6-Netz weiterreicht. In diesem 6in4-Tunnel sind die IPv6-Pakete nur Nutzdaten eines IPv4-Pakets, das seinen Weg in der bestehenden Infrastruktur findet.

Der Sixxs-Client baut Tunnel auf, die das **Verfahren AYIYA**[17] nutzen, das auch hinter NAT-Router und Firewalls funktioniert. Die bestehende IPv4-Konfiguration aus DSL-Router, **Switch**[18] und LAN bleibt also bestehen und funktionsfähig.

Formalitäten

Zuerst muss man sich jedoch auf der Sixxs-Website anmelden und die Bestätigung des Betreibers abwarten.

Nach der ersten Anmeldung auf der Website beantragt man einen Tunnel, wobei man seinen Standort und einige Gründe für die Nutzung angeben muss. Genehmigt der Betreiber den Tunnel, landen die Zugangsdaten kurze Zeit später im eigenen Postfach.

Damit füttert man nun den Sixxs-Client, der für den Aufbau der IPv6-Verbindung zuständig ist. Auch das weltweit erreichbare IPv6-Präfix beantragt man per Formular und bekommt es nach einer Prüfung per E-Mail zugesandt.

Arbeitet der eigene Rechner unter Debian oder Ubuntu, reicht für die Installation des Aiccu-Clients das Kommando `apt-get install aiccu`. Pakete und Quelltexte für andere Distributionen wie Fedora oder OpenWRT, Unixe wie FreeBSD oder Mac OS X finden sich **auf der Sixxs-Homepage**[19].

Für Windows steht zusätzlich ein grafisches Frontend bereit. Steht der eigene Tunnelendpunkt hinter einem IPv4-Router, benötigt aiccu einen TUN/TAP-Treiber, der virtuelle Netzwerkschnittstellen erstellt und verwaltet. Ein aktuelles Linux bringt den Treiber bereits mit. Sollte er nicht automatisch starten, hilft

auf einer Root-Konsole ein Aufruf von `modprobe tun`.

Für die verschiedenen Windows-Versionen finden sich beim Sixxs Zip-Archive eines TUN/TAP-Treibers. Die Einrichtung erledigt eine Batch-Datei, die allerdings von einer englischen Windows-Installation ausgeht. Sie ruft die benötigten Inf-Dateien und Programme über den Pfad `c:\Program Files` auf, was bei einem deutschen Windows XP fehlschlägt. Ändert man den Pfad in `%Program-Files%`, ist der Fehler behoben. Nach der Installation von `aiccu` nimmt die Datei `aiccu.conf` die Zugangsdaten für den Sixxs-Tunnel auf: Ist sie noch nicht vorhanden, finden sich Vorlagen auf Linux unter `/usr/share/doc/aiccu/examples` oder auf der [Sixxs-Website](#)[20].

Unter Windows sucht die Textversion des Sixxs-Clients im Windows-Verzeichnis nach der Datei `aiccu.conf` – also unter `%windir%`. Auf Linux muss sie im Verzeichnis `/etc` liegen. Für eine Grundeinrichtung reichen in der Datei `aiccu.conf` korrekte Werte für `username`, `password` und `server`, der in unserem Fall `tic.sixxs.net` lautet und den Anmelde-server bei `sixxs.net` benennt.

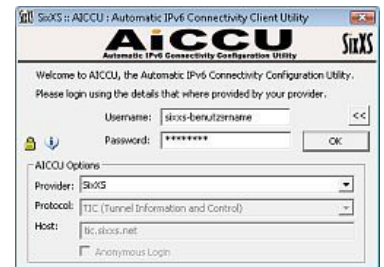
Soll das Programm den Tunnel automatisch aktivieren, muss `automatic true` vorhanden sein. Mit dem Eintrag `daemonize false` verhindert man, dass das Programm im Hintergrund arbeitet, was für Tests nützlich sein kann.

```
username SIXXS-USERNAME
password SIXXS-PASSWORD
server tic.sixxs.net
automatic true
verbose true
daemonize false
behindnat true
```

Die Option `behindnat true` signalisiert `aiccu`, dass der Rechner hinter einem NAT-Router arbeitet. Über `setupscript` soll `aiccu` zudem ein zusätzliches Einrichtungsskript aufrufen können, was im Test allerdings nicht gelang.

Anschließend startet der Tunnel-Client auf einer Unix-Konsole oder der Windows-Eingabeaufforderung mit dem Kommando `aiccu start`. Treten dabei Fehler auf, überprüft man mittels `aiccu test` die Tunnelverbindung. Das Programm führt dabei mehrere Tests durch und gibt nach jedem Schritt einige Hinweise auf mögliche Fehler. Arbeitet `aiccu` im Hintergrund oder als Systemdienst, beendet der Befehl `aiccu stop` einen Tunnel. Arbeitet es im Vordergrund, reicht die Tastenkombination `Strg-C`.

Unter Windows übernimmt alternativ eine grafische Aiccu-Version diese Aufgaben: Sie fragt per Dialogbox Benutzername und Passwort ab und baut auf Knopfdruck oder vollautomatisch die Verbindung auf. Diese Einstellungen speichert das Programm jedoch nicht automatisch. Der Menüpunkt "Speichern" versteckt sich hinter dem Fenster-Icon oben links.



Unter Windows steht `aiccu` auch als grafisches Programm bereit, andere Betriebssysteme müssen sich mit der Textversion begnügen.

Fallstricke

Nachdem `aiccu` den Tunnel aufgebaut hat, zeigt die Web-Seite www.sixxs.net/main[21], ob man per IPv4 oder IPv6 unterwegs ist. IPv6-taugliche Anwendungen wie Firefox oder der IE 7 suchen sich zuerst per IPv6 den Weg, sofern man diese Fähigkeit nicht abgeschaltet hat. Auf einem Windows XP funktionierte das auf Anhieb, ein Debian-Rechner mit dem Kernel 2.6.20 benötigte jedoch noch etwas Nachhilfe.

Die von `aiccu` gesetzte Default-Route auf das Tunnel-Interface scheint dem IPv6-Stack des Linux-Kernels nicht zu reichen – er verweigerte jegliche Netzwerkverbindung. Das erwünschte Ergebnis brachte erst eine zusätzliche Route auf das Präfix `2000::/3`, die man als Root-Benutzer mittels `ip -6 route add 2000::/3 dev sixxs` hinzufügt. Das Präfix `2000::/3` bezeichnet alle momentan nutzbaren, also nicht **durch die Internet Engineering Task Force reservierten IPv6-Unicast-Adressen**[22].

Die gesamte aktuelle Routing-Tabelle zeigen unter Linux die Befehle `route -6` oder `ip -6 route list an`, auf der Windows-Eingabeaufforderung erledigt das Kommando `route PRINT -6` die gleiche Aufgabe.

Soll der Sixxs-Tunnel automatisch unter Debian oder einem seiner Abkömmlinge starten, bindet man `aiccu` in die Netzwerk-Skripte unterhalb von `/etc/network` ein. Der Sixxs-Client `aiccu` muss dabei als Daemon arbeiten, was der Eintrag `daemonize true` in der Datei `/etc/aiccu.conf` erzwingt. Anschließend erstellt man einen neuen Eintrag in der Datei `interfaces`:

```
iface sixxs inet6 manual
    up /etc/init.d/aiccu start
    post-up ip -6 route add 2000::/3 dev sixxs
    pre-down ip -6 route del 2000::/3 dev sixxs
    down /etc/init.d/aiccu stop
```

Ein vollständiges Beispiel und weitere Skripte für die Einrichtung finden Sie in [dieser Zip-Datei](#)[23], weitere Details zu Debian-Netzwerkeinrichtung verraten die Manualpage von `ifupdown` und der Artikel [Spurenleser](#)[24].

IPv6 per DSL

Bislang gibt es nur wenige Provider, die einen IPv6-Zugang für Privatkunden anbieten. Dazu gehören in Deutschland Titan-DSL, Tal.de und Spacenet, die alle eine T-DSL-Leitung voraussetzen. Tal.de erklärte der `c't` allerdings, dass das Unternehmen für Geschäftskunden auch **SDSL**[25]-Zugänge über Leitungen von Arcor, QSC und Telefonica mit IPv6 versorgen kann. Andere große DSL-Anbieter wie Arcor und Hansenet oder Mobilfunk-Provider wie Vodafone und E-Plus bekunden bislang nur Interesse an dem neuen Protokoll, Angebote haben sie aber noch nicht im Programm.

Wir nutzen für die Tests einen Zugang des DSL-Providers Titan-DSL, dessen IPv6-Zugang über jede T-DSL-Leitung funktioniert und ohne IPv4 auskommt. Neben Benutzernamen und Passwort erhält man außerdem ein IPv6-Netzwerk-Präfix, das Titan-DSL auf die Einwahlverbindung routet. Benutzern von Windows XP bleibt die IPv6-Einwahl verwehrt: Der IPv6-Stack von XP arbeitet über **Ethernet**[26] oder Tunnel, versteht sich jedoch nicht mit Einwahlverbindungen. Microsofts neuestes Kind Vista, die verschiedenen Linux-Distributionen und andere Unixe wählen sich jedoch mit Bordmitteln ein.



Die Sixxs-Website zeigt im unteren Bereich an, ob man per IPv4 oder übers IPv6-Netz surft – samt IP-Adresse und Whois-Anfrage, die den Besitzer der Adresse identifiziert.

Linux nutzt für die Einwahl den Point-to-Point-Protocol-Daemon `pppd` und den PPP-over-Ethernet-Client `pppoe`[27], die allen modernen Distributionen beiliegen. Zugangsdaten wie DSL-Benutzername und Passwort hinterlegt man in der Datei `/etc/ppp/pap-secrets`. Die DSL-Zugangsparameter nimmt die Datei `/etc/ppp/peers/dsl-provider` auf, aus der das Kommando `pon` per Vorgabe seine Einstellungen liest. Besitzt man mehrere DSL-Zugänge, legt man unterhalb von `/etc/ppp/peers` eine zusätzliche Konfigurationsdatei an – beispielsweise `dsl-v6`. Diese Datei enthält neben den üblichen Einstellungen für die Verbindungssteuerung (`mtu`, `mru`, `persist`) die Optionen `noip` und `+ipv6`, die IPv4 ab- und IPv6 anschalten:

```
pty "pppoe -I eth0 -m 1404 -U "
linkname default0
noip
defaultroute
...
+ipv6 ipv6cp-use-ipaddr
debug
```

Möchte man sich mit diesen Einstellungen einwählen, ruft man `pon` mit dem Dateinamen `dsl-v6` als Parameter auf. Bei Titan-DSL erhält der Einwahlrechner im Unterschied zu einer Einwahl über IPv4 keine IP-Adresse vom Provider zugewiesen, was man dem PPP-Daemon mit der Option `noip` mitteilt. Die lokalen Adressen setzt man stattdessen per Skript: Liegt es unterhalb des Verzeichnisses `/etc/ppp/ipv6-up.d`, startet der `pppd` es bei der Einwahl automatisch.

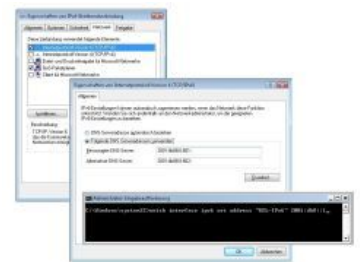
```
#!/bin/sh
PPP_IFACE="$1"
PRG="/sbin/ip -6"
IPV6ADDR=2001:db8::1/48
$PRG addr add $IPV6ADDR dev $PPP_IFACE
$PRG route add default dev $PPP_IFACE
$PRG route add 2000::/3 dev $PPP_IFACE
```

Das Skript verpasst dem PPP-Interface eine global gültige IPv6-Adresse und legt eine Default-Route auf das PPP-Interface sowie eine Route auf das IPv6-Präfix `2000::/3`. Der `pppd`-Daemon übergibt dem Skript außerdem einige Parameter, von denen hier nur der Interface-Name `PPP_IFACE` wichtig ist.

Vista ruft IPv6

Die Einwahl für eine Breitbandverbindung unter Windows Vista konfiguriert man über einige Dialoge, die sich im "Netzwerk und Freigabecenter" verstecken und die der Menüpunkt "Verbindung mit einem Netzwerk herstellen" startet. Der erste Punkt "Verbindung mit dem Internet herstellen" ruft anschließend den Assistenten für DSL- und Einwahlverbindungen auf. Dort schaltet man beim Anlegen der Verbindung in den Eigenschaften das bei Titan-DSL unnötige IPv4 ab.

Leider bringt Vista keine Dialoge für die Eingabe von festen IPv6-Adressen bei Einwahlverbindungen mit, wie sie für Ethernet- oder **WLAN**[28]-Verbindungen bereitstehen. Bei der IPv6-Einwahl greift man daher auf das Konsolenprogramm `netsh` zurück, dessen Kommandos über einen Neustart des Rechners erhalten bleiben.



Die Vista-Dialoge für eine IPv6-Einwahlverbindung nehmen lediglich die Adressen der DNS-Server entgegen. Routing und lokale Adresse legt man per Kommandozeile fest.

```
netsh interface ipv6 set address "DSL-IPv6" 2001:db8::1
```

Die letzten beiden Parameter beziehen sich auf den Namen der Einwahl-Verbindung und eine gültige Adresse aus dem vom Provider zugeteiltem IPv6-Bereich. Beide muss man für eigene Experimente anpassen.

Des Weiteren benötigt der Rechner einige Standard-Routen, wobei das zweite Kommando das eigene Präfix enthalten muss:

```
netsh interface ipv6 add route ::/0 "DSL-IPv6"
netsh interface ipv6 add route 2001:db8::/64 "DSL-IPv6"
```

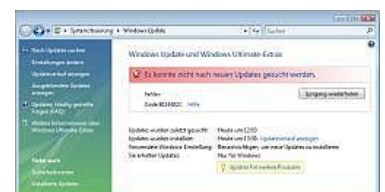
Danach zeigt ein `ping -6 www.sixxs.net`, ob Namensauflösung und Routing funktionieren und die IPv6-Gegenstelle antwortet. Ein Aufruf der Website im Browser führte im Test jedoch zu einer Fehlermeldung. Erst als die Netzwerkkarte, die die physische Verbindung zum DSL-Modem herstellt, eine gültige IPv6-Adresse erhielt, zeigt der Browser die Seite an:

```
netsh interface ipv6 set address "LAN-Verbindung" 2001:db8::2
```

Übergangshilfen

Übermittelt das Netz seine Daten ausschließlich per IPv6, ist der Zugang zu Diensten und Servern blockiert, die nur IPv4 verstehen. Momentan gehören die allermeisten Web- und Mailserver dazu, IM-Dienste wie ICQ oder MSN funktionieren **nur über Umwege**[29]. Auf den Sixxs-Webseiten finden sich zwar einige Hinweise auf reine IPv6-Angebote wie einen **IPv6-Bittorrent-Tracker**, **mehrere Web-Radios**, **Newsserver**, **VoIP-Dienste**, **Video-on-Demand-Angebote**[30] oder **Spiele wie Quake oder das Multi User Dungeon Fatal Dimension**[31]. Die IPv6-Angebote und -Dienste erreichen hingegen noch längst nicht die Fülle des heutigen IPv4-Netzes.

Für Webseiten stellt das Sixxs **einen Proxy bereit**[32], der die Browser und Server der beiden Protokolle zusammenbringt. Erweitert man den Namen eines IPv4-Webservers um den Zusatz `".sixxs.org"` oder `".ipv6.sixxs.org"`, lädt der Proxy sie und übergibt sie an den Browser. Surft man hingegen über IPv4, erledigt der Adresszusatz `".ipv4.sixxs.org"` die gleiche Aufgabe für reine IPv6-Webseiten.



Microsoft betreibt anscheinend keine IPv6-tauglichen Update-Server für sein neues Betriebssystem Vista.

Wer jedoch die eingangs erwähnte Tunnel-Methode nutzt, benötigt derartige Hilfsmittel nicht. Die beschriebene, direkte IPv6-DSL-Verbindung verzichtet

völlig auf IPv4, sodass man viele Angebote nicht erreichen könnte.

Eine zweite, gleichzeitig aufgebaute DSL-Verbindung per IPv4 behebt dieses Manko und ebnet den Weg zu den IPv4-Angeboten: Dazu benötigt man keine zweite DSL-Leitung, nur einen zweiten DSL-Zugang. Andere Anbieter und Titan-DSL bieten jedoch auch DSL-Zugänge an, die sowohl IPv6 als auch IPv4 mitbringen.

Sicherheit

Unter IPv6 entfällt die Firewall-Funktion von NAT. IPv6-Netze und -Rechner benötigen daher wenigstens eine Firewall, die unerwünschte Gäste fernhält. Unter Vista, Windows Server 2003 und 2008 sowie Windows XP (SP2) filtert die eingebaute Desktop-Firewall den Netzwerkverkehr für IPv6, was Hersteller Microsoft ausdrücklich wegen möglicher Viren- und Trojaner-Angriffe empfiehlt. Auf Apples aktuellem Mac OS X 10.5 schützt eine neue von Apple selbst entwickelte **Application Firewall**[33] das Betriebssystem. Gleichzeitig steht jedoch der **Paketfilter ipfw**[34] bereit, der ausschließlich per Konsole bedient wird und den das aktuelle Mac OS X kaum nutzt. FreeBSD setzt ebenfalls auf ipfw, OpenBSD benutzt hingegen **den Paketfilter pf**[35]. Linux filtert den IPv6-Datenverkehr über das **Programm ip6tables**[36], zu dem das Kernelmodule ip6_tables gehört. Ist es geladen, was sich über das Kommando `modprobe ip6_tables` nachholen lässt, existiert unter `/proc/net` die Datei `ip6_tables_names`. Anschließend errichten einige **wenige Kommandos eine einfache Schutzmauer**[37].

Die ersten drei Befehle erstellen Richtlinien für ein- und ausgehende sowie weitergeleitete Pakete. Der Aufruf `ip6tables -A INPUT -p tcp --dport 22 -j ACCEPT` öffnet auf dem Router einen Port für den SSH-Server. Die `ip6tables`-Aufrufe mit den Parametern `-m state --state ESTABLISHED,RELATED` schalten für eingehende und weitergereichte Pakete das Connection Tracking ein, sodass die Firewall Verbindungen beobachtet, die der Rechner ins Internet aufbaut. Antworten auf diese Anfragen lässt die Firewall passieren, nicht angeforderte Pakete verwirft sie.

IPv6-Firewall mit Connection Tracking ...

```
ip6tables -P INPUT ACCEPT
ip6tables -P FORWARD ACCEPT
ip6tables -P OUTPUT ACCEPT
ip6tables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
ip6tables -A INPUT -p tcp --dport 22 -j ACCEPT
ip6tables -A INPUT -i ${INTERFACE_EXTERN} -j DROP
ip6tables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
ip6tables -A FORWARD -i ${INTERFACE_EXTERN} -j DROP
```

... und ohne Connection Tracking

```
ip6tables -P INPUT ACCEPT
ip6tables -P FORWARD ACCEPT
ip6tables -P OUTPUT ACCEPT
ip6tables -A INPUT -p tcp --dport 22 -j ACCEPT
ip6tables -A INPUT -i ${INTERFACE_EXTERN} -p tcp --dport :1023 -j DROP
ip6tables -A INPUT -i ${INTERFACE_EXTERN} -p udp --dport :1023 -j DROP
ip6tables -A FORWARD -i ${INTERFACE_EXTERN} -p tcp --dport :1023 -j DROP
ip6tables -A FORWARD -i ${INTERFACE_EXTERN} -p udp --dport :1023 -j DROP
```

Linux vor Version 2.6.20 beherrschen diese Verbindungsüberwachung in der IPv6-Firewall noch nicht: Das zweite `ip6tables`-Beispiel blockiert bis auf SSH alle Dienste unterhalb des Ports 1024 (Privilegierte Ports), alles andere lässt die Firewall passieren. Weitere Hilfen stellen die **Manual-Seiten von ip6tables**[38] und die **Projektseite von Netfilter**[39] bereit.

Mittendrin

Die Vorteile von IPv6 liegen in Europa weniger beim Zugang zum Internet. Interessanter erscheint jedoch der umgekehrte Weg, der die heimischen Netzwerkgeräte per IPv6 ohne altbekannte Tricks weltweit erreichbar macht. Mit IPv6-tauglicher Software wie dem Webserver Apache oder Microsofts IIS lässt sich also schnell ein Server aufsetzen, der Dateien, Bilder oder andere Daten bereitstellt. Die eigenen Rechner stehen nun mitten im Internet und jeder kann sie erreichen – Schutzmaßnahmen wie eine Firewall und das Aktualisieren von Software sollte man also nicht vernachlässigen.

Literatur

Deutsches Linux-IPv6-HOWTO[40] von Peter Bieringer

URL dieses Artikels:

<http://www.heise.de/netze/artikel/IPv6-Internetzugang-221479.html>

Links in diesem Artikel:

- [1] <http://www.heise.de/netze/IPv4-Adressen-werden-knapp-/artikel/108743/>
- [2] <http://www.heise.de/glossar/entry/Network-Address-Translation-399059.html>
- [3] <http://www.heise.de/glossar/entry/Port-Forwarding-397535.html>
- [4] <http://www.ietf.org/>
- [5] <http://www.personal.psu.edu/dvm105/blogs/ipv6/2008/06/ipv6-as-a-social-justice-issue.html>
- [6] <http://www.heise.de/glossar/entry/Router-399395.html>
- [7] <http://www.heise.de/netze/artikel/IPv6-Das-Mega-Netz-221708.html>
- [8] <http://www.heise.de/netze/artikel/IPv6-fuer-kleine-Netze-221783.html>
- [9] <http://www.heise.de/netze/artikel/IPv6-Adressen-221445.html>
- [10] <http://www.sixxs.net>
- [11] <https://www.sixxs.net/tools/aiccu/>
- [12] <http://www.heise.de/glossar/entry/Point-to-Point-Protocol-396689.html>
- [13] http://www.heise.de/netze/software/download/cfos_ipv6_link/56472
- [14] <http://mirrors.bieringer.de/Linux%2BIPv6-HOWTO-de>

- [15] http://wiki.openwrt.org/IPv6_howto
- [16] <http://www.dd-wrt.com/wiki/index.php/IPv6>
- [17] <https://www.sixxs.net/faq/aiccu/>
- [18] <http://www.heise.de/glossar/entry/Switch-396339.html>
- [19] <https://www.sixxs.net/tools/aiccu>
- [20] <https://www.sixxs.net/archive/sixxs/aiccu/aiccu.conf>
- [21] <http://www.sixxs.net/main>
- [22] <http://www.heise.de/netze/artikel/IPv6-Adressen-221445.html>
- [23] <ftp://ftp.heise.de/pub/ct/listings/0806-214.zip>
- [24] http://www.heise.de/kiosk/archiv/ct/2008/01/176_Netzwerk-Profilе_unter_Linux_automatisch_aktivieren
- [25] <http://www.heise.de/glossar/entry/Symmetric-Digital-Subscriber-Line-397545.html>
- [26] <http://www.heise.de/glossar/entry/Ethernet-399379.html>
- [27] <http://www.heise.de/glossar/entry/Point-to-Point-Protocol-over-Ethernet-396829.html>
- [28] <http://www.heise.de/glossar/entry/Wireless-Local-Area-Network-395526.html>
- [29] <http://im.sixxs.net/>
- [30] <http://www.sixxs.net/tools/tracker>
- [31] <http://www.sixxs.net/misc/coolstuff>
- [32] <http://ipv6gate.sixxs.net/>
- [33] <http://www.heise.de/meldung/Apple-dokumentiert-Funktion-und-Luecken-der-Leopard-Firewall-193449.html>
- [34] <http://www.freebsd.org/cgi/man.cgi?query=ipfw&sektion=8>
- [35] <https://solarflux.org/pf/pf%2BIPv6.php>
- [36] <http://www.netfilter.org/documentation/index.html>
- [37] [#ipv6-firewall-linux](#)
- [38] <http://linux.die.net/man/8/ip6tables>
- [39] <http://www.netfilter.org/documentation/index.html>
- [40] <http://mirrors.bieringer.de/Linux%2BIPv6-HOWTO-de>