Heim-Router (Debian/VMware)

Belegarbeit für die Veranstaltung RN/OD im Sommersemester 2007

Mathias Slawik

Version: 1.0

Dokumentation einer Debian-Installation auf einer virtuellen VMware-Maschine als Beispielimplementierung einiger in der Veranstaltung besprochener Technologien.

1 INHALT

2 EINLEITUNG

2.1 AUFGABENSTELLUNG

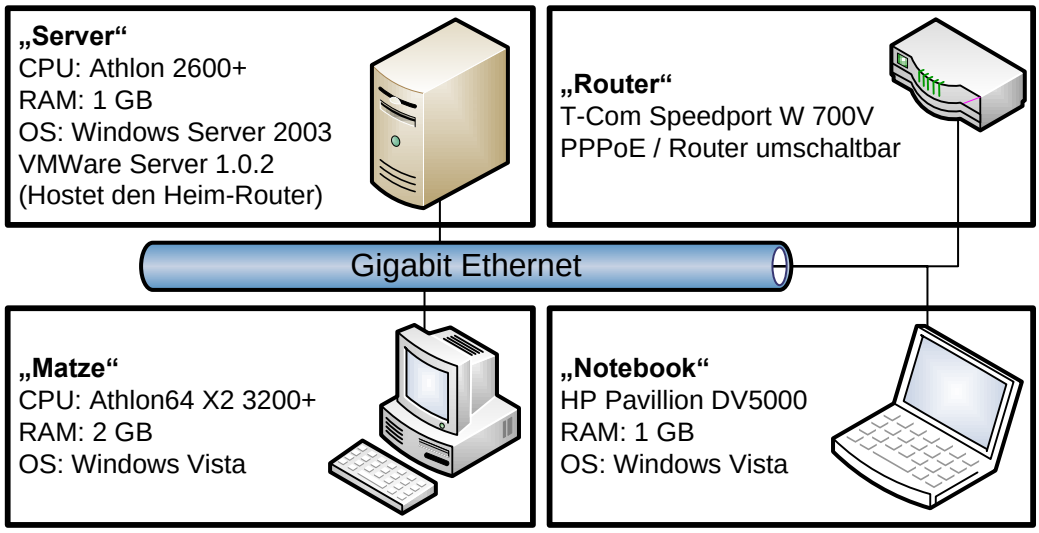
Implementierung und Dokumentation eines virtuellen Linux Routers für den Heimgebrauch als Beispiel der in der Veranstaltung besprochenen Technologien.

2.1.1 FUNKTIONALITÄTEN

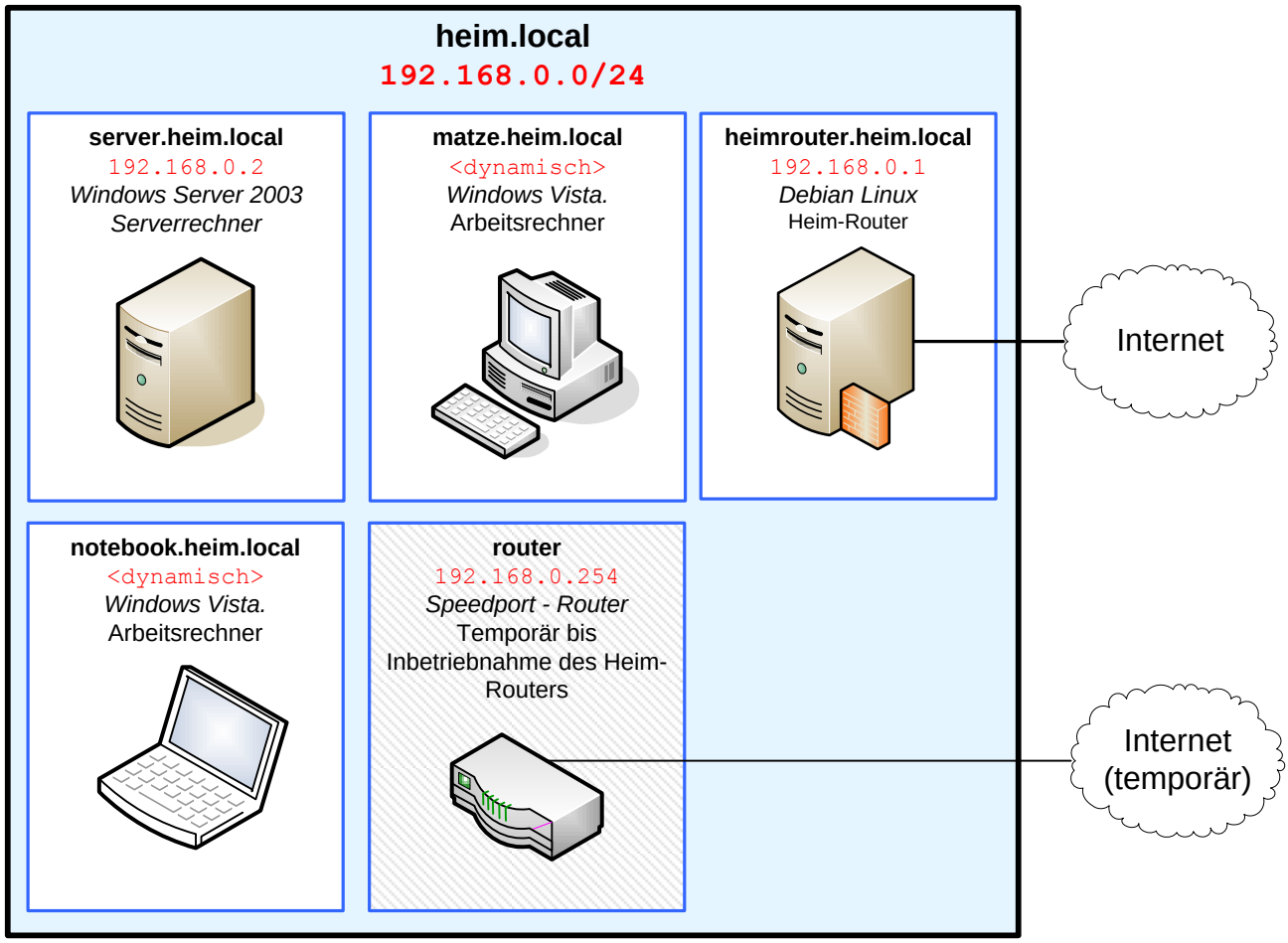
- DNS – Server für lokale Rechner mit Weiterleitung an DNS – Server des ISPs
- DHCP – Server mit automatischer Eintragung der Rechnernamen in den DNS - Server
- DSL-Internetverbindung über PPPoE
- Universal Plug & Play – Unterstützung
- Transparenter Web-Proxy mit Werbeunterdrückung
- Updatefunktion für ein DynDNS - Konto

2.2 AUFBAU DES TESTLABORS

Das Testlabor, in welchem die Referenzimplementierung aufgebaut wurde, sieht wie folgt aus:



1 - Physische Netzübersicht



2 - Logische Netzwerkübersicht

Es ist zu beachten, dass der Speedport W700V sowohl als Router konfiguriert werden kann, als auch als normales PPPoE-Terminator (sog. „DSL-Modem“).

2.3 PRÄMISSEN

Bei der Erstellung des Heim-Routers und dieser Dokumentation mussten diverse Prämissen getroffen werden, da ansonsten der Rahmen einer Belegaufgabe gesprengt worden wäre:

- Internetzugang über DSL
- Hardware, die vorhanden sein muss:
  - mind. 1 x Windows-Rechner mit installiertem VMware – Server
  - 1 x DSL-Modem (bzw. Router im DSL-Modem - Modus)
- Vorhandensein von Kenntnissen über die in der Vorlesung behandelten Themen
- Übung im Umgang mit Linux
  - Basiswissen Kommandozeile
  - Basiswissen Netzwerk
- Dokumentation nicht als „HOWTO“, sondern als Vorgangsbeschreibung

2.4 EINLEITUNG VMWARE

VMware, genauer VMware Inc., ist ein US-amerikanisches Unternehmen, welches Software im Bereich der Virtualisierung herstellt. Die Firma wurde 1998 mit dem Ziel gegründet, eine Technik zu entwickeln, virtuelle Maschinen auf Standard–Computern zur Anwendung zu bringen.

Die Produktpalette von VMware reicht von den kostenlosen Produkten Player und Server bis zu Produkten wie VMotion, welche es ermöglicht, virtuelle Maschinen im laufenden Betrieb zwischen Servern umzuziehen.

Die Produkte sind ausgereift, stabil und werden selbst in Hochverfügbarkeitsumgebungen eingesetzt, um z.B. die Serverauslastung zu optimieren.

Für die Referenzimplementierung wurde das kostenlose Produkt VMware Server eingesetzt. Es bietet folgende Funktionalitäten:

- Windows- und Linux-Kompatibel
- Ausführung des Heim-Routers im Hintergrund
- Hoch- und Herunterfahren des Heim-Routers mit dem Hostrechner
- Schnappschuss-Funktion, um Änderungen zurückzunehmen

VMware Server kann hier heruntergeladen werden:

http://www.vmware.com/download/server

Weiterhin wird eine Seriennummer benötigt, für die man sich hier kostenlos registrieren kann:

http://register.vmware.com/content/registration.html

Hier gibt es allgemeine Informationen über den VMware Server:

http://www.vmware.com/products/server/

2.5 EINLEITUNG DEBIAN

Das Debian-Projekt wurde am 16. August 1993 ins Leben gerufen. Debian GNU/Linux ist eine Linux-Distribution, welche zurzeit von über 1.000 Personen weiterentwickelt wird. Sie zeichnet sich durch große Stabilität aus, ist mittlerweile äußerst ausgereift und basiert komplett auf freien Softwarekomponenten.

Debian kann sowohl als Serversystem als auch als Desktopsystem eingesetzt werden. Mit seinen vielen Paketen bietet es die Basis einiger anderer Distributionen – wie z.B. Ubuntu. Eine weitere Besonderheit an Debian sind die flexible Paketverwaltung und die vielen unterstützten Hardwareplattformen.

Daher bietet sich diese Distribution für die Referenzimplementierung an.

3 DOKUMENTATION

3.1 INSTALLATION VMWARE SERVER

Bei der Installation des VMware Servers sind keine besonderen Eigenheiten zu beachten.

Der VMware Server selbst läuft unter Windows als Systemdienst. Die Administration dieses Systemdienstes erfolgt über das mitgelieferte Programm „VMware Server Console“.

Dieses Programm ist separat verfügbar und arbeitet über TCP/IP, so dass der Router nicht auf dem gleichen Rechner laufen muss, wie die Serverkonsole.

3.2 KONFIGURATION DER VIRTUELLEN MASCHINE

Nach der Installation des VMware Servers muss eine neue virtuelle Maschine erstellt werden. Dies geschieht über die „VMware Server Console“. Auf einer logischen Ebene besteht eine VMware virtuelle Maschine aus ein paar Konfigurationsdateien und beliebig vielen Festplatten-Images.

Die virtuelle Maschine der Referenzimplementierung wurde wie folgt konfiguriert:

- Hauptspeicher
  - 256 MB
- Festplatte
  - Maximale Kapazität: 8.0 GB
  - Eingehängt als SCSI 0:0 (LSI Logic HBA)
  - Das Festplatten-Image nimmt auf der Festplatte des Hostrechners allerdings nur soviel Platz ein, wie auf ihr (virtuell) gespeichert ist
- CD-Rom
  - ISO-Image: debian-40r0-i386-businesscard.iso (“Visitenkarten” Installations-CD)
  - Heruntergeladen von: http://cdimage.debian.org/debian-cd/4.0_r0/i386/iso-cd/debian-40r0-i386-businesscard.iso
  - Eingehängt als IDE 1:0
- Ethernet
  - „Bridged“ – Die VMware Maschine enthält direkten Zugriff auf das Netzwerk des Hostrechners
  - Bei mehreren Netzwerkkarten kann die gewünschte Netzwerkkarte über „Host“ -> „Virtual Network Settings“ verändert werden
- Prozessoren
  - One – für den Heim-Router reicht eine virtuelle CPU
- Optionen
  - Guest operating system: Linux (Other Linux 2.6.x Kernel)
  - Start- und Stoppoptionen können je nach Bedarf verändert werden (z.B. Hoch- und Herunterfahren mit dem Hostsystem)

3.3 BASISINSTALLATION DEBIAN

Zur Installation wurde das „Visitenkarten“ CD-Image der aktuellen Version 4.0 verwendet. Dieses enthält nur das Debian-Installationsprogramm. Die Softwarepakete werden hierbei während der Installation aus dem Internet heruntergeladen. Deshalb ist eine funktionierende Netzwerkkonfiguration erforderlich. Diese wurde im Testlabor über den Speedport-Router mit eingeschalteten Diensten (Internet-Routing, DHCP, DNS, etc.) realisiert.

Sollte dies bei einer anderen Installation keine Option darstellen, kann natürlich auch ein DVD-Satz mit allen Installationspaketen heruntergeladen und verwendet werden. Dann ist während der Installation keine Internetverbindung erforderlich.

Weiterführende Informationen:

- Herunterladen während der Installation: http://www.debian.org/CD/netinst/
- Installationsmedien mit allen Paketen: http://www.debian.org/CD/

Vorgehen bei der Installation der Referenzimplementierung:

- Virtueller Rechner fährt hoch und bootet von der Installations-CD
- Eingabe: **`installgui`** (Verwendung des graphischen Installers, da es Darstellungsprobleme beim textbasierten Installer unter VMware gibt)
- Sprache: Deutsch
- Gebiet: Deutschland
- Tastaturbelegung: Deutsch
- Netzwerkkonfiguration per DHCP (im Testlabor über Speedport)
    - Alternativ auch manuelle Angabe der IP-Konfiguration:
      IP-Adresse: 192.168.0.1
      Netzmaske: 255.255.255.0
      Gateway: keines
- Rechnername: **`heimrouter`**
- Domainname: heim.local
- Spiegelserver: beliebig
- Proxy: falls benötigt
- Festplatten partitionieren:
    - Geführt – Verwende vollständige Festplatte
    - Festplatte SCSI1 (0,0,0) … VMware
    - Alle Dateien auf eine Partition
    - „Weiter“ -> Änderungen auf Festplatte schreiben
- Root-Account erstellen
    - Passwort: **`root`** (sollte bei Betrieb des Servers aus Sicherheitsgründen geändert werden)
- Neuen Benutzer erstellen
    - Voller Name: **`Beispielnutzer`**
    - Benutzername: **`user`**
    - Passwort: **`user`** (sollte bei Betrieb des Servers aus Sicherheitsgründen geändert werden)
- Jetzt werden die für die Installation benötigten Pakete heruntergeladen, entpackt und installiert
- An der Paketverwendungserfassung teilnehmen: beliebig
- Softwareauswahl
    - Alle Häkchen entfernen
- GRUB-Bootloader (das Programm, welches beim Hochfahren des Rechners den Linux-Kernel lädt)
    - Ja, den GRUB-Bootloader in den Master Boot Record installieren
- „Weiter“ -> System neustarten
- Beim Neustarten die Virtuelle Maschine ausschalten
- Das CD-Laufwerk aus der Konfiguration entfernen
    - „Edit virtual machine settings“
    - CD-ROM anclicken
    - „Remove“

Damit ist die Basisinstallation abgeschlossen

3.4 SCHRITTE NACH DER INSTALLATION

- **less** installieren (komfortablere Version von **more**)
  - Kommando: **apt-get install less**
- Farbige **ls** – Anzeige
  - ~/.bashrc mit Texteditor (z.B. **nano**) bearbeiten
  - Kommentare vor zugehörigen Zeilen in Datei entfernen
- Maussteuerung installieren
  - Kommando: **apt-get install gpm**
  - Markieren und Kopieren mit der linken Maustaste, Einfügen mit der rechten Maustaste
- SSH installieren (zur einfacheren Administration des Servers)
  - Kommando: **apt-get install ssh**
  - Nun kann der Server per SSH von Außen erreicht werden (im Testlabor von „Matze“ über PuTTY)

3.5 NETZWERKKONFIGURATION

Die Netzwerkkonfiguration wird unter Debian hauptsächlich über die Datei /etc/network/interfaces geregelt. Hier werden Interfaces, IP-Adressen, Gateways, etc. konfiguriert.

Weitere Informationen darüber hier:

- Debian Reference: http://www.debian.org/doc/manuals/reference/ch-gateway.en.html

Der Heim-Router wurde wie folgt konfiguriert:

- IP-Adresse: 192.168.0.1
- Netzwerkmaske: 255.255.255.0

Der Inhalt der Konfigurationsdatei:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface

# Fahre Interface eth0 beim Starten automatisch hoch
auto eth0

# Konfiguration des Interfaces
iface eth0 inet static
        address 192.168.0.1
        netmask 255.255.255.0
```

Die Änderungen an der Netzwerkkonfiguration können über den Befehl **/etc/init.d/networking restart** sofort angewendet werden.

3.6 DIENSTKONFIGURATION

3.6.1 PPPOE - EINLEITUNG

In Zeiten, in denen die Verbindung zum Internet über Modems und ISDN – Geräte hergestellt wurde, waren die Geräte meist über eine serielle Schnittstelle angeschlossen, die meist etwas schneller war, als die Geschwindigkeit der Verbindung ins Internet. Das Protokoll, welches zwischen dem Internetprovider und dem Modem / ISDN-Gerät zum Herstellung der Internetverbindung benutzt wurde, war fast immer das PPP (Point to Point Protocol).

Da DSL-Zugänge fast immer schneller sind, als die maximale Geschwindigkeit der seriellen Schnittstelle war es nicht mehr möglich, die Internetverbindung über diese herzustellen. Man wollte aber nicht vom weit weit verbreiteten PPP abrücken, und so wurde PPPoE (Point to Point Protocol over Ethernet) entwickelt. PPPoE enkapsuliert das PPP über eine Ethernet-Verbindung.

3.6.2 PPPOE – KONFIGURATION

Damit man über das PPPoE eine Verbindung ins Internet herstellen kann, muss sowohl der Systemdienst, als auch die Konfigurationsprogramme installiert werden. Grob gesagt erzeugt der PPP-Dämon ein virtuelles Interface (ppp0), welches die Verbindung ins Internet darstellt. Über dieses fließt dann der IP-Verkehr ins Internet, genau so, wie der IP-Verkehr im internen Netz über die physische Schnittstelle eth0.

- Installation des PPPoE – Dienstes und des Konfigurationsprogramms
  - Kommando: **`apt-get install pppoeconf`**
- Ausführen des Konfigurationsprogramms: **`pppoeconf`**
  - Ja, übliche Optionen akzeptieren
  - Benutzername und Passwort eingeben
  - Ja, Addressen automatisch in /etc/resolv.conf eintragen
  - Ja, MSS-Größe festlegen
  - Ja, Verbindung während des Bootvorganges starten
  - Ja, Verbindung starten

Nun zeigt **`ifconfig`** das bereits besprochene virtuelle Interface an. Jetzt können schon vom Server aus ICMP ping – Signale an Rechner im Internet abgesetzt werden. Die PPP – Verbindung kann manuell über die Befehle **`pon dsl-provider`** und **`poff dsl-provider`** gesteuert werden.

In der Referenzimplementierung müssen in der letzten Zeile von /etc/ppp/pap.secrets das Benutzername und das Passwort der eigenen Internetverbindung angegeben werden!

3.6.3 NAT

Damit später auch andere Rechner im internen Netz Zugang ins Internet erhalten können, muss der Heim-Router angewiesen werden, IP-Pakete, die nicht für ihn selbst bestimmt sind, weiterzuleiten. Diesen Vorgang nennt man IP-Forwarding.

Die Steuerung des IP-Forwardings geschieht unter Linux normalerweise über die virtuellen Dateien /proc/sys/net/ipv4/conf/<Name des Interfaces>/forwarding. Da eine Änderung an der Datei nur bis zum nächsten Systemstart Bestand hätte, wurde bei der Referenzimplementierung die Datei /etc/sysctl.conf verwendet, die Einstellungen enthält, welche bei jedem Systemstart angewendet werden.

- Aktivieren von IP-Forwarding
    - Auskommentieren der vorletzten Zeile in /etc/sysctl.conf

Nun können Pakete von Rechnern im internen Netz an Rechner im Internet verschickt werden. Das Problem besteht jetzt aber darin, dass Rechner im Internet als Quelle der Pakete die interne IP-Adresse des Rechners empfangen, der das Paket verschickt hat. Da interne Adressen im Internet nichts verloren haben (nicht geroutet werden) werden diese wieder verworfen.

Jetzt muss der Heim-Router angewiesen werden, die interne Herkunft des Pakets zu verschleiern und die Quelladresse des Pakets auf seine eigene Internetadresse zu setzen. Bei Ankunft einer Antwort muss er dann das Ziel des Pakets mit der verschleierten Adresse ersetzen damit es eine funktionierende Kommunikation zwischen Rechnern im internen Netz und im Internet geben kann.

Durch diese „Maskierung“ der Pakete heißt dieses Verfahren auch „IP-Masquerading“. Dieses IP-Masquerading ist eine Funktion, die in der eingebauten Linux-Firewall vorhanden ist. Über den Befehl **`iptables`** lässt sich diese Firewall steuern. Die Maskierung kann über den Befehl **`iptables -t nat -A POSTROUTING -j MASQUERADE -o ppp0`** eingeschaltet werden.

Dieser Befehl müsste jedesmal ausgeführt werden, sobald eine neue Internetverbindung aufgebaut wird, da sich die IP-Adresse des Interfaces ppp0 bei jeder Einwahl verändert und somit auch die Firewall eine andere IP-Adresse verwenden muss, um die Pakete zu verschleiern.

Die bereits besprochene Datei /etc/network/interfaces kann dazu verwendet werden. Die Konfigurationsoption „post-up“ nimmt einen beliebigen Befehl an, welcher bei jedem Hochfahren des Interfaces ausgeführt wird.

- Aktivieren von IP Masquerading
    - Hinzufügen der folgenden Zeile zu /etc/network/interfaces:
      `post-up iptables -t nat -A POSTROUTING -j MASQUERADE -o ppp0`

Also sieht das Ende der Konfigurationsdatei /etc/network/interfaces wie folgt aus:

```
# Fahre Internetverbindung bei jedem Starten automatisch hoch
auto dsl-provider

# Konfiguration des Interfaces
iface dsl-provider inet ppp
pre-up /sbin/ifconfig eth0 up # line maintained by pppoeconf
# Konfiguration (Benutzername, Passwort) des Internet-Interfaces
provider dsl-provider
# Aktualisiere Masquerading bei jeder neuen Verbindung ins Internet
post-up iptables -t nat -A POSTROUTING -j MASQUERADE -o ppp0
```

3.6.4 DNS

Jetzt können Pakete von internen Rechnern ins Internet gesendet werden und Antworten empfangen werden. Jedoch können Rechner im internen Netz noch keine Namen von Rechnern im Internet auflösen. Dafür muss auf dem Heim-Router ein DNS-Server installiert werden.

Bei der Referenzimplementierung wird der am meisten verwendete DNS-Server verwendet, der BIND – Server des Internet Systems Consortium.

Heim-Router (Debian/VMware)

- Installation des BIND – DNS Servers
  - Kommando: **`apt-get install bind`**

Jetzt können Internetnamen von Rechnern im internen Netz aufgelöst werden. Jedoch werden alle Anfragen an die DNS Root-Server weitergeleitet, was einen relativ langsamen Vorgang darstellt. Viel schneller sind die DNS-Server des eigenen Internetanbieters, die bei jeder Verbindung ins Internet übergeben werden.

- Eintragen der DNS-Server des ISP
  - In /etc/resolv.conf stehen bei einer bestehenden Internetverbindung die DNS-Server des ISP
  - Diese wie folgt in die /etc/bind/named.conf.options eintragen:

```
forward only;
forwarders {
      < DNS Serveradresse 1>;
      < DNS Serveradresse 2>;
};
```

In der Referenzimplementierung sollten diese DNS-Adressen (Arcor-DNS) durch die des eigenen ISPs ersetzt werden!

Nun sollten noch die Einträge des Heimrouters in den DNS-Server eingetragen werden, damit dieser später über seinen FQDN heimrouter.heim.local aufgelöst werden kann.

- Eintragen des Heimrouters in den DNS Server
  - Erzeugen der Zonen-Information in /etc/bind/named.conf.local
  - Eintragen der A und PTR – Records

Damit später die Einträge der internen Rechner im DNS dynamisch aktualisiert werden können, muss ein Schlüssel erzeugt werden, der als Authentifizierung des DHCP-Servers am DNS-Server dient. Die beiden Server sind deswegen sehr kompatibel, da sie beide am Internet Systems Consortium entwickelt werden.

- Erzeugen eines DNS-Schlüssels
  - **`dnskeygen -H 128 -h -n heim.local.`**
  - Eintragen des Schlüssels in /etc/bind/named.conf.local

So sieht der Inhalt der /etc/bind/named.conf.local aus:

```
//
// Add local zone definitions here.
key heim.local. { (Definition des Schlüssels. Der Name muss gleich dem Namen sein, der dnskeygen angegeben wurde)
        algorithm hmac-md5; (Verschlüsselungsalgorithmus)
        secret "6w93vZPQVHgpFNR2Xfh9ew=="; (Von dnskeygen erstellt)
};

zone "heim.local" IN { (Zone zum Auflösen von Hostnamen nach IP-Adressen)
        type master; (Diese Zone ist auf dem DNS-Server beheimatet)
        file "/etc/bind/heim.local.zone"; (Daten der Zone)
        allow-update { key heim.local.;}; (Erlaube Updates durch Schlüssel)
};

zone "0.168.192.in-addr.arpa" IN { (Zone zum Auflösen von IP-Adressen nach Hostnamen)
        type master; (Diese Zone ist auf dem DNS-Server beheimatet)
        file "/etc/bind/0.168.192.in-addr.arpa"; (Daten der Zone)
        allow-update { key heim.local.;}; (Erlaube Updates durch Schlüssel)
};
```

Heim-Router (Debian/VMware)

Der Inhalt der /etc/bind/heim.local.zone:

```
$ORIGIN local.
heim     604800   IN       SOA      localhost. root.localhost. (6 604800 86400
2419200 604800)  (SOA – Start of authority … wir sind für diese Zone verantwortlich)

         604800   IN       NS       heimrouter.heim.local.heim.local.  (Nameserver)

$ORIGIN heim.local.
heimrouter       604800   IN       A        192.168.0.1  (Das Herzstück – der A-Eintrag des
Heimrouters)
```

Der Inhalt der /etc/bind/0.168.192.in-addr.arpa (Rückwärtsauflösung von IP-Adressen):

```
$ORIGIN 168.192.in-addr.arpa.
0        604800   IN       SOA      localhost. root.localhost. ( 4 604800 86400
2419200 604800 )  (SOA – Start of authority … wir sind für diese Zone verantwortlich)

         604800    IN           NS           heimrouter.heim.local.0.168.192.in-
addr.arpa.  (Nameserver)

$ORIGIN 0.168.192.in-addr.arpa.
1         604800   IN        PTR       heimrouter.  (Das Herzstück – der PTR-Eintrag des
Heimrouters)
```

- Weitere Informationen über BIND:
  http://www.isc.org/sw/bind/

Nun können sowohl Internetnamen, als auch der FQDN des Heim-Routers von Rechnern im internen Netzwerk aufgelöst werden.

3.6.5 UPNP

UPnP steht für „Universal Plug and Play“ und beschreibt ein Protokoll, welches eine einfache Konfiguration von Routern und anderen Netzgeräten im Heimbereich ermöglicht.

IP-Masquerading ermöglicht es vom Heimnetz ins Internet Pakete zu verschicken. Allerdings gibt es Probleme, wenn ein Rechner aus dem Internet versucht mit einem Rechner im internen Netz eine Verbindung herzustellen. Der Rechner aus dem Internet schickt Pakete an das externe Interface des Heim-Routers. Dieser bietet auf dem angesprochenen Port jedoch keinen Dienst an und weist das Paket ab.

Dieser Vorgang tritt meist bei folgenden Diensten auf:

- Filesharing
- Videokonferenzen / VoIP
- Instant Messenger / IRC
- Spiele

Damit diese Dienste korrekt funktionieren, muss dem Router die Information geliefert werden, welche Ports an welche Rechner im Internen Netzwerk weitergeleitet werden. Dieser Vorgang unterscheidet sich von Router zu Router sehr erheblich und stellt für die meisten Internetnutzer eine lästige Angelegenheit dar.

UPnP ermöglicht es nun, dass Programme, die solche Dienste anbieten, den Router automatisch für Ihre Zwecke konfigurieren. Seit Windows ME unterstützt auch das Betriebssystem selbst den Umgang mit UPnP fähigen Endgeräten. Diese tauchen im Ordner „Netzwerkverbindungen“ als eigenständige Geräte auf. Dadurch kann man komfortabel über den Explorer das Portforwarding aller UPnP-fähigen Endgeräte konfigurieren – auch das des Heim-Routers.

Microsoft selbst bietet über die „Internetverbindungsfreigabe“ eine eigene UPnP-Implementierung an. Sobald ein Windows-Rechner eine Internetverbindung freigibt, können Benutzer aus dem Netzwerk das Portforwarding dieses Rechners konfigurieren.

Unter Linux gibt es eine dazu weitgehend kompatible UPnP – Implementierung, die auf dem Heim-Router verwendet wurde. Die Kompatibilität ist besonders wichtig für Microsoft Produkte, wie z.B. den MSN Messenger.

- Installation linux-igd
  - Kommando: **`apt-get install linux-igd`**
  - Installation schlägt aufgrund einer schlampigen Paketkonfiguration fehl
- Auskommentieren der Standardoptionen in /etc/default/upnpd:

```
# Defaults for upnpd initscript
# sourced by /etc/init.d/upnpd
# installed at /etc/default/upnpd by the maintainer scripts

# External interface name
EXT_IFACE=ppp0

# Internal interface name
INT_IFACE=eth0
```

- Wiederholen der Installation
  - Kommando: **`apt-get install linux-igd`**
- Das Gerät wird nun im Ordner „Netzwerkverbindungen“ angezeigt und kann dort auch konfiguriert werden.

Im Testlabor war die UPnP-Implementierung vollständig kompatibel zu den Filesharing-Programmen Azureus und µTorrent.

Es sei allerdings an dieser Stelle noch auf die Nachteile von UPnP hingewiesen. Zum Beispiel gibt es keinerlei Authentifizierung seitens des UPnP-Gerätes, ob die Anfrage zum Port-Forwarding auch wirklich von einem Programm stammt, das vertrauenswürdig ist. So könnte zum Beispiel ein Schadprogramm automatisch alle Ports öffnen, und damit den Rechner angreifbar machen.

In der Praxis ist dieses Szenario allerdings eher theoretischer Natur.

3.6.6 DHCP MIT DDNS

Im Augenblick muss ein Rechner im internen Netzwerk noch manuell konfiguriert werden, um alle Dienste des Routers verwenden zu können. Die Übergabe von IP-Adresse, Domainname, Gateway und Nameserver-Adresse sollte automatisch über das DHCP geschehen.

- Installation des DHCP-Servers
  - Kommando: **`apt-get install dhcp3-server`**

Nun muss der DHCP-Server angewiesen werden, die korrekten Informationen an die Clients weiterzugeben und den Hostnamen, den die Clients übergeben in den DNS-Server einzutragen

- Bearbeiten der Konfigurationsdatei /etc/dhcpd.conf:

```
# Update den DNS - Server
ddns-update-style interim;

# Globale Optionen
option domain-name "heim.local";
option domain-name-servers heimrouter.heim.local;
default-lease-time 86400;
max-lease-time 604800;

# Fuehle Dich bei jeder Frage nach einer IP angesprochen
authoritative;

# Der Schluessel zum Update des DNS - Servers
key "heim.local." {
        algorithm HMAC-MD5;
        secret "6w93vZPQVHgpFNR2Xfh9ew==";
}

# Die Domainnamen, mit denen der DNS - Server upgedated werden soll
ddns-domainname "heim.local";
ddns-rev-domainname "in-addr.arpa";

# Die Zonen, die im DNS - Server upgedated werden sollen
zone heim.local. {
        primary 192.168.0.1;
        key "heim.local.";
}

zone 0.168.192.in-addr.arpa. {
        primary 192.168.0.1;
        key "heim.local.";
}

# Unser Heimnetz
subnet 192.168.0.0 netmask 255.255.255.0 {
        # Vergib IP-Adressen in dieser Range
        range 192.168.0.10 192.168.0.49;

        # Uebergib den Clients diese Optionen
        option routers 192.168.0.1;
        option domain-name-servers 192.168.0.1;
}
```

- Bearbeiten der DHCP-Defaultkonfiguration: /etc/default/dhcp3-server
  - Ändern der letzten Zeile
    INTERFACES="eth0"
- Server neu starten

Weitere Informationen über den DHCP-Server:

- http://www.isc.org/sw/dhcp/

3.6.7 DYNDNS – UPDATER

Bei jeder Verbindung ins Internet bekommt der Router eine neue IP-Adresse. Sollte der Heim-Router später um weitere Dienste ergänzt werden, würde es sich anbieten, wenn er über einen eindeutigen Namen im Internet angesprochen werden könnte. Es gibt diverse Anbieter im Internet, die einen solchen Dienst anbieten, unter anderem dyndns.org.

Ein Programm, welches die IP-Adresse eines zugewiesenen Hostnamens aktualisiert ist ddclient. Dieses Programm wird über das Skript /etc/ppp/ip-up.d/ddclient bei jeder Verbindung ins Internet aufgerufen und aktualisiert den konfigurierten Hostnamen anhand der Informationen, welche in /etc/ddclient.conf hinterlegt sind.

- Installation des DynDNS – Updateprogramms:
  - Kommando: apt-get install ddclient
- Datei /etc/ddclient.conf anpassen

```
# Configuration file for ddclient generated by debconf
#
# /etc/ddclient.conf

pid=/var/run/ddclient.pid
protocol=dyndns2
use=if, if=ppp0
server=members.dyndns.org
login=Matzes Login
password='Matzes Passwort'
Matzes Hostname
```

Dieser Dienst muss über den Befehl `dpkg-reconfigure ddclient` neu mit den eigenen Optionen versorgt warden!

3.6.8 SQUID

Squid ist ein Internet-Proxy und Cache. Squid ermöglicht es, aus dem Internet abgerufene Internetseiten zwischenzuspeichern und diese dann den Rechnern im internen Netzwerk zur Verfügung zu stellen. Dies kann teilweise den Seitenaufbau deutlich beschleunigen und Bandbreite einsparen. Weiterhin lässt sich Squid und der Router so konfigurieren, dass alle Zugriffe auf Internetseiten aus dem internen Netzwerk automatisch zwischengespeichert werden, ohne dass der aufrufende Rechner davon informiert wird. Dieses Verhalten nennt man „transparent proxying".

Aber eine der herausragenden Funktionen von Squid ist die Möglichkeit, Werbung und unerwünschte Internetinhalte zu filtern. Dadurch kann nicht nur ein Großteil der Werbung auf Internetseiten unterdrückt werden, sondern auch z.B. Werbung in anderen Programmen (im Testlabor z.B. in Winamp und ICQ). Dieses Filtern geschieht über ein externes Programm „squidGuard", welches von Squid bei jeder Abfrage aufgerufen wird und dann entscheidet, ob der Inhalt geblockt werden soll, oder nicht. Dafür benötigt das Programm

allerdings eine sogenannte „Blacklist“ welche Domainnamen und URLs enthält, die bestimmten Kategorien entsprechen (z.B. Werbung, Spyware, Pornos, etc.)

- Installation von Squid und squidGuard
  - Kommando: **apt-get install squid**
  - Kommando: **apt-get install squidguard**
- Anpassen der Konfiguration /etc/squid.conf (Auszüge der Änderungen):

```
# Squid normally listens to port 3128
http_port 127.0.0.1:8080 transparent (Einschalten von transparent proxying auf Port 8080)
http_port 192.168.0.1:8080 transparent (Einschalten von transparent proxying auf Port 8080)
```

```
#       By default, a URL rewriter is not used.
#
#Default:
# none
url_rewrite_program /usr/bin/squidGuard -d -c /etc/squid/squidGuard.conf (Einschalten des Inhaltsfilters)
```

```
#       By default, a Location rewriter is not used.
#
#Default:
# none
location_rewrite_program         /usr/bin/squidGuard         -d         -c
/etc/squid/squidGuard.conf (Einschalten des Inhaltsfilters)
```

```
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS

# Example rule allowing access from your local networks. Adapt
# to list your (internal) IP networks from where browsing should
# be allowed
acl our_networks src 192.168.0.0/24
http_access allow our_networks (Zugriff aus dem eigenen Netzwerk erlauben – jeder andere Zugriff wird abgewiesen)
http_access allow localhost
```

```
#  TAG: visible_hostname
#         If you want to present a special hostname in error messages,
etc,
#       define this.  Otherwise, the return value of gethostname()
#       will be used. If you have multiple caches in a cluster and
#          get errors about IP-forwarding you must set them to have
individual
#       names with this setting.
#
#Default:
# none
visible_hostname heimrouter.heim.local (Offensichtlich nötig für transparent proxying)
```

- Anweisen der Firewall, alle Zugriffe auf den Port 80 (http) von Rechnern im Internet an den Proxy weiterzuleiten
    - Anpassen der /etc/network/interfaces, damit bei jedem Hochfahren des internen Interfaces die Firewall-Regel gesetzt wird

```
# Konfiguration des Interfaces
iface eth0 inet static
        address 192.168.0.1
        netmask 255.255.255.0

# Transparentes Proxying einschalten
post-up iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j
REDIRECT --to-port 8080
```

- Konfiguration von squidGuard
    - Anpassen der /etc/squid/squidGuard.conf

```
#
# CONFIG FILE FOR SQUIDGUARD
#

# Verzeichnis mit der Blacklist
dbhome /var/lib/squidguard/db
logdir /var/log/squid

# Werbung
dest ads {
        # Liste von Domains mit Werbung
        domainlist blacklists/ads/domains
        # Liste von URLs mit Werbung
        urllist blacklists/ads/urls
}

# Spyware-Seiten
dest spyware {
        # Liste von Domains mit Spyware-Seiten
        domainlist blacklists/spyware/domains
        # Liste von URLs mit Spyware-Seiten
        urllist blacklists/ads/urls
}

# Konfiguration der Zugriffsverwaltung
acl {
        # Standard
        default {
                # Werbung und Spyware - Seiten werden geblockt,
                # ansonsten wird alles durchgelassen
                pass     !ads !spyware all

                # block.html ist eine leere Datei. Jeder Zugriff
                #  auf  Werbung  und  Spyware  wird  an  diese  Datei
umgeleitet
                redirect
http://www.matzelworkz.de/block.html
        }
}
```

- Nun muss noch die Blacklist (Liste der zu blockenden Internetseiten) heruntergeladen werden
  - http://squidguard.mesd.k12.or.us/blacklists.tgz nach /var/lib/squidguard/db herunterladen (z.B. über **`wget`**) und entpacken
  - Zugriffsrechte setzen:
    - **`chown proxy:proxy /var/lib/squidguard/db/blacklists –R`**
- Server neu starten

Damit ist der Heim-Router komplett konfiguriert.

3.7 WEITERE SCHRITTE

Nun, da der Heim-Router fertig konfiguriert ist, könnten die folgenden Schritte durchgeführt werden, die allerdings den Umfang der Belegarbeit sprengen würden:

- Anpassen von Squid auf optimale Performance

  Squid ist ein extrem flexibler und komplexer Dienst. Die derzeitige Konfiguration ist funktionstüchtig, jedoch gibt es im Internet viele „Best Practice“ – Beispiele, wie man noch mehr Performance aus dem Dienst entlocken kann

- Einführung von IPv6 – Kompatibilität

  Linux, Windows und alle Dienste des Heim-Routers sind bereits zur neuen Version des Internetprotokolls kompatibel. Allerdings gibt es zurzeit keine zwingenden Gründe auf IPv6 umzusteigen. Leider unterstützen auch noch keine Internetprovider in Deutschland den Zugang zu Internet über IPv6, jedoch könnte der Router jederzeit um IPv6 – Funktionalität erweitert werden.

- VPN – Funktionalität

  Der Router könnte um VPN – Funktionalitäten erweitert werden, damit er z.B. als Zugangspunkt in das interne Netzwerk, z.B. von der Arbeit aus, dient.

4 IMPRESSUM

4.1 COPYRIGHTS

4.2 VERSIONSHISTORIE

Version 1.0 vom 03.07.2007

- Erstveröffentlichung dieses Dokuments