Namensumsetzung in IP-Nummern

Wie heißt du?

Im dritten Teil der WCM-Linux-Box schaffen wir die Voraussetzungen für einen Webserver und wir lernen der Box als Domain-Name-System-Server zu funktionieren. Den Webserver brauchen wir damit wir ein grafisches Interface für die in der letzen Ausgabe installieren MySQL-Datenbank benutzen können, der Nameserver erleichtert die Arbeit für viele andere Tätigkeiten und sorgt für weniger Netzwerkverkehr nach außen.

von Martin Müller

Dass Computer schlecht mit uns vertrauten Ausdrücken umgehen können, bemerkt man schon daran, dass sie intern nur mit Nullen und Einsen arbeiten können. Auch das Navigieren durch Netzwerke erfolgt hinter den Kulissen ausschließlich durch Nummern. Damits für uns Menschen einfacher wird, brauchts einen Dienst der Eingaben wie www.wcm.at in die IP-Adresse 194.112.128.106 umsetzt.

Dieser Dienst nennt sich Domain Name System, kurz DNS, welcher erstmalig 1983 standartisiert festgeschrieben wurde. Das DNS löste die bis dahin üblichen *hosts*-Dateien ab, in denen händisch Rechner und deren zugehörige IP-Adressen gespeichert wurden. Die *hosts*-Datei ist aber noch heute bei nahezu jedem Betriebssystem vorhanden, auch heute können dort Rechnername und IP-Adresse angegeben werden.

Diese Änderungen müssen händisch auf jedem Rechner durchgeführt werden, ein Aufwand der für eine kleine Rechnergruppe, die auch Zonen genannt werden, noch überschaubar ist, aber sicher nicht für Millionen Teilnehmer. DNS-Server funktionieren wie ein Telefonbuch in dem nach Teilnehmern gesucht wird.

bind9 ein schneller DNS

bind9 nennt sich die aktuelle Nameserver-Software, die wir auf der Linux-Box einsetzen. Wir installieren diese und helfende Tools mit *apt-get install bind9 dnsutils* und begeben uns anschließend in das Konfigurationsverzeichnis /etc/ bind (*cd /etc/bind/*). Wir wollen einen Nameserver einrichten, der für unser lokales Netzwerk zuständig ist. Dafür schauen wir erst in die zentrale Konfigurationsdatei *named.conf.* In dieser Datei sind die Zonen definiert die wir verwalten wollen oder die durch gereicht werden sollen.

In dieser Datei finden Sie lauter Verweise (zB *file "/etc/bind/ db.local";)* zu externen Zonen-Dateien, die die eigentlichen Zonendefinitionen enthalten.

Wir schreiben nun in named.conf.local unsere eigene lokale Zone dazu (siehe Kasten Eigene Zone). Übrigens der Eintrag notify no; verhindert, dass unser DNS-Server Informationen unseres lokalen Netzes an externe DNS-Server weiter gibt.

Danach tragen wir in der Datei named.conf.options die DNS-Server unseres Internetproviders ein. Dazu löschen Sie die beiden Schrägstriche vor dem Eintrag forwarders und den nachfolgenden Zeilen. Die Einträge sind notwendig, damit unser DNS-Server weiß wen er fragen kann, wenn er die notwendigen Informationen nicht selber hat. Die Einträge müssen folgende Form haben:

> forwarders { 192.58.160.2; 192.58.168.3; };

Wie sie sicher schon bemerkt haben, sind Strichpunkte an jedem Zeilenende nicht von Nachteil. Teilweise sind diese sogar unbedingt erforderlich!

Zonen: Die halbe Miete

Im nächsten Schritt gehts an die Erstellung der Zonendatei



(testserver1.at.local). Die Zonendatei beschreibt unsere lokale DNS-Zone und enthält der Reihe nach folgende Einträge:

Als erstes den TimeToLive-Eintrag (*TTL*) der angibt, wie lange die Zonendatei Gültigkeit hat, bis sie neu eingelesen wird.

Danach folgen die Ressource Records, die aus Schlüsselnamen, optionalem Wert, Datensatztyp und Datenelement bestehen. Sie sehen in der Box *Zonendatei* die Elemente die hier beschrieben werden. Bitte beachten Sie den abschließenden Punkt bei der ersten Angabe in der zweiten Zeile! Diesen Schlusspunkten werden Sie öfter in den Dateien begegnen, Sie dürfen sie nicht weglassen!

Wir beschreiben die Zone testserver1.at, der Datensatz beginnt SOA (Start Of Authority), gefolgt vom völlständigen Rechnernamen sowie der Email-Adresse des Administrators der Domäne. Die Emailadresse ist nicht durch ein @ von der Domäne getrennt, sondern durch einen Punkt. Der volle Rechnername setzt sich aus Rechnern-

amen, auszulesen aus der Datei *hostname (less /etc/ hostname*) und der gewünschten Domäne zusammen.

Jetzt folgt die Seriennummer des Files, die nach jeder Änderung um mindestens eine Zahl erhöht werden muss. Bind liest diese Nummer aus und orientiert sich an Ihr. Wenn Sie diese Nummer als unverändert lassen, meint bind dass er den Inhalt schon kennt und arbeitet nicht weiter. Bewährt hat sich eine Seriennummer die sich aus dem aktuellen Datum und einer Tagesänderungsnummer zusammen setzt (Muster: JJJJMMTTNN).

Jetzt kommen Einträge zur Steuerung des Abfrageverhaltens anderer Nameserver, gefolgt vom vollem Rechnernamen der für unsere Zone zuständig ist. Anschlie-Bend haben wir einen optionalen MX-Eintrag nachgelegt. Der MX-Record beinhaltet den Namen für den Mailserver der für diese Zone zuständig ist. Die Ziffer vor dem MX-Eintrag gibt Auskunft wie bevorzugt er behandelt werden soll.

Je niedriger die Ziffer ist, desto wichtiger ist er. Wenn der MX-Server mit der niedrigsten Ziffer nicht erreichbar ist, dann werden Emails auf den MX-Server mit der nächst höheren Nummer weitergegeben.

Danach legen wir ein Alias für den Eintrag www.testserver1.at an.

Eigene Zone zone "testserver1.at" IN { type master; notify no; file "/etc/bind/testserver1.at.local"; }; zone "123.168.192.in-addr.arpa" IN (type master; notify no; file "/etc/bind/ 123.168.192.zone"; };

Reverse Zone

NEVELSE LU	IIC	
#/etc/bind/123.168.192.reverse	ezone	
123.168.192.in-addr.arpa.	ISOA	
Wie heißt du?		
debian.testserver1.at. root.test .at. (server1	
2005110801; Seriennumm	ner	
1D: Refreshdauer 1 Ta	a	
2H: Betryzeit 2 Stunde	n n	
1W: Expire Zoitroum 1	Woohon	
OD Minimum Nachfra		
	gewen z	age
) IN NS debian.testserver1	.at.	
1.123.168.192.in-addr.arpa.	IN	PTR
1 123 168 102 in addr arpa	INI	DTR
ne testervor1 at	IIN	1 111
45 122 169 102 in oddr arna	INI	DTD
45.125.108.192.111-auur.arpa.	IIN	FIN
perspierrecrimer.testserver1.at		

Weiters werden die IP-Adressen für die lokalen Rechner eingetragen. Diese lesen Sie aus den aktuellen Einstellungen Ihrer Clients aus. Auf Windows-Betriebssystemen öffnen Sie dazu die Eingabeaufforderung und setzten den Befehl *ipconfig /all* ab. In einer späteren Ausgabe werden wir das DNS-System so umstellen, dass diese Daten automatisch durch den DHCP-Server generiert werden.

Und jetzt umgekehrt

Durch Einträge in eine neue Datei names 123.168.192.reversezone legen wir den umgekehrten Weg fest. Wir reversieren den DNS-Prozess und lösen IP-Adressen in Hostnamen auf (siehe Box Reverse-Zone). Außerdem tragen wir die Reverse-Zone auch in die *named.conf.local* ein. Die Datei enthält die Adressangaben in umgekehrter Reihenfolge, gefolgt vom historisch notwendigen Zusatz in-addr.arpa., dieser darf nicht weggelassen werden!

Fertig, nun müssen wir noch dafür sorgen, dass unser DNS-Server auch benutzt wird. Als erste Anlaufstelle tragen wir in /etc/ resolv.conf die Zeile nameserver 127.0.0.1 ein

und setzten vor die bereits eingetragenen Nameserverzeilen ein Raute-Zeichen (#). Dies bewirkt dass die alten Einträge als Kommentar behandelt werden. Im Falle dass unser DNS einmal ausfällt, brauchen Sie nur die Kommentarzeichen zu löschen und das Netzwerk neu zu starten. Somit werden wieder die Provider-DNS-Server gefragt.

Hardwarerouter

Das Netzwerk muss nun neu gestartet werden (/etc/init.d/networking restart). Sollten Sie einen Hardware-Router im Einsatz haben, der IP-Adressen für die Clients durch einen DHCP-Server verteilt, dann müssen Sie nachsehen ob er auch die DNS-Server den Clients zuweist. Ist dies der Fall, so stellen Sie die entsprechenden Einträge auf die IP-Adresse

Zonendatei		
#/etc/bind/testserver1.at.local		
testserver1.at. IN SOA debian.testserver1.at. dnsmaster.testserver1.at. (
2005110801 ; Seriennummer		
10800 ; Refresh - Zeit in Sekunden nach der beim		
; ersten NS nach neuen Daten gefragt wird		
3600 ; Retry - Wenn ein autorativer NS nicht erreichbar ist,		
; wird die Zone auf den zweiten NS transferiert		
604800 ; Expire - Wir der primäre NS nicht erreicht,		
wird die Zone auch vom zweiten NS gelöscht		
86400 : Minimum - Ein sekundärer NS darf frühestens		
nach diesem Sekundenwert wieder dem ersten NS		
nach aktuellen Werten fragen		
/ IN NS debien testsenver1 at		
IN MY 6 debiantesterior at		
IN INA J UEDIAILLESISEIVEIT.a.		
www A 192.108.123.1; DIE IF-Adresse des NetzwerkInterfaces		
beispielrechner A 192.168.123.45; Adresse des Rechners "Beispielrechner"		

unserer WCM-Linux-Box um (192.168.123.1).

Wenn Sie die Clients händisch konfigurieren, dann müssen Sie dort die DNS-Server Ihres Internetproviders löschen und unseren Server angeben (192.168.123.1).

Alles fertig? Wir richten uns in einer neuen Konsole einen Monitor der Syslog-Datei ein. Mit STRG+ALT+F2 wechseln wir in eine neue Konsole, melden uns als root an und lassen uns durch tail-fn20 /var/log/syslog die letzten 20 Zeilen der syslog-Datei anzeigen. Mit STRG+ALT+F1 wechseln wir in die erste Kon-

sole zurück, und schon gehts los: / etc/init.d/bind restart sorgt für einen Neustart des DNS-Servers. In der zweiten Konsole sehen Sie die Meldungen von bind9. Haben Sie Fehler gemacht, meldet Ihnen dies bind9 nun mit der Zeilennummer in der der Fehler aufgetreten ist.

Fehlersuche

Im Falle von Fehlern müssen Sie die Zonen-Dateien korrigieren. Bitte vergessen Sie nicht die Seriennummer in diesen Dateien zu erhöhen, sonst nimmt bind9 die Änderungen nicht wahr.

bind9 läuft, Sie können ihn jetzt mit *host testserver1.at* befragen. Unser DNS sollte mit *testserver1.at has address* 192.168.123.1 richtig antworten. Um den Reverse-Lookup zu testen, tippen Sie *host* 192.168.123.1 ein. Unter Umständen kann es ein

wenig dauern bis etwaige Änderungen auch umgesetzt werden, da der DNS-Cache erst aktualisiert werden Weitere muss. Hilfsmittel sind nslookup hostname oder dig hostname. Beide geben weitere Aufschlüsse über die DNS-Struktur, welcher DNS-Server für diese Abfrage befragt wurde und zeigen Informati-



onen über die Zonen.

Nummernspeicher

Wenn wir nun unseren DNS-Server erstmalig nach der IP-Adresse von wcm.at fragen, dann weiß dieser die Adresse noch nicht. Er befragt die DNS-Server die wir in der *forwarders*-Sektion

genschaften von Internetprotokoll (TCP/IP)		<u>? ×</u>
Allgemein		
IP-Einstellungen können automati: Netzwerk diese Funktion unterstüt den Netzwerkadministrator, um die beziehen.	sch zugewiesen werden, wenn das zt. Wenden Sie sich andemfalls an geeigneten IP-Einstellungen zu	
C IP-Adresse automatisch bezi	ehen	
Folgende IP-Adresse verwer	nden.	
IP-Adresse:	192 . 168 . 123 . 5	
Subnetzmaske:	255 . 255 . 255 . 0	
Standardgateway:	192 . 168 . 123 . 1	
C DNS-Serveradresse automat		
Folgende DNS-Serveradress	ien verwenden.	_
Bevorzugter DNS-Server	192 . 168 . 123 . 1	
Alternativer DNS-Server.		
	Erweitert	
	OK Abbr	echen

angegeben haben und schreibt sich die Antwort in den Cache. Wenn wir nun eine neue Anfrage stellen, dann schaut er in seinem Cache nach und gibt von dort das Ergebnis zurück. Wir haben eine Reduktion des Netzwerkverkehrs nach außen erreicht!

Das DNS ist ein wichtiger Bestandteil unserer WCM-Linux-Box. Bitte testen Sie die Funktion von vielen Rechnern innerhalb des lokalen Netzes, damit Sie sicher sein können dass alles funktioniert. Wenn der DNS-Server nämlich nicht erreichbar ist, erhalten Sie von den Clients Fehlermeldungen die sie in die Irre führen könnten (z.B. Netzwerk nicht erreichbar).

Apache - der freie Webserver

Warum der Indianer der verbreitetste seiner Art ist

Jeder nahm mit ihm schon mal Kontakt auf. Jeder nahm seine Services schon mal in Anspruch und die wenigsten wissen, dass er erst vor kurzem zehn Jahre alt wurde. Mittlerweile erreicht Apache mit der Versionsnummer 2 an die siebzig Prozent markanteil. Absolut sind das siebzig Millionen Websites die durch die Dienste des Indianers der Öffentlich zugänglich gemacht werden.

von Martin Müller

Apache ist eine Abkürzung für "A Patchy Server", da durch patchen (verändern) seines Vorgängers NCSA-HTTPD entstand. Bitte beachten Sie: wir verwenden bei der WCM-Linux-Box den Apache in der Version 2.0.54-5 und nicht die Version 1.3. Diesen Umstand müssen Sie sich immer bewusst machen, da Sie sonst schnell mal die falschen Konfigurationsdateien bearbeiten oder die falschen Module installieren.

Zusammengefasste Vorteile:

Apache 2 kann verschiedene Laufzeitmodelle einsetzen. Ein Laufzeitmodell beschreibt die Art und Weise, wie der Webserver neue Prozesse erzeugt und Reserven schafft, wenn zu viele Anfragen an ihn gleichzeitig gestellt werden.

Apache 2 nutzt die Apache Portable Runtime. Sie stellt eine plattformunabhängige Schnittstelle zum Kern des Webservers her. Entwickler von zusätzlichen Modulen brauchen sich nicht um Speicherverwaltung, Datei- und Netzwerkfunktionen und ähnliches zu kümmern.

Die Möglichkeit Multi Processing Modules einzusetzen, ist ein weiterer Unterschied zwischen Apache 2 und seinem Vorgänger. MPMs können sich spezielle Fähigkeiten des Betriebssystems, auf dem Apache 2 läuft, zu nutze machen und somit sein Laufzeitmodell besser auszunutzen. Endlich sind auch mehrsprachige Fehlermeldungen, die an den Client ausgegeben werden, möglich. Und die vereinfachte Konfiguration sollte auch nicht unter den Tisch gekehrt werden.

Weiters sind die Implementierung des neuen Internetprotokolls IPv6, die neue Programmierschnittstelle (API) und die aktualisierte Bibliothek für reguläre Ausdrücke erwähnenswert.

Vor der Installation bringen wir die Paketlisten des Paketmanagers apt dur *apt-get update* auf den letzten Stand. Anschließend installieren wir auf unserer Maschine durch

apt-get install apache2-common apache2-utils libapache2-mod-php5 libapache2-mod-auth-mysql libapache2-webauth libapache2-mod-authpam den Webserver Apache 2, dazugehörige Werkezuge, die PHP5-Erweiterung, eine Erweiterung die es erlaubt sich über eine MySQL-Datenbank an passwortgeschützten Seiten anzumelden und schlussendlich zwei Erweiterungen für andere Authentifizierungsarten.

apt wird melden, dass das Modul apache2-mpm-worker entfernt wird. Das mpm-worker-Modul ist ein hybirdes Laufzeitverhalten, dass Client-Anfragen durch sogenannte Threads und gleichzeitig durch Prozesse abarbeitet. Dieses Modul bietet bei Servern mit vielen Anfragen sicherlich einige Vorteile, ist aber zu PHP nicht kompatibel. Es wird durch das Standardlaufzeitverhal-

ten des Apache 1.3X mpm-prefork ersetzt. Das Prefork-Modul legt sofort nach dem Apache gestartet ist, ein in der Konfigurationsdatei /etc/apache/httpd.conf festgelegte Anzahl an Kindprozessen an. Diese Prozesse warten nun auf Anfragen von Clients. Gehen die Prozesse zur neige weil Anfragen gestellt werden, startet der Hauptprozess weitere Kindprozesse auf Reserve. Diese werden wieder automatisch geschlossen wenn die Last abnimmt

Um nun sicher zu gehen, dass auch tatsächlich nur Apache 2 werkt, versuchen wir durch *apt-get remove* —*purge apache* den Apache in der Version 1.3 zu entfernen. Der Befehl wird durch eine Fehlermeldung abgeschlossen wenn

die Version 1.3 nicht installiert ist. Anderfalls wird Apache 1.3 und seine Konfigurationsdateien entfernt.

Die Konfiguration

Die zentrale Konfigurationsdatei für Apache 2 findet sich im Verzeichnis */etc/apache2/* und nennt sich apache2.conf. In dieser Datei können Sie einige wichtige Einstellungen vornehmen. Die Datei ist hervorragend in englischer Sprache kommentiert. Editieren Sie die Datei mit dem Editor pico und navigieren Sie zu dem Eintrag <IfModule prefrok.c>. Erhöhen Sie den Wert MaxClients auf 40, dadurch verdoppeln wir

den Wert der möglichen gleichzeitigen Zugriffe auf den Server. Sind mehr als 40 gleichzeitige Zugriffe, werden diese nicht verworfen, sondern in eine Warteschlange gereiht und von dort aus bei freier Kapazität abgearbeitet. Der Benutzer bekommt die angeforderte Seite also erst nach einer kleinen Verzögerung ausgeliefert. Stellen Sie den MaxClients-Wert nie höher als 256, die Performance wird davon nachhaltig schlecht beinflusst! Selbst auf großen Systemen wird meist nicht mehr als 156 verwendet!

Jetzt trennt uns ein wichtiger Schritt von vielen anderen Distributionen. Wir manipulieren nicht die Datei httpd.conf im selben



Verzeichnis, sondern erstellen eine eigene Konfigurationsdatei im bereits vorhanden Unterverzeichnis sites-enabled. In unserer neuen Datei 000-httpd_own definieren wir den Server, nehmen Einfluss auf seine Performance sowie auf sein Verhalten bei Fehlern und Aufrufen.

Bei anderen Betriebssystemen manipulieren Sie dazu die Datei httpd.conf, unter Debian ist diese nur mehr aus Kompatibilitätsgründen vorhanden. Da Apache 2 mehr als nur eine Site hosten kann, ist es sinnvoll diese jeweils in einer separaten Datei zu konfigurieren. So würden wir für die virtuellen drei Hosts www.wcm.at, www.apache.org und www.flightxpress.de jeweils eine eigene Datei unter sites-enabled erstellen. Übersichtlich und kompakt!

Virtuelle Hosts

pico /etc/apache2/sites-enabled/ 000-*httpd_oun* öffnet uns eine Datei in der wir mit den Konfigurationsanweisungen beginnen.

Der wichtigste Eintrag ist der Name dieses virtuellen Hosts, definiert durch NameVirtualHost \$IP-Adresse. Der Eintrag \$IP-Adresse muss durch die IP-Adresse ersetzt werden, an der der Server ansprechbar sein soll.

Danach beginnen wir mit der Eröffnung des Host-Eintrages. Die Anweisungen für den Host werden von <VirtualHost *> und </VirtualHost> eingeschlossen. Alle Einträge zwischen diesen beiden Angaben beziehen sich auf diesen einen Host. Bitte beachten Sie, dass in den Konfigurationsdateien jede Zeile einen neuen Eintrag darstellt. Sie dürfen also nicht nach Belieben Zeilenumbrüche setzen.

Für unseren ersten Host starten wir also mit <VirtualHost *> und legen in der nächsten Zeile

Beispiel der Konfig-Datei

#/etc/apache2/sites-enabled/000-httpd_own ServerName www.testserver1.at

<VirtualHost *>

DocumentRoot "/var/www/server1" ServerName www.testserver1.at ServerAlias www.testserver1.at beispielserver.at *.testing.at </VirtualHost>



den ServerNamen fest.

Dieser sollte ein FQDN (Full Qualified Domain Name) sein, wie er im DNS eingetragen wurde. Haben Sie keinen Eintrag im DNS, weil Sie zum Beispiel keinen öffentlichen Server betreiben wollen oder kein DNS einsetzen, so geben Sie bitte die IP-Adresse des localhost ein. Definiert wird der Name durch den Syntax ServerName testserver1.at oder für localhost ServerName 127.0.0.1. Erlaubt ist natürlich auch die Angabe der öffentlichen IP-Adresse. Vorteil der IP-Adresseneingabe ist, dass dadurch keine DNS-Anfrage gestartet wird, weshalb der Server ein wenig schneller betriebsbereit ist

Als nächsten Schritt legen wir ServerAliase fest. Dies sind Adresseinträge, bei deren Aufruf sich unser Apache 2 zuständig fühlt. Server Alias www.testserver1.at beispielserver.at *.testing.at bewirkt, dass beim Aufruf einer der ersten beiden angegebenen Adressen, unser Server die Auslieferung der Dokumente startet. Der dritte Eintrag wird durch einen Stern angeführt. Dieser Stern ist Platzhalter für alle Zeichen die vor der Domain angegeben werden. Eine einfache Art, den Apache2 zu einem Catch-All-Server zu machen.

> Wichtig ist auch dass fsich ür die angeführten Domänen Verweise im DNS befinden (IP-Adressen "zeigen" auf Apache 2). Mehr dazu im DNS-Artikel in dieser Ausgabe.

Danach legen wir das Stammverzeichnis fest, in dem sich die HTML-Dateien und andere Daten die zur Veröffentlichung bestimmt sind, befinden. Standardmäßig ist dies /var/www/, sie können aber auch jeden beliebigen anderen Ort angeben. Wir verwenden den Syntax *DocumentRoot "/var/www/server1*" um das Unterverzeichnis *server1* anzugeben.

Wir legen mit *mkdir /var/www/* server1/ ein neues Verzeichnis an und vergeben durch chown wwwdata /var/www/server1 && chgrp www-data /var/www/server1 die Dateirechte für den Apache 2.

Einige Tuningmaßnahmen

ServerAdmin admin@testserver1.at gehört zum guten Ton bei der Installation. Treten bei der Darstellung Ihrer Seiten Probleme auf, wird auf den Fehlerseiten diese Email-Adresse angeben.

Außerdem erreichen wir durch HostnamesLookup off dass der Server auf keinen Fall den Hostnamen des anfragenden Clients ermitteln will. Sie sollten diese Option unbedingt setzen, um unnötige DNS-Abfragen zu verhindern.

Des Weiteren setzten wir die Anweisung *EnableSendfile On*. Dadurch liefert unser Apache 2 den Inhalt der Daten an den Client, ohne dass das Betriebssystem die Datei selbst ausliest. Wir erreichen somit ein besseres Antwortverhalten. Beenden Sie den Editor und sichern Sie die Datei.

Die Konfiguration ist somit fast abgeschlossen. Der Ordnung halber legen wir noch einen symbolischen Link unserer Konfigurationsdatei in das Verzeichnis /etc/ apache2/sites-available (*ln -s /etc*/ apache2/sites-enabled/000-httpd_own /etc/apache2/sites-available/).

Wir stoppen unseren Server durch /etc/init.d/apache2 stop und testen die Konfigurationsdatei auf Syntaxfehler durch Aufruf von apache2 -t. Als Ergebnis erhalten Sie hoffentlich Syntax OK, sie können Apache 2 nun starten (/etc/ init.d/apache2 start).

Weitere Hilfsmittel

Im Konfigurationsverzeichnis befindet sich außerdem noch die Datei magic. Wenn Sie diese Option in der Datei /etc/apache2/ httpd.conf in einem Server aktivieren (MimeMagicFile /etc/apache/magic), dann prüft Apache 2 jede Datei vor der Auslieferung ob die Dateiendung (.jpg) auch tatsächlich mit dem Dateiinhalt übereinstimmt. Wenn keine Dateiendung vorhanden ist, wir auch dies erkannt und der MimeTyp richtig gesetzt. Jede Datei wird also mit Angaben aus der Datei magic verglichen und entsprechend ausgeliefert. Auf großen System sollten Sie dieses Modul aus Performancegründen lieber nicht aktivieren sondern von Haus aus auf die richtige Benennung achten.

Der Apache 2 Webserver bietet weit über 200 Konfigurationsoptionen, die ihm zu einem wahren Alleskönner ausbauen. Der Apache darf kostenlos genutzt werden und hat seine eigene Lizenz unter deren geringen Einschränkungen er verteilt werden darf. Wir werden ihn in Zukunft aufgrund seiner guten Performance und seines modulares Systems zu einem wichtigen Bestandteil der WCM-Linux-Box ausbauen.