

Linux als Webserver

› Der Webserver Apache wird von vielen Administratoren als das Non-Plus-Ultra für den eigenen Webauftritt gefeiert. Niedrige Ausfallzeiten und hohe Sicherheit sind tatsächlich gute Argumente für Apache.

› VON OLIVER MÜLLER

Vor dem eigenen Internet- oder Intranetauftritt steht zunächst die Einrichtung und Konfiguration des Servers mit Betriebssystem und des eigentlichen HTTP-Servers. Mit dem Gespann aus Linux und Apache steht dabei eine preisgünstige, sichere und leistungsfähige Plattform zur Verfügung. Auch tecChannel.de basiert aus gutem Grund auf dem Apache-Webserver unter Linux.

Wie die wichtigsten Bestandteile des Linux-Betriebssystems eingerichtet werden, erläutern Ihnen frühere Beiträge von tecChannel.de, die Sie im Online-Archiv finden. In diesem Artikel lesen Sie, wie Sie Apache auf dem Linux-Server installieren und konfigurieren, so dass dem eigenen Webauftritt eigentlich nichts mehr im Wege steht.

› Das erste Rüstzeug

Die meisten Linux-Distributoren konfigurieren Apache schon vor, so dass Sie in aller Regel nicht dazu gezwungen sind, den Webserver von Scratch zu konfigurieren. Sie werden vielmehr Apache an Ihre Bedürfnisse anpassen. Selbst wenn Sie Apache direkt von der [Homepage](http://www.apache.org) (<http://www.apache.org>) downloaden, finden Sie eine grundlegende Konfiguration vor.

Die Konfiguration von Apache erfolgt für Unix typisch über Textdateien. Das erste Problem, das sich dem angehenden Webadministrator stellt, ist die Frage nach dem Verzeichnis, in dem die Konfigurationsdateien zu finden sind. Diese können je nach Distribution an ganz unterschiedlichen Stellen liegen:

| Distribution | Position der Dateien |
|-----------------|------------------------|
| Original Apache | /usr/local/apache/conf |
| Red Hat | /etc/httpd/conf |
| SuSE | /etc/httpd |

Falls Sie eine andere Distribution verwenden, deren Softwareverwaltung auf RPM aufbaut, können Sie über das betreffende RPM-Paket die Lage der Konfigurationsdateien herausfinden. Vorausgesetzt Apache findet sich bei Ihrem Linux-System in einem Paket `apache*.rpm`, erfahren Sie das Konfigurationsverzeichnis durch folgenden Shell-Befehl:

```
» rpm -ql apache | grep httpd.conf «
```

Die zentrale Datei zum Setup Ihres Webserver ist `httpd.conf`. Die meisten Einstellungen führen Sie über Einträge in dieser Datei durch. Weiter gibt es `access.conf`, über die Sie Zugriffsrechte und Dienste der einzelnen Verzeichnisse festlegen. Mittels `srm.conf` legen Sie den Namensbereich fest, den die Benutzer Ihres Webserver sehen.

› Skripte für den Webserver

Den Start des Webserver überlassen Sie am besten einem Initscript. Unter Linux-Systemen, die zu Red Hat kompatibel sind, tragen Sie mit Tools wie `linuxconf` schlicht den `httpd`-Dienst zum automatischen Start ein. Bei SuSE Linux verwenden Sie entweder YaST oder Sie setzen in der Datei `» /etc/rc.config «` den Eintrag `START_HTTPD` auf "yes".

In Ihren Verzeichnissen mit den Initscripts können Sie Apache auch direkt starten. Das betreffende Initscript finden Sie bei Red Hat unter » `/etc/rc.d/init.d/httpd` « . Bei SuSE liegt es unter » `/sbin/init.d/httpd` « . Diesem Script können Sie die Argumente "start", "stop" und "reload" übergeben, um Apache zu starten, anzuhalten oder neu zu starten.

Immer wenn Sie die Konfiguration des Apache-Servers durch Bearbeiten der betreffenden Dateien verändern, starten Sie den Webserver erneut, damit die Modifikationen wirksam werden.

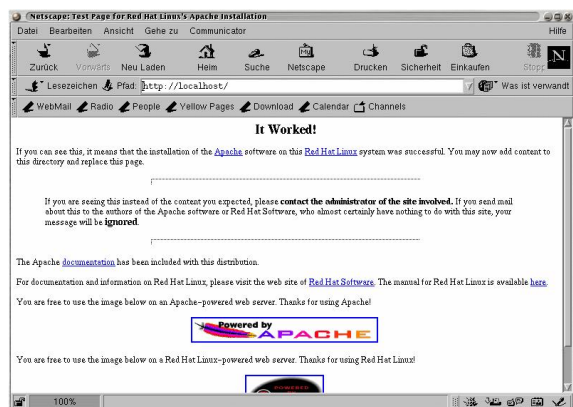
Es wäre jedoch sehr nervenaufreibend beziehungsweise sogar absolut unpraktikabel, wenn Sie hierzu den Computer jedes Mal neu booten müssten. Stattdessen verwenden Sie einfach das betreffende Initscript mit dem Argument "reload". Für Red Hat sieht das dann wie folgt aus:

```
» /etc/rc.d/init.d/httpd reload «
```

In einigen Fällen kann das Script auch `apachectl` heißen. Die Bedienung bleibt jedoch dieselbe.

› Der erste Start

Nach dem Start von Apache können Sie Ihren Browser aufrufen und die Seiten des lokalen Webserver ausprobieren. Geben Sie hierzu als URL `http://localhost` ein und schauen Sie was passiert. Sollte Ihre Distribution den Webserver bereits in einen funktionstüchtigen Zustand versetzt haben, sehen Sie jetzt eine Seite mit distributionsabhängigen Informationen.



Kurzer Test: Wenn Apache nach dem ersten Start diese Seite anzeigt, hat es funktioniert.

Die erste anstehende Änderung ist das Einstellen der eigenen Webseiten. Hierzu findet sich in der Datei » `httpd.conf` « oder » `srm.conf` « der Eintrag `DocumentRoot`. Das hier angegebene Verzeichnis verwendet Apache als Wurzelverzeichnis - und damit als Eintrittspunkt - für den HTTP-Server.

Sie können nun das hier angegebene Directory entleeren, also sämtliche Verzeichnisse und Dateien löschen. Diesem Prozedere würde jedoch auch in den allermeisten Fällen die Apache-Dokumentation zum Opfer fallen. Der bessere Weg ist es, ein separates Verzeichnis für Ihre Website anzulegen und einzustellen. Darin können Sie Ihre Webseiten hochziehen.

› Standarddokumente

Sowie Sie Ihre Domain ohne explizite Angabe eines HTML-Dokumentenpfades in einem Browser als URL eingeben, wird automatisch das Standarddokument aus dem mit `DocumentRoot` angegebenen Verzeichnis geöffnet. Sicherlich fragen Sie sich jetzt, woher Apache weiß, welches HTML-File das Standarddokument ist.

Hierzu existiert der Eintrag `DocumentIndex` in » `httpd.conf` « beziehungsweise » `srm.conf` « . Die Dateinamen, die Sie in diesem Eintrag auflisten, werden als Standarddokument verwendet. Dies gilt dabei nicht nur für das Wurzelverzeichnis, sondern für alle

Verzeichnisse Ihrer Website. Sobald als URL ein Verzeichnis statt eines HTML-Files angegeben wird, sucht Apache eine solche über » *DocumentIndex* « angegebene Datei in dem betreffenden Verzeichnis. Erst wenn in diesem Verzeichnis keine solche Datei existiert, listet Apache je nach Konfiguration entweder den Verzechnisinhalt auf oder präsentiert eine Fehlermeldung.

Angenommen in Ihrer Konfiguration findet sich die Zeile

```
» DocumentIndex index.html index.htm index.cgi «
```

Apache sucht dann zuerst nach einer Datei *index.html*. Ist diese nicht vorhanden, wird nach *index.htm* gesucht. Zu guter Letzt wird *index.cgi* verwendet, soweit vorhanden.

› **Andere Verzeichnisse im HTTP-Baum**

Neben dem Wurzelverzeichnis können Sie auch andere Verzeichnisse in den HTTP-Baum einbinden. Diese Verzeichnisse stehen dann über einen von Ihnen festgelegten Pfad zur Verfügung.

Allgemein lassen sich über die Alias-Direktive Verzeichnisse aus dem gesamten Linux-Dateibaum in Apache einbinden. Diese Direktive kann in allen drei Konfigurationsdateien von Apache verwendet werden.

Angenommen Sie wollen das Verzeichnis */mnt/www/SuperProdukt* unter der URL <http://www.IhreDomain.de/SuperProdukt> zur Verfügung stellen. In diesem Fall verwenden Sie die Zeile

```
» Alias /SuperProdukt /mnt/www/SuperProdukt
```

Eine weitere Möglichkeit, zusätzliche Verzeichnisse in den HTTP-Baum einzufügen, sind die HOME-Verzeichnisse der jeweiligen Benutzer. Wenn Sie die Zeile

```
» UserDir public_html «
```

in » *httpd.conf* « oder » *srm.conf* « eintragen, kann jeder Benutzer seine Homepage anlegen. Er muss lediglich seine Webseiten in dem Unterverzeichnis *public_html* des eigenen HOME-Verzeichnisses erstellen. Danach sind die Seiten über die URL <http://www.IhreDomain.de/~fritz> verfügbar.

Wollen Sie nicht, dass die Benutzer eigene Seiten ins Internet/Intranet/Extranet stellen, so lässt sich diese Funktion über die Zeile

```
» UserDir disabled «
```

auch komplett ausschalten. Führen Sie hingegen zusätzlich bei dieser Zeile noch Benutzernamen auf, so können nur die betreffenden Benutzer keine Homepage freischalten. Um root und fritz auszuschalten, geben Sie

```
» UserDir disabled root, fritz «
```

an.

Auch der umgekehrte Weg ist denkbar. Über

```
» UserDir enabled fritz, marina, theo «
```

gestatten Sie den Benutzern *fritz*, *marina* und *theo*, eine eigene Homepage aufzubauen. Dazu ist zusätzlich über die Option » *UserDir disabled* « den anderen Benutzern die eigene Homepage zu verweigern.

› **Das Ruder in der Hand - Zugriff steuern**

Die Zugriffsrechte auf die einzelnen Verzeichnisse Ihres HTTP-Baumes steuern Sie über die Konfigurationsdatei » *access.conf* « . Sie können zu jedem Verzeichnis über eine Directory-Struktur festlegen, welche Möglichkeiten für Zugriff und Dienste (wie beispielsweise Ausführen von CGI-Skripts) bestehen.

In dem Directory-Tag geben Sie das Verzeichnis aus dem Linux-Dateibaum an, für das Sie die Rechte festlegen wollen. Hier ist jetzt vor allem auch wichtig, dass Sie - sofern Sie vorher *DocumentRoot* verändert haben - die betreffende Directory-Struktur ändern. Überschreiben Sie einfach den alten Pfad mit Ihrem neuen *DocumentRoot*.

Eine Directory-Struktur besteht immer aus einem einleitenden und abschließenden Tag. Für das Verzeichnis /mnt/www/SuperProdukt wird die Struktur beispielsweise mit

```
» <Directory /mnt/www/SuperProdukt> «
```

eingeleitet und durch

```
» </Directory> «
```

abgeschlossen. Alle Zeilen zwischen diesen Tags beziehen sich dann ausschließlich auf das Verzeichnis /mnt/www/SuperProdukt.

Einschränkungen in den Diensten können Sie über den Optionseintrag innerhalb der Directory-Struktur bewirken. Fehlt ein solcher Eintrag sind alle Dienste möglich. Hier sind die folgenden Angaben denkbar:

| Argument von Options | Beschreibung |
|----------------------|--|
| All | Alle Optionen (außer MultiViews) werden eingeschaltet (Default). |
| ExecCGI | Ausführen von CGI-Scripts ist erlaubt. |
| FollowSymLinks | Befinden sich in dem Verzeichnis symbolische Links, wird diesen gefolgt. Beachten Sie, dass diese quer durch Ihr System führen können. |
| Includes | Serverseitige Includes sind gestattet. |
| IncludesNOEXEC | Serverseitige Includes sind erlaubt, aber #exec und #include von CGI-Scripts ist ausgeschaltet. |
| Indexes | Wird kein Standarddokument (vgl. DirectoryIndex) in dem Verzeichnis gefunden, wird der Inhalt des Verzeichnisses aufgelistet. Wenn diese Option fehlt, kommt es stattdessen zur Ausgabe einer Fehlermeldung. |
| MultiViews | Durch den Inhalt ausgelöste MultiViews sind erlaubt. |
| SymLinksIfOwnerMatch | Der Server folgt nur symbolischen Links, wenn das Ziel die gleiche User-ID hat, wie der Link selbst. |

Wollen Sie beispielsweise, dass der Inhalt eines Verzeichnisses ausgegeben wird, keine Fehlermeldung erscheint und außerdem symbolischen Links gefolgt wird, so geben Sie » *Options Indexes FollowSymLinks* « an. Das Ausführen von CGI-Scripts ist in diesem Fall dann beispielsweise aus diesem Verzeichnis nicht möglich.

Eine Directory-Struktur bezieht auch immer gleich alle Unterverzeichnisse mit ein, sofern für diese nicht eine separate Directory-Struktur verwendet wird.

› Wer darf und wer nicht?

Selbst in einem Intranet - und schon gar nicht im Internet - ist es allen erlaubt, alles zu sehen. Angenommen, eine Firma stellt vertrauliche Geschäftsberichte ins Intranet, Sinn der Sache wird allerdings nicht sein, dass alle Mitarbeiter diese lesen können. Hier muss der Zugriff beschränkt werden.

Am einfachsten beschränken Sie den Zugriff auf die IP-Adressen der Mitarbeiter im Netzwerk, die solche vertraulichen Informationen verwenden sollen. Hierzu gibt es drei wichtige Klauseln - *order*, *allow from* und *deny from*. Über *allow from* geben Sie an, welche IP-Adressen (oder Host-Namen) auf das Verzeichnis Zugriff haben. Die Option *deny from* wirkt umgekehrt und legt fest, welchen IP-Adressen der Zugriff untersagt ist. Via *order* geben Sie an, ob zuerst die allow- oder die deny-Regeln ausgewertet werden sollen.

Wenn Sie den Rechnern Oxygen, Nitrogen und Argon Zugriff auf ein Verzeichnis geben wollen, anderen jedoch nicht, geben Sie folgende Konstellation an:

```
» order allow,deny
```

» *deny from all*
» *allow from Oxygen Nitrogen Argon*

Diese Angaben bewirken, dass zunächst die allow- und danach die deny-Regel ausgewertet werden. Die allow-Klausel ermöglicht den Zugriff für die drei Hosts. Greift ein anderer Host zu, findet ihn Apache in dieser allow-Regel nicht und wendet die deny-Anweisung an. Hier wird allen Hosts der Zugriff verweigert.

Bei herkömmlichen Webseiten fürs Internet, die alle sehen sollen, verwenden Sie folgende Konstellation:

» *order allow,deny*
» *allow from all*

Eine deny-Klausel ist hier nicht erforderlich. Nachdem für alle Hosts der Zugriff freigegeben wurde und die allow- vor den deny-Regeln ausgewertet werden, würden deny-Regeln ohnehin niemals greifen.

Hinter deny und allow können Sie auch Domains und [Subnetze](#) (<http://www.tecchannel.de/internet/209/0.html>) angeben. Eine Sperr-Regel für alle Hosts der Domain `.warenausgang.firma.de` würde wie folgt aussehen:

» *deny from .warenausgang.firma.de* «

Um das Subnetz 192.168.0.0 mit der Netzmaske 255.255.255.0 freizuschalten, geben Sie zum Beispiel

» *allow from 192.168.0.0/255.255.255.0* «

in der Directory-Struktur an.

› Gezielter steuern

Sie können die Angaben der Directory-Strukturen auch gezielter setzen. Wenn Sie für eine ganze Reihe von Unterverzeichnissen die Zugriffsrechte individuell setzen müssen, kann die Bearbeitung der Datei » *access.conf* « sehr schnell zur Sisyphusarbeit werden. Sie müssten ja für jedes Verzeichnis eine Directory-Struktur schreiben. Wenn sich dann noch Ihr Webcontent extrem häufig ändert, werden Sie Ihres Lebens wohl kaum mehr froh werden.

Sie können daher die Zugriffsrechte auch in separaten Dateien in den jeweiligen Verzeichnissen verstauen. Schreiben Sie hierzu eine Directory-Struktur für das übergeordnete Verzeichnis und nehmen Sie die Zeile

» *AllowOverride all* «

darin auf. Ab jetzt können Sie die Optionen der Directory-Struktur in den einzelnen Verzeichnissen überschreiben, indem Sie im betreffenden Verzeichnis die Datei » *.htaccess* « anlegen. Darin können Sie nun Anweisungen mit *order*, *allow from* und *deny from* ablegen, die sich auf das entsprechende Verzeichnis auswirken.

Hinweis: Über *AllowOverride* sind noch wesentlich genauere Abstufungen möglich, auf die hier nicht weiter eingegangen wird. Genaueres entnehmen Sie der Dokumentation von Apache.

› Sicherheit eine Stufe höher

Den Zugriff auf Webinhalte können Sie bei Apache selbstverständlich auch durch Passwörter absichern. Hierzu müssen Sie jedoch für Apache eine eigene Passwort-/Benutzerdatei anlegen. Diese erstellen Sie mit dem Kommando

» *htpasswd -c /etc/httpd/passwd firstuser* « .

Hierbei wird die Datei » */etc/httpd/passwd* « erstellt und zugleich der erste Benutzer mit dem Namen `firstuser` eingerichtet. Nach Bestätigung des Befehls auf der Shell werden Sie zur Eingabe des Passworts aufgefordert.

Weitere Benutzer legen Sie an durch

» *htpasswd /etc/httpd/passwd <neuerNutzer>* « .

Damit die Passwortabfrage auch funktioniert, müssen Sie noch in die Directory-Struktur oder die .htaccess-Datei des entsprechenden Webserver-Verzeichnisses folgende Zeilen aufnehmen:

```
» AuthType basic
» AuthName "Das Passwort bitte"
» AuthUserFile /etc/httpd/passwd
```

Über AuthType legen Sie das Autorisierungsverfahren fest. Derzeit existieren hier lediglich basic und digest. Mit AuthName setzen Sie einen Prompt, der im Eingabedialog angezeigt wird. Last but not least gibt AuthUserFile die Passwort-/Userdatei an, die Sie mit *htpasswd* angelegt haben. Um mehrere Passwortdateien für verschiedene Zwecke anzulegen, verwenden Sie als Argument für *htpasswd* und AuthUserFile einfach andere Dateinamen.

› Fazit

Der erste Schritt zum eigenen Inter-/Intranet-Auftritt ist die Einrichtung und Konfiguration des Webserver. Unter Linux ist das mit Apache inzwischen kein großes Problem mehr. Man muss nur wissen, an welchen Knöpfen man zu drehen hat, um einerseits den Benutzern maximalen Komfort und andererseits größtmögliche Datensicherheit zu gewährleisten. Die Absicherung der Website durch Passwörter ist dabei nur ein Schritt unter vielen. Wie Sie den Server durch eine Firewall sichern, ist Bestandteil eines späteren Beitrags, und auch wie Sie die Funktionalität des Internetservers etwa um E-Mail erweitern, werden wir noch darlegen (siehe Online-Archiv von tecChannel.de). (mha)

› Weitere Themen zu diesem Artikel:

[Linux als Dial-up-Router \(http://www.tecchannel.de/betriebssysteme/322/index.html\)](http://www.tecchannel.de/betriebssysteme/322/index.html)

[Linux als Windows-Printserver \(http://www.tecchannel.de/betriebssysteme/392/index.html\)](http://www.tecchannel.de/betriebssysteme/392/index.html)

[Linux als Windows-Server \(http://www.tecchannel.de/betriebssysteme/248/index.html\)](http://www.tecchannel.de/betriebssysteme/248/index.html)

Copyright © 2001
IDG Interactive GmbH
Alle Rechte vorbehalten. Jegliche Vervielfältigung oder Weiterverbreitung in jedem Medium in Teilen oder als Ganzes bedarf der schriftlichen Zustimmung der IDG Interactive GmbH. DPA-Texte und Bilder sind urheberrechtlich geschützt und dürfen weder reproduziert noch wiederverwendet oder für gewerbliche Zwecke verwendet werden. Für den Fall, dass in tecChannel unzutreffende Informationen veröffentlicht oder in Programmen oder Datenbanken Fehler enthalten sein sollten, kommt eine Haftung nur bei grober Fahrlässigkeit des Verlages oder seiner Mitarbeiter in Betracht. Die Redaktion übernimmt keine Haftung für unverlangt eingesandte Manuskripte, Fotos und Illustrationen. Für Inhalte externer Seiten, auf die von tecChannel aus gelinkt wird, übernimmt die IDG Interactive GmbH keine Verantwortung.