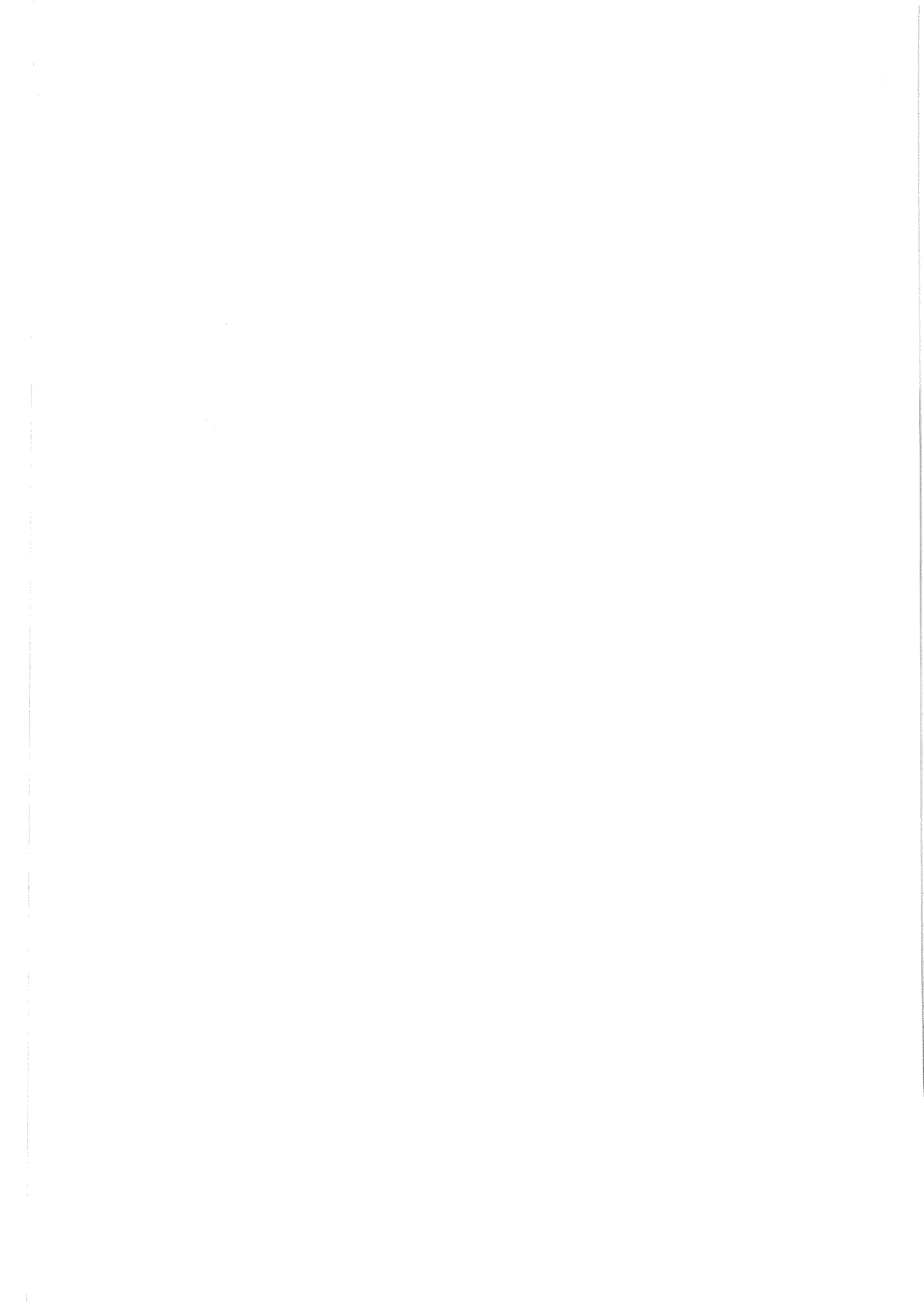


# Modul 239: Internetserver in Betrieb nehmen

## Grundlagen, Praxisbeispiele, Repetitionsfragen mit Lösungen

Umberto Annino





## **Modul 239: Internetserver in Betrieb nehmen**

---

Grundlagen, Praxisbeispiele, Repetitionsfragen mit Lösungen

---

Umberto Annino

---

Modul 239: Internetserver in Betrieb nehmen  
Grundlagen, Praxisbeispiele, Repetitionsfragen mit Lösungen  
Umberto Annino

Grafisches Konzept: dezember und juli, Wernetshausen  
Satz und Layout: Mediengestaltung, Compendio Bildungsmedien AG, Zürich  
Druck: Edubook AG, Merenschwand

Artikelnummer: 5632  
Auflage: 2. Auflage 2007  
Ausgabe: K1021  
Sprache: DE  
Code: ICTW 051

Verlag: Stiftung Wirtschaftsinformatikschule Schweiz, WISS

Alle Rechte, insbesondere die Übersetzung in fremde Sprachen, vorbehalten. Das Werk und seine Teile sind urheberrechtlich geschützt. Jede Verwertung in anderen als den gesetzlich zugelassenen Fällen bedarf der vorgängigen schriftlichen Zustimmung von der Stiftung Wirtschaftsinformatikschule Schweiz.

Copyright © 2006, Stiftung Wirtschaftsinformatikschule Schweiz, WISS



## Inhaltsverzeichnis

---

	<b>Über dieses Lehrmittel</b>	<b>5</b>
<b>Teil A</b>	<b>Grundlagen (Informationen beschaffen)</b>	<b>9</b>
	<b>Einleitung, Lernziele und Schlüsselbegriffe</b>	<b>10</b>
<b>1</b>	<b>Vorgehen bei der Informationsbeschaffung</b>	<b>11</b>
1.1	Wie beginne ich das Projekt?	11
1.2	Ausgangslage der Aufgabe	11
1.3	Anforderungen der Aufgabe	12
1.4	Wo liegt der Unterschied zwischen Projektmanagement und Projektarbeit?	12
1.5	Ergänzende Methoden	13
1.6	Zusammenfassung	13
<b>2</b>	<b>Wozu werden Internetserver eingesetzt?</b>	<b>15</b>
2.1	Einsatzgebiet	15
2.2	Anwendungen	16
<b>3</b>	<b>Wie werden Internetdienste erbracht?</b>	<b>19</b>
3.1	Welche Systeme werden benötigt?	19
3.2	Die Auswahl der Server-Hardware	19
3.3	Die Auswahl der Server-Software	21
3.4	Internetprotokolle	23
3.5	Technischer Aufbau: die Architektur	25
3.6	Anbieter und Abnehmer	27
<b>Teil B</b>	<b>Konzeption und Dimensionierung</b>	<b>29</b>
	<b>Einleitung, Lernziele und Schlüsselbegriffe</b>	<b>30</b>
<b>4</b>	<b>Ist-Situation analysieren</b>	<b>31</b>
4.1	Welche Dienstleistungen werden angeboten?	31
4.2	Wie sieht die aktuelle Serverumgebung aus?	32
<b>5</b>	<b>Soll-Zustand analysieren</b>	<b>34</b>
5.1	Welche Applikationen sollen «web-enabled» werden?	34
5.2	Allgemeine Sicherheitsrichtlinien im Unternehmen	34
5.3	Datenschutz und Personendaten	34
5.4	Anforderungen an die Verfügbarkeit (uptime) und Datenvolumen	35
5.5	Welche Benutzerprofile sind vorgesehen?	36
<b>6</b>	<b>Lösung entwerfen</b>	<b>38</b>
6.1	Benötigtes Know-how aneignen und planen	38
6.2	Benötigte Hardware festlegen	38
6.3	Sicherheit gewährleisten mit technischen und organisatorischen Massnahmen	39
6.4	Betriebssystem für den Internetserver definieren	39
6.5	Wie werden Applikationen an den Server angebunden?	40
6.6	Beschaffung der Software und Installationsabhängigkeiten	40
6.7	Zu verwendende Namen für Systeme, Dienste und Daten	41
6.8	Standardeinstellungen festlegen	42
<b>7</b>	<b>Systemtest und Dokumentation vorbereiten</b>	<b>43</b>
7.1	Technische Tests am Internetserver	43
7.2	Applikatorische Tests am Internetserver	43
7.3	Sicherheitstests rund um den Internetserver	44
7.4	Lasttest	45
7.5	Dokumentation des Internetserver	45

<b>Teil C</b>	<b>Internetserver realisieren</b>	<b>47</b>
	<b>Einleitung, Lernziele und Schlüsselbegriffe</b>	<b>48</b>
<b>8</b>	<b>Software installieren, Default-Einstellungen und Benutzer konfigurieren</b>	<b>49</b>
8.1	Software besorgen (Quelle, Version, Plattform)	49
8.2	SuSe Linux 10 OSS	49
8.3	Apache Webserver	54
8.4	DNS für Linux	77
8.5	TFTP für Linux	89
8.6	SMTP und Postfach für Linux mit Stalker CommuniGate	89
<b>9</b>	<b>Log-Services und Sicherungsprozeduren gemäss Lösungsentwurf aufsetzen</b>	<b>101</b>
9.1	Logfile-Rotation	101
9.2	Externer Logserver	101
9.3	Logfile-Analyse	101
9.4	Sicherungsprozeduren und Back-up	105
<b>10</b>	<b>Fremde Ressourcen anbinden</b>	<b>106</b>
10.1	Authentifizierungsserver	106
10.2	Datenbanken	106
10.3	Sicherheitsaspekte bei fremden Ressourcen	107
<b>Teil D</b>	<b>Internetserver testen, Betriebsübergabe</b>	<b>109</b>
	<b>Einleitung, Lernziele und Schlüsselbegriffe</b>	<b>110</b>
<b>11</b>	<b>Anforderungen an die Wartung definieren</b>	<b>111</b>
11.1	Betrieb des Internetservers	111
11.2	Dienstprogramme zur Systemüberwachung in SuSE Linux	111
<b>12</b>	<b>Systemabnahme und Betriebsübergabe planen</b>	<b>126</b>
12.1	Abnahmeprotokoll vorbereiten und prüfen	126
12.2	Systemabnahme bzw. -abgabe durchführen	126
12.3	Abnahmeprotokoll	126
<b>Teil E</b>	<b>Anhang</b>	<b>127</b>
	<b>Gesamtzusammenfassung</b>	<b>128</b>
	<b>Antworten zu den Repetitionsfragen</b>	<b>129</b>
	<b>Glossar</b>	<b>132</b>
	<b>Stichwortverzeichnis</b>	<b>139</b>

## Über dieses Lehrmittel

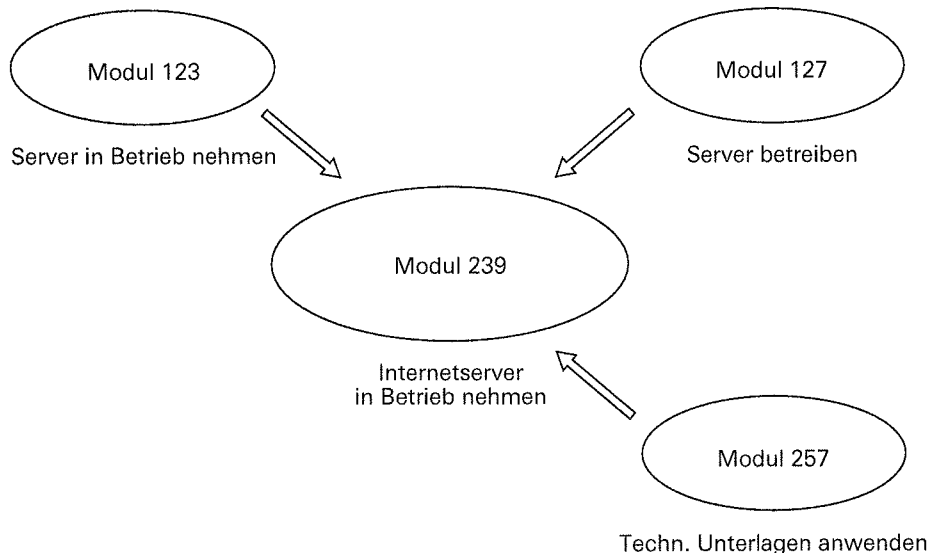
---

### Positionierung des Themas

---

Internetserver sind heute nicht mehr wegzudenken. Als zentrale Elemente der Informationstechnologie werden Internetserver täglich weltweit branchenübergreifend in Politik und Industrie benötigt.

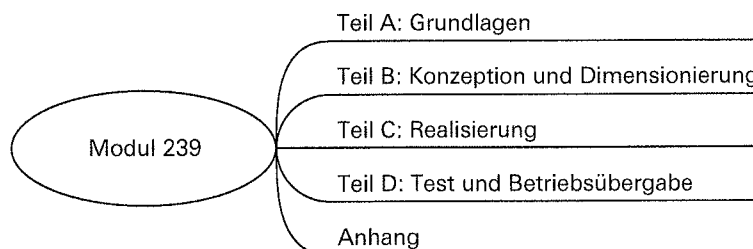
Das vorliegende Modul 239 bildet die Schnittstelle zu den Modulen 123, 127 und 257.



Dieses Lehrmittel vermittelt die Kenntnisse, um die Möglichkeiten und Einsatzgebiete der unterschiedlichen Protokolle und Internetdienste sinnvoll zu planen und umzusetzen. Das Hauptthema ist der Internetserver mit verschiedenen Diensten wie Webservice, E-Mail und Fileservice (FTP up- und Download).

### Inhalt und Aufbau dieses Lehrmittels

---



- **Teil A** macht Sie mit den Grundlagen der Materie vertraut. Sie lernen Einsatzgebiet, Hintergründe, Protokolle und Dienste von Internetservern kennen.
- **Teil B** befasst sich mit der konzeptionellen Ebene. Auftragserteilung, detaillierte Planung, Installation und Konfiguration der Serversoftware und entsprechenden Diensten.
- **Teil C** beinhaltet die Realisierung des Projekts. Darunter wird die Durchführung von Installation und Konfiguration inkl. Dokumentation des Systems verstanden. Eine Systemdokumentation soll gemäss dem installierten System erstellt werden.
- **Teil D** beinhaltet den Systemtest und das Lastverhalten der gesamten Installation vor Betriebsübergabe an den Abnehmer/Kunden.

- **Im Anhang** sind die Antworten der Lernfragen abgelegt. Zur Unterstützung der Lernenden wird zudem ein Glossar zum Thema angeboten.

## Dieses Lehrmittel liefert die Grundlage für den Erwerb folgender Kompetenz

Kompetenz: Internetserver mit HTTP und weiteren Diensten konfigurieren und in Betrieb nehmen, und dabei Sicherheitsvorgaben und betriebliche Anforderungen beachten.

Dieser Kompetenz sind die folgenden Handlungsziele zugeordnet:

- Anforderungen (Sicherheit, Lastprofil, Datenvolumen, Verfügbarkeit notwendiger Dienste, zu integrierende Applikationen) an einen Internetserver analysieren und dokumentieren.
- Bestehende Infrastruktur (Server, Netzwerk, Dienste) mit den Anforderungen abgleichen und bei Bedarf notwendige Anpassungen bzw. Erweiterungen vorschlagen.
- Erforderliche Standardeinstellungen gemäss Sicherheits- und Betriebskonzept realisieren.
- Software installieren, konfigurieren und notwendige Dienste einrichten.
- Zugriffsberechtigungen vergeben und Log-Services sowie Sicherungsprozeduren einrichten.
- Internetserver testen (Lasttest, Sicherheit, Crashtest).

## Technische/methodologische Voraussetzungen

Die technischen und methodologischen Voraussetzungen an den Lernenden und die Schulorganisation sehen wie folgt aus:

- Ein Internetserver mit entsprechender HW und SW pro Arbeitsgruppe
- Notebook oder eigener PC mit Dualbootfunktion für Installationen von Software unter Windows und Linux pro Lernenden
- Netzwerkequipment für das Erstellen einer Client/Server-Umgebung
- Konfigurationen von Internetservices

Begriff	Betriebssysteme	Einsatzschwerpunkte
Unix basierend	Bezieht sich auf: Linux/Unix Derivate und unterschiedliche Unixvarianten wie BSD, Free BSD und Solaris	Verzeichnisstruktur und Konfigurationsmöglichkeiten, unterschiedliche Installationsvarianten und Sicherheitsaspekte
Windows basierend	Windows 9X, ME, 2000, Server, XP Home/Pro, 2003 Server	Unterschiede bei Installation und Konfiguration von Windows und Unix/Linux
Webbasierend	plattformunabhängig	Zugriff nach Serverberechtigung, öffentlichen und privaten Bereich trennen

Da in der Praxis die unterschiedlichsten Webserverinstallationen zu erwarten sind, werden die im vorliegenden Lehrmittel beschriebenen Arbeiten auf Basis folgender Technologien durchgeführt:

- Webserver: der frei verfügbare und für mehrere Plattformen verfügbare Apache Webserver
- Betriebssysteme: SuSE Linux 9/10 und Windows 2000 (SP4) resp. XP (SP2)

## Literatur

---

Als Nachschlagewerk wird die technische Dokumentation des Apache Servers empfohlen. Grundsätzlich findet sich zum Thema Internetserver sehr viel Information in Form von «howto»-Anleitungen im World Wide Web.

## Für die Arbeit mit diesem Lehrmittel werden folgende Kenntnisse vorausgesetzt

---

- Server aufsetzen und installieren (Hardware)
- Kenntnisse der Hardware und generelle Kenntnisse für SW-Installationen
- Technische Dokumentationen
- Internet und Netzwerkdienste

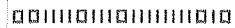
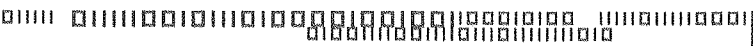
Besuchte Module	
Server in Betrieb nehmen	Modul 123
Server betreiben	Modul 127
Techn. Unterlagen anwenden	Modul 257

## Nützliche «Links» zum Thema

---

Die vorausgesetzten Kenntnisse, sowie Ergänzungen zu bestehendem Wissen, können mithilfe folgender Literaturhinweise und Internet Links angeeignet werden:

<a href="http://httpd.apache.org/docs/2.0/de/">http://httpd.apache.org/docs/2.0/de/</a>	Umfangreiche Dokumentation zum Apache Webserver (Deutsch)
<a href="http://www.communiGate.com/CommuniGatePro/">http://www.communiGate.com/CommuniGatePro/</a>	Umfangreiche Dokumentation zum Mailserver CommuniGate Pro (Englisch)
<a href="http://www.novell.com/de-de/documentation/suse10/">http://www.novell.com/de-de/documentation/suse10/</a>	Dokumentation zum SuSE Linux Betriebssystem und Funktionen (Deutsch)





## **Teil A Grundlagen (Informationen beschaffen)**

---

## Einleitung, Lernziele und Schlüsselbegriffe

---

### Einleitung

---

In diesem Teil werden mögliche Vorgehensmethoden für die Erarbeitung der Anforderungen an einen Internetserver vorgestellt. Anschliessend werden Internetserver und deren Einsatzgebiete näher vorgestellt. Es folgt eine Vorstellung der Dienste eines Internetserver und der dazu notwendigen Systeme und Software. Die für den Internetserver benötigten Protokolle sowie Architekturen und Bauweisen werden ebenfalls näher erläutert.

### Lernziele und Lernschritte

---

Lernziele	Lernschritte
<input type="checkbox"/> Kann die Anforderungen an Internetserver bez. Dienste und Verfügbarkeit ermitteln und begründen.	<ul style="list-style-type: none"><li>• Welche Anforderungen an einen Internetserver werden durch den Auftraggeber gegeben?</li><li>• Welche Verfügbarkeit muss der Internetserver aufweisen?</li></ul>
<input type="checkbox"/> Kann Anforderungen strukturieren, quantifizieren und deren Machbarkeit und Auswirkung beurteilen und dokumentieren.	<ul style="list-style-type: none"><li>• In welche Kategorien können die Anforderungen unterteilt werden?</li><li>• Wie sieht das Mengengerüst der Anforderungen aus?</li><li>• Welches sind die Auswirkungen auf Hard- und Software die von den Anforderungen ausgehen?</li></ul>

---

### Schlüsselbegriffe

---

http, dns, ftp, Mail, Datenschutz, Datensicherheit, IPERKA, SEUSAG, Dienst, Service, Hosting, Housing

# 1 Vorgehen bei der Informationsbeschaffung

---

Dieses Kapitel macht Sie mit der Vorgehensmethode im vorliegenden Lehrmittel vertraut. Das methodische Vorgehen im Projekt gehört zu den Grundlagen in diesem Lehrmittel.

## 1.1 Wie beginne ich das Projekt?

---

Eine gute Vorgehensmethode um das Projekt in Umfang und Tiefe zu bestimmen, ist es die erforderlichen Informationen des Auftraggebers zu beschaffen. Nachdem die Dimension des Projekts ersichtlich ist, können Sie entscheiden, welche Elemente in welcher Reihenfolge dazugehören.

Lassen Sie sich in der Praxis den Projektauftrag in schriftlicher Form, d. h. als unterschriebenes Auftragsblatt, geben.

## 1.2 Ausgangslage der Aufgabe

---

Beispiel: Sie arbeiten als Informatiker im Serverbereich der Firma «Info-Sales», einer Firma, die Informatiklösungen für Handelsfirmen realisiert. Die neue, firmeneigene Branchenlösung ist eine webbasierende Applikation. Sie soll baldmöglichst auf dem Markt eingeführt werden.

«Info-Sales» erteilt Ihnen den Auftrag, einen Internet/Intranetserver im bestehenden Netzwerk zu installieren. Für die Realisierung werden folgende Serverdienste benötigt:

- **http** hypertext transfer protocol: Dienst für Webserver (Client: Browser)
- **dns** domain name service: Dienst für Namensauflösung eines Domainnamens in IP-Adresse. Optional, kann auch durch einen separaten DNS-Server bereitgestellt werden oder mit sogenanntem «dynamischen DNS» betrieben werden: der Internet-Server hat dabei keine feste (statische) IP-Adresse.
- **ftpd** file transfer protocol daemon: Dienst für Dateiserver (Fileserver, FTP-Server), für Dateiaustausch (upload, Download)
- **Mail**: Elektronische Nachrichten, Dienst für versenden (SMTP simple mail transfer protocol) und abfragen von E-Mail-Nachrichten (POP post office protocol), Postfach für Benutzer. Optional, E-Mail-Server kann auch separat betrieben werden. Oft wird das Postfach auf einem Server geführt (z. B. Exchange-Server mit Microsoft Betriebssystem) und das Empfangen und Versenden von E-Mail-Nachrichten von und an andere E-Mail-Server wird durch den SMTP-Server erledigt.
- **https** hypertext transfer protocol secure: Verschlüsselte Variante von http, um die Vertraulichkeit der Daten während dem Transport zu gewährleisten (Abhörsicherheit), damit werden alle Datenpakete des http-Protokoll mit dem SSL (Secure Socket Layer) Protokoll verschlüsselt. Optional, nur wenn Verschlüsselung benötigt wird.

Damit die Realisierung des Projekts anhand von Ausgangslage und Auftragsziel durchgeführt werden kann ist eine Projektplanung unumgänglich.

### 1.3 Anforderungen der Aufgabe

Anhand des nachfolgenden Beispiels werden mögliche Anforderungen aufgezeigt.

#### Anforderungen an den Internetserver

- **Investitionskosten:** Für den neuen Internetserver wird ein Budget von max. CHF X'xxx.-, für Hard und Software, bereitgestellt.
- **Verfügbarkeit:** Die Verfügbarkeit des neuen Internetserver muss für die 5 Testkunden während den Bürozeiten gewährleistet sein.
- **Datenschutz:** Alle kundenrelevanten Daten die auf dem Internetserver gespeichert werden, müssen vor Zugriff durch Unbefugte sicher sein.
- **Datensicherheit:** Datenverlust durch Fehlmanipulation, Harddiskausfall und Viren muss mittels Bandsicherung oder Harddiskspiegelung (RAID 0 oder 5) gewährleistet sein. Die Sicherheit des Internetserver wird durch eine vorgeschaltete Paketfilter-Firewall gewährleistet, welche bereits angeschafft wurde und vom Provider konfiguriert wird. Der Internetserver ist dennoch mit minimalen Diensten zu betreiben (hardened Internet Server), d. h., der Server wird einer Härting unterzogen, damit unnötige Dienste keine Schwachstellen darstellen.
- **Interne und externe Abläufe und Prozesse:** Kunden, die auf Ihrem Webserver arbeiten werden, müssen Ihre Aufträge mit den gewohnten Werkzeugen erledigen können.
- **Termine:** Die Integration des Internetserver muss innerhalb von 14 Tagen abgeschlossen sei. Das Projekt muss mit der bestehenden Organisation und vorhandener Informatikmittel (Ausnahme ist der neue Server) durchführbar sein.

### 1.4 Wo liegt der Unterschied zwischen Projektmanagement und Projektarbeit?

Für die Durchführung des Realisierungsprojekts Internetserver wird eine gängige Vorgehensmethode angewendet. Die Vorgehensmethode IPERKA geht davon aus, dass die meisten Projekte im Informatikbereich in sechs Phasen abgewickelt werden können. Die Funktionen von Projektmanager und Projektmitarbeiter unterscheiden sich dabei grundsätzlich.

[1-1] Zusammenfassung der Phasen und Schritte der IPERKA Methode

Phase	Schritte	Tätigkeiten
		Fragen
1	I (Information)	<ul style="list-style-type: none"> <li>• Sie klären ab, was der Auftraggeber möchte.</li> <li>• Sie beschaffen sich die notwendigen Informationen und werten diese aus.</li> </ul>
		<ul style="list-style-type: none"> <li>• Wie lauten Auftrag und Vorgaben?</li> <li>• Welche Informationen muss ich mir beschaffen?</li> </ul>
2	P (Planen)	<ul style="list-style-type: none"> <li>• Sie planen das Projekt.</li> <li>• Sie schlagen Verantwortliche für Projektteile vor.</li> <li>• Sie legen Ziel, Konzept und Lösungsweg fest.</li> </ul>
		<ul style="list-style-type: none"> <li>• Welche Aufgaben sind zu lösen?</li> <li>• Wie lässt sich das Projekt realisieren?</li> </ul>
3	E (Entscheiden)	<ul style="list-style-type: none"> <li>• Sie vergleichen Varianten und entscheiden sich für eine Lösung.</li> </ul>
		<ul style="list-style-type: none"> <li>• Kann ich mein Vorhaben fortsetzen?</li> <li>• Wer trägt die Verantwortung?</li> </ul>

Phase	Schritte	Tätigkeiten
		Fragen
4	R (Realisieren)	<ul style="list-style-type: none"> <li>Sie beginnen mit der Umsetzung.</li> <li>Dabei arbeiten Sie genau nach Ihrem Plan.</li> </ul>
		<ul style="list-style-type: none"> <li>Arbeiten wir nach den Vorgaben?</li> <li>Arbeiten wir nach Zeitplan?</li> </ul>
5	K (Kontrollieren)	<ul style="list-style-type: none"> <li>Projektergebnis mit den Anforderungen vergleichen.</li> </ul>
		<ul style="list-style-type: none"> <li>Welche Mängel sind zu beheben?</li> </ul>
6	A (Abschliessen)	<ul style="list-style-type: none"> <li>Protokoll zur Übergabe Ihres Produkts erstellen.</li> </ul>
		<ul style="list-style-type: none"> <li>Wann sind die ersten Unterhaltsarbeiten durchzuführen?</li> </ul>

**Projektmanagement** wird durch den Projektleiter oder Projektmanager angewendet, um Strukturierung und Durchführung eines ganzen Projekts zu planen.

**Projektarbeit** befasst sich mit Teilproblemen einzelner Projektphasen.

Der Projektmitarbeiter ist diesbezüglich meistens ein Fachspezialist, während der Projektleiter ein Organisator und Generalist ist.

## 1.5 Ergänzende Methoden

Die SEUSAG-Methode (nach Dr. Goetz Schmidt) bildet eine Ergänzung zu IPERKA.

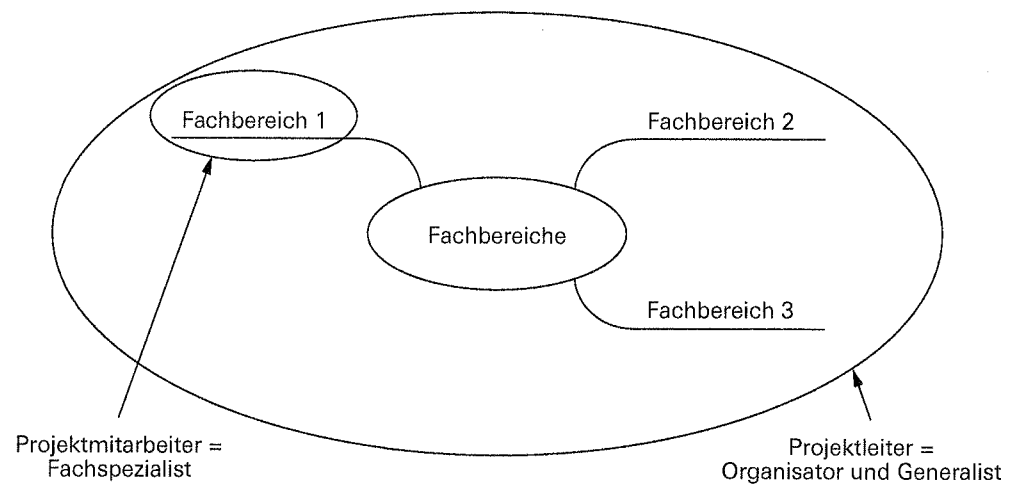
SEUSAG bedeutet:

Kurzbezeichnung	Erläuterung
Systemgrenzen	Was gehört zu ihrem Projekt und was nicht?
Einflussgrößen ermitteln	Welche Größen üben welchen Einfluss aus?
Unter- und Teilsysteme abgrenzen	Bereiche, die innerhalb des Projekts zusammenspielen, voneinander trennen.
Schnittstellen bestimmen	Wo werden Technologien gewandelt, codiert? Wo werden personelle Verantwortungen übergeben?
Analysieren von Unter- und Teilsystemen	Können getrennte Bereiche optimaler Zusammenspielen (z. B. neue und bestehende Internetdienste)?
Gemeinsamkeiten ermitteln	Welche Teile Ihres Auftrags haben Gemeinsamkeiten, die bei der Realisierung berücksichtigt werden können?

## 1.6 Zusammenfassung

In diesem Kapitel erhielten Sie einen Einblick in die Arbeit des **Projektmanagement** im Informatikbereich. Sie haben die sechs Phasen der Projektmethode **IPERKA** kennengelernt. Anhand der Projektvorgaben können mit **IPERKA** die meisten Projekte im Informatikbereich praxisgerecht durchgeführt werden.

Unterscheidung Projektmanagement vs. Projektarbeit: **Projektmanagement** wird durch den Projektleiter oder Projektmanager angewendet, um Strukturierung und Durchführung eines ganzen Projekts zu planen. Die **Projektarbeit** befasst sich ergänzend mit Teilproblemen einzelner Projektphasen.



### Repetitionsfragen

- 
- 1 Was bedeutet die Abkürzung IPERKA?
- 
- 6 Nennen Sie die Unterschiede zwischen Projektarbeit und Projektmanagement in eigenen Worten?
-



## 2 Wozu werden Internetserver eingesetzt?

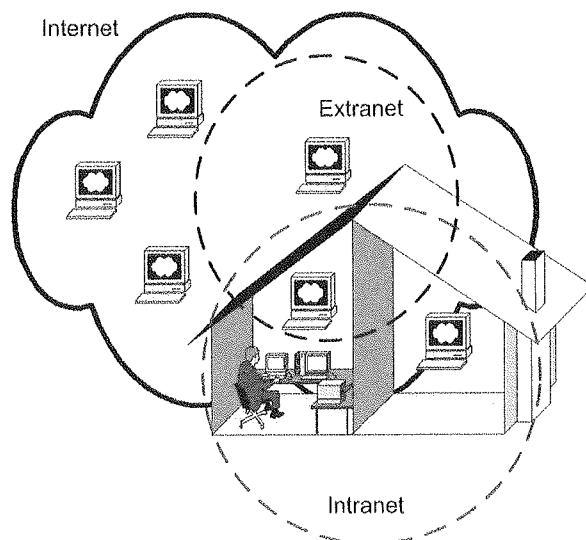
Im allgemeinen Sprachgebrauch herrscht oft Verwirrung bzw. Unwissen über den Unterschied zwischen «World Wide Web» und «Internet». Je nach Anwendung und Tätigkeit, die man auf dem weltweiten Netzwerk durchführt, nutzt man entweder das WWW oder das Internet. Das WWW (oder abgekürzt: Web) bezieht sich ausschliesslich auf denjenigen Teil des Internets, der mit dem http-Protokoll betrieben wird (oft sagt man auch «das sichtbare Internet ist das Web», mit «sichtbar» sind die HTML-Seiten gemeint). Alle anderen Anwendungen, die im Internet möglich sind, laufen eben auch über das Internet und nicht über das WWW. Das WWW ist somit lediglich ein Teil des Internet.

Im Rahmen des Moduls 239 werden Sie lernen, einen Internet-Server aufzusetzen, der mehr kann als «nur» Webseiten bereitstellen (sonst würde man von einem Webserver reden). In diesem Kapitel werden die Einsatzgebiete und verschiedene Typen des Internets aufgezeigt und die grundlegenden Anwendungen, die mit dem Internet möglich sind.

### 2.1 Einsatzgebiet

Nachdem in der Einleitung bereits der Unterschied zwischen WWW und Internet erläutert wurde, schauen wir die einzelnen «Typen» von Netzwerken an, die man als «I-Net» bezeichnen kann. Das «I» steht als Platzhalter für:

- Internet: Dies ist das bekannte, weltumspannende, öffentlich verfügbare Netzwerk, welches hauptsächlich mit der Protokollfamilie «TCP/IP» betrieben wird.
- Intranet: Das Intranet ist technisch genau gleich wie das Internet aufgebaut, bedient aber eine sogenannte «closed user group». Das Intranet ist somit ein kleines Internet mit einer definierten Grenze gegen aussen (Internet). Die «closed user group» können beispielsweise die Mitarbeiter einer Firma sein.
- Extranet: Das Extranet ist der Teil des Intranet, der über das Internet erreichbar ist. Oft möchte ein Unternehmen Inhalte aus dem Intranet auch Kunden, Lieferanten oder externen Mitarbeitern zur Verfügung stellen, aber nicht der ganzen Welt über das Internet zugänglich machen.



Im weiteren Verlauf dieses Lehrmittels wird einheitlich der Begriff «Internetserver» verwendet werden.

## 2.2 Anwendungen

Mit einem Internetserver sind verschiedenste Anwendungen möglich. Im Rahmen dieses Lehrmittels werden wir auf diejenigen eingehen, welche gemäss den Handlungszielen relevant sind.

Der grosse Vorteil des Internet ist die weltweite Erreichbarkeit. Seitens der Benutzer ist dazu lediglich ein Zugang zum Internet und ein Programm nötig, das den «Programmcode» in sichtbare Zeichen und Grafiken umwandelt: der Browser. Der Browser ist nichts anderes als eine «schlanke» Client-Applikation, das ganze Internet ist somit eine riesige Client-/Server-Umgebung.

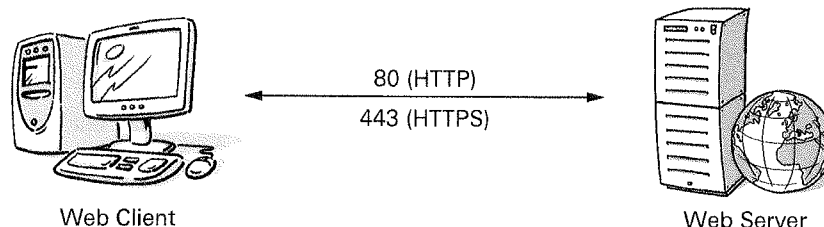
### Web-Anwendungen

Webbasiert, d. h. über das WWW, sind praktisch alle Anwendungen (Applikationen) möglich, die programmierbar sind. HTML bietet zwar keine unendlichen Möglichkeiten an, erlaubt aber die Implementation verschiedenster anderer Technologien wie JavaApplets, Flash, Scripts etc. Dadurch ist es auch möglich, dass ein Teil der Datenverarbeitung auf dem Client erfolgt. Dies passiert meist, um den Server zu entlasten und um gewisse einfachere Aufgaben, die sehr oft anfallen, am Ort des Problems durchzuführen (Beispiel: Plausibilisierung von Benutzer-Eingaben durch JavaScript im Browser, bevor die fehlerhaften Daten überhaupt zum Server übermittelt werden und so unnötig Bandbreite und Serverkapazität verbraucht wird). Der Browser macht grundsätzlich keine Datenverarbeitung (der fachliche Begriff lautet «der Browser führt keine Applikationslogik aus»), sondern «parsed» den HTML-Inhalt in lesbare Zeichen (vom englischen Wort «parsing»=Syntaxanalyse). Das Web stellt somit den grössten Teil der Anwendungen zur Verfügung, die im Internet genutzt werden.

Der WWW-Dienst benutzt standardmässig den Port 80 (http; oder Port 443, falls verschlüsselt mit https).

Anwendungen bzw. Webseiten werden grob in folgende Kategorien unterteilt:

- Statisch: Der Inhalt der Webseite ändert nicht (automatisch) und ist für alle Benutzer gleich. Beispiel: Informationsseite einer Firma mit Telefonnummern und E-Mail-Adressen.
- Dynamisch: Der Inhalt der Webseite ändert je nach Art der Anfrage. Der Inhalt wird dabei oft von einer Datenbank geholt und dynamisch aufbereitet. Das bedeutet, dass der Inhalt der Webseite für jede Anfrage individuell aufbereitet wird. (Beispiel: Je nachdem welche Spracheinstellung ein Benutzer im Browser hat und ob ein Cookie vorher abgespeichert wurde, wird der Benutzer persönlich in der jeweiligen Sprache und mit seinem akademischen Titel begrüsst.)
- Transaktions-orientiert: Eine transaktionsorientierte Webseite ist eine besondere Form der dynamischen Webseite. Der Benutzer hat die Möglichkeit, Transaktionen auszulösen. Eine Transaktion ist eine abgeschlossene Einheit von Tätigkeiten eines Geschäftsprozesses, beispielsweise ein Einkauf im E-Shop oder eine Überweisung von Geld mit E-Banking.



## Andere Anwendungen

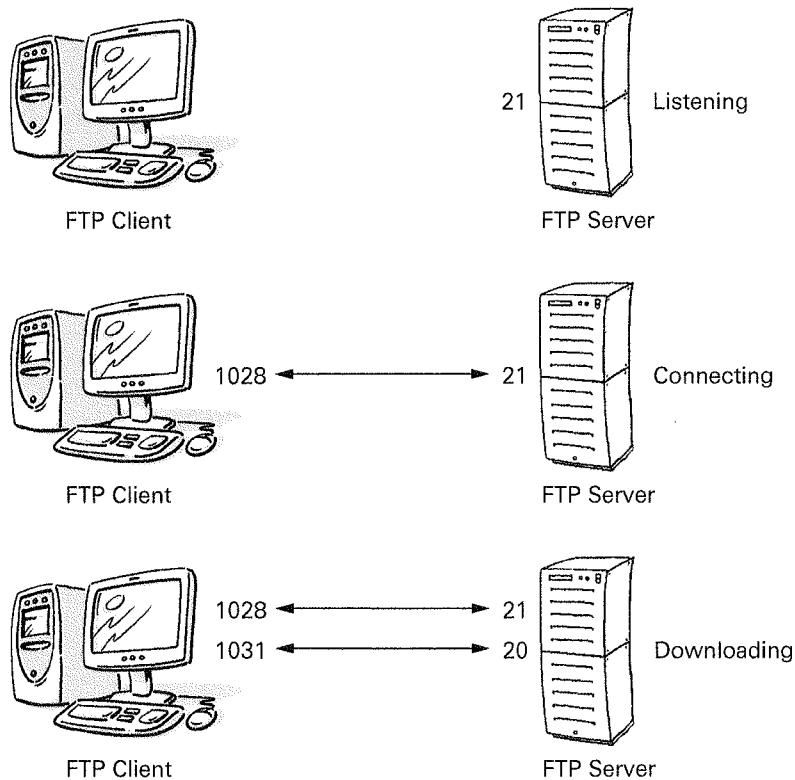
Andere Anwendungen, wie beispielsweise Filetransfers mit FTP-Protokoll, Chats über Internet Relay Chat (IRC) oder Netzwerk-Monitoring mit Simple Network Management Protocol (SNMP), erfolgen nicht über das WWW, sondern über das Internet mit den dafür nötigen Protokollen.

Von den vielen weiteren Anwendungsmöglichkeiten interessieren wir uns im Rahmen dieses Moduls neben dem WWW für die folgenden:

### Datenaustausch mit dem File Transfer Protocol (FTP)

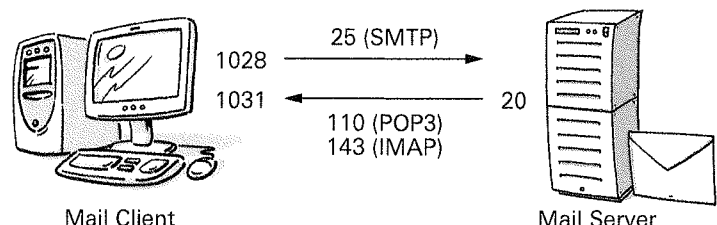
Fileserver als FTP-Server, um File up- und Downloads über das FTP-Protokoll zu ermöglichen. FTP wird über Port 20 und 21 betrieben, und kennt zwei Betriebs-Modi:

- **Active:** Beim Active Mode baut der Server von seinem Port 20, dem Data Port, eine Datenverbindung zu einem vom Client gewählten Endpunkt auf. Dieser Endpunkt ist typischerweise ein Port des Clients, der jenseits 1023 liegt, kann aber auch ein anderer Server sein, der seinerseits in den **Passive Mode** geschaltet wurde, also auf eine Verbindung wartet (sogenanntes FXP). Die Kommunikation mit Befehlen erfolgt auf dem Port 21. Man spricht auch von der Steuerung «Out of Band». Somit bleibt es möglich, dass während der Datenübertragung die Partner noch immer miteinander kommunizieren können.
- **Passive:** Beim Passive Mode baut der Client eine Datenverbindung zum vom Server gewünschten Port auf. Hier wird typischerweise von beiden Seiten ein Port jenseits 1023 benutzt. Diese Technik wird eingesetzt, wenn der Client für den Server nicht erreichbar ist. Dies ist beispielsweise der Fall, wenn der Client sich hinter einem Router befindet, der die Adresse des Clients mittels NAT umschreibt, oder wenn eine Firewall das Netzwerk des Client vor Zugriffen von aussen abschirmt.



### E-Mail-Anwendungen

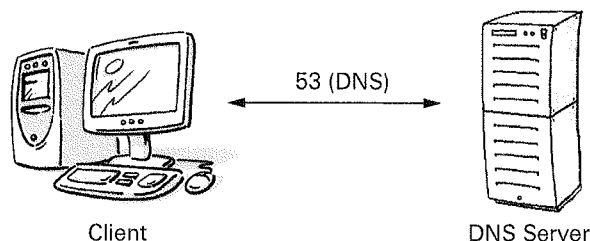
Im Wesentlichen das Senden von E-Mails mit dem Simple Mail Transfer Protocol (SMTP) und das Empfangen von Mails auf Client-Seite mittels Post Office Protocol (POP3) und Internet Message Access Protocol (IMAP) und das dazu nötige Postfach auf dem Server. E-Mail wird über Port 25 (SMTP) verschickt, und vom Client mittels POP3 auf Port 110 oder mit IMAP auf Port 143 abgeholt.



### Namensauflösung

Namesauflösung von Domain-Namen (z. B. [www.tages-anzeiger.ch](http://www.tages-anzeiger.ch)) mit dem Domain Name Service (DNS) Protokoll, indem der Internetserver auch als DNS-Server konfiguriert wird. DNS wird über Port 53 betrieben. Dabei wird primär zwischen drei Typen von DNS-Servern unterschieden:

- **Authoritative oder Primary:** Der Haupt-DNS Server für eine bestimmte Domain, d. h., hier wird der Eintrag mit allen Details verwaltet und bei Bedarf geändert. Wenn man also eine «wahre» Aussage möchte (d. h. die garantiert richtige IP-Adresse einer Domain), so muss der Authoritative-DNS angefragt werden, da dieser immer über die aktuellste Information verfügt.
- **Secondary:** Der Secondary-DNS Server ist der Stellvertreter des Primary-Server und «spiegelt» dessen Einträge. Je nach Einstellung werden die Einträge innerhalb von Stunden regelmässig aktualisiert, beispielsweise alle 3h wird der Eintrag mit dem Primary-Server abgeglichen.
- **Forwarding:** Der Forwarding-DNS Server ist eigentlich kein richtiger DNS-Server, sondern lediglich der «nächstbeste» Ansprechpartner für DNS-Queries (DNS-Anfragen) in einem Netzwerk. Dabei leitet er die Anfrage an einen «richtigen» DNS-Server (secondary oder primary) weiter, wo er die Information bekommt. Diese wird an den ursprünglichen «Requestor» (Anfragender) weitergegeben (forwarding). Dabei kann der Forwarding-DNS Server die Information zwischenspeichern, damit er nicht jedes Mal einen forward machen muss (sogenanntes DNS caching).



### Repetitionsfragen

11 Nennen Sie die Einsatzgebiete eines Internetservers?

16 Welche Anwendungen stellt ein Internetserver üblicherweise bereit?

## 3 Wie werden Internetdienste erbracht?

---

Bei der Erbringung der Internetdienste (damit ist die Gesamtheit aller Möglichkeiten des Internets gemeint) sind Server notwendig, die mit zweckmässiger Software und Hardware ausgerüstet sein müssen.

Die hier aufgeführten Anforderungen für einen Internetserver beziehen sich auf einen professionellen Server, der für mehrere Kunden und mehrere Dutzend gleichzeitige Benutzer ausgelegt ist. Solche Server sind vor allem bei einem Internet Service Provider im Einsatz. Für den Betrieb eines Internetserver für eine einzelne, kleinere KMU-Firma reicht es durchaus, die Werte für den professionellen Betrieb zu halbieren, womit der Preis eines eigenen Internet-Servers auch für eine kleinere Unternehmung erschwinglich wird.

### 3.1 Welche Systeme werden benötigt?

---

Je nach Budget und Anforderungen empfiehlt es sich in der Praxis, für die einzelnen Dienste jeweils separate (sogenannt «dedizierte») physische Server aufzusetzen. Gründe dafür sind:

- eine erhöhte Leistungsfähigkeit,
- eine verbesserte Sicherheit, die für jedes einzelne System separat konfiguriert werden kann,
- und ein kleineres Ausfallrisiko. Ein «single point of failure» kann vermieden werden, d. h., wenn ein physischer Rechner abstürzt, sind nicht alle Dienste (WWW, E-Mail, FTP etc.) davon betroffen.

Um die Verfügbarkeit noch weiter zu erhöhen, besteht die Möglichkeit, mehrere Server zu einer logischen Einheit zusammenzuschliessen: der Fachbegriff dafür lautet «Cluster». Dabei stehen zwei oder mehr Systeme miteinander in Verbindung, treten aber gegenüber dem Besucher als ein System auf (welches auch unter einer Domain bzw. IP-Adresse erreichbar ist). Ein sogenannter «Load-Balancer» verteilt dabei die Anfragen gleichmässig auf die dahinter liegenden physischen Rechner. Die Last wird so gleichmässig verteilt, und falls ein Rechner ausfällt, übernimmt der «Nachbar» einfach die anstehenden Anfragen und verarbeitet diese weiter. Zu diesem Zweck müssen die beiden Systeme synchronisiert werden; die Software der einzelnen Dienste muss ein solches «clustering» vorzugsweise unterstützen.

Im Rahmen dieses Moduls werden wir die verschiedenen Dienste auf einem einzelnen physischen Rechner aufsetzen und konfigurieren. Clustering und dedizierte Systeme werden in diesem Modul nicht aufgesetzt.

### 3.2 Die Auswahl der Server-Hardware

---

#### Auswahlkriterien

Die für einen Internet-Server benötigte Hardware richtet sich nach verschiedenen Faktoren:

- Dienste, die angeboten werden (WWW, E-Mail etc.)
- Anzahl der Benutzer, die gesamthaft auf den Server zugreifen können (sogenannte «named user») und Anzahl der Benutzer, die gleichzeitig auf den Server zugreifen (sogenannte «concurrent user» oder «concurrent sessions»)

Die «named user» sind alle berechtigten Benutzer auf einem Server; die «concurrent

user» sind alle Benutzer, die gleichzeitig zugreifen. Es kann beispielsweise 100 berechnete Benutzer auf einem Server geben, aber es werden maximal 20 gleichzeitig zugreifen.

- Menge an Daten (Volumen), die up- und gedownloadet werden (z. B. hat ein FTP-Server typischerweise eine grosse Datenmenge, die meist heruntergeladen wird; ein statischer Webserver wird ein mittelmässiges Volumen erzeugen und beim Mailserver kommt es stark auf das Benutzerverhalten und die Konfiguration an, insbesondere wenn grosse Anhänge (Attachments) erlaubt sind)
- Geschwindigkeit der Anbindung an das Netzwerk (diese hängt auch von den zu erwartenden Benutzermengen ab)
- Für die Auslastung nicht zu vergessen sind auch andere Systeme, die an den Internetserver angeschlossen sind und sich quasi «selbstständig» bedienen, also Daten abholen oder Daten zur Verfügung stellen (z. B. Datenbank-Server)
- Zukünftige Erwartungen an die oben genannten Faktoren und Investitionsschutz (d. h. man plant und budgetiert von vornherein ein eher grosszügiges und für die aktuellen Bedürfnisse überdimensioniertes System, das auch für zukünftige Anforderungen ausreicht). Dieser Faktor wird oft mit dem Fachbegriff «Skalierbarkeit» ausgedrückt.
- Zur Verfügung stehendes Budget für die Hardware

### Rechnerarchitektur und Peripherie

Grundsätzlich unterscheidet man zwischen der «Intel-Architektur», d. h. Servern die auf x86-Architektur aufbauen (Intel, AMD Prozessoren) und SUN-Systemen, die auch oft eingesetzt werden. Grössere Systeme (IBM AS/400 etc.) können zwar auch als Internet-Server betrieben werden, sind aber nicht primär dafür ausgelegt. Mac OS von Apple bietet auch die Möglichkeit, einen Internetserver zu betreiben.

Die verwendete Architektur hat zum Teil direkten Einfluss auf das Betriebssystem, das installiert wird:

- x86-Server werden mit Windows (Version für Server oder Workstation) oder mit Linux betrieben.
- SUN-Server werden mit SUN's eigenem Betriebssystem, Solaris, betrieben.

Für die in diesem Lehrmittel verwendeten Beispiele wird aus lizenzrechtlichen Gründen das freie Betriebssystem Linux (Distribution SuSE, bzw. openSUSE) auf x86-Hardware verwendet.

Ein Internetserver stellt weniger Anforderungen an die Peripheriegeräte als ein herkömmlicher PC/Workstation. So wird im Normalfall weder ein Drucker benötigt, noch sind verschiedene Anschlüsse wie USB, Firewire, Infrarot oder Ähnliches nötig. Spezialisierte Systeme, die sich für den Einbau in Serverschränke eignen (sogenannte «rack-mountable devices») werden in verschiedenen Höhen (gemessen in «height units, HU oder einfach U» oder «HE, Höheneinheiten») angeboten und haben eine Standard-Breite von 19 Zoll. Oft wird für solche 1HU-Einheiten die Bezeichnung «Pizzabox» verwendet, wegen der Ähnlichkeit mit einer Pizzaschachtel.

### Hardware-Komponenten

Die wichtigsten Hardware-Komponenten eines Internetserver sind folgende:

- RAM: Je mehr Benutzer vorhanden sind, desto mehr Arbeitsspeicher wird benötigt. Grundsätzlich sollte ein moderner Internetserver, unabhängig von der gewählten Hardware-Architektur, mit mindestens 1 GB RAM betrieben werden.
- Prozessor: Der Prozessor spielt bei einem Webserver eine untergeordnete Rolle. Je mehr Applikationen auf einem Server betrieben werden, desto schneller sollte der Pro-



zessor sein. Ein Internetserver wird aber typischerweise keine hochkomplexen Berechnungen durchführen müssen (das ist die Aufgabe des Applikationsservers). Es wird jedoch empfohlen, einen Server-Prozessor einzusetzen (z. B. Intel's Xeon) und weniger einen Prozessor für mobile Geräte (Intel's Celeron).

Eine Ausnahme gibt es, wenn der Internetserver die Verschlüsselung von Daten vornimmt (z. B. für https/SSL oder als E-Mail-Verschlüsselungs-Gateway). Verschlüsselungsoperationen sind sehr arbeitsintensiv und benötigen deshalb einen entsprechend schnellen Prozessor.

- **Harddisk:** Die Harddisk sollte so ausgelegt sein, dass sie mindestens doppelt so gross ist wie die zu verwaltende Datenmenge inklusiv Betriebssystem. Besonders berücksichtigt werden muss die anfallende Menge an Log-Dateien, um die Harddisk nicht schon nach wenigen Tagen volllaufen zu lassen. Es empfiehlt sich daher, Log-Dateien auf einem separaten Server zu speichern (sogenannter «Logserver») und nicht auf dem Internetserver. Besonders grosse Platzansprüche haben insbesondere Fileserver (FTP) und E-Mailserver (je nach Anzahl E-Mail-Benutzer und erlaubte Attachments). Wichtig ist bei der Auswahl der Harddisk der Typ. In einem Server sollten grundsätzlich SCSI-Harddisks eingebaut werden, da diese eine höhere Beständigkeit haben als anderen Typen (aber auch teurer sind).
- **Netzwerk:** Ein Internetserver verfügt über einen oder auch zwei bis mehrere Netzwerkanschlüsse. Dabei wird die Bezeichnung «single homed» (1 Netzwerkanschluss), «dual homed» (2 Netzwerkanschlüsse) oder «multi-homed» (mehrere Netzwerkanschlüsse) verwendet. Für jeden einzelnen (physischen) Netzwerkanschluss besteht die Möglichkeit, eine separate IP-Adresse zu konfigurieren. So kann der Internetserver gleichzeitig in verschiedenen Netzwerken betrieben werden, um so aus Sicherheitsgründen verschiedene Dienste über verschiedene Adressen laufen zu lassen. Typischerweise werden die Benutzerdienste auf dem «externen» Interface betrieben, während die Administration des Servers nur über das «interne» Interface möglich ist. Beim Netzwerk-Interface ist darauf zu achten, dass es für die benötigte Geschwindigkeit ausgelegt ist. Es werden heutzutage folgende drei Typen von Netzwerk-Interfaces unterschieden: 10/100 Mbit, 10/100/1000 Mbit (sogenannte Gigabit-Interfaces) und Kupfer und Gigabit-Lichtleiter (Glasfaser) Interface. Für den normalen Einsatz reicht heutzutage ein 10/100 Mbit Interface grundsätzlich aus, da ein Internet-Server in einer KMU-Firma selten mit einer Gigabit-Leitung ans Internet angebunden wird.

### 3.3 Die Auswahl der Server-Software

---

#### Begriffserklärung

In diesem Zusammenhang muss auf die Doppeldeutigkeit des Wortes «Server» hingewiesen werden: Sowohl der physische Rechner als auch entsprechende Software wird in der Informatikwelt als «Server» bezeichnet. Der Grund ist die Bedeutung des Wortes in der englischen Sprache: «to serve» bedeutet «zu dienen», die Hauptaufgabe des Servers: er dient dem Client (Kunden), der Anfragen an den Server schickt. So ist sowohl die Hardware ein (physischer) Server, wie auch die Software, die die Anfragen des Client beantwortet, ein (software)-Server. Das gleiche gilt übrigens auch für den Client: Der physische Rechner und die Software, die eine Anfrage stellt, werden als (physischer oder Software)-Client bezeichnet. Beispiel: Der Browser (Software-Client), der auf einem physischen Client (Notebook) installiert ist, stellt dem (physischen) Server beim Internet Service Provider eine Anfrage (zeige mir die Webseite der Domain [www.symlink.ch](http://www.symlink.ch)). Diese Anfrage wird vom Software-Webserver beantwortet.

## Server-Betriebssysteme

Für den Betrieb des Internetserver ist ein Betriebssystem notwendig, das zur Software gezählt wird. Je nach Betriebssystem sind gewisse Software-Pakete für den Betrieb bereits inbegriffen.

Es ist empfehlenswert ein «serverbasiertes» Betriebssystem einzusetzen. Damit ist z. B. Windows 2003 Server oder eine Server-Edition eines Linux-Betriebssystems gemeint. Mac OS verfügt ebenfalls über eine Server-Version. Andere (unix-artige) Betriebssysteme wie beispielsweise SUN's Solaris sind von Grund auf als Serversystem ausgelegt.

Serverbasierte Betriebssysteme haben den Vorteil, dass sie für einen ständigen Betrieb des Systems ausgelegt sind. Überwachungs- und Monitoringmöglichkeiten sind vorhanden, oder entsprechende Schnittstellen sind vorgesehen. Allgemein ist die Stabilität des Betriebssystems für den Dauerbetrieb optimiert. Dieser Vorteil wird durch Verzicht auf Fähigkeiten erkauft, die typischerweise bei Desktop-Betriebssystemen zu finden sind: Audio- und Multimedia-Unterstützung, Treiber für eine Vielfalt von (desktop-typischen) Peripheriegeräten wie z. B. Drucker, hochauflösende Bildschirme, Multimedia-Geräte etc. Die Serversysteme sind ausserdem oft «gehärtet», d. h. grundsätzlich ist nur ein Minimum an Diensten automatisch eingeschaltet, um Sicherheitsrisiken möglichst auszuschliessen.

## Desktop-Betriebssysteme

Dies bedeutet nicht, dass heutige moderne Desktop-Betriebssysteme (Windows XP Professional, Linux Workstation Edition) nicht fähig wären, als Internetserver betrieben zu werden. Nötigenfalls müssen aber Konfigurationen angepasst und Dienste deaktiviert werden, damit die nötige Leistungsfähigkeit und Sicherheit erreicht werden kann.

## Übungs-Betriebssysteme

Für die Beispiele des Lehrmittels wird das Betriebssystem SuSE Linux 10 OSS (open source system) eingesetzt. Es kann von der Website <http://www.opensuse.org> heruntergeladen und installiert werden.

Dieses (desktop-basierte) Betriebssystem enthält ausschliesslich open-source-Komponenten und ist deshalb frei von lizenzrechtlichen Einschränkungen. Andere Betriebssysteme müssen sowohl in der Desktop- wie auch in der Server-Variante lizenziert werden. Eine Ausnahme bildet SUN's Solaris, welches inzwischen auch als open-source-Edition verfügbar ist. Da Solaris aber spezielle Anforderungen an Hardware, Installations- und Betriebsvorgehen erfordert, wurde darauf verzichtet.

## Software für Anwendungen

Für die Realisierung der verschiedenen Dienste des Internetserver werden die folgende Softwarekomponenten benötigt:

- Webserver: Bekannte Webserver sind Microsoft's Internet Information Server (IIS) und der open-source Webserver «Apache»
- Mailserver: Der Mailserver setzt sich einerseits aus dem sogenannten «Mail Transfer Agent» (MTA) zusammen, der nichts anderes macht als die E-Mails zu versenden. Der MTA wird oft auch als «SMTP-Server» bezeichnet, in Anlehnung an das für den Mailversand benötigte Protokoll SMTP. Neben dem MTA wird für den sinnvollen Betrieb eines Mailserver auch ein Postfach benötigt, wo die E-Mails für die Benutzer gespeichert werden. Das Postfach ist der eigentliche «Briefkasten» und ist (technisch gesehen) unabhängig vom MTA. Der MTA ist sozusagen der «Pöster», der die Briefe zu-

stellt (entweder an einen anderen, entfernten Mailserver, oder an den «benachbarten» Briefkasten).

- **FTP-Server:** Der File-Transfer-Server ist technisch gesehen nichts anderes als ein Software-Aufsatz auf das Dateisystem (file system), das die Übertragung der Dateien über das FTP-Protokoll ermöglicht. Neben der Kommunikation mit einem FTP-Client (der nötig ist, um eine Datei über das FTP-Protokoll zu senden oder empfangen) verwaltet der FTP-Server oft auch die Benutzerberechtigungen, d. h. welcher Benutzer auf welchem Verzeichnis lesen oder schreiben darf.
- **DNS-Server:** Der DNS-Server ist eine Software, die auf dem Port 53 auf Anfragen wartet und diese (sofern der Client der die Anfrage stellt, berechtigt ist) mit den Informationen beantwortet, die der DNS-Server im Konfigurationsfile (das sogenannte «Zonen-File») gespeichert hat. Der DNS-Server ist so gesehen eine relativ einfache Software, die im «Telefonbuch DNS» nach der IP-Adresse sucht, die für einen Domainnamen angefragt wurde.

### Die Bearbeitung der Anfrage der verschiedenen Anwendungen

Sie erfolgt mit einem «daemon». Damit wird ein Server-Prozess auf dem Betriebssystem gemeint, der auf bestimmte Anfragen/Ports horcht und diese an den entsprechenden Serverprozess weiterleitet (oder selbst bearbeitet und beantwortet). Der httpd (http-daemon) horcht somit auf dem Port 80 des Webservers auf Anfragen von Clients (Browser). Sobald eine Anfrage eintrifft, erfolgt ein TCP-handshake und die Verbindung wird aufgebaut. Dabei erfolgt der Verbindungsaufbau vom Client aus über einen Port >1023, der http-daemon horcht auf Port 80. Die eigentliche Kommunikation wird daraufhin vom daemon auf einen Port >1023 gelegt, damit der Port 80 für weitere Anfragen frei bleibt. Server und Client kommunizieren nun beide mit Ports >1023, die innerhalb eines definierten Bereichs frei gewählt werden (z. B. Client sendet Anfragen von Port 14411, Server nimmt diese entgegen auf Port 28330). Nur die ursprüngliche Verbindungsanfrage erfolgte auf Port 80, danach einigen sich daemon und Client auf einen nicht-belegten Port >1023.

### Weitere Softwarekomponenten

Für den reinen Betrieb des Internetserver wird neben der oben aufgeführten Software der Anwendungen grundsätzlich keine weitere Software benötigt. Oft wird für die Überwachung/Monitoring des Servers weitere Software installiert, die beispielsweise die Fernwartung erlaubt oder bestimmte Meldungen an ein remote-System schickt. Zudem kann der Webserver so konfiguriert werden, dass er auf verschiedenen Ports auf Verbindungsanfragen horcht, sodass z. B. auf Port 80 normale Client-Anfragen an den Webserver beantwortet werden und über Port 8080 ein Administrations-Interface für den Entwickler der Website zur Verfügung gestellt wird.

## 3.4 Internetprotokolle

---

Im vorangegangenen Kapitel wurden bereits einige der Internetprotokolle erwähnt. Nachfolgend sind die wichtigsten Grundlagen dazu zusammengefasst. Das reibungslose Zusammenspiel der verschiedenen Protokolle ist durch die eingesetzte Software gewährleistet. Für das Verständnis ist es wichtig, den stufenweisen Aufbau der beteiligten Komponenten zu kennen.

Die Kommunikationsdienste sind auf 7 Stufen (Layers) angeordnet. Diese sind als OSI-Schichtenmodell bekannt geworden. Die TCP/IP-Protokollfamilie hat aber nur 4 Schichten, die dem OSI-Schichtenmodell zugeordnet werden können. Die folgenden detaillierten Erklärungen zu den Schichten und den zugeordneten Diensten sind dem Wikipedia entnommen (<http://de.wikipedia.org/wiki/OSI-Modell>).

OSI-Schicht Layer	Schichtname Englisch	Schichtname Deutsch	Einordnung	TCRIP-Schicht	Protokollbeispiele
7	Application	Anwendung	Anwendungsorientiert	Application	HTTP FTP HTTPS NCP
6	Presentation	Darstellung			
5	Session	Sitzung			
4	Transport	Transport	Transportorientiert	Transport oder Host to Host	TCP; UDP SPX
3	Network	Vermittlung			
2	Data Link	Sicherung			
1	Physical	Bitübertragung		Link oder Network	Ethernet Token Ring FDDI ARC NET

- **Layer 1 (Physical Layer, Bitübertragung):**

- Koaxial, Kupfer- oder Glasfaser-Verkabelungen mit entsprechenden Steckern (RJ45, Koax etc.)
- Hardware: Auf dieser Schicht werden Hubs und Bridges eingesetzt.
- Vergleichbar mit der Strasse in der realen Welt

- **Layer 2 (Data Link Layer, Sicherung):**

- Für TCP/IP-basierte Kommunikation kommt auf Layer 2 vor allem das Ethernet-Protokoll (802.16) sowie je nach Einsatzzweck Wireless LAN (802.11) zum Einsatz. Um Netzwerke in logische Einheiten zu unterteilen, werden sogenannte «Virtual LANs» gebaut, mit der Abkürzung «VLAN» (nicht zu verwechseln mit WLAN für Wireless LAN).
- Hardware: Auf dieser Schicht werden Switches eingesetzt.
- Vergleichbar mit den Strassennummern in der realen Welt (MAC-Adressen als physische, eindeutige Adresse der Netzwerkkarte)

- **Layer 3 (Network Layer, Vermittlung):**

- IP Internet Protocol (die IP-Adresse)
- Hardware: Auf dieser Schicht werden Router und Paketfilter eingesetzt.
- Vergleichbar mit den Namen an den Haustüren (die logische Adresse, d. h. diese kann ändern und ist nicht fest zugeteilt wie die physische MAC-Adresse)

- **Layer 4 (Transport Layer, Transport):**

- ICMP Internet Control Message Protocol (time to live, traceroute, PING)
- TCP Transport Control Protocol (stateful, d. h. eine Verbindung wird aufgebaut mit Informationen, ob ein Paket angekommen ist oder nicht; vergleichbar mit einem eingeschriebenen Brief)
- UDP User Datagram Protocol (stateless, ohne Verbindungsinformation; keine Informationen, ob ein Paket ankommt; vergleichbar mit einer Postkarte)
- Hardware: Auf dieser Schicht werden Firewalls eingesetzt.

- **Layer 5 (Session Layer, Sitzung):**

- SSL Secure Session Layer (ursprünglich 1995 von Netscape entwickelt, um eine Verschlüsselung primär für http zu ermöglichen; SSL ist kein offizieller Standard)
- TLS Transport Layer Security (die offizielle Nachfolge von SSL, von der Internet Task Force verabschiedeter Standard)

- Durch die Positionierung von SSL und TLS auf Layer 5 ist es möglich, die darüberliegenden Protokolle damit abzusichern (dadurch nicht nur http, welches damit zu https wird, sondern auch andere Protokolle wie z. B. POP3 etc. je nach Eignung)
- Hardware: keine besondere, evtl. SSL/TLS-Accelerator (spezieller Proxy, der die Ver- und Entschlüsselung der Daten vornimmt um andere Geräte nicht zu belasten, insbesondere eingesetzt bei stark besuchten Webseiten mit SSL/TLS-Verschlüsselung)
- **Layer 6 (Presentation Layer, Darstellung):**
  - Auf der Präsentationsschicht wird geregelt, in welcher Form die Daten auf der Applikationsschicht (Layer 7) ausgetauscht werden; also z. B. um welchen Datentyp es sich bei einem E-Mail-Attachment handelt (Text, binäres Objekt, Bild, PDF-File etc).
  - Hardware: keine besondere
- **Layer 7 (Application Layer, Anwendung):**
  - **http** Hypertext Transfer Protocol (das Fundament für das World Wide Web, Besonderheit: Hyperlink ermöglicht es, auf andere http-Adressen zu springen), Standardport: 80 (TCP)
  - **https** (durch SSL oder TLS abgesicherte Variante von http), Standardport: 443 (TCP)
  - **ftp** File Transfer Protocol (wird eingesetzt, um Dateien up- oder downloaden), Standardport: 20 (data port bei «active FTP») und 21 (control port bei «active FTP», Standardport für «passive FTP») (TCP)
  - **dns** Domain Name Service (zur Auflösung von Domainnamen in IP-Adressen und umgekehrt), Standardport: 53 (UDP oder TCP; immer TCP für Zonentransfers)
  - **smtp** Simple Mail Transfer Protocol (Dienst, um Mails im Internet zu verschicken und empfangen), Standardport: 25 (TCP)
  - **pop3** Post Office Protocol v3 (Dienst, um Mails von einem Postfach mittels eines E-Mail-Client abzuholen; es gibt auch eine sichere Variante: POP3/S mit SSL/TLS), Standardport: 110, POP3/S: 995 (TCP)
  - **imap4** Internet Message Access Protocol (alternativer Dienst zu POP3, der es erlaubt, das Postfach mittels Befehlen zu verwalten, ohne die Mails downloaden zu müssen; benötigt einen entsprechenden Mailclient, der IMAP beherrscht), Standardport: 143, IMAP mit SSL/TLS: 993
  - Hardware: Auf Schicht 7 kommen Proxies oder sogenannte Application Level Gateways zum Einsatz und natürlich die entsprechenden (physischen) Server wie Webserver, Mailserver, Fileserver etc.

### 3.5 Technischer Aufbau: die Architektur

---

Für den Aufbau eines Internetserver stehen verschiedene Architekturen zur Verfügung, die nachfolgend vorgestellt werden:

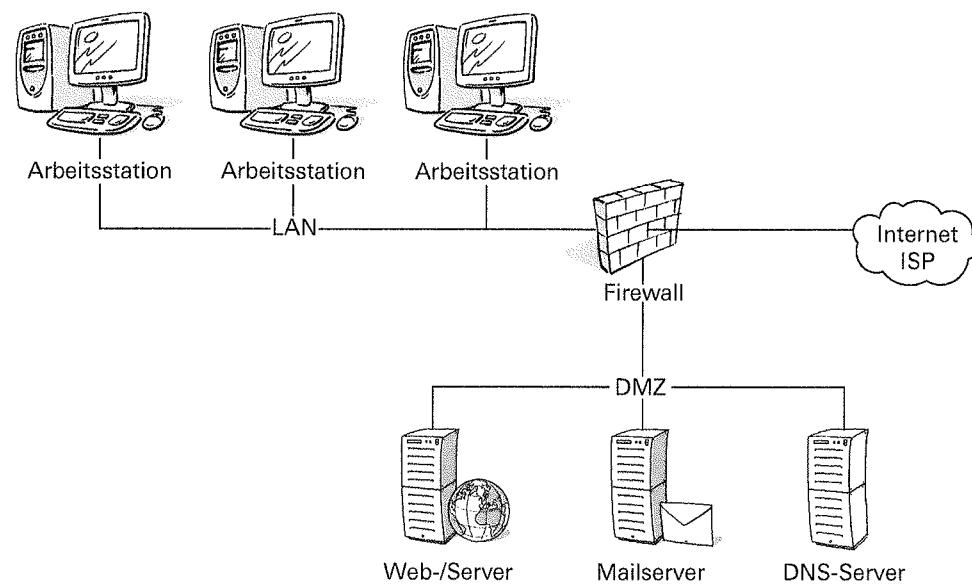
#### Internetserver in einer demilitarisierte Zone (DMZ)

Mit dieser Zone ist ein geschützter Netzwerkteil gemeint. Dort werden exponierte Systeme (Webserver, Mailserver, DNS Server) aufgebaut und betrieben. Dabei handelt es sich um ein vom restlichen LAN getrenntes Netzwerk mit separater IP-Range bzw. eigenem Subnetz. Die Abtrennung erfolgt durch eine Firewall bzw. Paketfilter. Grundsätzlich erfolgt die Konfiguration wie folgt:

- Von aussen (Internet) in die DMZ: erlaubt ist Web-, Mail- und wenn nötig DNS-Verkehr; weitere Dienste nur soweit nötig
- Von aussen (Internet) in das LAN: keine Verbindungen erlaubt

- Vom LAN nach draussen (Internet): Verbindungen erlaubt (evtl. Restriktion auf Webverkehr, port 80 und 443)
- Vom LAN zur DMZ: je nach Bedarf (z. B. wenn das Intranet ebenfalls auf dem DMZ-Webserver läuft dann Webverkehr erlauben), zudem müssen die E-Mails abgeholt werden können. Oft befindet sich aber der Server mit dem Postfach im LAN und in der DMZ ist nur der SMTP-Server aufgesetzt, der die Mails nach draussen verschickt und entgegennimmt und an das Postfach weiterleitet.

Die demilitarisierte Zone ist die üblichste Architektur für den Betrieb eines Internetservers.



### Internetserver beim Internet Service Provider (ISP)

Hosting: Der Internetserver wird nicht vom Unternehmen selber, sondern von einem Provider betrieben. Entsprechend liegt der Internetserver nicht im unternehmenseigenen Netzwerk. Typischerweise sind Dienste wie SMTP und POP3 eingeschaltet (für berechtigte Benutzer) und Webverkehr natürlich auch (da keine Unterscheidung gemacht wird ob interner oder externer Benutzer, da aus Sicht des Providers alle Benutzer von aussen kommen).

### Internetserver im LAN

Internetserver im LAN betreiben: Falls keine DMZ möglich ist, kann der Internetserver auch innerhalb des LAN betrieben werden. Dabei handelt es sich aber um eine «gefährliche» Architektur, da ein Angreifer dadurch zumindest auf den freigegebenen Diensten einen Zugriff auf einen Server im LAN hat (den Internetserver). Durch einen Bug kann er so Zugriff aufs interne Netzwerk erlangen, was bei der DMZ nur bedingt möglich ist.

Beim internen Betrieb müssen die entsprechenden Dienste auf der Firewall freigeschaltet werden. Am besten werden die Pakete NUR an den Internetserver weitergeleitet. Oft ist auch NAT (Network Adress Translation) im Einsatz, weshalb ein Port-Forwarding nötig ist, da der Internetserver durch das NAT on aussen nicht sichtbar ist.



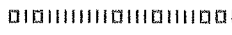
### 3.6 Anbieter und Abnehmer

Ein Internetserver kann in Eigenregie betrieben werden (inhouse), womit sämtliche Administrationsaufgaben vom Administrator durchgeführt werden müssen. Eine andere Möglichkeit besteht darin, den Betrieb des Internetserver an einen Internet Service Provider (ISP) auszulagern. Dabei werden drei hauptsächliche Betriebsarten unterschieden:

- **Hosting:** Beim Hosting wird beim ISP ein sogenannter virtueller Webserver für eine monatliche oder jährliche Gebühr gemietet. Der ISP richtet dafür auf einem physischen Server einen Webserver ein, der für mehrere Domains (und so für verschiedene Kunden) konfiguriert ist. Der Kunde bekommt in diesem Angebot einen definierten Speicherplatz und die Möglichkeit, via FTP oder z. T. auch WebDAV auf das für ihn definierte Verzeichnis zuzugreifen und den Webauftritt zu gestalten. Oft sind auch Datenbanken, Scripts und Auswertungsdienstleistungen im Angebot enthalten sowie die Möglichkeit, ein Postfach mit mehreren Adressen unter der reservierten Domain einzurichten. Die Administrationsmöglichkeiten beschränken sich im Normalfall auf den Zugriff auf das Dateisystem und einer webbasierten Konfigurationsoberfläche. Dabei ist es wichtig, dass es sich physisch um einen einzigen Rechner handelt. Dementsprechend besteht bis auf Layer 4 keine wirkliche Trennung zwischen den einzelnen Domains, d. h. alle Domains sind unter der gleichen IP-Adresse erreichbar (und alle offenen Ports sind für alle Domains geöffnet). Ein Ausfall der Harddisk oder des Netzteils betrifft so alle Domains, die auf diesem Rechner verwaltet werden; gleiches gilt für eine Fehlkonfiguration auf der Netzwerk- oder Benutzerverwaltungsebene und natürlich im Web- oder Mailserver (Layer 7) selbst.
- **Root-Server (Leasing, Renting):** Beim sogenannte root-Server stellt der Provider einen Server zur Verfügung, der voll im Zugriff des Kunden liegt (root = Administrator). Der Provider führt evtl. zusätzliche Dienstleistungen aus (reboot, Back-up), da ein physischer Reboot aus Distanz nicht möglich ist. Für das aufsetzen und pflegen des Servers ist der Kunde selber verantwortlich. Je nach Bedarf kann auch eine Firewall dazu gemietet werden, die ebenfalls selber oder durch den Provider gepflegt wird.
- **Housing/Co-location:** Beim sogenannten Housing stellt der Provider lediglich die Infrastruktur des Rechenzentrums zur Verfügung (Strom, Klima, Platz, Internet-Zugang, Routing). Der Server wird vom Kunden gebracht und an das Netz angeschlossen und völlig unabhängig vom Provider betrieben. Oft stellt der Provider für die Kunden eine Möglichkeit des (physischen) Reset/Reboot zur Verfügung, welche von Remote (d. h. aus Entfernung) genutzt werden kann, da ansonsten für den Reboot der Kunde ins Rechenzentrum müsste.

### Repetitionsfragen

20	Welche Systeme werden für einen Internetserver benötigt?
24	Welches sind die zwei Haupt-Einflussfaktoren für die Dimensionierung der Hardware?
2	In welche zwei Typen lässt sich die für einen Internetserver eingesetzte Software einteilen?
7	Welches ist das grundlegende Protokoll, das für einen Internetserver verwendet wird?
12	Was ist der Unterschied zwischen Hosting und Housing?



**28**

Teil A Grundlagen (Informationen beschaffen)  
**3** Wie werden Internetdienste erbracht?





## Einleitung, Lernziele und Schlüsselbegriffe

### Einleitung

Für die Konzeption und Dimensionierung des Internetservers muss zuerst die IST-Situation analysiert werden. Insbesondere sind die benötigten Dienste für den Internetserver festzulegen. Danach folgt eine Beschreibung der angebotenen bzw. vom Auftraggeber geforderten Dienstleistungen, die der Internetserver anbieten soll. Darunter versteht man beispielsweise Informationen (statisch), Produktkataloge und Preislisten, Bestellungen bzw. E-Shop-Systeme für den Einkauf via Internet. Anschliessend wird der SOLL-Zustand festgelegt. Dabei kann es nötig sein, bestehende Applikationen für den Einsatz im Internet «fit» zu machen, d. h. möglicherweise sind Schnittstellen nötig. Allgemeine Sicherheitsrichtlinien, insbesondere Anforderungen bezüglich Datenschutz, werden ebenfalls dargestellt. Eine wichtige Anforderung beim Internetserver ist die Verfügbarkeit der Dienste (da der Internetserver ständig im Internet verfügbar ist) sowie das Datenvolumen, welches zwischen Benutzer und Internetserver übertragen wird. Die Benutzerprofile und die externen Zugriffe sind ebenfalls Bestandteil der Kapitel im Teil B. Nachdem alle Anforderungen erhoben wurden, wird die Lösung (Hard- und Software, Konfiguration) entworfen und dokumentiert. Dabei werden auch Namenskonventionen und Standardeinstellungen angesprochen. Die Vorbereitung der System- und Sicherheitstests (Testfälle, Dokumentation des Testablaufs) sind ebenfalls Bestandteil der Planung.

### Lernziele und Lernschritte

Lernziele	Lernschritte
<input type="checkbox"/> Kann die bestehende Serverinstallation mit dem Anforderungsprofil vergleichen und begründete Verbesserungsvorschläge machen.	<ul style="list-style-type: none"> <li>• Wie ist eine bestehende Serverinstallation konfiguriert?</li> <li>• Entspricht die aktuelle Konfiguration den Anforderungen des Auftraggebers?</li> <li>• Welche Anpassungen müssen noch vorgenommen werden?</li> </ul>
<input type="checkbox"/> Kennt die erforderlichen Arbeitsschritte und Softwarekomponenten zur Installation der verlangten Dienste.	<ul style="list-style-type: none"> <li>• In welcher Reihenfolge werden die Softwarekomponenten installiert?</li> <li>• Welche zusätzlichen Arbeitsschritte sind für die Installation notwendig?</li> </ul>
<input type="checkbox"/> Erkennt die grundsätzlichen Sicherheitsrisiken und kann deren Relevanz bezüglich dem Anforderungsprofil einschätzen. (Sicherheitskonzept)	<ul style="list-style-type: none"> <li>• Welche Sicherheitsrisiken bestehen für einen Internetserver?</li> <li>• Welche Sicherheitsanforderungen lassen sich aus den Anforderungen des Auftraggebers ableiten (Vertraulichkeit, Verfügbarkeit, Integrität)?</li> </ul>
<input type="checkbox"/> Schlägt geeignete Sicherheitsmassnahmen vor.	<ul style="list-style-type: none"> <li>• Welche Sicherheitsmassnahmen müssen für den Internetserver umgesetzt werden (bezüglich Verfügbarkeit, Integrität, Vertraulichkeit)?</li> <li>• Welche zusätzlichen Massnahmen helfen, die Sicherheit zu erhöhen (organisatorische Massnahmen)?</li> </ul>

### Schlüsselbegriffe

Replikation, Transaktion, Applikation, Web-enabled, Uptime, Downtime, Traffic, Download, Upload, Benutzerprofil, CRUD, ACL

## 4 Ist-Situation analysieren

«Info-Sales» erteilt Ihnen den Auftrag im bestehenden Netzwerk einen Internetserver für ca. 500 Benutzer zu installieren. Das Projekt soll durch Sie geplant und ausgeführt werden.

Stellen Sie zu diesem Zweck einen eigens dafür verwendeten Webserver zusammen. Bis der Server geliefert wird, verwenden sie einen bestehenden PC, um die Installation in allen Phasen zu planen und durchzuführen.

Für die Realisierung werden folgende Serverdienste benötigt:

- http
- dns
- ftp
- mail (smtp und pop3)

Als Beispiel werden typische Eckdaten eines Internet-Servers sowie eines dafür geeigneten Test-Servers (handelsüblicher PC, der als Testserver konfiguriert wird) beschrieben. Je nach Anwendungen und Benutzerverhalten können diese Daten natürlich ändern. Bei grafik-intensiven Tätigkeiten (Aufbau von Charts) oder Berechnungen und Operationen (Zugriffe auf Datenbank, Berechnung von Angeboten) wird oft ein Vielfaches der genannten Angaben benötigt.

### [4-1] Eckdaten des neuen Servers

Maximale Anzahl Benutzer	500
Arbeitsspeicher pro Applikation und Benutzer	64 MB
Anzahl gleichzeitiger Benutzer	max. 50
Arbeitsspeicher pro Serverdienst	16 MB
Arbeitsspeicher für Betriebssystem	1024 MB
Harddiskspeicher total (inkl SWAP)	80 GB

### [4-2] Eckdaten für den Testserver (PC der als Server eingesetzt wird)

Maximale Anzahl Benutzer	5
Arbeitsspeicher pro Applikation und Benutzer	64 MB
Anzahl gleichzeitiger Benutzer	max. 1
Arbeitsspeicher pro Serverdienst	16 MB
Arbeitsspeicher für Betriebssystem	128 MB
Harddiskspeicher total (inkl SWAP)	min. 12 GB

### 4.1 Welche Dienstleistungen werden angeboten?

Je nach Angebot auf dem Internetserver werden verschiedene Dienste angeboten. Grundsätzlich stellen sich folgende Ausgangslagen:

#### Web (port 80, 443):

- Reiner Webauftritt (statisch) mit Informationen für die Kunden (Produkte, Preislisten etc.), evtl. mit Download-Möglichkeit (Wird diese durch FTP realisiert?)
- Webauftritt mit dynamischen Inhalten (Anbindung an eine Datenbank/CRM ist nötig)
- Transaktionsorientierter Webserver (e-shop, e-business), d. h., es werden zusätzlich zu den Informationen auch Waren (Bücher, CD, DVD etc) verkauft. Dabei stellt sich die

Frage, wie der Server in das Produktionssystem (das sogenannte Backend mit der produktiven Datenbank) angebunden wird.

- Replikation der produktiven Datenbank: Die produktive Datenbank wird nicht direkt an den Webserver angebunden, sondern regelmässig kopiert. Bestellungen erfolgen dadurch ebenfalls nicht direkt in der Datenbank, sondern über eine Schnittstelle (z. T. auch als E-Mail-Formular etc).
- Direkte Anbindung an das Backend: Der Webserver wird direkt mit dem Backend verknüpft. Dadurch stehen immer die aktuellen Daten zur Verfügung, das Risiko eines Einbruchs auf das produktive System (durch einen externen Angreifer) ist aber dadurch stark erhöht. Entsprechend sollten zusätzliche Schutzmassnahmen getroffen werden.

#### **Mailserver:**

- Mailserver für ein Unternehmen oder eine sogenannte «closed user group» (geschlossene Benutzergruppe), d. h., Mails werden typischerweise von einem geschützten Netzwerk (LAN) zum Internet verschickt und von dort empfangen. Vom Internet aus können aber keine Mails mithilfe des Mailserver ins Internet verschickt werden (sogenanntes «relaying»). Falls dies trotzdem (durch Fehlkonfiguration) möglich sein sollte, wird ein solcher Mailserver sehr rasch zum Versenden von unerwünschten Werbebotschaften (Spam) missbraucht.
- Öffentliche Mailedienstleistungen: Gegen Bezahlung oder auch kostenfrei wird die Möglichkeit angeboten, ein Postfach zu eröffnen und E-Mails zu empfangen und versenden. Dabei kann sich «jeder» für ein Postfach anmelden. Die Konfiguration des jeweiligen Benutzers und des Postfachs geschieht automatisch (Scripts).
- Unabhängig von den obengenannten Betriebsarten muss auch entschieden werden, ob direkt beim Mailempfang auf Viren geprüft werden soll (d. h. auf dem Mailserver und evtl. Auch noch auf dem E-Mail-Client / Empfänger-PC). Zusätzlich zur Virenprüfung werden ankommende E-Mails auch bezüglich ihrem Inhalt und ihrer Herkunft analysiert, um unerwünschte Werbebotschaften (Spam) zu identifizieren und auszufiltern.

## **4.2 Wie sieht die aktuelle Serverumgebung aus?**

---

Die aktuelle Serverumgebung ist bei der Planung des Internetservers ebenfalls zu berücksichtigen. Oft (meist bei kleineren Unternehmen) soll möglichst wenig zusätzliche Hardware (Server) angeschafft werden. Dies führt dazu, dass verschiedene Dienste auf derselben Hardware betrieben werden.

Dadurch ergeben sich zwei Hauptprobleme:

- Verfügbarkeit durch «single point of failure»: Durch Ausfall der Hardware oder Stromausfall werden alle auf dem Server betriebenen Dienste lahmgelegt.
- Anfälligkeit gegen Angriffe: Ein Angreifer muss nur eine physische Maschine erobern und hat dadurch Zugriff auf eine grosse Menge von Daten verschiedenster Art (z. B. Kundendaten, die eigentlich nicht im Web verfügbar sind, aber auf dem gleichen physischen Server verwaltet werden).
- Kapazitätsprobleme aufgrund hoher Last durch externe (Web)-Besucher und durch die Verwendung als interner Server

Nebst diesen direkten Problemen ergeben sich auch Fragestellungen zur Anbindung anderer Systeme an den Internetserver. Bei komplexeren Umgebungen werden die Arbeiten auch auf verschiedene Server verteilt, d. h., der Webserver präsentiert die Daten lediglich,

während ein Applikationsserver die Verarbeitung vornimmt und ein weiterer Datenbankserver die Daten verwaltet und speichert.

## Repetitionsfrage

---

17 Welche Faktoren werden bei der Ist-Aufnahme berücksichtigt?

---

## 5 Soll-Zustand analysieren

---

Die Definition des Soll-Zustands ist zugleich das Pflichtenheft für den Internetserver. Welche Dienste sollen angeboten werden und für wen? Sind die Dienste auf mehrere physische Server zu verteilen? Müssen Datenbanken angebunden werden? Welche Sicherheitsanforderungen bestehen bezüglich Verfügbarkeit und Vertraulichkeit? Müssen Daten konvertiert werden oder sind ältere Systeme überhaupt in der Lage, mit dem Internetserver zu kommunizieren? Müssen Schnittstellen und Programme geschrieben werden, um ältere Applikationen an das Internet anzubinden?

### 5.1 Welche Applikationen sollen «web-enabled» werden?

---

Anhand einer Liste der bestehenden Applikationen wird entschieden, welche Daten und Funktionen auch im Web zur Verfügung stehen sollen. Möglicherweise müssen neue Schnittstellen geschaffen werden, um Datenbanken an den Internetserver anzubinden.

Bestehende Applikationen können oft nicht einfach mit dem Internetserver verbunden werden. Die Durchführung einer Bestellung aus dem Web bedingt eventuell, dass eine neue Applikation für den Web-Shop geschrieben wird, und innerhalb des Unternehmen weiterhin eine Client-Server-Applikation eingesetzt wird. Oft ist auch ein ERP-System im Einsatz. Moderne ERP-Systeme verfügen meist über vorbereitete (optionale) Schnittstellen für die Anbindung an das Internet. (Ein sogenanntes Enterprise Resource Planning (ERP) System besteht aus mehreren Anwendungen wie z. B. Lagerverwaltung, Personalverwaltung, Kundenstammdaten, Buchhaltung, welche miteinander verknüpft (integriert) sind und somit die Verwaltung der Daten eines Unternehmens mit einer (grossen) Anwendung, dem ERP-System, erledigt werden kann.)

### 5.2 Allgemeine Sicherheitsrichtlinien im Unternehmen

---

Falls allgemeingültige Sicherheitsrichtlinien im Unternehmen bestehen, müssen diese auch auf den Internetserver angewendet werden. Benutzerrichtlinien (Passwortvergabe, Benutzerverwaltung) müssen je nach Bedarf für den Internetserver separat erstellt werden (insbesondere wenn die Benutzer des Internetserver nicht mit bestehenden Verzeichnisdiensten wie Active Directory etc. verwaltet werden).

Vorgaben bezüglich der Verfügbarkeit von Systemen und der Integrität und Vertraulichkeit von Daten müssen ebenfalls eingehalten oder für den Internetserver spezifisch erstellt werden.

### 5.3 Datenschutz und Personendaten

---

Daten über Personen (z. B. Kunden- oder Mitarbeiterdaten) unterliegen besonderen Schutzbestimmungen des Datenschutzgesetzes. Die Leitfäden des Datenschutzbeauftragten geben detaillierte Hinweise, wie solche Daten bearbeitet und geschützt werden müssen: <http://www.edsb.ch/d/doku/leitfaeden/index.htm>.



Die wichtigsten Kriterien sind:

- Personendaten sind vor unbefugter Bearbeitung (Einsicht, Manipulation, Löschung) mit angemessenen technischen und organisatorischen Massnahmen zu schützen (z. B. Verschlüsselung, Regelung/Weisung zum Umgang mit Personendaten).
- Personendaten müssen richtig sein (Integrität der Daten) und sind bei Bedarf oder auf Anfrage der betroffenen Person zu berichtigen.
- Personendaten dürfen nur zu dem Zweck bearbeitet werden, für den sie gesammelt wurden (also keine Werbung mit Rechnungsadressen aus dem e-shop).
- Der betroffenen Person ist auf Anfrage Auskunft über die gesammelten Daten zu geben.

Ein wichtiger Punkt, der sich für den Betrieb eines Internetservers aus dem Datenschutzgesetz ableitet, ist der Umgang mit Logfiles, die personenbezogene Daten enthalten (d. h. Benutzernamen, die direkte Rückschlüsse auf Personen ermöglichen, z. B. Benutzername «hans.muster»). Die Bearbeitung dieser Logfiles muss so sichergestellt sein, dass keine personenbezogenen Auswertungen durch unbefugte Personen möglich sind, und dass die betroffenen Personen bei solchen Auswertungen darüber informiert werden (z. B. welcher Benutzer welche Webseite wann besucht hat, Einsicht in das Mailserver-Logfile um zu sehen, wer wem E-Mails schreibt).

## 5.4 Anforderungen an die Verfügbarkeit (uptime) und Datenvolumen

---

Die Anforderungen der Kunden an die Verfügbarkeit des Internetservers und der darauf verarbeiteten Daten müssen definiert werden. Insbesondere ist darauf zu achten, dass Backend-Systeme z. T. ausser Betrieb sind (z. B. nachts während des Back-up-Zyklus der Datenbank). Der Internetserver ist typischerweise ständig im Internet erreichbar und die Kunden aus verschiedenen Zeitzonen wissen nicht, dass im Hintergrund ein Back-up läuft und dadurch gewisse Informationen evtl. nicht zur Verfügung stehen.

Falls solche Unterbrüche zu erwarten sind, soll dies klar kommuniziert werden (Text auf der Homepage); bzw. kann heute mit entsprechenden Mitteln dafür gesorgt werden, dass solche Unterbrüche keine Auswirkung auf den Internetserver haben (z. B. Zwischenspeichern/caching der Datenbank auf dem Internetserver während der Back-up läuft).

### 5.4.1 Wie viel Traffic/Volumen wird erwartet; zukünftige Entwicklung?

---

Anhand von Messungen des Verkehrs (Anzahl Anfragen pro Minute, Stunde, Tag; Download-Datenvolumen pro Stunde, Tag etc.) kann festgestellt werden, wie der Internetserver benutzt wird. Durch ständige Auswertung dieser Daten kann man auch rasch erkennen, wenn die Belastung des Systems zunimmt und entsprechende Massnahmen treffen (mehr Speicherplatz, grösserer Server etc.).

Oft ist es schwierig abzuschätzen, wie hoch der Verkehr nach der Inbetriebnahme ist. Je bekannter die Webseite, je mehr Benutzer auf dem Mailserver, desto höher die Auslastung. Typischerweise nimmt die Auslastung in den ersten Wochen stark zu (bei bekannt werden der Webseite; Benutzer fangen an, E-Mail zu benutzen). Nach einigen Wochen ist der Normalzustand erreicht und die Messungen können beginnen.

Zur Messung eignet sich am besten die Auswertung der Logfiles des Web- und E-Mail-Servers; wichtige Kennzahlen sind:

- Anzahl Hits pro Tag (oder kürzere Abstände wenn nötig, z. B. pro Stunde), Verteilung über verschiedene Tageszeiten (dabei sieht man oft, wann in welchen Teilen der Welt morgen oder abend ist, da der Verkehr dann stark zu- oder abnimmt)
- Anzahl und Menge der Downloads, falls solche Angeboten werden
- Anzahl der eingeloggten Benutzer (sofern eine Möglichkeit für Login besteht, z. B. in einen geschlossenen Benutzerbereich). Diese Kennzahl ist insbesondere wichtig bei SSL-Verbindungen (über https), da die Ver- und Entschlüsselung der Daten zusätzliche Last erzeugt. Je mehr solche verschlüsselten Verbindungen offen sind, desto höher die Prozessorlast. Unverschlüsselte Verbindungen verursachen nicht solch grosse Last auf dem Prozessor. Diese Kennzahl wird auch «concurrent user» genannt, was so viel heisst wie «gleichzeitige Benutzer» und ist die einzige wichtige Kennzahl bezüglich der Benutzer. Unabhängig wie viele Benutzer auf dem System erfasst sind, ist es nur ausschlaggebend, wie viele das System gleichzeitig nutzen. Die Anzahl der «named user», d. h. der definierten/erfassten Benutzer, stellt lediglich die höchstmögliche Anzahl «concurrent user» dar, wird aber in der Praxis nie ausgeschöpft. Zwischen 20–50 % aller Benutzer sind normalerweise gleichzeitig online.
- Anzahl verschickter und empfangener E-Mails; Grösse und Anzahl der verschickten Attachments
- Anzahl und Datenmenge der Downloads via FTP
- Anzahl der Anfragen an den DNS-Server

## 5.5 Welche Benutzerprofile sind vorgesehen?

Je nach Einsatz des Internetserver gibt es verschiedene Benutzer auf dem System, die über entsprechende Rechte verfügen. Nachfolgend werden Möglichkeiten beschrieben, um diese Rechte zu verwalten.

### 5.5.1 CRUD: Create Read Update Delete

Die sogenannte CRUD-Matrix eignet sich vor allem für **Applikationen**, um die verschiedenen Autorisierungen (Berechtigungen) der **Benutzer** übersichtlich aufzuzeigen.

Diese Matrix soll zeigen, welche Rechte den verschiedenen Benutzern erteilt werden:

- C: Create: Dateien erzeugen
- R: Read: Dateien lesen
- U: Update: Dateien verändern
- D: Delete: Dateien löschen

	Kundenstamm	Produkte-DB	Benutzerverwaltung	E-Mail
<b>Kunde</b>	R	R		CRD
<b>Administrator</b>	CRUD	CRUD	CRUD	CRUD
<b>Internerbenutzer</b>	CRUD	CRUD	RU	CRD
<b>Support</b>	RU	RU	RUD	CRD

### 5.5.2 Zugriffskontrolle (Access Control Lists)

---

Eine Zugriffskontroll-Liste zeigt listenförmig auf, welcher Benutzer über welche Zugriffe verfügt. Das folgend Beispiel ist eine ACL für einen Webserver:

- Benutzer: Zugriffsart
- admin: Administrator mit Vollzugriff
- b.kaelin: Supportfunktionen
- c.meier: Supportfunktionen
- m.luethi: Kundenfunktionen

### 5.5.3 Administration

---

Der Administrator verfügt im Normalfall über weitgehende Berechtigungen. Üblicherweise wird dabei das Prinzip der «Gewaltentrennung» eingesetzt, d.h. der Administrator darf nur Änderungen durchführen oder Benutzer erstellen, wenn er von einer anderen (berechtigten) Person dazu ermächtigt wurde. Dadurch wird vermieden, dass der Administrator ohne Kontrolle auf dem System Arbeiten durchführt.

Beispielsweise stellt ein Mitarbeiter der Personalabteilung den Antrag auf ein E-Mail-Account. Die Personalabteilung erteilt danach dem Administrator den Auftrag, den E-Mail-Account zu eröffnen.

Ebenfalls wichtig ist die Dokumentation und Logging der Tätigkeiten des Administrators, damit jederzeit nachvollzogen werden kann, welcher Admin wann was geändert hat. Im Idealfall soll auch der (schrittliche) Antrag des Benutzers archiviert werden, damit nachvollziehbar ist, aufgrund welchen Auftrags der Administrator ein neues Benutzerkonto eröffnet hat.

Bei wichtigen Änderungen (z. B. Upgrade des Systems, löschen eines Benutzeraccounts) kann zudem das 4-Augen-Prinzip angewendet werden, d. h., ein zweiter Administrator gibt die Änderung nach einer Kontrolle frei. Dadurch wird vermieden, dass aus Versehen eine Änderung am System zu Ausfällen oder Problemen führt.

### 5.5.4 Externe Zugriffe

---

Zugriffe von aussen auf ein System sind immer mit grosser Vorsicht zu betreiben. Im Normalfall darf ein externer Zugriff nur durch gesicherte Kommunikationsverbindungen vorgenommen werden (secure shell ssh, https, VPN-Tunnel), und die Tätigkeiten des externen Zugreifers sollen immer protokolliert (Logfile) werden.

Eine übliche Vorgehensweise ist, den externen Zugriff prinzipiell zu sperren und nur auf Anfrage zu öffnen. Dadurch ist sichergestellt, dass immer bekannt ist, wann ein Zugriff erfolgt.

## Repetitionsfragen

---

21 Welche Anforderungen sind bei der Definition des Soll-Zustands zu berücksichtigen?

---

25 Erklären sie kurz den Unterschied zwischen Datenschutz und Datensicherheit?

---

## 6 Lösung entwerfen

---

In den folgenden Kapiteln werden die notwendigen Schritte für den Entwurf eines Internet-servers, von der Hardware über Software sowie Setup- und Betriebsdokumentation, im Detail dargelegt.

### 6.1 Benötigtes Know-how aneignen und planen

---

Das vorliegende Lehrmittel soll genau das Know-how vermitteln, das für das aufsetzen und betreiben eines Internetservers benötigt wird. Dennoch gibt es in der Praxis oft Fälle, bei denen professionelle Hilfe von aussen benötigt wird oder neue, noch nicht bekannte Technologien oder Systeme eingesetzt werden sollen. Besonders wenn der Internetserver sehr «exponiert» ist, das heisst, er erfüllt eine besonders geschäftskritische Aufgabe, muss der stabile Betrieb absolut gewährleistet werden.

Je nach zur Verfügung stehender Zeit und Budget kann das benötigte Know-how entweder angeeignet werden (mit Kursen, Ausbildungen etc.) oder es wird (zusätzliche) externe Hilfe von professionellen Anbietern beigezogen. Wenn der Internetserver für geschäftskritische Anwendungen (E-Business etc.) eingesetzt wird, empfiehlt es sich oft eine zusätzliche Sicherheitsüberprüfung durch eine spezialisierte Drittfirma vornehmen zu lassen. Damit erhält man eine unabhängige Bestätigung, dass die Anforderungen an das System erfüllt sind.

### 6.2 Benötigte Hardware festlegen

---

Bevor die Software für den Internetserver ausgewählt wird, muss der Server physisch aufgebaut werden. Die dazu benötigte Hardware ist zu bestimmen und anzuschaffen. Je nach Einsatzzweck kostet die Hardware für den Internetserver einiges an Geld. Es lohnt sich daher auch, auf die Herkunft und Garantie der Teile zu achten.

Als erstes wird eine Stückliste für die Hardware erstellt:

- Server-Einheit: Prozessortakt, -marke; RAM, Anzahl Netzwerkkarten, Bus-System und -geschwindigkeit, interne und externe Harddisks (Volumen, Typ), CD oder DVD Laufwerk, serieller Anschluss, KVM (Konsole, Video, Maus). Es ist bei der Auswahl der Server-Hardware darauf zu achten, dass diese für einen Dauerbetrieb ausgelegt ist. Eine billige Anschaffung von PC-Komponenten lohnt sich für einen richtigen Internetserver nicht, da solche Komponenten weniger für den Dauerbetrieb und die Anforderungen eines Servers optimiert sind.
- Netzwerk-Anschlüsse, Leitungen, Kabel, Kabelführung, Stromzufuhr, unterbruchfreie Stromversorgung bzw. Notstromaggregat
- Serverschrank (Rack), Zugangsmöglichkeiten, physischer Verschluss des Racks und je nach Modell auch des Servers mit Schlüssel, Belüftung/Ventilation, Klimatisierung des Raumes, Positionierung des Internetserver im Rack zusammen mit anderen Servern (Zugangsmöglichkeiten, Ein- und Ausbau)

Neben der eigentlichen Hardware ist mit den Lieferanten allenfalls auch eine Vereinbarung über Ersatzteillieferung zu treffen. Je nach Hardware kann es einige Tagen oder Wochen dauern, bis diese geliefert werden kann. Bei einem Ausfall muss dafür gesorgt sein, dass Ersatzteile in der benötigten Zeit geliefert werden können oder ab Lager verfügbar sind.

Falls der Internetserver hohe Anforderungen an verschlüsselte Verbindungen (SSL/TLS mit vielen concurrent sessions / aktiven Benutzern) stellt, sollte die Anschaffung von soge-

nannter Verschlüsselungs-Beschleuniger Hardware überlegt werden (sog. «SSL Accelerator» Hardware). Diese spezialisierte Hardware entlastet den Hauptprozessor von Verschlüsselungsaufgaben und sorgt für eine hohe Durchsatzrate (Performance) der verschlüsselten Verbindungen. Der Nachteil solcher Hardware ist meist der hohe Anschaffungspreis.

### 6.3 Sicherheit gewährleisten mit technischen und organisatorischen Massnahmen

---

Die Grundanforderungen an die Sicherheit des Servers (Verfügbarkeit, Integrität, Vertraulichkeit) müssen mit organisatorischen und technischen Massnahmen gewährleistet werden. Folgende Punkte sind abzuklären und zu dokumentieren:

Organisatorische Massnahmen:

- Verantwortlichkeiten für den Internetserver (Betrieb/Administration, Back-up, Benutzerverwaltung, Notfallplanung, Ersatzteile, digitale Zertifikate)
- Eskalationsprozedur (Alarmierungsorganisation) im Notfall, z. B. bei einem Virus- oder Hackerangriff aus dem Internet
- Regeln für den Betrieb und die Benutzer, Weisungen für korrektes Verhalten, Regelung, welche Daten auf dem Server wo gespeichert werden (dürfen) und welche Daten keinesfalls auf dem Internetserver gespeichert werden sollen

Technische Massnahmen:

- Firewall (Verantwortung für Konfiguration und Pflege der Firewall ebenfalls festlegen)
- Intrusion Detection System innerhalb der DMZ
- Spamfilter
- Content-Filter (Inhalts-Filter für ein- und ausgehenden Internet- und Mailverkehr)
- Proxy, Reverse Proxy
- Notstromversorgung (USV)
- Sicherer Standort und Zugang (abgeschlossenes Server-Rack)
- Brand- und Wasserschutz

### 6.4 Betriebssystem für den Internetserver definieren

---

Nach der Auswahl der geeigneten Hardware (je nach eingesetztem Betriebssystem hat dies Einfluss auf die Hardware, z. B. SUN Solaris), muss das Betriebssystem für den Internetserver ausgewählt werden. Folgende Betriebssysteme eignen sich für den Betrieb eines Internetserver:

- Windows 2000 Server
- Windows 2003 Server (verschiedene Ausführungen) mit Internet Information Server (IIS), jeweils aktuellste Version und Service Pack auswählen!
- Verschiedene Linux-Distributionen (SuSE, RedHat etc.)
- Unix-Derivate der BSD-Familie (FreeBSD, NetBSD etc.)
- SUN's Solaris (benötigt spezielle Hardware)

Für die Beispiele in diesem Lehrmittel wurde SuSE Linux «Open Source Edition» OSS v10 ausgewählt, da dieses Betriebssystem einfach aufzusetzen ist, die meisten benötigten Software-Komponenten für den Internetserver bereits mitbringt und frei erhältlich ist (open source). In der OSS-Edition sind keine nicht-open-source Komponenten enthalten. Alternativ kann auch die «Evaluation Edition» von SuSE Linux v10 verwendet werden. Diese ent-

hält proprietäre Anwendungen wie z. B. Adobe Reader etc. und ist ansonsten gleich zu konfigurieren wie die OSS-Edition.

## 6.5 Wie werden Applikationen an den Server angebunden?

---

Werden auf dem Internetserver verschiedene Applikationen angeboten, ist zu überlegen, wie diese an den Internetserver angebunden werden:

- **Direkt:** Die Applikation wird direkt auf dem Internetserver installiert und läuft evtl. auf eigenen Ports. Dabei muss lediglich sichergestellt werden, dass die Applikation auf dem gleichen Server keine Störungen mit anderen Applikationen (Webserver, Mail etc.) verursacht.
- **TCP/IP:** Die Applikation(en) ist auf einem anderen Server installiert und wird über das TCP/IP-Protokoll angesprochen. Dabei ist sicherzustellen, dass die Adresse der Applikation vom Internetserver erreichbar ist. Je nach Bedarf können weitere Sicherheitsmassnahmen getroffen werden, um die Applikation vor Angreifern von aussen zu schützen (z. B. zusätzliche Firewall vor dem Applikationsserver).
- **Proprietäre Protokolle:** Es gibt auch Applikationen, welche über proprietäre Protokolle mit dem Internetserver kommunizieren. Solche Applikationen sind eher selten, die Anbindung und Kommunikation zwischen Internetserver und Applikation soll vor Inbetriebnahme ausführlich getestet werden.

## 6.6 Beschaffung der Software und Installationsabhängigkeiten

---

Bei der Wahl der Software für den Internetserver sind einige Fragen zu klären:

- Welche Dienstleistungen werden für welches Publikum angeboten? Wird der Internetserver nur als Webserver gegen aussen und als interner Mailserver, evtl. zusätzlich mit interner DNS-Funktionalität betrieben oder sind Bereiche des Internetserver (Mail, FTP etc.) auch von aussen her zugänglich; von wem und in welcher Art (verschlüsselte Verbindung, ständige Verbindung oder nur temporär)?
- Lizenzmodell des Anbieters? Entscheidet man sich für lizenzpflichtige Software, sind einerseits einmalige Lizenzkosten zu entrichten und oft jährliche Gebühren für Upgrades und Garantie fällig. Solche Kosten sind in der Budgetierung des Internetserver aufzunehmen. Eine wichtige Einflussgrösse auf den Lizenzpreis hat einerseits die eingesetzte Hardware (Anzahl Prozessoren, Server-Typ) sowie die Anzahl der Benutzer (je nach Software werden die konfigurierten Benutzer (Gesamte Anzahl Benutzer) oder die «concurrent user» für die Preisgestaltung verwendet). Bei open-source Software entfallen die Lizenzgebühren, allerdings können Kosten für Pflege und Wartung (Fachpersonen; Anbieter der Software bietet kostenpflichtige Dienstleistungen an wie z. B. SuSE) anfallen.

Wird die Software zudem online gekauft, kann man diese oft downloaden, ohne dass ein zusätzliches Original-Softwarepaket geliefert wird (wobei meist der Preis dadurch etwas günstiger wird). Es ist dabei wichtig, eine oder mehrere Sicherheitskopien der Original-SW zu erstellen, damit auch zu einem späteren Zeitpunkt der Internetserver mit der ursprünglichen SW-Version wiederhergestellt werden kann; unter anderem auch um entsprechende ältere Back-ups wieder erfolgreich aufspielen zu können.

Bei der Auswahl der Software für die Beispiele in diesem Lehrmittel wurde ausschliesslich auf frei erhältliche Software geachtet, um keine Einschränkungen für die Lernenden einzugehen.

### 6.6.1 Webserver

---

Der im Lehrmittel verwendete Webserver ist der open-source Webserver Apache 2.0. Für Windows-Server wird üblicherweise der Microsoft-eigene Webserver «Internet Information Server» IIS 6.0 verwendet, wobei auch Apache gut auf Windows-Systemen funktioniert. Grundsätzlich ist Apache auf fast jedem Betriebssystem lauffähig.

Daneben gibt es Dutzende weitere Webserver, die in verschiedenen Sprachen geschrieben sind. Diese werden meist in spezialisierten Projekten/Applikationen eingesetzt, seltener im kommerziellen Umfeld, wo sich Apache und IIS stark verbreitet haben.

### 6.6.2 FTP Server

---

FTP-Server gibt es ebenfalls in einer grossen Anzahl, open-source oder kommerziell. Oft ist ein FTP-Server bereits auf dem Betriebssystem vorhanden, v. a. wenn es sich um ein Server-Betriebssystem handelt.

### 6.6.3 DNS Server

---

Die wohl bekannteste DNS-Serversoftware ist die Implementierung «BIND». Sie ist zugleich der Ur-FTP-Server und wird heutzutage oft eingesetzt. Daneben gibt es ebenfalls weitere, frei verfügbare DNS-Serversoftware.

### 6.6.4 Mailedienste: SMTP und Postfach/Mailbox

---

Beim E-Mail wird unterschieden zwischen dem SMTP-Server, der Mails versendet und entgegennimmt, um an die Postfächer zu verteilen. Der SMTP-Server muss nicht zwingend der gleiche Server sein, wo die Postfächer für die Benutzer verwaltet werden.

Den Mail Transfer Agent (MTA, wie der SMTP Server auch genannt wird) gibt es in mehreren Varianten und für verschiedene Betriebssysteme. Die bekanntesten sind:

- MS Exchange für Windows
- Exim (Unix)
- Lotus Domino (Linux, Unix, Windows)
- Postfix, Qmail, Sendmail (Unix/Linux)

Der Mail Delivery Agent (MDA) ist zuständig für die Verteilung der eingehenden Mails in die Mailboxen/Benutzerkonten. Der MDA ist auch zuständig für die Verwaltung der E-Mails wenn diese vom MUA (Mail User Agent, Mailclient) geholt werden mittels POP3 oder IMAP-Protokollen. Die Mailclients sind nicht Bestandteil dieses Moduls.

Der bekannteste MDA unter Linux ist procmail. Unter Windows erfüllt Exchange diese Funktion.

## 6.7 Zu verwendende Namen für Systeme, Dienste und Daten

---

Bei der Planung des Internetservers soll auf eine konsistente Namengebung geachtet werden. Bestehen bereits entsprechende Regelungen im Unternehmen, so sind diese soweit anwendbar zu gebrauchen. Ansonsten empfiehlt es sich, gewisse Standards zu definieren:

- Servernamen werden nach einem bestimmten Muster vergeben, z. B. «svr\_i01» für «Server, Internet, Nummer 01» oder «srv\_db\_i01» für «Server: Datenbankserver, Gruppe: i01 (Internetserver)», d. h. Datenbankserver der die Internetserver bedient.
- Die Dienste (http-daemon, ftp-daemon etc.) haben oft bereits voreingestellte Namen, die teilweise änderbar sind. Werden diese geändert, ist auf Abhängigkeiten zu anderen Systemen und Applikationen zu achten, die möglicherweise den Standard-Namen suchen und nicht finden und dadurch eine Kommunikation verhindert wird. Es empfiehlt sich, die Namen der Dienste und Prozesse in der Standardeinstellung zu belassen, sofern dies keine Konflikte mit bereits erfolgten Installationen hervorruft.
- Daten und Verzeichnisse sind ebenfalls nach einheitlichen Mustern zu verteilen und zu bezeichnen. So empfiehlt es sich, Standard-Verzeichnisse z. B. für Installationsdateien, Daten, Applikationen etc. festzulegen. Auch die Namengebung für die Dateien soll soweit möglich aufschlussreich sein, damit leicht erkannt wird, um was für einen Dateityp es sich handelt. Dies ist insbesondere im Unix-/Linux-Umfeld nützlich, da dort nicht alle Dateien eine entsprechende Endung haben. Auch hier ist es wichtig, Abhängigkeiten mit anderen Applikationen oder Diensten zu prüfen. Die Umbenennung von Konfigurationsfiles führt z. B. oft dazu, dass eine Konfiguration der Systeme mit GUI-Werkzeugen nicht mehr möglich ist, da diese nach fest definierten Dateinamen suchen.

## 6.8 Standardeinstellungen festlegen

---

Neben den Namenskonventionen ist es sehr wichtig, eine Standard- oder Default-Einstellung für alle installierten Dienste zu definieren. Diese Einstellungen sind entweder schon bei der Installation aufzufinden oder müssen direkt nach der Installation vorgenommen werden. Fortgeschrittene Administratoren können die Installationsscripts bzw. -routinen so anpassen, dass die Software auf vorher festgelegte Verzeichnisse installiert wird und gegebenenfalls von den «Werkseinstellungen» abweichende Konfigurationen aufweist.

Mithilfe der vordefinierten Standardeinstellungen ist es auch für andere Systembetreuer nachvollziehbar und möglich, eine Installation vorzunehmen. Es ist deshalb wichtig, dass die Default-Einstellungen detailliert dokumentiert werden.

## Repetitionsfragen

---

- |   |   |
|---|---|
| 3 | Welche Hardware wird bei der Realisierung des Internetserver benötigt bzw. auf den Internetserver abgestimmt? |
| 8 | Was ist nach der Beschaffung der Software zu berücksichtigen?   |
-



## 7 Systemtest und Dokumentation vorbereiten

---

Bevor mit der Installation begonnen wird, wird die Dokumentation der in den vorherigen Kapiteln beschriebenen Schritte vorgenommen. Zusätzlich sind die Anforderungen an den Internetserver in Form von Testfällen zu dokumentieren, d. h., wie die Erfüllung der Anforderungen getestet wird.

Nebst der Dokumentation der Testfälle werden die Testresultate ebenfalls dokumentiert, um Probleme nachvollziehbar zu machen. Damit wird vermieden, dass durch Installation oder Änderungen am Internetserver andere Funktionen nicht mehr verfügbar sind. Anhand der Testdokumentation ist es so immer möglich, auf den letzten funktionierenden Stand des Systems zurückzugehen.

### 7.1 Technische Tests am Internetserver

---

Zu den technischen Tests am Internetserver gehören vor allem die Überprüfung der generellen Erreichbarkeit des Internetserver und die Berechtigungen auf Netzwerkebene innerhalb des Subnetzes und vom Internet her.

Folgendes sollte überprüft werden:

- **Routing:** Kann die IP-Adresse des Internetserver von den notwendigen Zugangspunkten aus erreicht werden (Intranet/LAN, DMZ, Internet, andere Netzwerke innerhalb des Unternehmens)? Erfolgt der Zugang über die korrekten Routen (mithilfe von `tracert/traceroute` überprüfen)? Ebenfalls überprüft werden muss, ob der Server von Netzen erreichbar ist, von welchem er nicht erreichbar sein sollte. Dieser Test soll zusätzlich auch umgekehrt erfolgen: Welche Netze kann der Internetserver erreichen (ein von einem Angreifer kompromittierter Internet-Server im internen Netzwerk kann dadurch Schaden auf anderen Systemen anrichten)?
- **Umgebung:** Stromversorgung (Test unterbrechfreie Stromversorgung), Belüftung und Klimatisierung (Test der Temperatur; auch nach ein paar Tagen ununterbrochenen Betriebs), Zugang zu Kabeln und Server (für Reparaturen, Austausch); dies besonders wenn der Server andernorts «housed» und nicht in eigenen Räumlichkeiten betrieben wird.
- **Burn-in Test:** Hardware über mehrere Tage laufen lassen, um Funktionsstörungen zu beobachten die sich evtl. erst nach einer gewissen Betriebsdauer einstellen.

### 7.2 Applikatorische Tests am Internetserver

---

Die applikatorischen Tests umfassen die Überprüfung der Funktionen der einzelnen installierten Applikationen. Beim Internetserver ist dies:

- **Webserver (Port 80 und 443, evtl. andere Ports je nach Konfiguration; Administrations-Interface z. B. auf Port 8080 falls vorhanden):** Werden die richtigen Seiten angezeigt? Bei Konfiguration von mehreren Domains: Werden alle richtig angezeigt? Spezielle Konfigurationen testen (individuelle Error-Messages). Zusätzlich sollen die Zugriffe von nicht authentisierten Benutzern getestet werden: Auf welche Verzeichnisse haben die Benutzer Zugriff? Bei verschlüsselten Verbindungen die Zugänglichkeit und Gültigkeit der Zertifikate überprüfen. Möglicherweise wurde konfiguriert, dass nur verschlüsselte Verbindungen möglich sind: Testen der Umleitung bei Verbindung mit nicht-verschlüsselter Verbindung. Wird diese auf die verschlüsselte Verbindung umgeleitet (redirect von port 80 zu 443)? Das Management-Interface auf einem anderen Port (z. B. 8080, 8081 etc.) sollte nicht von extern (Internet) zugänglich sein. Dies soll ebenfalls getestet

werden. Der Webserver ist generell sehr detailliert zu testen (insbesondere Sicherheitstests), ganz besonders wenn eine Datenbank mit vertraulichen Daten mit dem Webserver verbunden ist. Weitere wichtige Informationen zu diesem Thema liefern verschiedene Webseiten, darunter: <http://www.webappsec.org> und <http://www.osstmm.org>

- **Mailserver (port 25 SMTP, 110 POP3, 143 IMAP etc.):** Dieser soll ebenfalls ausführlich getestet werden:
  - Können E-Mails von aussen her versendet werden? (Spam-Relaying vom Internet aus, E-Mail versenden ohne Authentifikation, E-Mail senden mit falschem Absender)
  - Werden E-Mails von verschiedenen externen (und internen) Absendern akzeptiert? Werden Spams blockiert, und werden berechnete E-Mails als Spam blockiert? Können E-Mails auch vom Internet aus abgefragt werden (POP3, IMAP), bzw. soll dies nicht möglich sein? Sind verschlüsselte Verbindungen konfiguriert und sollen diese auch von extern erreichbar sein?
- **FTP-Server (port 20, 21):** Ist dieser von extern erreichbar und welche Benutzer können einloggen? Ist anonym Zugriff (anonymous) möglich? Wenn ja, welche Verzeichnisse sind sichtbar? Können auch grössere Dateien übermittelt werden, oder unterbricht die Verbindung? Wie viele Sessions/Sitzungen können gleichzeitig geöffnet werden (wichtig bei starkem Verkehr auf dem FTP-Server, wenn viele Benutzer gleichzeitig Downloads machen möchten)? Welche Benutzer dürfen schreiben, welche nur lesen? Welche Verzeichnisse sind sichtbar?
- **DNS-Server (udp/tcp port 53):** Werden von extern auch interne Domains aufgelöst (und somit interne Adressen nach aussen preisgegeben), bzw. werden externe Anfragen beantwortet (oft ist der DNS-Server nur für internen Gebrauch, d. h. Anfragen von aussen sollen nicht möglich sein)? Erlaubt der DNS-Server gar einen zone-transfer von extern (dies sollte nur für den secondary DNS server möglich sein)? Ist die Konfiguration des DNS-Servers geschützt (v. a. bei Web-GUI zur Konfiguration) oder besteht für Angreifer die Möglichkeit Einträge zu ändern?

### 7.3 Sicherheitstests rund um den Internetserver

---

Die Sicherheitstests rund um den Internetserver umfassen einerseits die oben genannten Tests und sollen vor allem die Anforderungen an Verfügbarkeit, Vertraulichkeit und Integrität sicherstellen.

- **Verfügbarkeit:** Wie verhält sich der Internetserver unter Last? Ist der Internetserver immer verfügbar? (Dieser Test lässt sich einfach von einem benachbarten System automatisieren, um die ständige Erreichbarkeit des Internetserver dauernd zu überwachen.)
- **Vertraulichkeit:** Sind die schützenswerten Daten entsprechend sicher und können nur berechnete Benutzer darauf zugreifen? Welche Daten über Mitarbeiter, Kunden oder das Unternehmen sind sichtbar?
- **Integrität:** Ist das Zertifikat des Webserver aktuell (Ablaufdatum), und entspricht es der Domain der WWW-Adresse? Sind die DNS-Einträge korrekt und können diese nicht verändert werden?

Neben diesen generellen Hinweisen finden sich im Internet detaillierte Anleitungen für die Absicherung der verschiedenen Internet-Server-Dienste. Die Sicherheit des Internetserver kann auch mit automatisierten Werkzeugen getestet werden. Die bekanntesten sind:

- Nmap ([www.insecure.org](http://www.insecure.org)): Portscanner zur Überprüfung der Dienste auf einem Server
- Nessus (Unix), NeWT (Windows) ([www.nessus.org](http://www.nessus.org)): Vulnerability Scanner, der die Schwachstellen von einer Vielzahl von Diensten aufspürt und testet; sehr empfehlenswertes Werkzeug

- **Ethereal** ([www.ethereal.org](http://www.ethereal.org)): Sniffer, um Daten und Verbindungen im lokalen Netzwerk zu überprüfen
- **Nikto** ([www.cirt.net](http://www.cirt.net)), **N-Stealth** ([www.nstalker.com](http://www.nstalker.com)): Webserver-Scanner, die speziell Webserver überprüfen
- **Microsoft Baseline Security Analyzer (MBSA)**, ([www.microsoft.com](http://www.microsoft.com)): nur für Windows-Systeme
- **IIS Lockdown Tool** ([www.microsoft.com](http://www.microsoft.com)): schliesst nicht benötigte Dienste und Lücken im Internet Information Server

## 7.4 Lasttest

---

Das Verhalten des Internetservers unter Last ist oft nicht einfach zu testen, da das Benutzerverhalten sehr schwierig vorauszusagen ist. Dennoch gibt es verschiedene Tools, um solche Tests durchzuführen und mehrere Benutzeranfragen zu simulieren.

Die beste Möglichkeit für einen Lasttest auf dem Internetserver ist die Benutzung durch echte User. Ein solcher Test kann während der Pilotphase durchgeführt werden, wenn immer mehr Benutzer für den Internetserver zugelassen werden. Während dieser Phase werden die Logs und Auslastungsprotokolle genau beobachtet, um bei Problemen sofort eingreifen zu können.

Microsoft bietet für den IIS das «Web Capacity Analysis Tool» an, um Benutzersitzungen zu simulieren. Daneben gibt es auch das «Web Application Stress Tool». Es gibt des Weiteren viele Stresstest-Tools im Internet. Die meisten davon sind jedoch kostenpflichtig, können aber als Testversion oft 30 Tage lang ausprobiert werden.

## 7.5 Dokumentation des Internetservers

---

Die Dokumentation der Installationsparameter (Auswahl Hardware, Software, Versionen, Speicher- und Lagerorte der SW und HW) sowie des Installationsvorgehens ist ebenso wichtig wie die Beschreibung der Tätigkeiten im täglichen Betrieb des Servers. Wenn nötig, wird auch eine Dokumentation für die Benutzer des Systems erstellt bzw. bei den entsprechenden Applikationsverantwortlichen in Auftrag gegeben.

Eine vollständige und aktuelle Dokumentation erleichtert die Installation und Pflege eines Internetservers ungemein und ist daher als zwingend zu erachten.

Detailliertere Angaben zur Dokumentation eines Servers sind im Modul 127 «Server betreiben» behandelt.

### 7.5.1 Installationsdokumentation

---

Die Installationsdokumentation umfasst die gesamte Hardware, die zur Installation benötigten wurde sowie die Software, die installiert wurde. Es lohnt sich auch, die Original-Lieferscheine bzw. Garantie-Vereinbarungen zentral aufzubewahren (zusammen mit der Installationsdoku), um bei Beschaffung von Ersatzteilen die benötigten Dokumente zusammen zu haben.

Seriennummern von Hard- und Software werden idealerweise ebenfalls in die Installationsdokumentation aufgenommen, um diese immer im Zugriff zu haben (elektronisch).

Die Installationsdokumentation beschreibt Schritt für Schritt die notwendigen Installationspunkte und Anfangs-Konfigurationen und idealerweise Möglichkeiten, um die Installations-

tion jeweils zu testen (Funktionstest). Sehr geeignet sind dabei «screenshots», um das Vorgehen bildhaft zu dokumentieren.

### 7.5.2 Betriebsdokumentation

---

Die Betriebsdokumentation umfasst die Informationen für den Administrator eines Systems, die für den täglichen Betrieb benötigt werden. In der Betriebsdokumentation wird auch das Notfallkonzept beschrieben, damit die Informationen zentral (und nicht auf verschiedene Dokumente verteilt) verfügbar sind.

Konfigurationen, Konfigurationsdateien und für den Betrieb notwendige Einstellungen werden dokumentiert. Befehle und Hilfsprogramme, um den Server zu warten (Back-ups, Funktionstests, sog. «morning checks» um den Betrieb des Servers regelmässig zu überprüfen) werden ebenfalls beschrieben.

### 7.5.3 Benutzerdokumentation

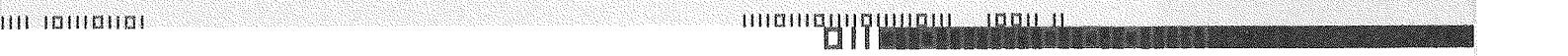
---

Die Benutzerdokumentation umfasst im Wesentlichen die Handbücher für die einzelnen Applikationen und idealerweise auch das Vorgehen bei Problemen (Hotline, Kontaktmöglichkeiten).

Ebenfalls in der Benutzerdokumentation enthalten sind allfällige Konfigurationen von Client-Software, die der Benutzer vornehmen muss, um mit dem Internetserver zu kommunizieren/arbeiten.

## Repetitionsfragen

- 
- |    |   |
|----|---|
| 13 | Welche Art von Tests sollen vor der Abnahme des Internetserver geplant werden?  |
| 18 | Nenne Sie die verschiedenen Arten von Dokumentationen für einen Internetserver? |
-



## Teil C Internetserver realisieren

---

© 2008 Pearson Education, Inc. All rights reserved. Printed in the United States of America. This book is published by Pearson Education, Inc., 221 Rte. 91, Northbrook, IL 60062-3202. ISBN 0-13-035957-9. ISBN 978-0-13-035957-2. Printed in the United States of America. This book is published by Pearson Education, Inc., 221 Rte. 91, Northbrook, IL 60062-3202. ISBN 0-13-035957-9. ISBN 978-0-13-035957-2.

## Einleitung, Lernziele und Schlüsselbegriffe

---

### Einleitung

---

Im Teil C wird detailliert auf die einzelnen Installationsschritte und insbesondere die Konfiguration der einzelnen Dienste eingegangen. Die Installation basiert auf SuSE Linux und wird mit frei verfügbaren (open source) Softwarepaketen durchgeführt. Schritt für Schritt wird die Grundkonfiguration des Betriebssystems und anschliessend die Konfiguration der Dienste Webserver, Mailserver, DNS, FTP-/Fileserver aufgezeigt. Anschliessend werden die Log-Dienste (Logfiles) und Sicherungsprozeduren für Back-ups beschrieben. Speziell wird auf die Analyse der Logfiles mit entsprechenden Hilfsmitteln (Tools) eingegangen. Die Anbindung von fremden Ressourcen wie Authentifizierungsserver (z. B. LDAP oder Active Directory) und Datenbanken wird ebenfalls angesprochen.

### Lernziele und Lernschritte

---

Lernziele	Lernschritte
<input type="checkbox"/> Installiert, konfiguriert und dokumentiert den Internetserver gemäss Anforderungsprofil.	<ul style="list-style-type: none"> <li>• Wie ist bei der Installation des Internetserver vorzugehen (Hardware, Software, Abhängigkeiten)?</li> <li>• Welche Punkte sind bei der Konfiguration der verschiedenen Dienste zu berücksichtigen?</li> <li>• Welche Art von Dokumentation ist zu führen?</li> </ul>
<input type="checkbox"/> Ist in der Lage, mehrere virtuelle Server (verschiedene Domänen, sog. Namensauflösung) einzurichten.	<ul style="list-style-type: none"> <li>• Wie werden virtuelle Server konfiguriert?</li> <li>• Wie erfolgt die Namensauflösung (DNS) und wie wird diese konfiguriert?</li> </ul>
<input type="checkbox"/> Stellt die an die unterschiedlichen Dienste angepassten Zugriffsberechtigungen ein. Kenntnisse in Einsatz und Verwendung des Usermanagements.	<ul style="list-style-type: none"> <li>• Wie und wo werden die Berechtigungen für die Dienste und Benutzer eingestellt?</li> <li>• Welche Berechtigungen werden durch das Betriebssystem eingestellt, welche durch die Dienste?</li> </ul>
<input type="checkbox"/> Erstellt das Benutzerkonzept einer einfachen Mailumgebung.	<ul style="list-style-type: none"> <li>• Welche Benutzertypen sind bei einer Mail-Umgebung (E-Mail) zu berücksichtigen?</li> <li>• Wie greifen die verschiedenen Benutzer auf das Postfach zu und welche Dienste werden dafür benötigt und sind zu konfigurieren?</li> </ul>

### Schlüsselbegriffe

---

OSS, Evaluation, Domäne, Ressource, Direktive, Directory, Binärdatei, timeout, CGI, Perl, PHP, SSL, TLS, Root, URL, Logfile

## 8 Software installieren, Default-Einstellungen und Benutzer konfigurieren

Im folgenden Kapitel wird die Installation und Konfiguration des Internetservers detailliert beschrieben und anhand von Screenshots dokumentiert.

### 8.1 Software besorgen (Quelle, Version, Plattform)

SuSE Linux OSS ist verfügbar unter [www.opensuse.org](http://www.opensuse.org). In der Distribution sind die benötigten Software-Pakete enthalten. E-Mail wird mit der Software von Stalker konfiguriert, da dort sowohl der SMTP-MTA wie auch die Postfächer einfach webbasiert verwaltet werden können. Auf Linux gibt es dazu ebenfalls entsprechende Pakete (postfix MTA und procmail Postfächer), diese sind aber wesentlich komplizierter zu konfigurieren und werden deshalb abgeschaltet.

### 8.2 SuSe Linux 10 OSS

SuSE Linux 10 OSS kann unter [http://en.opensuse.org/Released\\_Version](http://en.opensuse.org/Released_Version) gedownloadet werden.

Es ist auch eine «Evaluation Edition» vorhanden, welche unlimitiert gültig ist (für Privatgebrauch). Die Evaluation Edition beinhaltet auch lizenzpflichtige Software (freeware) wie z. B. Adobe Reader und weitere. Da diese Pakete für die Konfiguration des Internetservers nicht benötigt werden, wurde die OSS Edition verwendet.

Die komplette SuSE Linux 10 OSS Edition verteilt sich auf 5 Installations-CDs von jeweils ca. 600 MB.

Um die Konfiguration der Dienste zu erleichtern, wird jeweils direkt als Administrator-Benutzer «root» unter Linux eingeloggt (in KDE). Grundsätzlich können die Konfigurationen über die grafische Oberfläche vorgenommen werden. In Ausnahmen sind die Konfigurationsfiles direkt anzupassen.

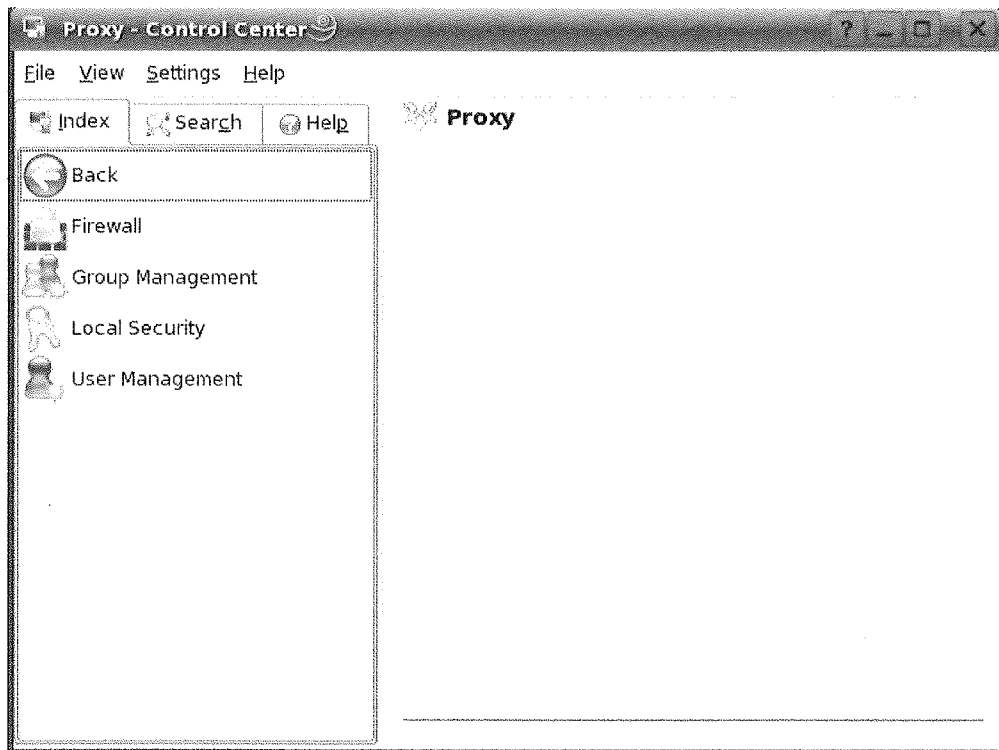
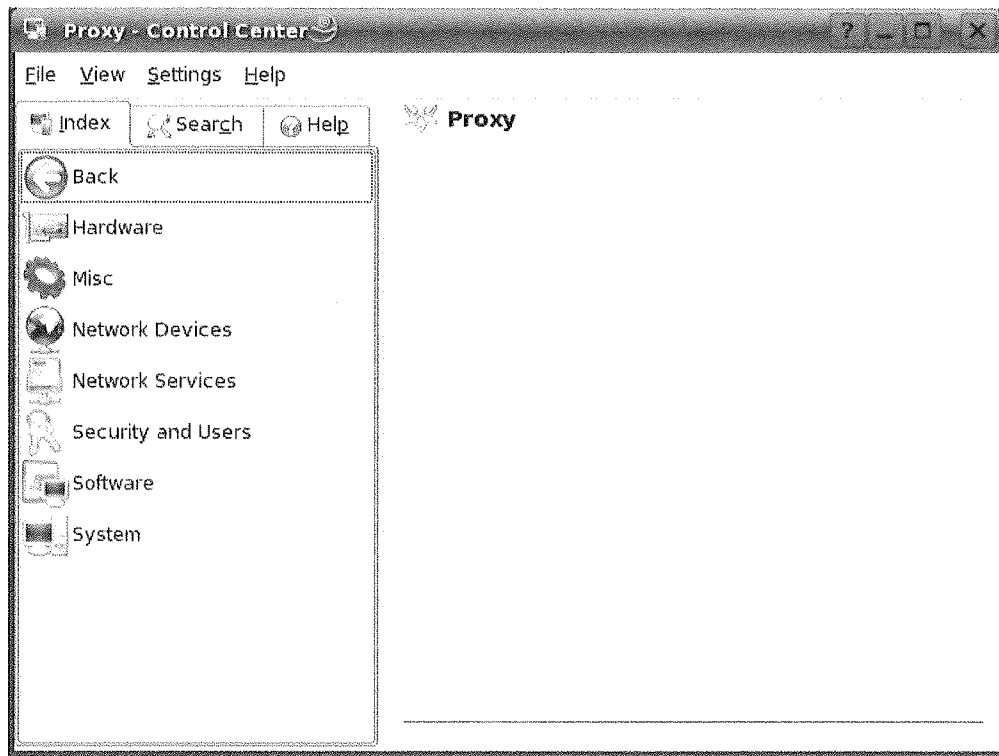
Nachdem SuSE Linux standardmässig installiert wurde, erfolgt die Konfiguration der einzelnen Dienste. Dabei muss SuSE Linux für den Betrieb des Internetservers umkonfiguriert werden, damit die einzelnen Dienste optimal betrieben werden können.

#### Grundsätzliche Einstellungen

Die Konfiguration der Dienste erfolgt entweder mithilfe des «Control Center», welches über das «K Menu» (grüner Punkt unten links) geöffnet werden kann.

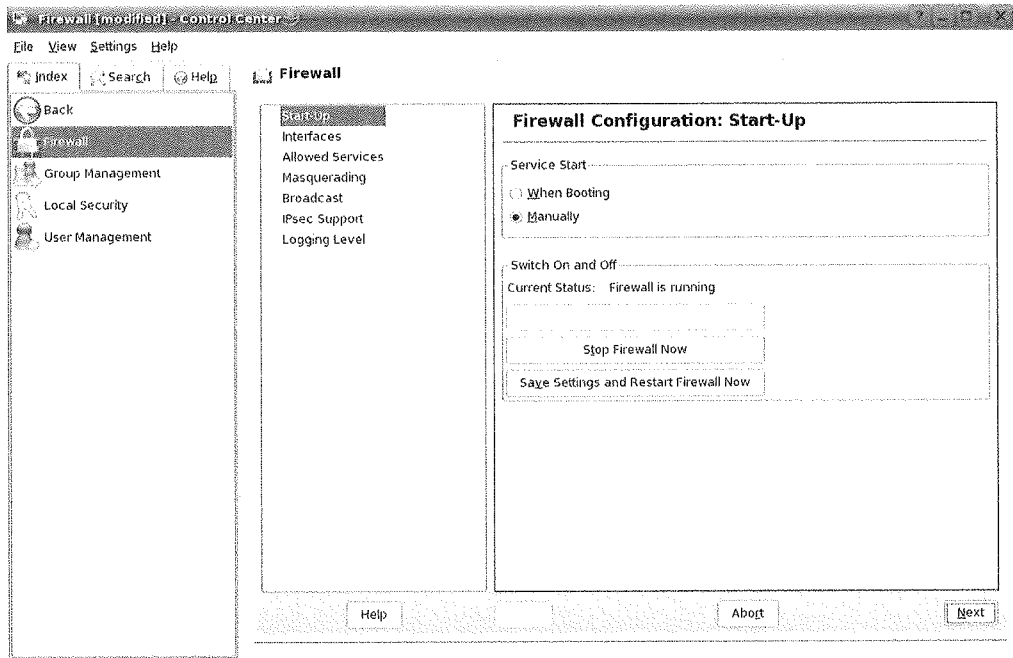


Im «Control Center» wird der Menüpunkt «YaST2 Modules» aufgerufen. Unter «Security and Users» wird die Firewall aufgerufen (Menüpunkt «Firewall») und ausgeschaltet.

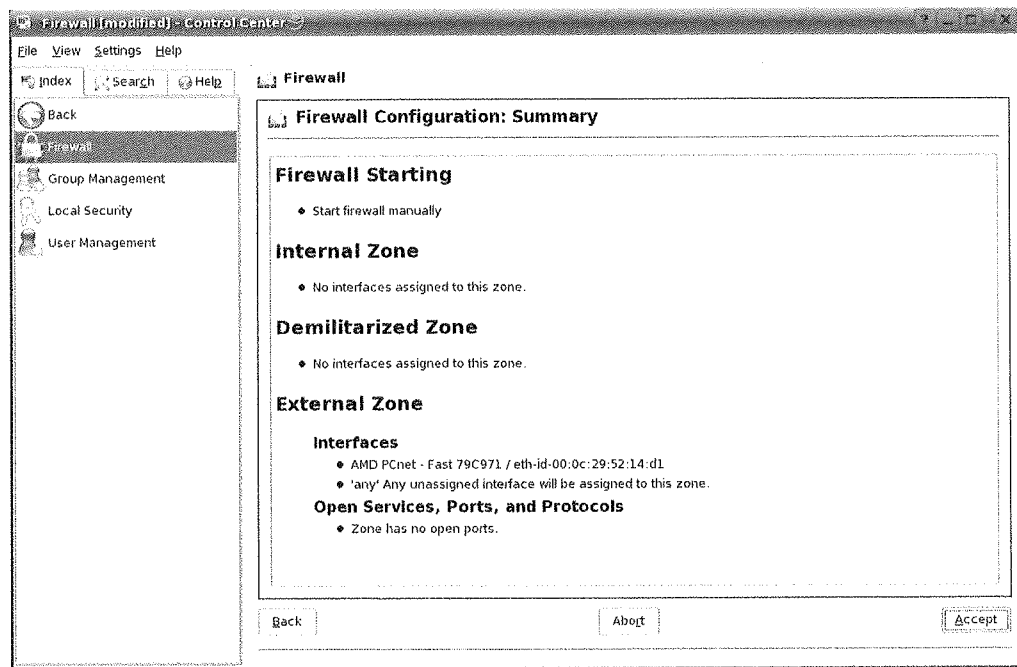


Service Start «Manual» auswählen, Button «Stop Firewall Now» auswählen und Button «Next» klicken.

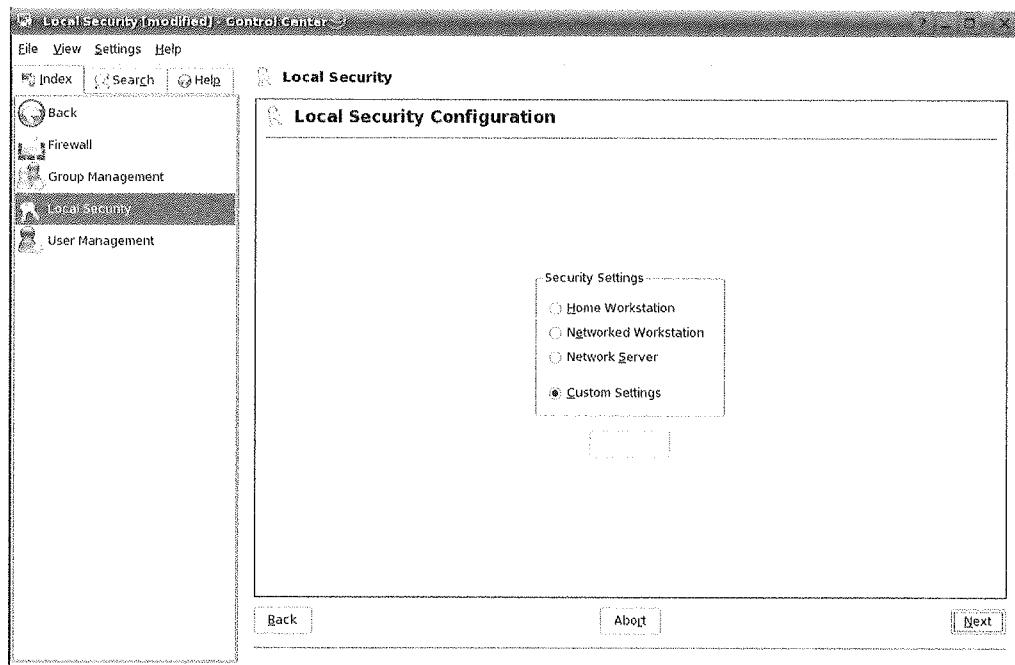




Danach werden die Änderungen angezeigt, welche mit »Accept“ durchgeführt werden.



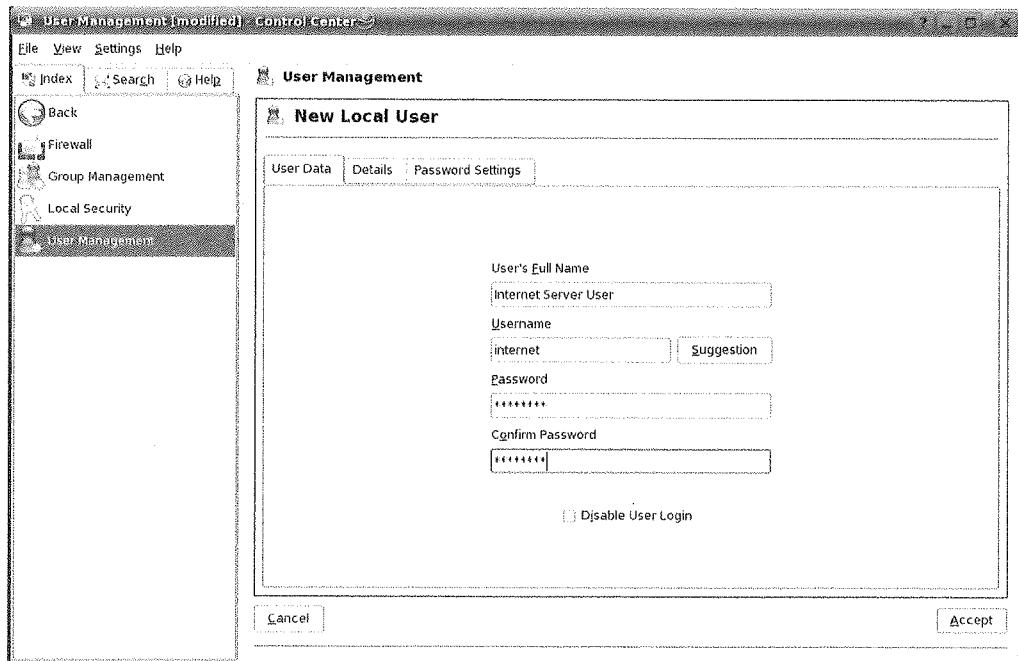
Unter dem Menüpunkt «Local Security» können nach Abschaltung der Firewall die Sicherheitseinstellungen durchgeführt werden. Hier wurde «custom settings» ausgewählt. Die Firewall kann nach erfolgter Konfiguration des Internetserver wieder eingeschaltet werden, wobei die notwendigen Ports für die Internet-Dienste zu öffnen sind. Es empfiehlt sich, zusätzlich eine separate Firewall vor dem Internetserver zu betreiben, da der Internetserver mit den vielen offenen Diensten im Internet sehr exponiert und ein Angriffsziel darstellt.

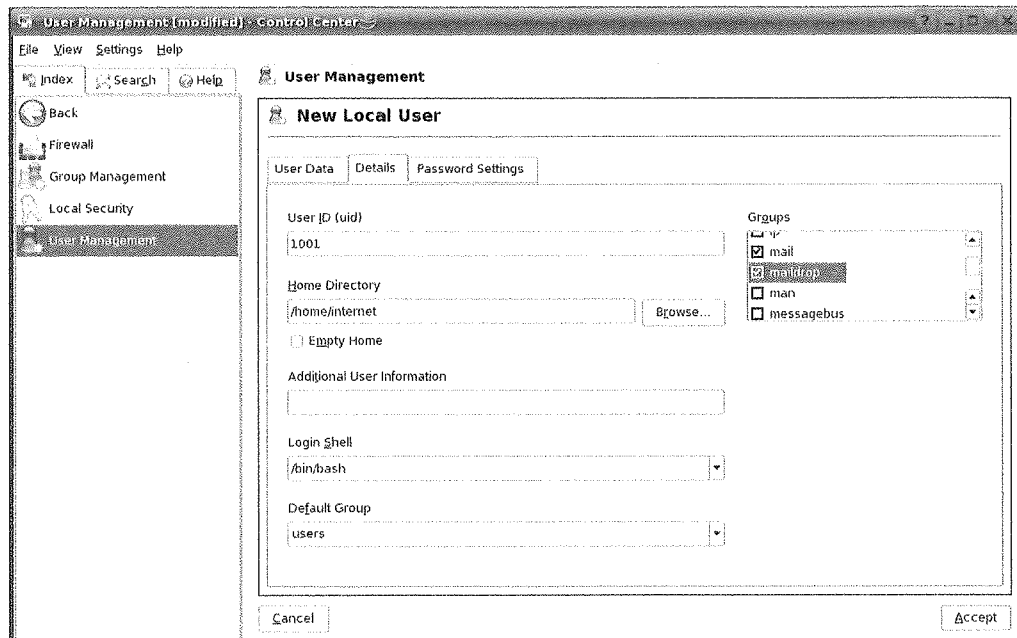


Die gewünschten Anpassungen vornehmen, wobei für den Betrieb als Internetserver die Default-Einstellungen belassen werden können, oder an die Sicherheitsrichtlinien angepasst werden sollen.

- Password Settings
- Boot Settings
- Login Settings
- User Addition
- Miscellaneous Settings
- Button «Finish» klicken

Danach werden die Benutzer auf dem Internetserver konfiguriert. Diese Benutzer werden auf Ebene des Betriebssystems erstellt und erhalten entsprechende Berechtigungen. Nicht alle Internetserver-Dienste nutzen die System-Benutzer; dort müssen die Benutzer separat erfasst werden. Der Menüpunkt heisst «User Management»:

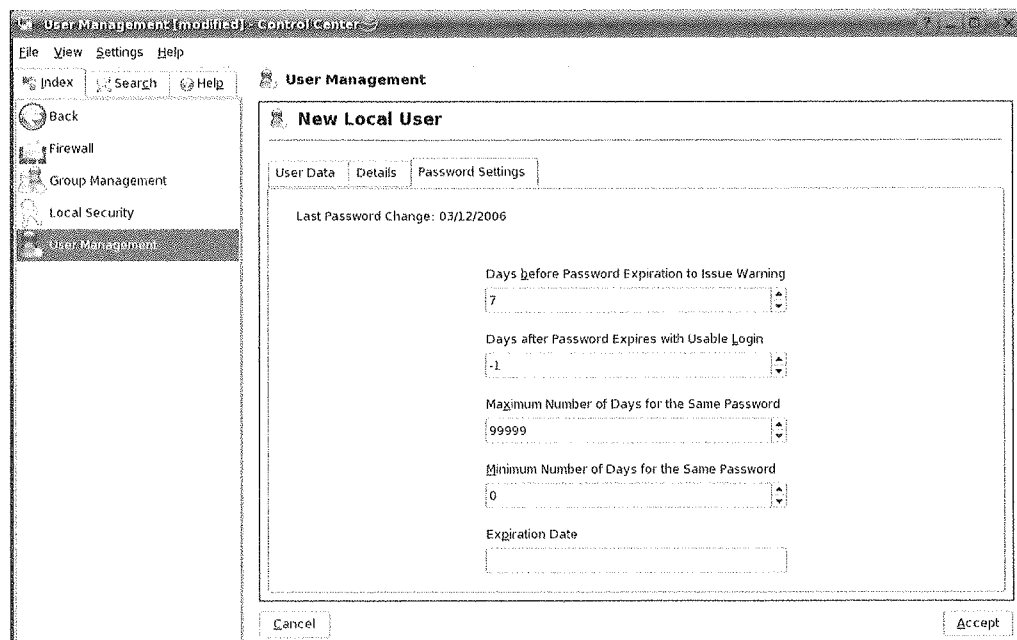




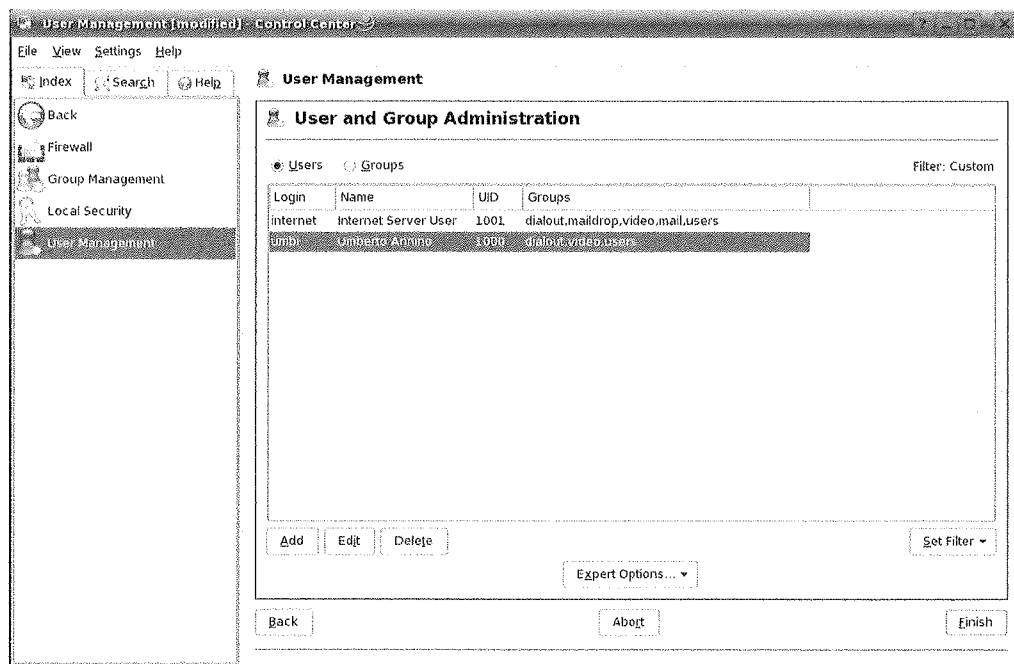
Bei den «Groups» kann der Benutzer in die entsprechenden Gruppen eingetragen werden. Folgende Einträge sind auszuwählen, um den Benutzer für die entsprechenden Dienste zu berechtigen (z. T. werden die Benutzer auch direkt im Dienst berechtigt, z. B. im verwendeten E-Mail-Server CommuniGate Pro, in diesem Fall ist der Benutzer unabhängig von diesen Einstellungen und wird direkt im entsprechenden Serverdienst konfiguriert):

- users
- ftp
- mail
- maildrop
- postfix
- www

Diese Auswahl erleichtert die Funktionstests der einzelnen Dienste. Je nach Sicherheitsvorkehrungen sind gewisse Gruppen wieder zu deaktivieren, nachdem die Funktionstests durchgeführt wurden. (Achtung: Gewisse Gruppenberechtigungen sind für Funktionen der Dienste weiterhin nötig; detailliert testen.)



Passwort-Einstellungen nötigenfalls anpassen und mit «Accept» bestätigen.



## 8.3 Apache Webserver

In diesem Kapitel wird Version 2.x des Web- und Anwendungsservers Apache vorgestellt. Neben Hinweisen zur Installation und Konfiguration von Apache finden Sie hier auch die Beschreibung einiger seiner Module.

### 8.3.1 Vorwort und Terminologie

Dieser Abschnitt definiert Begriffe, die in Verbindung mit dem Web, insbesondere aber in Zusammenhang mit Apache häufig genannt werden.

### 8.3.2 Webserver

Ein Webserver stellt auf Anfrage eines Clients Webseiten bereit. Beim Client kann es sich um einen Webbrowser wie Konqueror oder um jedes andere Gerät handeln, das eine Verbindung mit dem Internet herstellen kann. Diese Seiten können als Ganzes auf der Festplatte gespeichert werden (statische Seiten) oder als Ergebnis der Abfrage einer externen Entity, beispielsweise einer Datenbank oder eines Webdienstes, generiert werden (dynamische Seiten).

### 8.3.3 http

Die Kommunikation zwischen dem Client und dem Webserver erfolgt über http (Hypertext Transfer Protocol). Die aktuelle Version, HTTP 1.1, ist in RFC 2068 und dem zugehörigen Update RFC 2616 dokumentiert. Diese RFCs stehen unter <http://www.w3.org> zur Verfügung.

### 8.3.4 URLs

URL ist die Abkürzung von «Universal Resource Locator», einer eindeutigen Adresse im Internet. Clients verwenden URLs, beispielsweise <http://www.example.com/index.html>, zur Anforderung von Seiten von einem Server. Eine URL besteht aus folgenden Komponenten:

#### Protokoll

Gängige Protokolle:

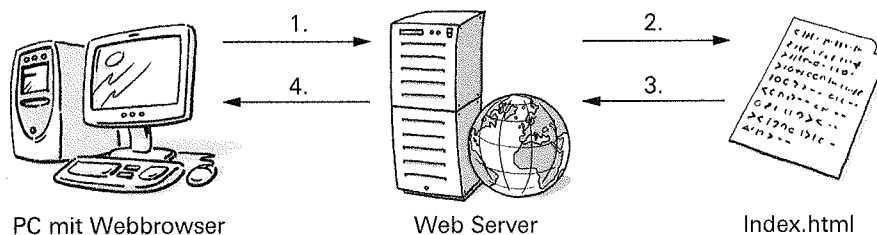
- **http://** (Das HTTP-Protokoll)
- **https://** (Sichere, verschlüsselte Version von HTTP)
- **ftp://** (FTP steht für «File Transfer Protocol» und ist ein Übertragungsprotokoll für das downloaden und Uploaden von Dateien.)

#### Domäne

In diesem Beispiel lautet die Domäne [www.beispiel.com](http://www.beispiel.com). Die Domäne ist der zu einer IP-Adresse gehörende Name. Die Domäne [www.beispiel.com](http://www.beispiel.com) lässt sich daher eindeutig einer IP-Adresse, zum Beispiel 123.456.789.1, zuordnen. Die Zahl hingegen ist die eindeutige Kennzeichnung des Computers, auf dem ein Webserver läuft. Die Zuordnung eines Domännennamens zu seiner IP-Adresse wird als *Namensauflösung* bezeichnet. Ein Domänenname ist in mehrere Komponenten unterteilt. Diese sind in unserem Beispiel: `www`, `beispiel` und `com`. Der letzte Teil des Domännennamens ist die Top-Level-Domäne (TLD). In unserem Beispiel ist `com` die TLD. Die TLD stellt die oberste Ebene des Namensauflösungsprozesses dar. TLDs können generisch sein (gTLDs), wie `com`, `org` und `net`, oder landesspezifisch (ccTLDs), wie `de` für Deutschland. Alle Teile einer Domäne gemeinsam werden als vollständig qualifizierter Domänenname (FQDN, Fully Qualified Domain Name) bezeichnet. Eine detaillierte Darstellung der Namensauflösung findet sich im Modul Serverdienste in Betrieb nehmen (123).

#### Ressource

In diesem Beispiel lautet die Ressource `index.html`. Dieser Teil gibt den vollständigen Pfad einer Ressource an. Bei der Ressource kann es sich, wie in diesem Beispiel, um eine Datei handeln. Es kann sich aber auch um ein CGI-Skript, eine JavaServer-Seite oder jede andere Ressource handeln. Der verantwortliche Internet-Mechanismus, beispielsweise das Domain Name System (DNS), leitet eine Anfrage nach der Domäne [www.beispiel.com](http://www.beispiel.com) an einen oder mehrere Computer weiter, auf denen sich die Ressource befindet (1.). Apache liefert daraufhin die betreffende Ressource, im Beispiel die Seite `index.html`, an den Client zurück (4.). In unserem Beispiel befindet sich die Datei im Top-Level-Verzeichnis. Ressourcen können sich aber auch in einem Unterverzeichnis befinden (z. B. in <http://www.example.com/linux/novell/suse>).



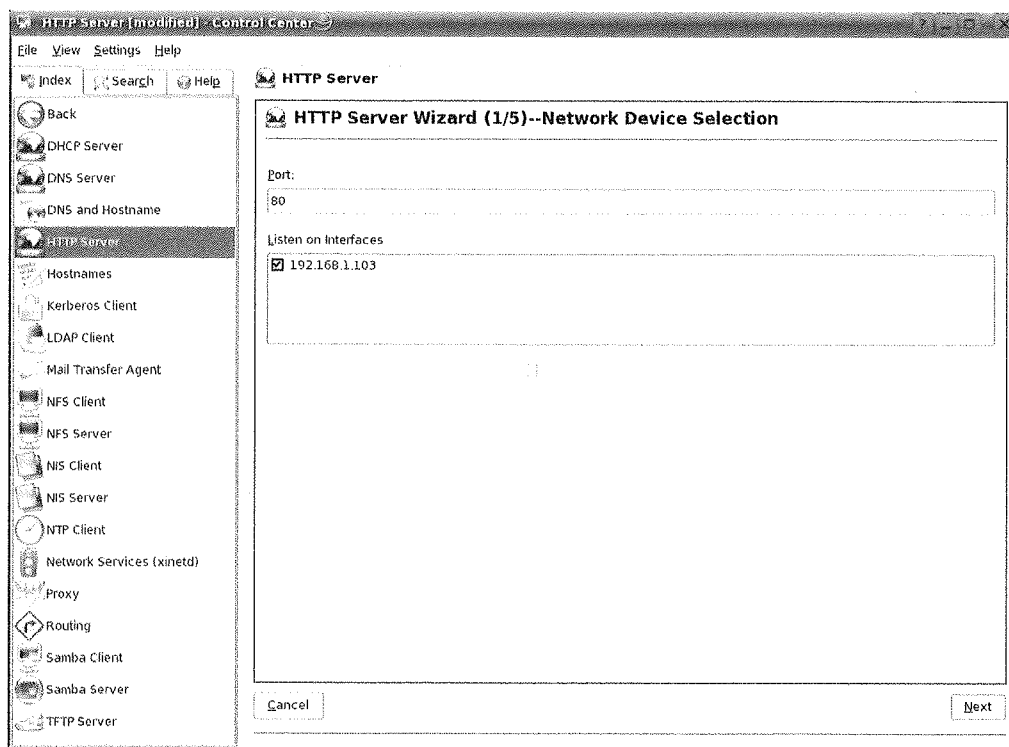
### 8.3.5 Direktive

Bei der Konfiguration von Apache wird der Begriff *Direktive* häufig als Synonym für «Konfigurationsoption» verwendet. Direktive ist ein spezieller, in Verbindung mit dem Apache-Webserver verwendeter technischer Begriff.

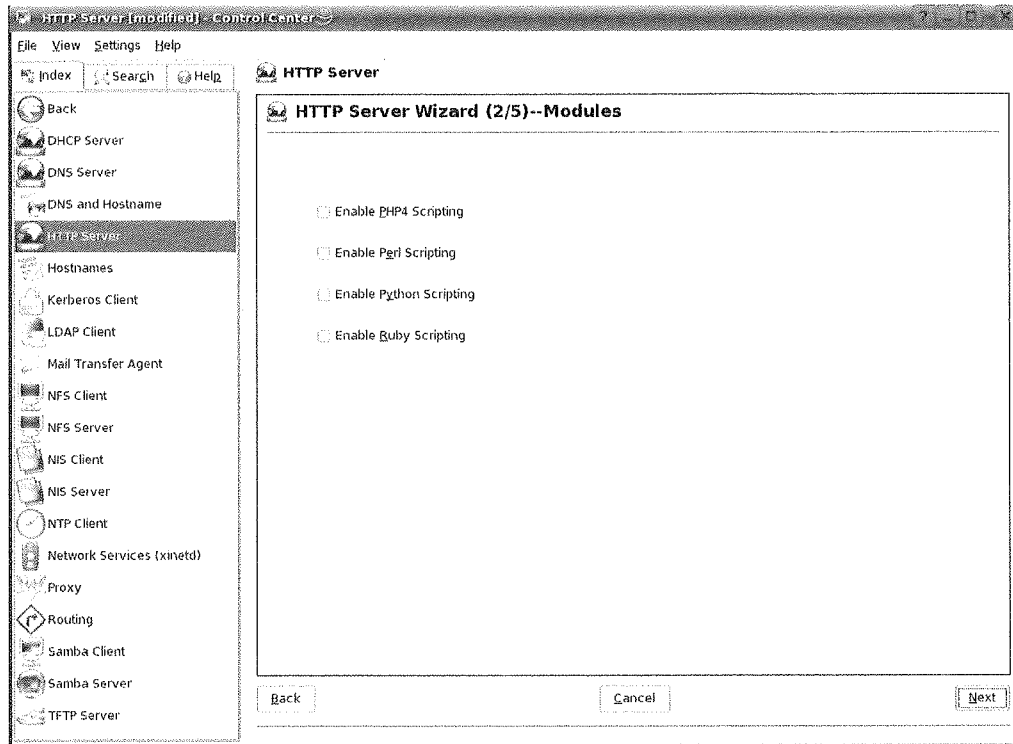
### 8.3.6 Installation

Apache läuft in SUSE Linux «Out of the Box», d. h. in der voreingestellten Standardkonfiguration. Wenn Sie die Anleitungen in diesem Kapitel befolgen, verfügen Sie innerhalb kürzester Zeit über einen funktionsfähigen Apache-Webserver. Zur Installation und Konfiguration von Apache müssen Sie `root`-Benutzer sein.

Über den Menüpunkt «Network Services» gelangt man zum Menü «http Server» für den Webserver:

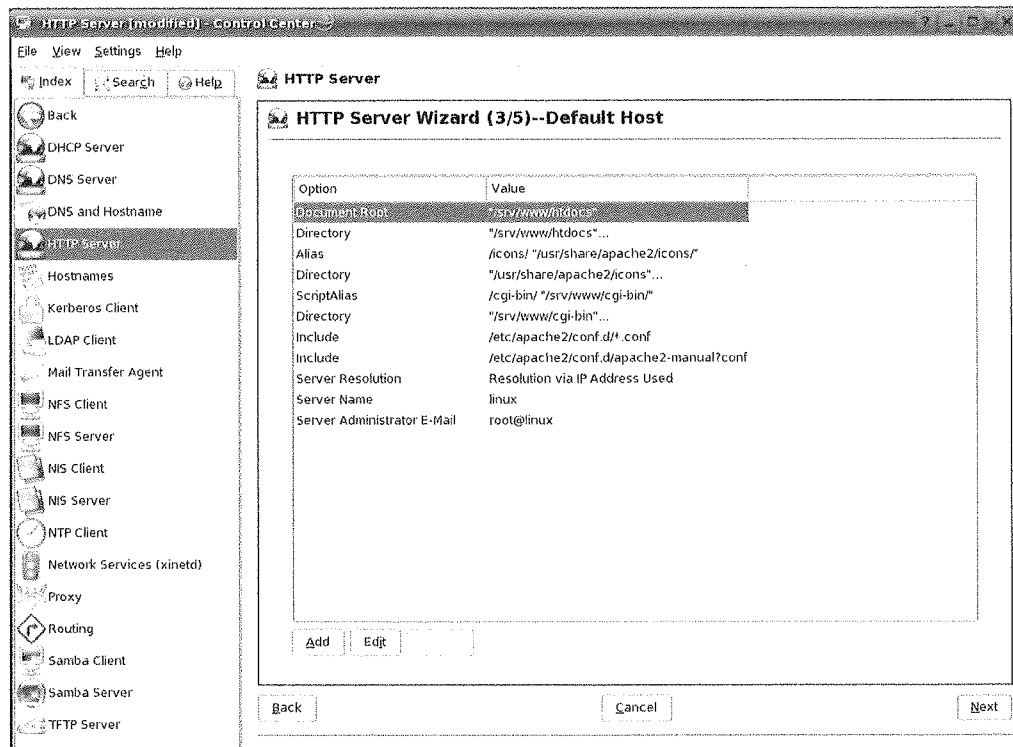


Der Port 80 und das Netzwerk-Interface, auf dem der `httpd` (daemon) läuft, sind bereits ausgewählt und können mit «Next» bestätigt werden. Die nächste Seite zeigt die Einstellungen für Scripting, welche nach Bedarf eingeschaltet werden können.

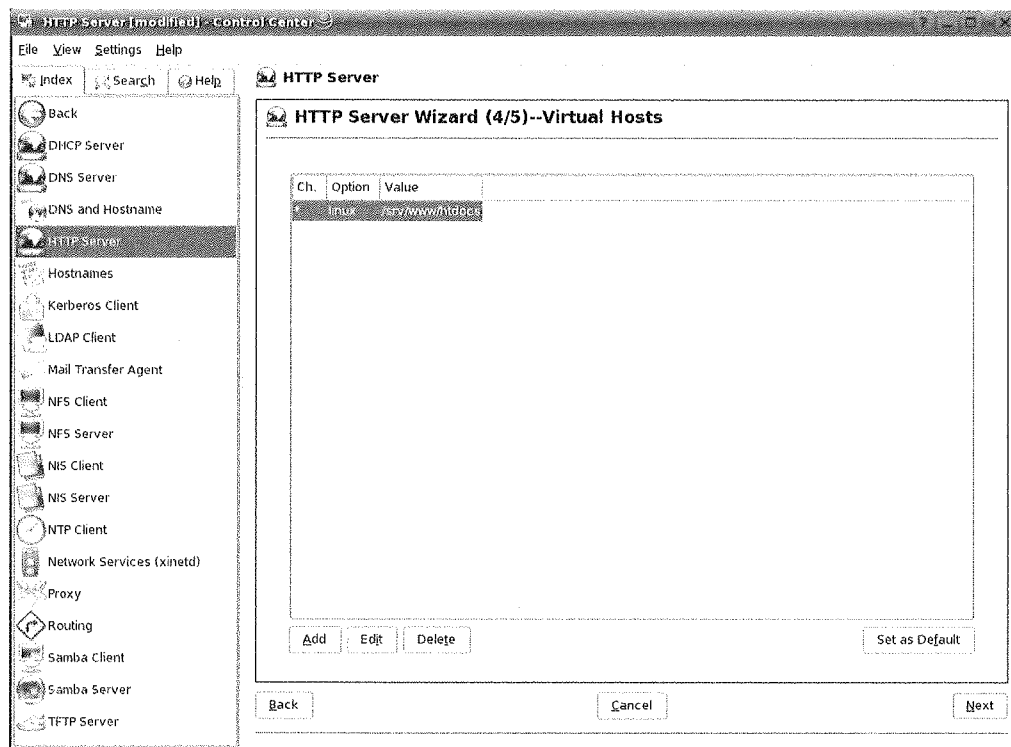


Die nachfolgende Übersicht zeigt die Default-Einstellungen sowie die Standardverzeichnisse:

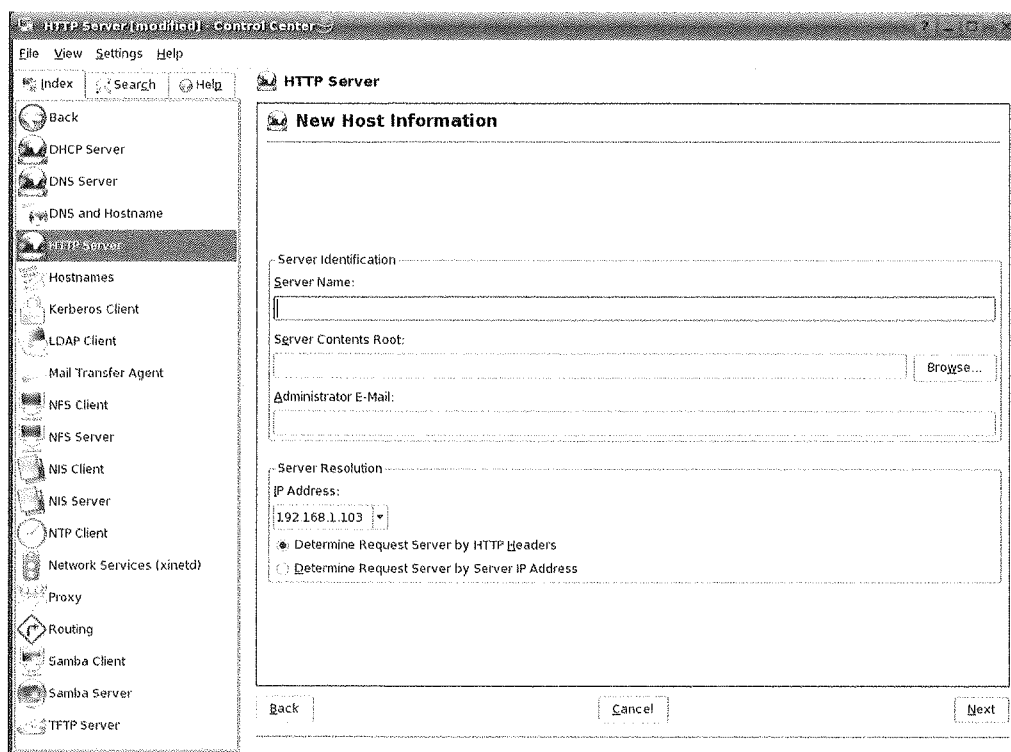
- DocumentRoot: Verzeichnis der HTML-Dokumente (wo standardmässig nach index.htm, index.html gesucht wird)
- Directory: Default-Ablageverzeichnis für Dokumente
- ServerName: Standardname des Servers (kann auf den Domainnamen geändert werden)
- Server Administrator E-Mail: E-Mail-Adresse des Administrators (für Fehlermeldungen etc.)



Mit «Next» gelangt man zur nächsten Übersicht, die «Virtual Hosts»:



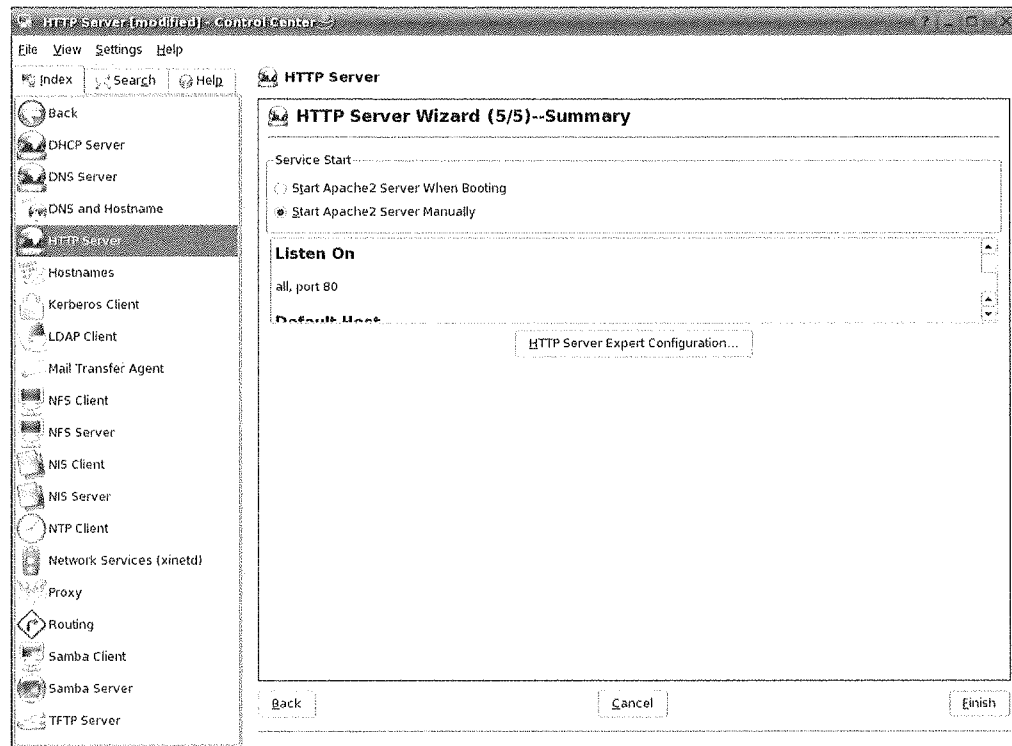
Bei Bedarf können weitere virtuelle Hosts (Hostnamen) zugefügt werden:



Die Server Resolution lautet dabei auf die gleiche IP-Adresse (ausser der Server verfügt über mehrere Netzwerkkarten). Beim Virtual Host sollte die Option «Determine Request Server by http Headers» eingeschaltet bleiben, da der Webserver ansonsten verwirrt ist, wenn der virtuelle Host aufgerufen wird (falls kein virtueller Host besteht oder der Webserver mit der IP-Adresse aufgerufen wird, wird der Default-Host angezeigt).



Auf der letzten Konfigurationsseite erfolgt die Überprüfung der Einstellungen sowie die Konfiguration des Startverhaltens des Webservers:



Die Auswahl «Service Start» ist auf «...when Booting» einzuschalten. Mit «Finish» beenden.

Der Webserver ist nun fertig konfiguriert und wird nach dem Reboot des Systems automatisch eingeschaltet.

### 8.3.7 Standarddateisystem und Anwendungslayout

SUSE Linux installiert die Dateien des Apache-Pakets in Standardverzeichnissen. Die Verzeichnisse der wichtigsten Dateien sind in den nachfolgenden Abschnitten aufgelistet.

#### Binärdateien

An die Namen der meisten ausführbaren Dateien von SUSE Linux Apache ist eine 2 angefügt. Dadurch lassen sich die Binärdateien paralleler Installationen von Apache 1.x und Apache 2.x leichter unterscheiden.

```
/usr/sbin/httpd2-prefork
```

Die eigentliche ausführbare Datei von Apache2.

```
/usr/sbin/apache2ctl
```

Steuerskript zum Starten und Beenden des Webservers, das vom Apache HTTPDProjekt bereitgestellt wird. Weitere Informationen erhalten Sie durch Ausführung von `/usr/sbin/apache2ctl help`.

```
/etc/init.d/apache2
```

Start- und Stoppskript, das Apache vollständig in die SUSE Linux-Installation integriert und den Webserver beim Hochfahren startet. Das Skript überprüft die Konfiguration vor dem Starten und Beenden des Servers und überschreibt den Speicherort der Konfiguration. Es

ermöglicht den Einschluss weiterer Konfigurationsdateien, das Laden von Modulen und sogar das Starten einer separaten Serverinstanz, ohne dass das Skript bearbeitet werden muss.

```
/usr/sbin/rcapache2
```

Ein bequemer Symlink für `/etc/init.d/apache2`, da `/etc/init.d/` standardmässig nicht im Pfad angegeben werden muss. Zum Starten von Apache brauchen Sie nur `rcapache2 start` einzugeben.

```
/usr/sbin/htpasswd2
```

Dienstprogramm zur Generierung verschlüsselter Passwörter für die `.htaccess`-basierte Authentifizierung. Informationen zur Verwendung des Tools finden Sie auf der Manualpage `htpasswd2(1)`.

### Konfigurationsdateien

Die meisten Konfigurationsdateien befinden sich in `/etc/apache2`.

```
/etc/apache2/httpd.conf
```

Die übergeordnete Konfigurationsdatei. An dieser Datei sollten Sie möglichst keine Änderungen vornehmen. Diese Datei legt in erster Linie die einzuschliessenden Konfigurationsdateien sowie globale Einstellungen fest.

```
/etc/apache2/*.conf
```

Einige externe Apache-Module legen ihre Konfigurationsdateien im Verzeichnis `/etc/apache2/` ab. Den Namen der Konfigurationsdateien wird in der Regel der Modulname vorangestellt (`mod_*.conf`).

```
/etc/apache2/conf.d/*
```

Verzeichnis für verschiedene andere Konfigurationsdateien aus bestimmten Paketen.

```
/etc/apache2/vhosts.d/*
```

Verzeichnis für die optionalen Konfigurationsdateien der virtuellen Hosts.

```
/etc/sysconfig/apache2
```

SUSE Linux-Konfigurationsdatei für Apache2. Diese Konfigurationsdatei enthält alle wichtigen Konfigurationsparameter für die Steuerung des Apache-Webservers. `/etc/sysconfig/apache2` wird von YaST zur Konfiguration von Apache verwendet. Die Datei kann auch manuell bearbeitet werden.

### Protokolldateien

In den folgenden Dateien zeichnet Apache verschiedene Informationen über seinen Laufzeitstatus auf:

```
/var/log/apache2/error_log
```

In dieser Datei protokolliert Apache die beim Starten und Herunterfahren ausgegebenen Meldungen sowie alle Laufzeitfehler.

```
/var/log/apache2/access_log
```

In dieser Datei werden alle Anforderungen an den Webserver protokolliert. Die Einträge in dieser Datei enthalten standardmässig Informationen über den Host und den Benutzeragenten, von dem die Anforderung stammt, sowie die zugehörige URI.

### Document Root (absoluter Pfad)

Das physische Verzeichnis `/srv/www/htdocs` ist der Standardspeicherort, aus dem Apache Webseiten ausgibt. Dieses Verzeichnis dient als «Root-Verzeichnis» für Client-Anforderungen. Wenn Sie Webseiten mit Apache veröffentlichen möchten, speichern Sie die Dateien hierarchisch in bzw. unter diesem Verzeichnis. Eine URL wie <http://www.beispiel.com/index.html> verweist in der Apache-Standardkonfiguration von SUSE Linux auf den Pfad `/srv/www/htdocs/index.html` einer Domäne namens `beispiel.com`.

Das Standarddokument ist `index.html`.

### 8.3.8 Aktivieren, Starten und Beenden von Apache

---

Zur Aktivierung des Apache-Webrowsers beim Hochfahren des Computers verwenden Sie den Runlevel-Editor von YaST. Um diesen zu starten, wählen Sie in YaST **System** → **Runlevel-Editor** aus. Navigieren Sie danach zum Eintrag **apache2**. Wählen Sie **Aktivieren** aus, wenn Apache beim Hochfahren des Computers automatisch gestartet werden soll. Erfahrene Benutzer können diese Einstellung auch mit dem Befehlszeilentool `chkconfig` vornehmen: `/sbin/chkconfig -a apache2`.

Zum Starten oder Beenden von Apache verwenden Sie das Skript `/usr/sbin/rcapache2` als Root-Benutzer. `/usr/sbin/rcapache2` akzeptiert zum Starten und Beenden des Apache-Webrowsers folgende Parameter:

`start`

Startet den Apache-Webserver.

`startssl`

Startet den Apache-Webserver mit SSL-Unterstützung.

`stop`

Beendet den Apache-Webserver.

`configtest`

Testet die Apache-Konfiguration, ohne die Stop-, Start- oder Neustartvorgänge tatsächlich auszuführen. Da dieser Test bei jedem Start, beim Laden oder bei einem Neustart des Servers automatisch ausgeführt wird, ist eine manuelle Ausführung des Tests in der Regel nicht erforderlich.

`restart`

Beendet den Webserver und startet ihn neu.

`try-restart`

Startet den Webserver neu, sofern er bereits läuft.

`restart-hup`

Startet den Webserver mittels `SIGHUP`-Signal neu. Dieses Signal wird normalerweise nicht verwendet.

`graceful` und `reload`

Beendet den Webserver erst, nachdem alle durch Forking erstellten Apache-Prozesse aufgefordert wurden, ihre Anforderungen vor dem Herunterfahren zu Ende zu führen. Anstelle der beendeten Prozesse werden neue Prozesse gestartet. Dies führt zu einem vollständigen «Neustart» von Apache.

`status`

Überprüft den Laufzeit-Status des Apache-Webservers.

Beispielausgabe beim Starten und Beenden von Apache

```
tux@sun # rcapache2 status
Checking for httpd2: unused

tux@sun # rcapache2 configtest
Syntax OK

tux@sun # rcapache2 start
Starting httpd2 (prefork) done

tux@sun # rcapache2 status
Checking for httpd2: running

tux@sun # rcapache2 graceful
Reload httpd2 (graceful restart) done

tux@sun # rcapache2 status
Checking for httpd2: running
```

Eine fehlerhafte Konfigurationsdatei kann dazu führen, dass Apache gar nicht oder nicht korrekt gestartet wird. Falls der Webserver gar nicht gestartet wird, erhalten Sie unter Umständen nicht einmal Fehlermeldungen. Überprüfen Sie bei jedem Start oder Neustart das Hauptfehlerprotokoll.

### 8.3.9 Manuelle Konfiguration von Apache

---

Wenn Sie den Apache-Webserver manuell konfigurieren möchten, müssen Sie die Klartext-Konfigurationsdateien als `Root`-Benutzer bearbeiten.

`/etc/sysconfig/apache2`

`/etc/sysconfig/apache2` steuert einige globale Einstellungen von Apache, beispielsweise die zu ladenden Module, die einzuschliessenden Konfigurationsdateien, die beim Serverstart zu verwendenden Flags sowie Flags, die der Befehlszeile hinzugefügt werden sollen. Die Konfigurationsoptionen dieser Datei sind hinreichend dokumentiert und werden daher an dieser Stelle nicht näher erläutert. Für die Konfigurationsanforderungen eines typischen Webservers dürften die Einstellungen der Datei `/etc/sysconfig/apache2` ausreichen.

**WICHTIG: Beim Serverstart automatisch erstellte Dateien**

`/etc/sysconfig/apache2` erstellt bzw. bearbeitet die folgenden Dateien automatisch bei einem Start oder Neustart des Webservers.

- `/etc/apache2/sysconfig.d/loadmodule.conf`: Während der Laufzeit geladene Module
- `/etc/apache2/sysconfig.d/global.conf`: Serverweite, allgemeine Einstellungen
- `/etc/apache2/sysconfig.d/include.conf`: Liste der eingeschlossenen Konfigurationsdateien

Diese Dateien dürfen nicht manuell bearbeitet werden. Ändern Sie stattdessen die entsprechenden Einstellungen in `/etc/sysconfig/apache2`.

Für spezielle Konfigurationen, besonders, wenn Sie Änderungen an der manuellen Konfiguration virtueller Hosts, der globalen Umgebung oder des Hauptservers vornehmen möchten, verweisen wir Sie auf die Dateien in `/etc/apache2/*`.

### Apache-Direktiven in `/etc/apache2/httpd.conf`: Global Environment

SUSE Linux verwendet `/etc/apache2/httpd.conf` als zentrale Referenz für andere Konfigurationsdateien. Bearbeiten Sie diese Datei nur, wenn Sie Funktionen aktivieren möchten, die in `/etc/sysconfig/apache2` nicht zur Verfügung stehen. Die Direktiven im Abschnitt *Global Environment* (globale Umgebung) der Datei `httpd.conf` wirken sich auf die gesamte Funktionalität von Apache aus.

Die folgenden Abschnitte befassen sich mit einigen der Direktiven, die nicht in YaST zur Verfügung stehen. Kerndirektiven wie `DocumentRoot` sind sowohl für `Global Environment` als auch für `VirtualHost` absolut notwendig.

Die folgenden Parameter und Direktiven sind nach logischem Zusammenhang und Bedeutung für die Konfiguration sortiert. Sie sollten in `/etc/apache2/httpd.conf` festgelegt werden.

```
LoadModule Modul_ID /Pfad/des/Moduls
```

Die `LoadModule`-Direktive bestimmt, welche Apache-Module während der Laufzeit geladen werden. `Modul_ID` ist der in seiner Dokumentation angegebene Name des Moduls. `/Pfad/des/Moduls` ist der absolute oder relative Pfad der Moduldatei.

#### Direktive `LoadModule`

```
LoadModule rewrite_module /usr/lib/apache2-prefork/mod_rewrite.so
```

In SUSE Linux sind keine direkten `LoadModule`-Anweisungen erforderlich. Stattdessen kann `APACHE_MODULE` in `/etc/sysconfig/apache2` verwendet werden.

```
MaxClients Zahl
```

Die maximale Anzahl an Clients, die Apache gleichzeitig bedienen kann. Die maximale Client-Anzahl muss einerseits die Anzahl der erwarteten, gleichzeitigen Anforderungen an die Website berücksichtigen, andererseits aber auch den zur Verfügung stehenden physischen RAM-Speicher. Dieser muss für alle Prozesse ausreichend ausgelegt sein.

```
Timeout Sekunden
```

Gibt die Dauer in Sekunden an, bevor Apache für eine Anforderung eine Zeitüberschreitung meldet.

## Apache-Direktiven in /etc/apache2/httpd.conf: Main Server

Die Direktiven im Abschnitt `Main Server` treten in Kraft, wenn Client-Anforderungen von keinem virtuellen Host (`VirtualHost`) beantwortet werden und daher von einem Standard- bzw. Hauptserver bearbeitet werden müssen. Darüber hinaus handelt es sich bei den in diesem Abschnitt angegebenen Parametern um die Standardwerte aller konfigurierten virtuellen Hosts. Die Direktiven des Abschnitts `Main Server` können also auch im `VirtualHost`-Kontext festgelegt werden. In diesem Fall überschreiben sie die Standardwerte.

`DirectoryIndex` Dateinamen

Legt fest, nach welchen Dateien Apache suchen soll, wenn in einer URL die Dateiangabe fehlt. Die Standardeinstellung ist `index.html`. Fordert ein Client beispielsweise die URL <http://www.beispiel.com/foo/> an und das Verzeichnis `foo` enthält eine Datei namens `index.html`, so gibt Apache diese Seite dem Client zurück. Bei der Angabe mehrerer Dateien müssen Sie die einzelnen Dateien jeweils durch ein Leerzeichen trennen.

### DirectoryIndex

```
DirectoryIndex index.html index.shtml start.php begin.pl
```

`AllowOverride` All | None | Option

Diese Direktive kann nur innerhalb einer `<Directory></Directory>`-Deklaration verwendet werden.

`AllowOverride` gibt an, welche Zugriffs- und Anzeigeoptionen eine `.htaccess`-Datei (oder andere in `AccessFileName` angegebene Dateien) überschreiben kann.

Mögliche Werte:

- All (Alle Optionen können von einer `.htaccess`-Datei überschrieben werden.)
- None (Keine Optionen können von einer `.htaccess`-Datei überschrieben werden.)
- AuthConfig (Verzeichnisse können mittels einer `.htaccess`-Datei durch ein Passwort geschützt werden.)
- FileInfo (Ermöglicht die Verwendung von Direktiven, die die Dokumenttypen in einer `.htaccess`-Datei steuern. Ein typisches Beispiel ist die Konfiguration von benutzerdefinierten Fehlerseiten mithilfe von `ErrorDocument` (siehe <http://httpd.apache.org/docs-2.0/mod/core.html#errordocument>).
- Indexes (Falls kein `DirectoryIndex`-Dokument gefunden wird, erlaubt dieser Parameter Apache die Steuerung der Anzeige von Verzeichnisinhalten.)
- Limit (Steuert den Client-Zugriff auf ein Verzeichnis bzw. auf bestimmte Dateien. Zu diesem Zweck werden in einer `.htaccess`-Datei die Direktiven `Allow`, `Deny` und `Order` verwendet. Eine Beschreibung dieser Direktiven finden Sie in der Dokumentation des Zugriffsmoduls ([http://httpd.apache.org/docs-2.0/mod/mod\\_access.html](http://httpd.apache.org/docs-2.0/mod/mod_access.html)).
- Options (Lässt die Verwendung der Direktiven `Options` und `XBitHack` in einer `.htaccess`-Datei zu. Die Direktive `Options` (<http://httpd.apache.org/docs-2.0/mod/core.html#options>) steuert, welche Serverfunktionen in einem bestimmten Verzeichnis verfügbar sind. Die Direktive `XBitHack` ([http://httpd.apache.org/docs-2.0/mod/mod\\_include.html#xbithack](http://httpd.apache.org/docs-2.0/mod/mod_include.html#xbithack)) lässt für Dateien mit `Execute`-Bit das Parsen als SSI (server side include) zu

**WICHTIG:** Diese Einstellungen werden rekursiv auf das aktuelle Verzeichnis und seine Unterverzeichnisse angewandt. Die Optionen können mit Ausnahme von `All` und `None` kombiniert werden, müssen dann aber durch ein Leerzeichen getrennt sein.

**Direktive** AllowOverride

```
<Directory /srv/www/htdocs>
AllowOverride None
</Directory>
<Directory /srv/www/htdocs/project>
AllowOverride All
</Directory>
<Directory /srv/www/htdocs/project/webapp>
AllowOverride Indexes Limit AuthConfig
</Directory>
```

**AccessFileName** Dateinamen

**AccessFileName** legt die Namen der Dateien fest, die globale Zugriffsberechtigungen und andere Verzeichniseinstellungen überschreiben können. Die Standardeinstellung ist `.htaccess`. Bei der Angabe mehrerer Dateien müssen Sie die einzelnen Dateien jeweils durch ein Leerzeichen trennen.

**Direktive** AccessFileName

```
AccessFileName .htaccess .acl permission.txt
```

```
ErrorLog Datei | "|Befehl"
```

Gibt den Namen der Datei an, in der Apache Fehlermeldungen aufzeichnet. Als Alternative können Sie für die Protokollierung auch einen Befehl oder ein Skript angeben. Die Standardeinstellung ist `/var/log/apache2/error_log`.

**Direktive** ErrorLog

```
ErrorLog /var/log/apache2/error_log
ErrorLog "|/path/to/script"
```

**LogLevel** Stufe

Legt die Ausführlichkeit der aufgezeichneten Fehlermeldungen fest. `Stufe` kann folgende Werte haben (wobei nachfolgende Liste in aufsteigender Reihenfolge nach Ausführlichkeit bzw. in absteigender Reihenfolge nach Schweregrad der Meldung sortiert ist).

- emerg
- alert
- crit
- error
- warn
- notice
- info
- debug

Die Standardeinstellung `warn` empfiehlt sich für alltägliche Vorgänge. Zur Problembehebung liefern `info` und `debug` hilfreiche Informationen.

**Direktive** LogLevel

```
LogLevel debug
```

## Apache-Direktiven in `/etc/apache2/httpd.conf`: Virtual Hosts

Wenn Sie mehrere Domänen oder Hostnamen auf einem physischen Gerät einrichten möchten, benötigen Sie `VirtualHost`-Container. Diese werden in den Konfigurationsabschnitten `Virtual Hosts` festgelegt.

### 8.3.10 Virtuelle Hosts

---

Virtueller Host bezieht sich auf die Fähigkeit von Apache, mehrere URIs (Universal Resource Identifiers) vom gleichen physischen Computer aus bedienen zu können. In anderen Worten: Mehrere Domänen wie [www.beispiel.com](http://www.beispiel.com) und [www.beispiel.net](http://www.beispiel.net) können von einem einzigen Webserver auf einem physischen Computer ausgeführt werden. Virtuelle Hosts werden häufig eingesetzt, um den Verwaltungsaufwand (nur ein Webserver muss verwaltet werden) und die Hardware-Kosten (für die einzelnen Domänen ist kein dedizierter Server erforderlich) zu sparen. Virtuelle Hosts können auf Namen, IP-Adressen oder Anschlüssen basieren.

Virtuelle Hosts können mit YaST oder manuell im Abschnitt `Virtual Host` der Datei `httpd.conf` konfiguriert werden.

In SUSE Linux ist Apache unter `/etc/apache2/vhosts.d/` standardmässig für eine Konfigurationsdatei pro virtuellen Host vorbereitet. Dieses Verzeichnis enthält auch eine allgemeine Vorlage für virtuelle Hosts (`vhost.template`). Die Konfiguration virtueller Hosts kann aber auch an anderer Stelle vorgenommen werden, zum Beispiel in einer Datei, die anschliessend der Konfiguration hinzugefügt wird.

**WICHTIG:** Es empfiehlt sich, die virtuelle Hostkonfiguration mit `httpd2 -s` zu überprüfen. Dieser Befehl gibt die virtuellen Hosteinstellungen so aus, wie sie von Apache interpretiert werden. Sie stellen damit sicher, dass Sie das gewünschte Ergebnis erhalten. Wenn Sie Apache mit Flags wie `-DSSL` verwenden, müssen Sie die gleichen Flags auch beim Testen verwenden. Zum Beispiel: `httpd2 -s -DSSL`.

### 8.3.11 Namensbasierte virtuelle Hosts

---

Namensbasierte virtuelle Hosts können an jeder IP-Adresse mehrere Websites bedienen. Apache verwendet das Hostfeld im vom Client übersandten HTTP-Header, um die Anforderung mit einem übereinstimmenden `ServerName`-Eintrag der virtuellen Hostdeklarationen zu verbinden. Wird kein übereinstimmender `ServerName` gefunden, dann wird der erste angegebene `VirtualHost` als Standard verwendet. Der `Virtual Host`-Bereich einer Apache-Konfiguration beginnt mit `NameVirtualHost`.

`NameVirtualHost`

`NameVirtualHost` teilt dem Apache-Webserver mit, welche IP-Adresse (und optional welcher Port) auf Client-Anforderungen überwacht werden soll, die den Domännennamen im HTTP-Header enthalten. Als erstes Argument kann der vollständig qualifizierte Domänenname eingegeben werden – empfohlen wird aber die IP-Adresse. Das zweite, optionale Argument ist der Port. Dieser ist standardmässig Port 80 und wird mit der `Listen`-Direktive konfiguriert.

Sowohl für die IP-Adresse als auch für die Port-Nummer kann ein Platzhalterzeichen (\*) eingegeben werden. In diesem Fall werden die Anforderungen an allen Schnittstellen empfangen. IPv6-Adressen müssen in eckigen Klammern eingeschlossen sein.



### Namensbasierte VirtualHost-Einträge

```
#
NameVirtualHost IP-Adresse[:Port]
NameVirtualHost 192.168.1.100:80
NameVirtualHost 192.168.1.100
NameVirtualHost *:80
NameVirtualHost * NameVirtualHost [2002:c0a8:164::]:80
```

<VirtualHost></VirtualHost> im namensbasierten Kontext

Der <VirtualHost></VirtualHost>-Block enthält die Informationen zu einer bestimmten Domäne. Wenn Apache eine Client-Anforderung für einen definierten VirtualHost empfängt, verwendet es die in diesem Bereich angegebenen Direktiven. In diesem Bereich kann jede Apache-Direktive verwendet werden, die im VirtualHost-Kontext zugelassen ist. In einer namensbasierten virtuellen Hostkonfiguration sind für das VirtualHost-Anfangstag die folgenden Argumente zulässig:

- IP-Adresse (oder vollständig qualifizierter Domänenname). Die Adresse muss zuvor mit der NameVirtualHost-Direktive deklariert worden sein.
- Optionale Port-Nummer. Diese muss zuvor mit der NameVirtualHost-Direktive deklariert worden sein. Anstelle der IP-Adresse wird auch ein Platzhalterzeichen (\*) akzeptiert. Diese Syntax ist allerdings nur in Verbindung mit einem Platzhalter in NameVirtualHost \* zulässig. IPv6-Adressen müssen in eckige Klammern eingeschlossen werden.

### Namensbasierte VirtualHost-Direktiven

```
<VirtualHost 192.168.1.100:80>
ServerName www.beispiel.com
DocumentRoot /srv/www/htdocs/beispiel.com
ServerAdmin webmaster@beispiel.com
ErrorLog /var/log/apache2/www.beispiel.com-error_log
CustomLog /var/log/apache2/www.beispiel.com-access_log common
</VirtualHost>

<VirtualHost 192.168.1.100:80>
ServerName www.beispiel.net
DocumentRoot /srv/www/htdocs/beispiel.net
ServerAdmin webmaster@beispiel.net
ErrorLog /var/log/apache2/www.beispiel.net-error_log
CustomLog /var/log/apache2/www.beispiel.net-access_log common
</VirtualHost>

<VirtualHost [2002:c0a8:164::]>
# 2002:c0a8:164:: is the IPv6 equivalent to 192.168.1.100
ServerName www.beispiel.org
```

```
DocumentRoot /srv/www/htdocs/beispiel.org  
ServerAdmin webmaster@beispiel.org  
ErrorLog /var/log/apache2/www.beispiel.org-error_log  
CustomLog /var/log/apache2/www.beispiel.org-access_log common  
</VirtualHost>
```

In diesem Beispiel befinden sich die Domänen [www.beispiel.com](http://www.beispiel.com) und [www.beispiel.net](http://www.beispiel.net) auf dem gleichen Computer mit der IP-Adresse 192.168.1.100. Der erste angegebene `VirtualHost` ist der Standardhost für alle ankommenden Anforderungen auf dem Webserver.

In den Direktiven `ErrorLog` und `CustomLog` (siehe [http://httpd.apache.org/docs-2.0/mod/mod\\_log\\_config.html#customlog](http://httpd.apache.org/docs-2.0/mod/mod_log_config.html#customlog)) muss der Domänenname nicht angegeben sein. Sie können hier einen beliebigen Namen eingeben.

### 8.3.12 IP-basierte virtuelle Hosts

---

Bei dieser alternativen virtuellen Hostkonfiguration werden auf einem Computer mehrere IPs eingerichtet. Auf einer Apache-Instanz befinden sich mehrere Domänen, denen jeweils eine eigene IP zugewiesen ist.

#### WICHTIG: IP-Adressen und IP-basierte virtuelle Hosts

Auf dem physischen Server muss für jeden IP-basierten virtuellen Host eine eigene IP-Adresse eingerichtet sein. Falls der Computer nicht über die entsprechende Anzahl an Netzwerkkarten verfügt, können auch virtuelle Netzwerkschnittstellen verwendet werden (IP-Aliasing).

#### Konfigurieren von IP-Aliasing

Damit Apache mehrere IPs handhaben kann, muss der physische Computer Anfragen für mehrere IPs akzeptieren. Dies wird auch als Multi-IP-Hosting bezeichnet. Zusätzlich muss im Kernel IP-Aliasing aktiviert sein. Dies ist die Standardeinstellung in SUSE Linux.

Wenn der Kernel für IP-Aliasing konfiguriert ist, können Sie mit den Befehlen `ifconfig` und `route` weitere IPs auf dem Host einrichten. Für diese Befehle sind `Root`-Berechtigungen erforderlich. Im folgenden Beispiel wird davon ausgegangen, dass auf dem Host bereits die IP-Adresse 192.168.0.10 eingerichtet und dem Netzwerkgerät `eth0` zugewiesen ist. Mit dem Befehl `ifconfig` können Sie die IP des Host anzeigen. Weitere IP-Adressen können mit den folgenden Befehlen hinzugefügt werden:

```
ip addr add 192.168.0.20/24 dev eth0
```

```
ip addr add 192.168.0.30/24 dev eth0
```

Alle diese IP-Adressen werden dem gleichen physischen Netzwerkgerät, nämlich `eth0`, zugewiesen.

#### <VirtualHost></VirtualHost> im IP-basierten Kontext

Apache kann konfiguriert werden, sobald auf dem System IP-Aliasing eingerichtet ist (oder auf dem Host eine ausreichende Anzahl an Netzwerkkarten zur Verfügung steht). Für jeden virtuellen Server wird ein eigener `VirtualHost`-Block benötigt. Das folgende Beispiel zeigt

Apache auf einem Computer mit der IP 192.168.1.10, auf dem sich zwei Domänen mit den zusätzlichen IPs 192.168.0.20 und 192.168.0.30 befinden. Dieses spezielle Beispiel funktioniert nur in einem privaten Netzwerk, da IPs von 192.168.0.0 bis 192.168.0.255 nicht in das öffentliche Internet weitergeleitet werden.

#### IP-basierte VirtualHost-Direktiven

```
<VirtualHost 192.168.0.20>
ServerName www.beispiel.com
DocumentRoot /srv/www/htdocs/beispiel.com
ServerAdmin webmaster@beispiel.com
ErrorLog /var/log/apache2/www.beispiel.com-error_log
CustomLog /var/log/apache2/www.beispiel.com-access_log common
</VirtualHost>
<VirtualHost 192.168.0.30>
ServerName www.beispiel.net
DocumentRoot /srv/www/htdocs/beispiel.net
ServerAdmin tux@beispiel.net
ErrorLog /var/log/apache2/www.beispiel.net-error_log
CustomLog /var/log/apache2/www.beispiel.net-access_log common
</VirtualHost>
```

In diesem Beispiel sind nur für die beiden zusätzlichen IP-Adressen (also nicht für 192.168.0.10) VirtualHost-Direktiven angegeben. Sollte für 192.168.0.10 auch eine Listen-Direktive konfiguriert sein, müsste ein eigener IP-basierter Host für die HTTP-Anforderungen an diese Schnittstelle eingerichtet werden. Anderenfalls fänden die Direktiven aus dem Abschnitt `Main Server` der Datei `/etc/apache2/httpd.conf` Anwendung.

### 8.3.13 Apache-Module

Die Apache-Software ist modulartig aufgebaut. Sämtliche Funktionen mit Ausnahme der wichtigsten Aufgaben werden in Modulen zur Verfügung gestellt. Dies geht sogar so weit, dass selbst HTTP durch ein Modul verarbeitet wird (`http_core`). Apache-Module können bei der Entwicklung in die Apache-Binaries kompiliert oder während der Laufzeit dynamisch geladen werden.

Apache wird in SUSE Linux mit den folgenden, im `apache2-RPM` sofort verfügbaren Modulen ausgeliefert (das Präfix «`mod_`» wurde in folgender Aufstellung weggelassen):

`access`, `actions`, `alias`, `asis`, `auth`, `auth_anon`, `auth_dbm`, `auth_digest`, `auth_ldap`, `autoindex`, `cache`, `case_filter`, `case_filter_in`, `cern_meta`, `cgi`, `charset_lite`, `dav`, `dav_fs`, `deflate`, `dir`, `disk_cache`, `dumpio`, `echo`, `env`, `expires`, `ext_filter`, `file_cache`, `headers`, `imap`, `include`, `info`, `ldap`, `log_config`, `log_forensic`, `logio`, `mem_cache`, `mime`, `mime_magic`, `negotiation`, `proxy`, `proxy_connect`, `proxy_ftp`, `proxy_http`, `rewrite`, `setenvif`, `speling`, `ssl`, `status`, `suexec`, `unique_id`, `userdir`, `usertrack` und `vhost_alias`.

Darüber hinaus stellt SUSE Linux folgende Apache-Module als RPM-Pakete bereit, die gesondert installiert werden müssen: `apache2-mod_auth_mysql`, `apache2-mod_fastcgi`,

apache2-mod\_macro, apache2-mod\_murka, apache2-mod\_perl, apache2-mod\_php4, apache2-mod\_php5, apache2-mod\_python und apache2-mod\_ruby.

Auf einige dieser Module wird in diesem Abschnitt näher eingegangen. Eine Beschreibung der übrigen, in der Basisausstattung enthaltenen Module, finden Sie auf der Apache-Website unter <http://httpd.apache.org/docs-2.0/mod/>.

Module von Drittanbietern werden unter <http://modules.apache.org/> beschrieben.

Apache-Module lassen sich in drei Kategorien einteilen: Basismodule, Erweiterungsmodule und externe Module.

### Basismodule

Basismodule sind standardmässig in Apache enthalten. Sie stehen in jedem Fall zur Verfügung, es sei denn, sie wurden bei der Entwicklung ausdrücklich weggelassen. In Apache von SUSE Linux sind nur die grundlegenden Basismodule kompiliert. Alle anderen Basismodule stehen jedoch als *shared objects* zur Verfügung: wenn sie nicht in der `/usr/sbin/httpd2-Binary` enthalten sind, können sie während der Laufzeit über `APACHE_MODULES` in `/etc/sysconfig/apache2` hinzugefügt werden.

### Serverseitige Includes (Einschlüsse) mit `mod_include`

`mod_include` ist ein Mittel zur Dateiverarbeitung, bevor Daten an den Client gesendet werden. In der Regel wird `mod_include` zum Einschliessen von Dateien in ein Dokument verwendet, die vor Erreichen des Clients als HTML geparkt werden. Aus diesem Grund werden diese Einschlüsse auch als serverseitige Includes (SSIs) bezeichnet. Bei SSIs werden spezielle Befehle auf dem Server ausgeführt, die von formatierten SGML-Kommentaren initiiert werden. Diese SGML-Befehle haben die folgende Syntax:

```
<!--#Element Attribut=Wert -->
```

Eine Liste der Element- und Attribut-Werte finden Sie in der Dokumentation von `mod_include` unter [http://httpd.apache.org/docs-2.0/mod/mod\\_include.html](http://httpd.apache.org/docs-2.0/mod/mod_include.html).

Wenn Sie `mod_include` in SUSE Linux verwenden möchten, fügen Sie `include` zu `APACHE_MODULES` in `/etc/sysconfig/apache2` hinzu oder verwenden Sie YaST.

### Common Gateway Interface: `mod_cgi`

`mod_cgi` befähigt Apache, Inhalte bereitzustellen, die in externen Common Gateway Interface (CGI)-Programmen oder -Skripten erstellt wurden. `mod_cgi` agiert somit als Instanz zwischen der Programmiersprache auf dem physischen Gerät und dem Apache-Webserver. Theoretisch können CGI-Skripte in jeder beliebigen Programmiersprache geschrieben sein. Üblich sind aber Sprachen wie Perl oder C. `mod_cgi` ist die gängigste Methode, dynamischen Inhalt in eine Website einzuschliessen. Die CGI-Programmierung unterscheidet sich von der herkömmlichen Programmierung insoweit, als CGI-Programme und -Skripte den MIME-Typ `Content-type: text/html` hervorbringen müssen, um eine HTML-Ausgabe zu produzieren.

#### Ein einfaches CGI-Skript in Perl

```
#!/Pfad/zu/perl

print "Content-type: text/html\n\n";

print "Hello, World.";
```

Der Unterschied zwischen Modulen, die an eine spezielle Programmiersprache gebunden sind (z. B. `mod_php5`), und `mod_cgi` liegt in der Möglichkeit, `mod_cgi` mit `mod_suexec` zu kombinieren. Durch diese Kombinationsfähigkeit können CGI-Skripts mit einer bestimmten Benutzer-ID ausgeführt werden. Skripts, die nur `mod_cgi` oder `mod_php5` verwenden, werden in der Regel mit der Benutzer-ID des Apache-Benutzers ausgeführt (Standardeinstellung in SUSE Linux: `wwwrun`). Module, die für eine bestimmte Programmiersprache (wie `mod_php5` oder `mod_ruby`) entwickelt wurden, betten in Apache einen persistenten Interpreter ein, der die Skripts unter der Benutzer-ID von Apache ausführt.

CGIs mit `mod_suexec` vereinfachen daher die Verwaltung, da die CGI-Prozesse statt dem Webserver individuellen Benutzern zugeordnet werden können. Ausserdem erhöht diese Kombination die Sicherheit des Dateisystems: Das Skript übernimmt nur die Dateisystemrechte des jeweiligen Benutzers. Dagegen werden dem Skript im Falle von Modulen die Dateiberechtigungen des Webserver-Benutzers zugeschrieben, was wiederum zu einer unerwünschten Datensichtbarkeit im Dateisystem führen kann. CGIs werden nach der Ausführung einer Client-Anforderung an den Webserver beendet. CGIs sind also nicht persistent und geben die belegten Ressourcen nach ihrer Beendigung frei. Gerade im Falle einer fehlerhaften Programmierung ist dies von Vorteil. Bei Modulen können sich die Auswirkungen von Programmierungsfehlern anhäufen, da der Interpreter persistent vorliegt. Dies kann dazu führen, dass Ressourcen, beispielsweise Datenbankverbindungen, nicht mehr freigegeben werden, wodurch letztlich ein Neustart von Apache erforderlich wird.

Wenn Sie `mod_cgi` in SUSE Linux verwenden möchten, fügen Sie `cgi` zu `APACHE_MODULES` in `/etc/sysconfig/apache2` hinzu oder verwenden Sie YaST. Das Standardverzeichnis für CGIs ist in SUSE Linux `/srv/www/cgi-bin/`. Falls Sie Ihre Apache-Konfigurationsdatei manuell bearbeiten möchten, verwenden Sie das folgende Beispiel als Anhaltspunkt für die Konfiguration von `mod_cgi`.

#### Manuelle Aktivierung von `mod_cgi`

```
# Global Environment

LoadModule cgi_module /Pfad/zu/mod_cgi.so

# Main Server and/or Virtual Host and/or
# Directory and/or .htaccess context

AddHandler cgi-script .cgi .pl

# Main Server and/or Virtual Host context

ScriptAlias /cgi-bin/ /srv/www/cgi-bin/

# Alternatively, explicitly allow CGI scripts in a directory
# Main Server and/or Virtual Host context

<Directory /srv/www/some/dir>

Options +ExecCGI

</Directory>
```

#### Erweiterungsmodule

Im Allgemeinen sind Erweiterungsmodule im Apache-Softwarepaket enthalten, jedoch nicht statisch im Server kompiliert. In SUSE Linux stehen diese Module als shared Objects zur Verfügung, die während der Laufzeit in Apache geladen werden können.

### Secure Sockets Layer und Apache: `mod_ssl`

`mod_ssl` bietet mittels der Protokolle Secure Sockets Layer (SSL) und Transport Layer Security (TLS) eine sichere Verschlüsselung für die HTTP-Kommunikation zwischen einem Client und dem Webserver. Zu diesem Zweck sendet der Server vor der Beantwortung von Anforderungen an eine URL ein SSL-Zertifikat mit Informationen, die die Identität des Servers nachweisen. Dies garantiert, dass der Server der eindeutig gekennzeichnete und richtige Endpunkt der Kommunikation ist. Ausserdem wird durch das Zertifikat eine verschlüsselte Verbindung zwischen dem Client und dem Server hergestellt, die sicherstellt, dass Informationen ohne das Risiko der Freigabe sensibler Klartextinhalte übertragen werden. Die Verwendung von `mod_ssl` in Apache erkennen Sie in URLs am Präfix `https://` (statt `http://`).

Auf dem Webserver ist Port 443 der Standard-Port für SSL- und TLS-Anforderungen. Zwischen einem «normalen» Apache-Webserver, der Port 80 überwacht, und einem SSL/TLS-aktivierten Apache-Server, der Port 443 überwacht, kommt es zu keinen Konflikten. In der Tat kann die gleiche Apache-Instanz sowohl HTTP als auch HTTPS ausführen. In der Regel ist ein virtueller Host eigens dafür abgestellt, die Anforderungen für Port 80 und Port 443 an separate virtuelle Server zu verteilen.

### WICHTIG: Namensbasierte virtuelle Hosts und `mod_ssl`

Auf einem Server mit nur einer IP-Adresse können nicht mehrere SSL-aktivierte virtuelle Hosts laufen. Benutzer, die versuchen, eine Verbindung mit einer solchen Konfiguration herzustellen, erhalten bei jedem Besuch der URL eine Warnung mit dem Hinweis, dass das Zertifikat nicht mit dem Namen des Servers übereinstimmt. Für die Kommunikation auf Grundlage eines gültigen SSL-Zertifikats ist eine separate IP-Adresse bzw. ein separater Port für jede SSL-aktivierte Domäne erforderlich. Trotz der Warnung erhalten Sie die gleiche Verschlüsselungsstufe wie auf jeder gültigen SSL-Site. Die Kommunikation zwischen dem Webserver und dem Client ist also trotz Warnung sicher. Ein wichtiges Konzept, das durch ein gültiges SSL-Zertifikat garantiert wird, nämlich der Identitätsnachweis des Servers, geht allerdings verloren.

Wenn Sie `mod_ssl` in SUSE Linux aktivieren möchten, fügen Sie `ssl` zu `APACHE_MODULES` in `/etc/sysconfig/apache2` hinzu oder verwenden Sie YaST. Ausserdem müssen Sie auf dem Webserver die Überwachung des HTTPS-Standardportes 443 konfigurieren. Diese Einstellung können Sie manuell in `/etc/apache2/listen.conf` oder in YaST mit dem Menüeintrag **Lauschen auf** vornehmen.

Mit `cd /usr/share/doc/packages/apache2; ./certificate.sh` als `root` können Sie ein SSL-Testzertifikat erstellen. Befolgen Sie hierzu die Anweisungen auf dem Bildschirm. Die zugehörigen Zertifikatdateien werden in den Verzeichnissen `/etc/apache2/ssl*` abgelegt. Ein «echtes» Zertifikat mit globaler Gültigkeit erhalten Sie von Zertifikatsausstellern wie Thawte (<http://www.thawte.com/>) oder Verisign ([www.verisign.com](http://www.verisign.com)).

Falls Sie Ihre Apache-Konfigurationsdatei manuell bearbeiten möchten, verwenden Sie das folgende Beispiel als Anhaltspunkt für die Konfiguration von `mod_ssl`.

#### Manuelle Konfiguration von `mod_ssl`

```
# Global Environment

# listen on the standard SSL port

Listen 443

# load module only if rcapache2 start-ssl was issued
```

```
<IfDefine SSL>
LoadModule ssl_module /Pfad/zu/mod_ssl.so
</IfDefine>

# Main Server context

# include global (server-wide) SSL configuration
# that is not specific to any virtual host
# only if ssl_module was loaded

<IfModule mod_ssl.c>
Include /etc/apache2/ssl-global.conf
</IfModule>
```

### Externe Module

Externe Module sind offiziell nicht in der Apache-Distribution enthalten. SUSE Linux bietet jedoch einige externe Module an, die ohne grossen Aufwand sofort verwendet werden können. Dieses Kapitel geht kurz auf einige dieser Module und deren Funktionen ein.

#### Verwenden von Perl zur Verwaltung von Apache: `mod_perl`

`mod_perl` bettet einen persistenten Perl-Interpreter in Apache ein. Perl umgeht den von `mod_cgi` verursachten Overhead, der bei jeder CGI-Anforderung eine externe ausführbare Datei aufruft. Zudem ermöglicht `mod_perl` die Steuerung zahlreicher Aspekte der Apache-Funktionalität mithilfe der Programmiersprache Perl. Wenn Sie `mod_perl` in SUSE Linux verwenden möchten, installieren Sie das `apache2-mod_perl-RPM` und aktivieren Sie das Modul mit YaST oder manuell in `/etc/sysconfig/apache2`. Nach der Installation und Aktivierung wird in `/etc/apache2/conf.d/` eine eigene Konfigurationsdatei (`mod_perl.conf`) für dieses Modul erstellt. Ausserdem wird das `mod_perl`-Startskript (`mod_perl-startup.pl`) installiert. Weitere Informationen zur Verwendung dieses Moduls finden Sie in der Dokumentation auf der Website zu `mod_perl` unter <http://perl.apache.org/>.

#### Unterstützung für PHP: `mod_php4`, `mod_php5`

PHP ist eine weit verbreitete Programmiersprache, die ursprünglich für das Web entwickelt wurde und in zwei Versionen vorliegt: PHP4 und PHP5. PHP4 repräsentiert das klassische Konzept und die ursprünglichen Vorgehensweisen von PHP, während PHP5 neue, objekt-orientierte Programmiermöglichkeiten mit zahlreichen erweiterten Funktionen bereitstellt. Beide Versionen stehen in SUSE Linux zur Verfügung. Sie betten den PHP-Interpreter als persistentes Modul in Apache ein. Wenn Sie `mod_php4` oder `mod_php5` in SUSE Linux verwenden möchten, installieren Sie das betreffende RPM (`apache2-mod_php4` oder `apache2-mod_php5`) und aktivieren Sie das Modul mit YaST oder manuell in `/etc/sysconfig/apache2`.

Nach der Installation und Aktivierung wird in `/etc/apache2/conf.d/` eine eigene Konfigurationsdatei für das jeweilige Modul (`php4.conf` oder `php5.conf`) erstellt. Die PHP-Website (<http://www.php.net>) ist ein hervorragendes Nachschlagewerk, wenn Sie Informationen zur Verwendung von Apache mit PHP suchen.

**Zugriff auf das native Dateisystem: mod\_dav**

`mod_dav` stellt in Apache WebDAV-Funktionalität (Web-Based Distributed Authoring and Versioning) bereit. WebDAV ist eine Erweiterung des HTTP-Protokolls, mit dem Benutzer Dateien auf entfernten Servern gemeinsam bearbeiten und verwalten können. Die Funktionalität von WebDAV ist vergleichbar mit der von FTP, allerdings mit dem Unterschied, dass HTTP als zugrunde liegendes Protokoll für den Serverzugriff verwendet wird. Im Prinzip wandelt `mod_dav` einen einfachen Apache-Webserver in ein erweitertes entferntes Dateisystem um. Wenn auch nicht erforderlich, empfiehlt es sich, den Zugriff auf die via WebDAV zur Verfügung gestellten Verzeichnisse einzuschränken. Als minimale Vorkehrung sollten Sie die WebDAV-Ressource durch eine grundlegende HTTP-Authentifizierung und Limit-Klauseln in `Location`-Direktiven schützen.

Für den Zugriff auf WebDAV-Ressourcen ist auf dem Client eine WebDAV-fähige Software erforderlich. SUSE Linux verfügt bereits über WebDAV-Fähigkeiten: Für die Verbindung mit einem Apache WebDAV-Dateisystem kann `Konqueror` mit dem Präfix `webdav://` oder `webdavs://` (Letzteres für WebDAV via SSL) verwendet werden.

`mod_dav` setzt das Modul `mod_dav_fs` voraus, das den eigentlichen Dateisystemzugriff für WebDAV bereitstellt. Wenn Sie `mod_dav` in SUSE Linux verwenden möchten, aktivieren Sie das Modul mit YaST oder manuell in `/etc/sysconfig/apache2`. Aktivieren Sie auf die gleiche Weise auch `mod_dav_fs`.

Weitere Informationen zur Verwendung dieses Moduls finden Sie in der Dokumentation auf der Website zu `mod_dav` unter [http://httpd.apache.org/docs-2.0/mod/mod\\_dav.html](http://httpd.apache.org/docs-2.0/mod/mod_dav.html).

**Anbieten von Benutzer-Homepages: mod\_userdir**

`mod_userdir` in SUSE Linux bietet standardmässig den Inhalt des `~/public_html`-Ordners eines jeden Benutzers als öffentliche Webseiten an. Die URL, mit der auf diese Seiten zugegriffen wird, lautet <http://www.beispiel.com/~Benutzername/>.

Wenn Sie `mod_userdir` in SUSE Linux verwenden möchten, aktivieren Sie das Modul mit YaST oder manuell in `/etc/sysconfig/apache2`. Weitere Informationen zur Verwendung dieses Moduls finden Sie in der Dokumentation auf der Website zu `mod_userdir` unter [http://httpd.apache.org/docs-2.0/mod/mod\\_userdir.html](http://httpd.apache.org/docs-2.0/mod/mod_userdir.html).

**Ändern des URL-Layouts: mod\_rewrite**

`mod_rewrite` wird gerne mit einem «Schweizer Präzisionsmesser für die URL-Manipulation» verglichen. Es schreibt angeforderte URLs in Windeseile nach bestimmten Regeln um. Aus umständlichen URLs wie <http://www.beispiel.com/display.php?cat=2&article=1&lang=de> ergibt sich so sehr schnell eine wesentlich einfachere Adresse wie <http://www.beispiel.com/2/1/de>.

Der `URL Rewriting Guide` fasst die Vorteile und Nachteile dieses leistungsstarken, aber komplexen Moduls mit wenigen Worten zusammen:

«Mit `mod_rewrite` schießen Sie sich beim ersten Versuch entweder in den Fuss und verwenden es nie wieder oder Sie schätzen seine Leistungstärke für den Rest Ihres Lebens.»

`RewriteRule`-Sätze können für jeden Konfigurationskontext festgelegt werden: für den Hauptserver, für virtuelle Hosts, für Verzeichnisse und für `.htaccess`-Dateien. Wenn Sie `mod_rewrite` zum ersten Mal verwenden, empfiehlt sich als Lektüre der «URL Rewriting Guide» unter <http://httpd.apache.org/docs-2.0/misc/rewriteguide.html>.



Wenn Sie `mod_rewrite` in SUSE Linux verwenden möchten, aktivieren Sie das Modul mit YaST oder manuell in `/etc/sysconfig/apache2`.

### 8.3.14 Sicherheit

---

Ein dem öffentlichen Internet ausgesetzter Webserver erfordert ständige Wartungs- und Verwaltungsarbeiten. Sicherheitsprobleme, verursacht durch die Software wie auch durch versehentliche Fehlkonfigurationen, sind kaum zu vermeiden. Im Folgenden einige Tipps zur Verbesserung der Sicherheit.

**Bleiben Sie stets auf dem neuesten Stand!** Bei Bekanntwerden von Sicherheitsrisiken in der Apache-Software veröffentlicht SUSE sofort einen entsprechenden Sicherheitshinweis. Dieser enthält Anleitungen zur Behebung der Risiken, die möglichst frühzeitig ausgeführt werden sollten. Die SUSE-Mailing-Liste zu Sicherheitsankündigungen ist unter [http://www.suse.com/us/private/support/online\\_help/maillinglists/](http://www.suse.com/us/private/support/online_help/maillinglists/) verfügbar.

Die neuesten Informationen zu Sicherheitsaspekten in SUSE Linux-Paketen werden ausserdem unter <http://www.novell.com/linux/security/securitysupport.html> online veröffentlicht.

Ausserdem sollten Sie sich in die Apache-Mailing-Liste eintragen (<http://httpd.apache.org/lists.html#http-announce>), über die neue Versionen und Bug Fixes veröffentlicht werden.

#### DocumentRoot-Berechtigungen

In SUSE Linux sind das DocumentRoot-Verzeichnis `/srv/www/htdocs` (absoluter Pfad) und das CGI-Verzeichnis `/srv/www/cgi-bin` standardmässig dem Root-Benutzer zugeordnet. Diese Berechtigungen sollten nicht geändert werden. Wenn diese Verzeichnisse für alle Benutzer modifizierbar wären, könnte jeder Benutzer Dateien darin ablegen. Diese Dateien würden dann von Apache mit `wwwrun`-Berechtigungen ausgeführt werden, was wiederum dem Benutzer unbeabsichtigt Zugriff auf die Ressourcen des Dateisystems gewähren würde. Verwenden Sie Unterverzeichnisse von `/srv/www/htdocs` und `/srv/www/cgi-bin` zur Organisation von benutzer- oder domänenspezifischen Daten in Kombination mit der `Directory`-Direktive.

#### CGI- und SSI-Verzeichnisse

Interaktive Skripts in Perl, PHP, SSI oder anderen Programmiersprachen können im Prinzip jeden beliebigen Befehl ausführen. Eine Möglichkeit, das damit einhergehende Sicherheitsrisiko zu vermindern, ist eine Ausführungsbeschränkung für CGIs und SSIs und «Serverseitige Includes (Einschlüsse) mit `mod_include`» auf bestimmte Verzeichnisse, statt einer globalen Zulassung dieser Skripts. Eine andere Möglichkeit ist die generelle Verwendung von `mod_suexec` für CGIs. Auch eine sicherheitsbewusste Interpreterkonfiguration für die jeweiligen Apache-Module, wie in «Unterstützung für PHP: `mod_php4`, `mod_php5`» beschrieben, ist bereits ein grosser Schritt in Richtung einer sicheren Web-Umgebung.

## Zugriffsberechtigungen

Besonders in Testumgebungen werden die Zugriffsberechtigungen für einen Webserver oft nachlässig behandelt, da es sich ja «nur» um einen Konfigurationstest handelt. Dies kann zur versehentlichen Freigabe sensibler Informationen, ja sogar zur Preisgabe eines vollständigen Servers an das falsche Publikum führen. Verwenden Sie die `Order`-Direktive ([http://httpd.apache.org/docs-2.0/mod/mod\\_access.html#order](http://httpd.apache.org/docs-2.0/mod/mod_access.html#order)) in Verbindung mit `.htaccess`-Dateien, um den Zugriff auf bestimmte Websites auf einen bestimmten Benutzer- oder Client-Kreis einzuschränken.

Zusätzlich können Sie nach dem Grundsatz «Sicherheit durch Verschleierung» vorgehen. Ein typisches Beispiel hierfür wäre die Ausführung von Apache an einem nicht standardgemässen Port. An die URLs würde in diesem Fall die Port-Nummer angefügt werden (z. B. <http://www.beispiel.com:8765>), was in Testumgebungen durchaus akzeptabel ist.

### 8.3.15 Fehlerbehebung

---

Wenn sich Apache nicht starten lässt, eine Webseite nicht angezeigt werden kann oder Benutzer keine Verbindung zum Webserver herstellen können, müssen Sie die Ursache des Problems herausfinden. Im Folgenden werden einige nützliche Ressourcen vorgestellt, die Ihnen bei der Fehlersuche behilflich sein können.

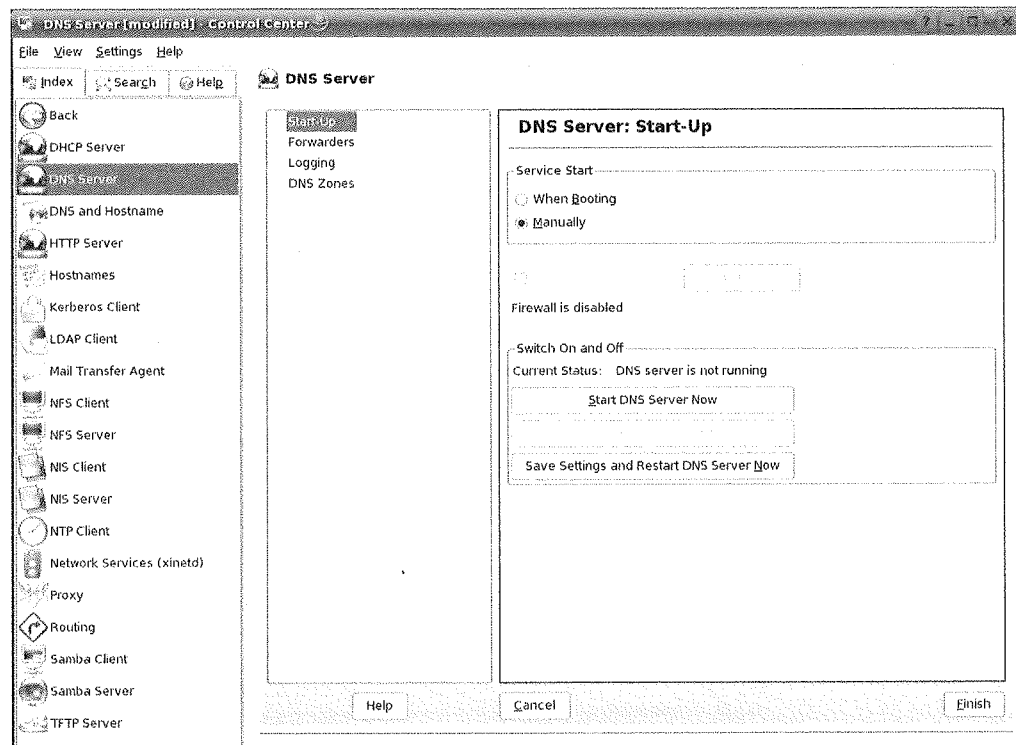
An erster Stelle sei hier das Skript `rcapache2` genannt, das sich sehr ausführlich mit Fehlern und deren Ursachen befasst und bei Problemen mit Apache wirklich hilfreich ist. Manchmal ist es eine Versuchung, die Binärdatei `/usr/sbin/httpd2` zum Starten oder Beenden des Webserver zu verwenden. Vermeiden Sie dies aber und verwenden Sie stattdessen besser das Skript `rcapache2`. Dieses gibt sogar Tipps und Hinweise zur Behebung von Konfigurationsfehlern.

An zweiter Stelle möchten wir auf die Bedeutung von Protokolldateien hinweisen. Sowohl bei geringfügigen als auch bei schwerwiegenden Fehlern sind die Protokolldateien von Apache der beste Ort, um nach Fehlerursachen zu fahnden. Mit der Direktive `LogLevel` (Protokollgenauigkeit) können Sie im Übrigen die Ausführlichkeit der protokollierten Meldungen einstellen. Dies ist zum Beispiel nützlich, wenn Sie mehr Details benötigen.

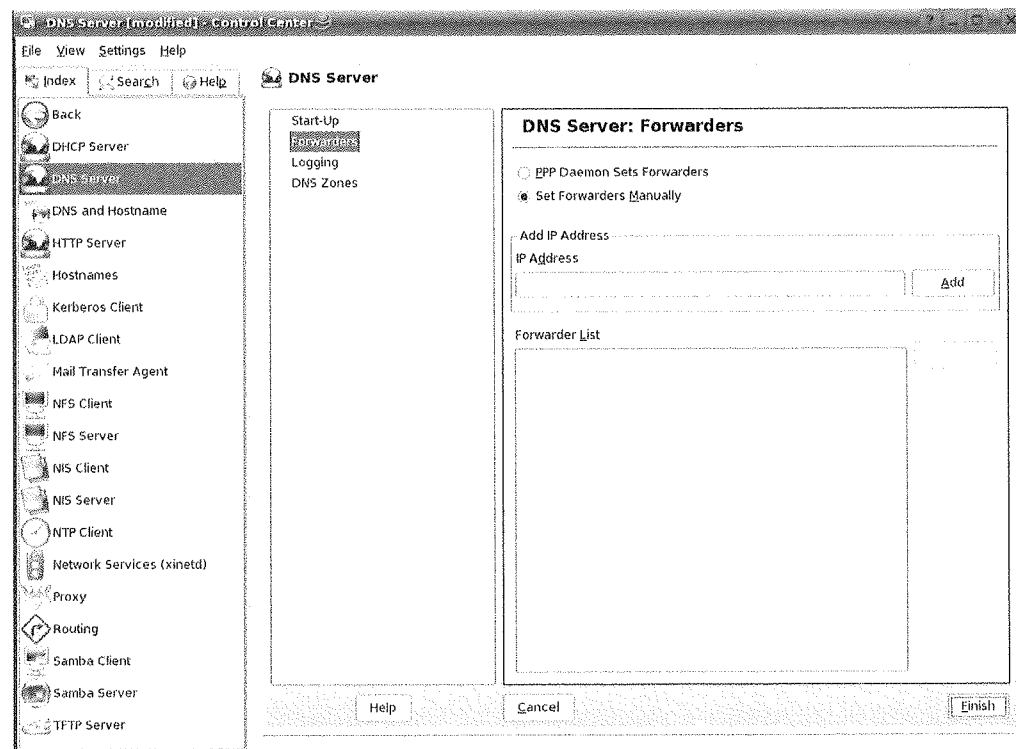
Häufig wird vergessen, die Ports für Apache in der Firewall-Konfiguration des Servers zu öffnen. YaST bietet bei der Konfiguration von Apache eine eigene Option, die sich dieses speziellen Themas annimmt. Falls sich Ihr Problem nicht mithilfe der vorgenannten Ressourcen beheben lässt, finden Sie weitere Informationen in der Apache-Fehlerdatenbank, die online unter [http://httpd.apache.org/bug\\_report.html](http://httpd.apache.org/bug_report.html) zur Verfügung steht. Sie können sich auch an die Apache-Benutzercommunity wenden, die Sie via Mailing-Liste unter <http://httpd.apache.org/userslist.html> erreichen. Des Weiteren empfehlen wir die Newsgroup `comp.infosystems.www.servers.unix`.

## 8.4 DNS für Linux

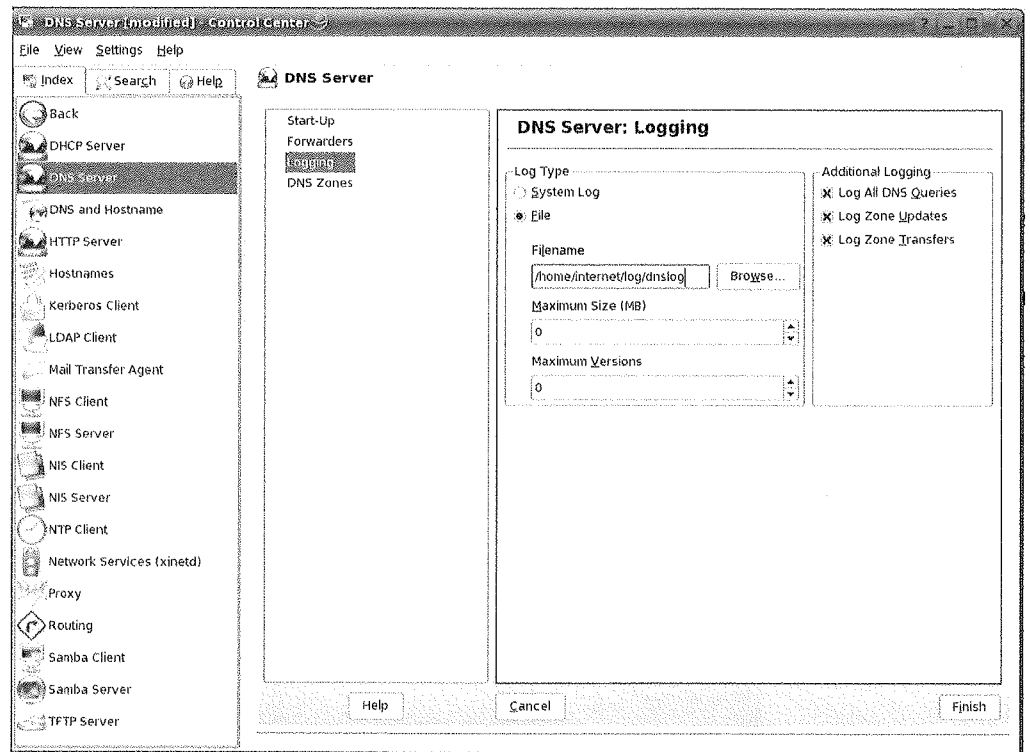
Der DNS-Server ist unter dem entsprechenden Menüpunkt «DNS Server» zu finden:



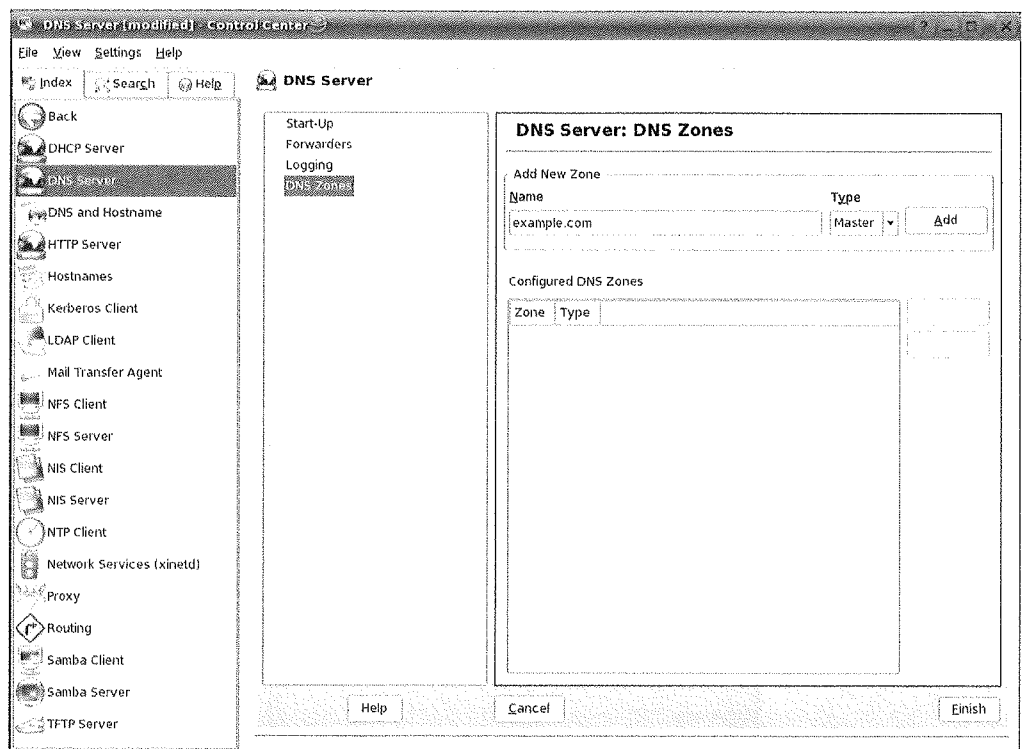
Auch hier wird der Service Start auf «When Booting» gesetzt. Da die Firewall ausgeschaltet ist, wird keine automatische Konfiguration derselben vorgenommen (Check Box ist grau). Bevor der DNS Server gestartet wird (passiert beim Reboot automatisch), werden noch die restlichen Einstellungen vorgenommen:



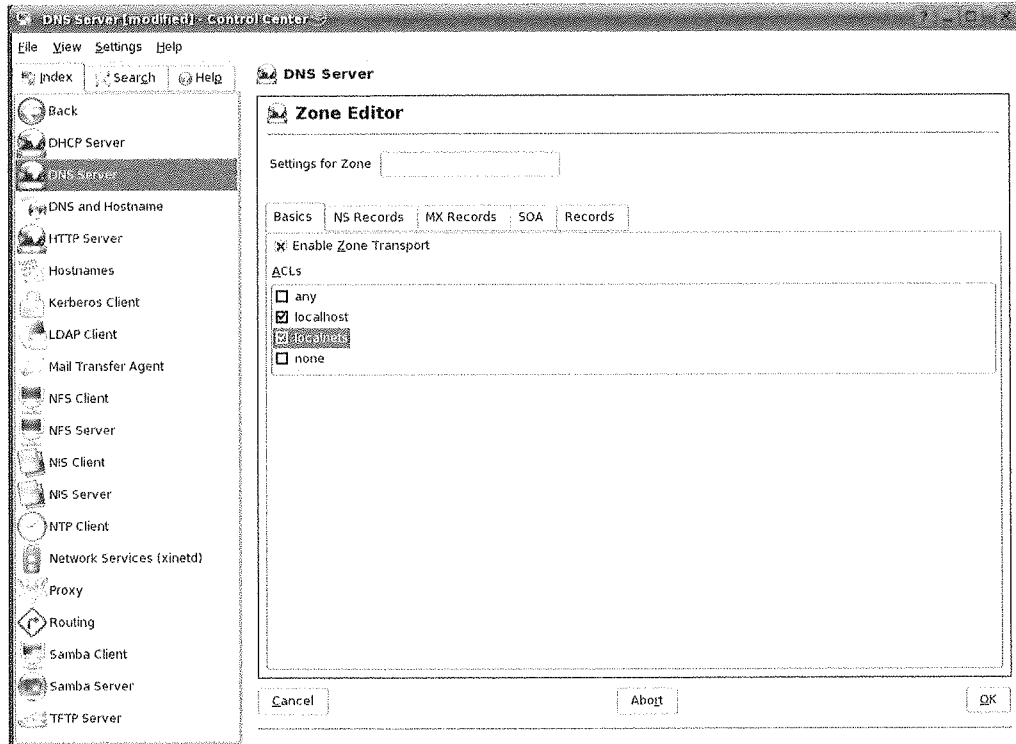
Hier wird nichts geändert, «Set Forwarders Manually» belassen. Bei «Logging» auf dem nächsten Bild empfiehlt es sich, ein separates Logfile für den DNS-Server zu konfigurieren:



Auf der Seite «DNS Zones» können nun die Einträge für die Domains vorgenommen werden, die von diesem DNS-Server aufgelöst werden sollen.

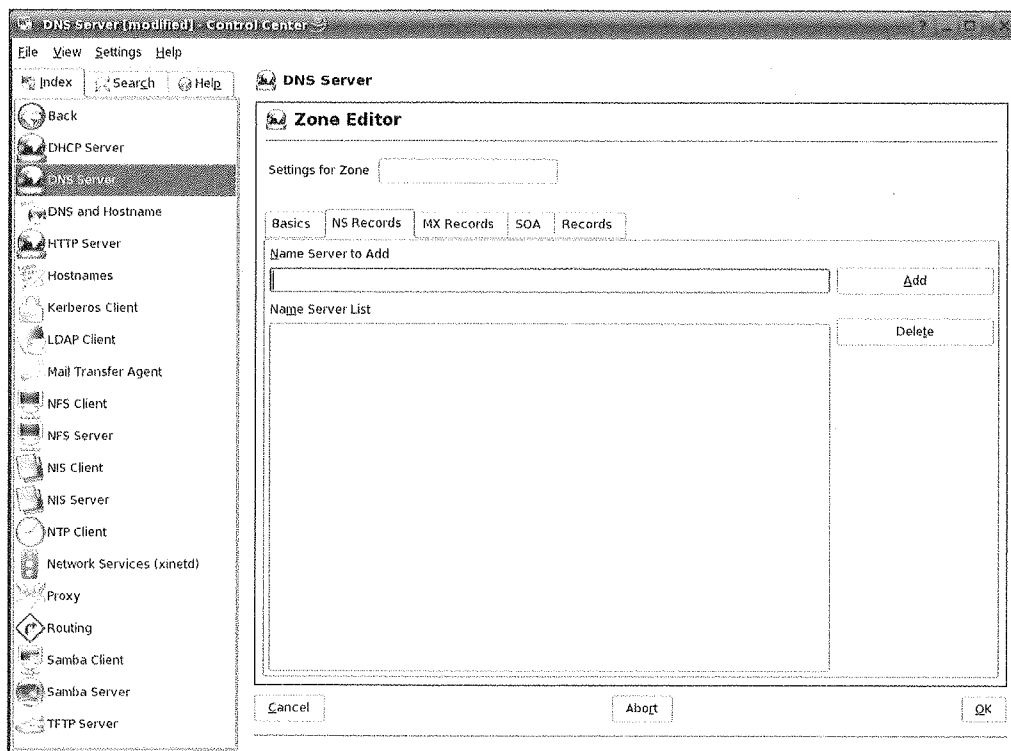


Die vorgeschlagene example.com Zone als «Master» belassen und «Add» klicken. Andernfalls «example.com» markieren, mit der «cut/auschneiden» Funktion entfernen und mit «Add» das Menü dazu bringen, den Fokus auf das Eingabefeld zu bringen (da der vorgeschlagene Name anderweitig nicht änderbar ist). Danach den Eintrag auswählen und mit «Edit» anpassen:

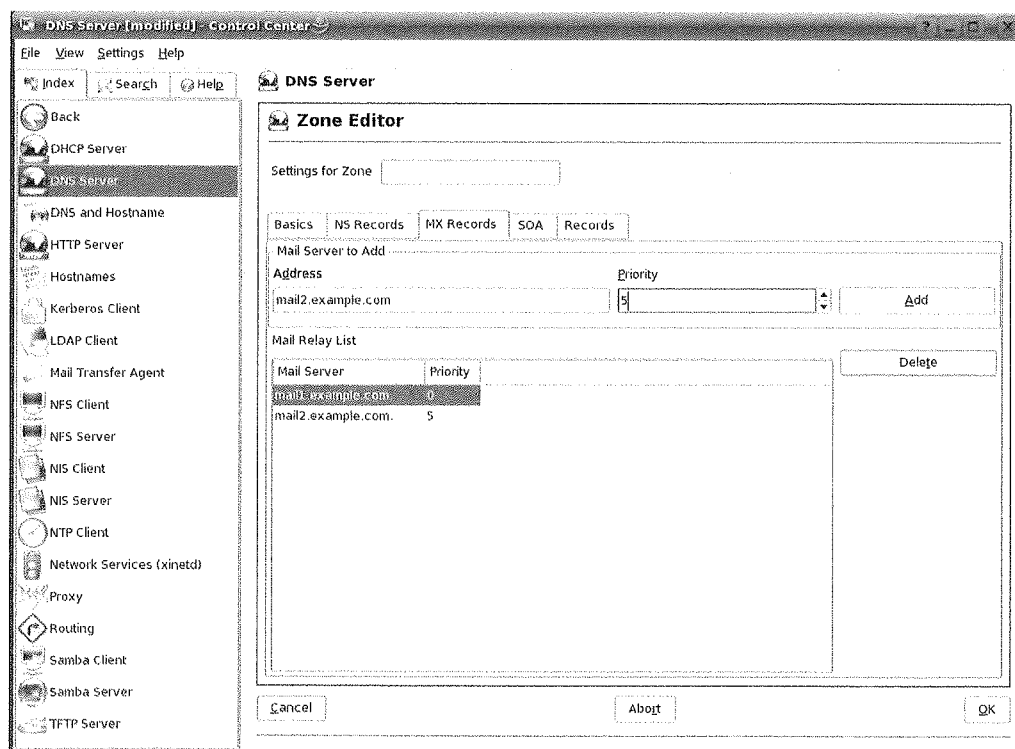


«Enable Zone Transport» ist einzuschalten, falls ein secondary DNS-Server die Zonen von diesem Primary-DNS-Server refreshen muss. Andernfalls ist die Option auszuschalten. Wenn eingeschaltet, kann ausgewählt werden, woher Zone-Transfer-Anfragen beantwortet werden. «Any» bedeutet, dass jegliche Anfragen beantwortet werden, es empfiehlt sich deshalb die Beschränkung auf localhost, localnets.

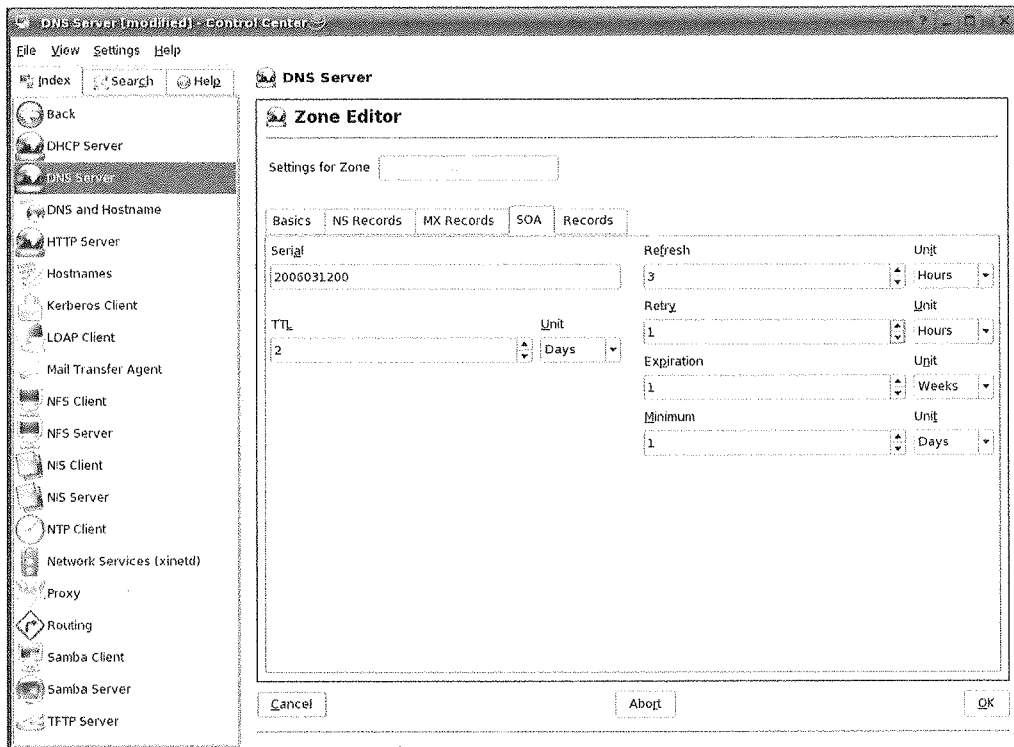
Auf dem nächsten tab «NS Records» können optional weitere Nameserver eingefügt werden. In diesem Fall sind keine Einträge vorgesehen.



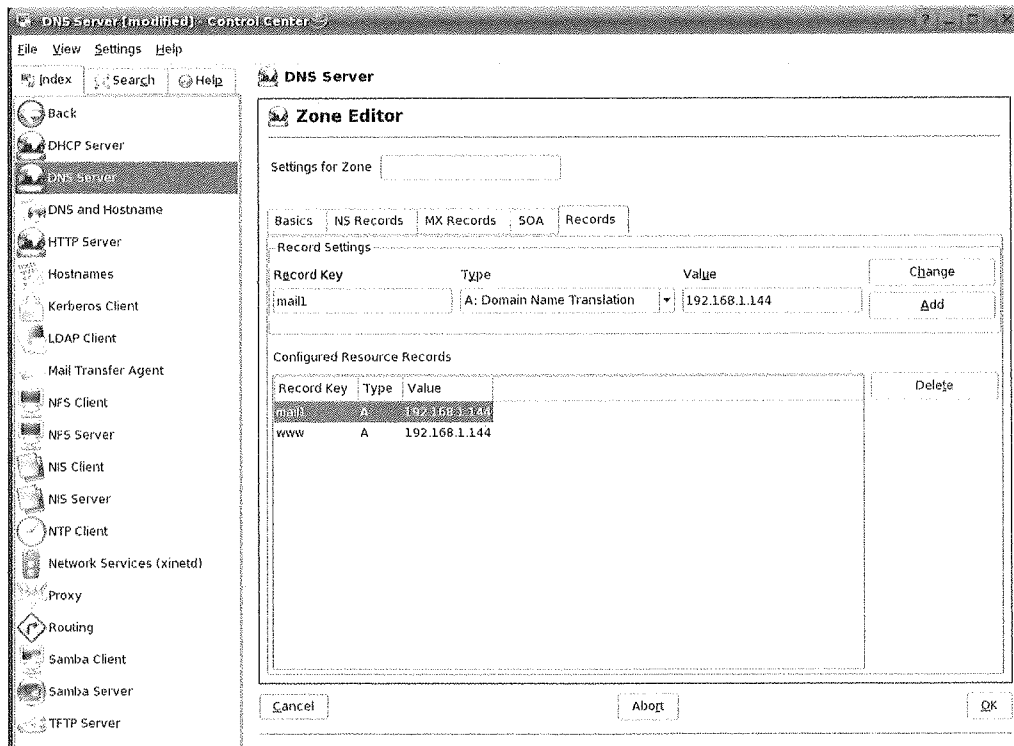
Im nächsten tab «MX Records» werden Einträge für Mailserver der Domain gemacht. Mit der Prioritätszahl kann eingestellt werden, in welcher Reihenfolge (bei nicht-antworten) die Mailserver benutzt werden sollen. Es empfiehlt sich, falls möglich einen Back-up-Mailserver einzurichten, falls der primäre Mailserver ausser Betrieb ist (ansonsten gehen Mails verloren).



Der tab «SOA» kann so belassen werden. Hier wird nötigenfalls eingestellt, nach welcher Zeit der Zone-Record abläuft und vom secondary-DNS-Server bzw. anderen DNS-Servern refreshed werden muss.



Auf dem letzten tab «Records» werden nun die Zuweisungen der eigentlichen Rechner und deren IP-Adressen vorgenommen. Es benötigt hier vor allem den Eintrag für den MX-Record (IP-Adresse des Mailserver) sowie des WWW-Server (Web-/http-Server). Da beide auf dem gleichen System laufen, ist die lokale IP-Adresse hier einzutragen. (Wir erinnern uns: der Internetserver ist zugleich Web-, Mail- und DNS-Server, die IP-Adresse ist somit immer gleich.)



Die Konfiguration wird mit «Finish» gespeichert und beendet.

### 8.4.1 Nameserver BIND starten

---

Es folgen nun detaillierte Konfigurationsanleitungen für den DNS-Server «BIND», der oben via grafischem Interface konfiguriert wurde. Was nun folgt, sind die Konfigurationsmöglichkeiten über die eigentlichen Konfigurationsdateien.

Der Nameserver BIND (Berkeley Internet Name Domain) ist auf SUSE Linux bereits soweit vorkonfiguriert, dass man ihn problemlos sofort nach der Installation starten kann. Hat man bereits eine funktionsfähige Internetverbindung und trägt in der `/etc/resolv.conf` als Nameserver `127.0.0.1` für `localhost` ein, hat man in der Regel schon eine funktionierende Namensauflösung, ohne dass man den DNS des Providers kennt. BIND führt so die Namensauflösung über die Root-Nameserver durch, was aber merklich langsamer ist. Normalerweise sollte man den DNS des Providers mit seiner IP-Adresse in der Konfigurationsdatei `/etc/named.conf` unter `forwarders` eintragen, um eine effektive und sichere Namensauflösung zu erhalten. Funktioniert das soweit, läuft der Nameserver als reiner «Caching-only»-Nameserver. Erst wenn man ihm eigene Zonen bereitstellt, wird er ein richtiger DNS werden. Ein einfaches Beispiel dafür, findet man im Dokumentations-Verzeichnis `/usr/share/doc/packages/bind/sample-config`.

Man sollte allerdings keine offizielle Domain aufsetzen, solange man diese nicht von der zuständigen Institution – für `.ch` ist das die SWITCH AG `nic.ch` – zugewiesen bekommen hat. Auch wenn man eine eigene Domain hat, diese aber vom Provider verwaltet wird, sollte man diese besser nicht verwenden, da BIND sonst keine Anfragen für diese Domain mehr forwarden (weiterleiten) würde und so zum Beispiel der Webserver beim Provider für die eigene Domain nicht mehr erreichbar wäre. Um den Nameserver zu starten, gibt man auf der Kommandozeile den Befehl `rndc start` als `root` ein. Erscheint rechts in grün «done», ist der `named`, so heisst der Nameserver-Prozess, erfolgreich gestartet. Auf dem lokalen System kann man die Funktionsfähigkeit des Nameservers sofort testen, indem man die Programme `host` oder `dig` verwendet. Als Default-Server muss `localhost` mit der Adresse `127.0.0.1` angezeigt werden. Sollte das nicht der Fall sein, steht wahrscheinlich in der `/etc/resolv.conf` ein falscher Nameserver oder diese Datei existiert gar nicht.

Für einen ersten Test gibt man `host 127.0.0.1` ein, das sollte immer funktionieren; erhält man eine Fehlermeldung, sollte man mit dem Befehl `rndc status` überprüfen, ob der `named` überhaupt läuft. Falls der Nameserver nicht startet oder ein fehlerhaftes Verhalten zeigt, findet man die Ursache in den meisten Fällen in `/var/log/messages` protokolliert.

Um den Nameserver des Providers oder um einen eigenen, der bereits im lokalen Netz läuft, als «Forwarder» zu verwenden, trägt man diesen oder auch mehrere, im Abschnitt `options` unter `forwarders` ein.

#### Forwarding-Optionen in `named.conf`

```
options {  
    directory "/var/lib/named";  
    forwarders { 10.11.12.13; 10.11.12.14; };  
    listen-on { 127.0.0.1; 192.168.0.99; };  
    allow-query { 127/8; 192.168.0/24; };  
    notify no;  
};
```

Nach den `options` folgen die Einträge für die Zonen, die Einträge für `localhost`, `0.0.127.in-addr.arpa`, sowie `.` vom `type hint` sollten immer vorhanden sein. Die zuge-



hörigen Dateien müssen nicht verändert werden, da sie so funktionieren wie sie sind. Beachten muss man auch, dass nach jedem Eintrag ein ; steht und die geschweiften Klammern korrekt gesetzt sind. Hat man nun Änderungen an der Konfigurationsdatei `/etc/named.conf` oder an den Zonen-Dateien vorgenommen, muss man BIND mit dem Kommando `rndc reload` dazu veranlassen, diese neu einzulesen.

Alternativ kann man den Nameserver auch komplett mit dem Befehl `rndc restart` neu starten. Mit dem Kommando `rndc stop` kann man den Nameserver jederzeit komplett beenden.

#### 8.4.2 Die Konfigurationsdatei `/etc/named.conf`

---

Alle Einstellungen zum Nameserver BIND sind in der Datei `/etc/named.conf` vorzunehmen. Die Zonendaten selbst, die Rechnernamen, IP-Adressen etc. für die zu verwaltenden Domains, sind in separaten Dateien im Verzeichnis `/var/lib/named` abzulegen, dazu aber später mehr.

Die `/etc/named.conf` unterteilt sich grob in zwei Bereiche, zum einen der Abschnitt `options` für allgemeine Einstellungen und zum anderen die `zone`-Einträge für die einzelnen Domains. Ausserdem kann man noch einen Bereich `logging`, sowie Einträge vom Typ `acl` (engl. Access Control List) definieren. Kommentarzeilen beginnen mit einem `#`-Zeichen, alternativ ist `//` auch erlaubt.

##### Minimalistische Datei `/etc/named.conf`

```
options {
directory "/var/lib/named";
forwarders { 10.0.0.1; };
notify no;
};
zone "localhost" in {
type master;
file "localhost.zone";
};
zone "0.0.127.in-addr.arpa" in {
type master;
file "127.0.0.zone";
};
zone "." in {
type hint;
file "root.hint";
};
```

### 8.4.3 Wichtige Konfigurationsoptionen

---

#### **directory "filename";**

gibt das Verzeichnis an, in dem der BIND die Dateien mit den Zonendaten findet; dies ist in der Regel `/var/lib/named`.

#### **forwarders { ip-address; };**

verwendet man, um den oder die Nameserver (meist des Providers) anzugeben, an den oder die die DNS-Anfragen weitergereicht werden, die nicht direkt beantwortet werden können. Anstelle von `ip-address` verwenden Sie eine IP-Adresse wie `10.0.0.1`.

#### **forward first;**

bewirkt, dass die DNS-Anfragen zu erst geforwarded werden, bevor versucht wird diese über die Root-Nameserver aufzulösen. Anstelle von `forward first` kann man auch `forward only` schreiben, dann werden alle Anfragen weitergeleitet und die Root-Nameserver werden gar nicht mehr angesprochen. Das kann für Firewall-Konfigurationen sinnvoll sein.

#### **listen-on port 53 { 127.0.0.1; ip-address; };**

sagt BIND, auf welchen Netzwerkinterfaces und welchem Port er Anfragen der Clients entgegen nehmen soll. Die Angabe `port 53` kann man sich dabei sparen, da 53 ohnehin der Standardport ist. Mit `127.0.0.1` lässt man Anfragen von localhost zu. Lässt man diesen Eintrag komplett weg, werden standardmässig alle Interfaces verwendet.

#### **listen-on-v6 port 53 { any; };**

sagt BIND, auf welchem Port er auf Anfragen der Clients horcht, die IPv6 verwenden. Ausser `any` ist alternativ nur noch `none` erlaubt, da der Server stets auf der IPv6-Wildcard-Adresse horcht.

#### **query-source address \* port 53;**

kann notwendig sein, wenn eine Firewall die externen DNS-Abfragen blockiert. So wird BIND dazu gebracht, Anfragen nach aussen von Port 53 aus und nicht von den hohen Ports > 1024 zu stellen.

#### **query-source-v6 address \* port 53;**

Dieser Eintrag muss für Anfragen über IPv6 verwendet werden.

#### **allow-query { 127.0.0.1; net; };**

bestimmt die Netze, aus denen Clients DNS-Anfragen stellen dürfen. Anstelle von `net` trägt man Adressenangaben wie `192.168.1/24` ein; dabei ist `/24` eine Kurzschreibweise für die Anzahl der Bits in der Netzmaske, in diesem Fall `255.255.255.0`.

#### **allow-transfer { ! \*; };**

regelt, welche Rechner Zonentransfers anfordern dürfen, dieses Beispiel unterbindet sie, aufgrund des `! *` komplett. Ohne diesen Eintrag können Zonentransfers ohne Einschränkungen von überall angefordert werden.

**statistics-interval 0;**

die Angabe von 0 bewirkt, dass diese komplett unterdrückt werden; hier kann man die Zeit in Minuten angeben. Ohne diesen Eintrag produziert BIND stündlich mehrere Zeilen Statusmeldungen in `/var/log/messages`.

**cleaning-interval 720;**

Diese Option legt fest, in welchem Zeitabstand BIND seinen Cache aufräumt. Die Aktivität führt jedes Mal zu einem Eintrag in `/var/log/messages`. Die Zeitangabe erfolgt in Minuten. Voreingestellt sind 60 Minuten.

**interface-interval 0;**

BIND durchsucht regelmässig die Netzwerkschnittstellen nach neuen oder nicht mehr vorhandenen Interfaces. Setzt man diesen Wert auf 0, so wird darauf verzichtet und BIND lauscht nur auf den beim Start gefundenen Interfaces. Alternativ kann man das Intervall in Minuten angeben. Voreingestellt sind 60 Minuten.

**notify no;**

Das `no` bewirkt, dass keine anderen Nameserver benachrichtigt werden, wenn an den Zonendaten Änderungen vorgenommen werden oder der Nameserver neu gestartet wird.

## 8.4.4 Zonen-Einträge

---

**Zone-Eintrag für meine-domain.ch**

```
zone "meine-domain.ch" in {  
type master;  
file "meine-domain.zone";  
notify no;  
};
```

Nach `zone` wird der Name der zu verwaltenden Domain angegeben, hier willkürlich `meine-domain.ch` gefolgt von einem `in` und einem in geschweiften Klammern gesetzten Block zugehöriger Optionen. Will man eine «Slave-Zone» definieren, ändert sich nur der `type` auf `slave` und es muss ein Nameserver angegeben werden, der diese Zone als `master` verwaltet – das kann aber auch ein «slave» sein.

**Zone-Eintrag für andere-domain.ch**

```
zone "andere-domain.ch" in {  
type slave;  
file "slave/andere-domain.zone";  
masters { 10.0.0.1; };  
};
```

Die Zonen-Optionen:

```
type master;
```

Das `master` legt fest, dass diese Zone auf diesem Nameserver verwaltet wird. Das setzt eine korrekt erstellte Zonendatei voraus.

```
type slave;
```

Diese Zone wird von einem anderen Nameserver transferiert. Muss zusammen mit `masters` verwendet werden.

```
type hint;
```

Die Zone `.` vom Typ `hint` wird für die Angabe der Root-Nameserver verwendet. Diese Zonendefinition kann man unverändert lassen.

```
file "meine-domain.zone" oder file "slave/andere-domain.zone";
```

Dieser Eintrag gibt die Datei an, in der die Zonendaten für die Domain eingetragen sind. Bei einem `slave` braucht die Datei nicht zu existieren, da ihr Inhalt von einem anderen Nameserver geholt wird. Um Master- und Slave-Dateien auseinander zu halten, gibt man für die Slave-Dateien das Verzeichnis `slave` an.

```
masters { server-ip-address; };
```

Diesen Eintrag braucht man nur für Slave-Zonen und er gibt an, von welchem Nameserver die Zonendatei transferiert werden soll.

```
allow-update { ! *; };
```

Diese Option regelt den Schreibzugriff von extern auf die Zonendaten. Damit wäre es Clients möglich, sich selbst im DNS einzutragen, was aus Sicherheitsgründen nicht wünschenswert ist. Ohne diesen Eintrag, sind Zonen-Updates generell untersagt, dieses Beispiel würde daran auch nichts ändern, da `! *` ebenfalls alles verbietet.

#### 8.4.5 Zonendateien

---

Man benötigt zwei Arten von Zonen-Dateien, die einen dienen dazu, einem Rechnernamen die IP-Adresse zuzuordnen und die anderen gehen den umgekehrten Weg und liefern zu einer gegebenen IP-Adresse den Rechnernamen.

Der folgende Teil ist aus dem Administrationshandbuch von Suse, Kapitel 13 Grundlagen der Vernetzung entnommen.

**WICHTIG:** Der Punkt `.` in Zonendateien! Eine wichtige Bedeutung hat der Punkt in den Zonendateien. Werden Rechnernamen, ohne abschliessenden `.` angegeben, wird immer die Zone ergänzt. Man muss also komplette Rechnernamen, die bereits mit vollständiger Domain angegeben wurden, mit einem `.` abschliessen, damit die Domain nicht noch einmal dran gehängt wird. Ein fehlender Punkt oder einer an der falschen Stelle, dürfte die häufigste Fehlerursache bei der Konfiguration von Nameservern sein.

Den ersten Fall betrachten wir die Zonendatei `welt.zone`, die für die Domain `welt.all` zuständig ist.

```

Datei /var/lib/named/welt.zone

$TTL 2D

welt.all. IN SOA gateway root.welt.all. (
2003072441 ; serial

1D ; refresh

2H ; retry

1W ; expiry

2D ) ; minimum

IN NS gateway

IN MX 10 sonne

gateway IN A 192.168.0.1

IN A 192.168.1.1

sonne IN A 192.168.0.2

mond IN A 192.168.0.3

erde IN A 192.168.1.2

mars IN A 192.168.1.3

www IN CNAME mond
    
```

- Zeile 1: `$TTL` definiert die Standard-TTL (engl. Time To Live), also zu deutsch Gültigkeitsdauer, die für alle Einträge in dieser Datei gilt: hier 2 Tage (2D = 2 days).
- Zeile 2: Hier beginnt der SOA control record (SOA = Start of Authority): An erster Stelle steht hier der Name der zu verwaltenden Domain `welt.all`, diese ist mit einem `.` abgeschlossen, da ansonsten die Zone noch einmal angehängt würde. Alternativ kann man hier ein `@` schreiben, dann wird die Zone dem zugehörigen Eintrag in der `/etc/named.conf` entnommen. Nach dem `IN SOA` steht der Name des Nameservers, der als Master für diese Zone zuständig ist. In diesem Fall wird der Name `gateway` zu `gateway.welt.all` ergänzt, da er nicht mit einem `.` abgeschlossen ist. Danach folgt eine E-Mail-Adresse, der für diesen Nameserver zuständigen Person. Da das `@`-Zeichen bereits eine besondere Bedeutung hat, ist hier stattdessen einfach ein `.` zu setzen, für `root@welt.all` trägt man hier folglich `root.welt.all` ein. Den `.` am Ende darf man hier nicht vergessen, da sonst die Zone noch angehängt würde. Am Ende folgt eine `(`, um die folgenden Zeilen, bis zur `)` mit in den SOA-Record einzuschliessen.
- Zeile 3: Die `serial number` ist eine willkürliche Zahl, die bei jeder Änderung an dieser Datei erhöht werden sollte. Sie wird benötigt, um sekundäre Nameserver (Slave-Server) über Änderungen zu informieren. Eingebürgert hat sich dafür eine zehnstellige Zahl aus Datum und fortlaufender Nummer in der Form `JJJJMMTTNN`.
- Zeile 4: Die `refresh rate` gibt das Zeitintervall an, in dem Sekundär-Nameserver die `serial number` der Zone überprüfen. In diesem Fall 1 Tag (1D = 1 day).
- Zeile 5: Die `retry rate` gibt den Zeitabstand an, in dem ein sekundärer Nameserver, im Fehlerfall versucht den primären Server erneut zu kontaktieren. Hier 2 Stunden (2H = 2 hours).
- Zeile 6: Die `expiration time` gibt den Zeitraum an, nachdem ein sekundärer Nameserver die gecacheten Daten verwirft, wenn er keinen Kontakt zum primären Server mehr bekommen hat. Hier ist das eine Woche (1W = 1 week).
- Zeile 7: Der letzte Eintrag im SOA ist die `negative caching TTL`. Er sagt aus, wie lange die Ergebnisse von DNS-Anfragen von anderen Servern gecached werden dürfen, die nicht aufgelöst werden konnten.
- Zeile 9: Das `IN NS` gibt den Nameserver an, der für diese Domain zuständig ist. Auch hier gilt, dass `gateway` wieder zu `gateway.welt.all` ergänzt wird, weil es nicht mit einem `.` abgeschlossen ist. Es kann mehrere Zeilen dieser Art geben, eine für den primären und jeweils eine für jeden sekundären Na-

meserver. Ist für diese Zone `notify` in der `/etc/named.conf` nicht auf `no` gesetzt, werden alle hier aufgeführten Nameserver über Änderungen der Zonendaten informiert.

- Zeile 10: Der MX-Record gibt den Mailserver an, der für die Domain `welt.all` die Mails annimmt und weiterverarbeitet oder weiterleitet. In diesem Beispiel ist das der Rechner `sonne.welt.all`. Die Zahl vor dem Rechnernamen ist der Präferenz-Wert, gibt es mehrere MX-Einträge, wird zuerst der Mailserver mit dem kleinsten Wert genommen und falls die Auslieferung an diesen scheitert, wird der mit dem nächst höheren Wert versucht.
- Zeile 12-17: Das sind jetzt die eigentlichen Adresseneinträge (engl. Address Records), in denen den Rechnernamen eine oder mehrere IP-Adressen zugeordnet werden. Die Namen stehen hier ohne abschliessenden `.`, da sie ohne angehängte Domain eingetragen sind und alle um `welt.all` ergänzt werden dürfen. Dem Rechner `gateway` sind zwei IP-Adressen zugeordnet, da er über zwei Netzwerkkarten verfügt. Das `A` steht jeweils für eine traditionelle Rechneradresse; mit `A6` trägt man IPv6-Adressen ein, und `AAAA` ist das obsoletere Format für IPv6-Adressen.
- Zeile 18: Mit dem Alias `www` kann auch `mond` (CNAME = canonical name) angesprochen werden. Für die Rückwärts-Auflösung (engl. reverse lookup) von IP-Adressen in Rechnernamen wird die Pseudo-Domain `in-addr.arpa` zu Hilfe genommen. Diese wird dazu an den in umgekehrter Reihenfolge geschriebenen Netzanteil angehängt. Aus `192.168.1` wird dann `1.168.192.in-addr.arpa`.

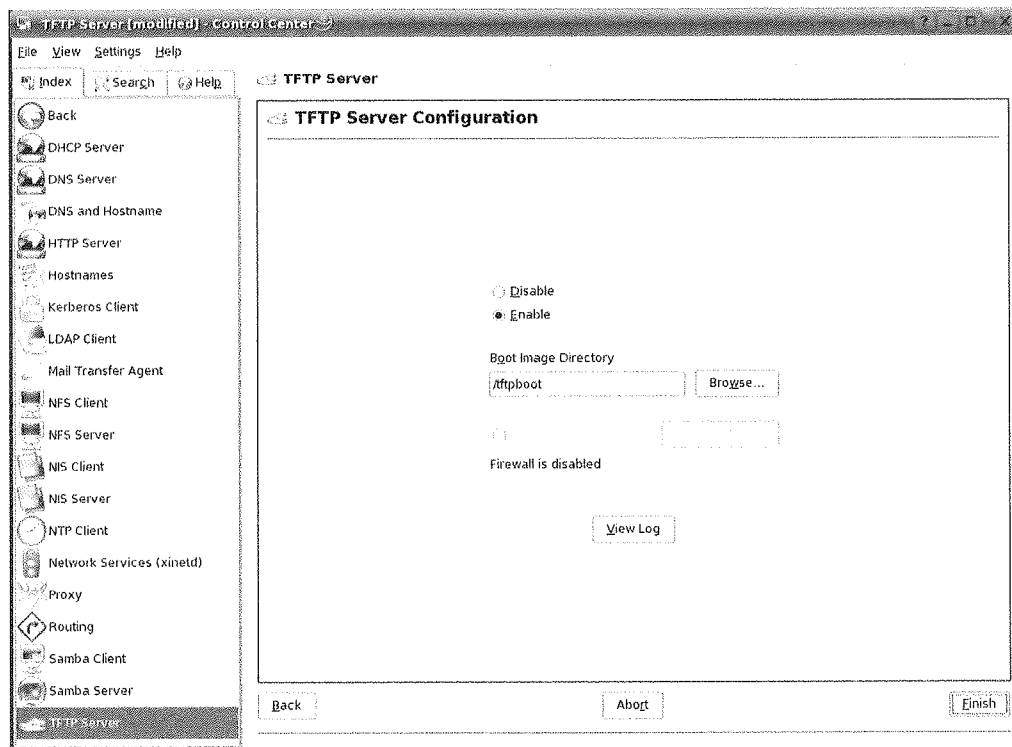
#### Umgekehrte Adressauflösung

```
$TTL 2D
1.168.192.in-addr.arpa. IN SOA gateway.welt.all. root.welt.all. (
2003072441 ; serial
1D ; refresh
2H ; retry
1W ; expiry
2D ) ; minimum
IN NS gateway.welt.all.
1 IN PTR gateway.welt.all.
2 IN PTR erde.welt.all.
3 IN PTR mars.welt.all.
```

- Zeile 1: `$TTL` definiert die Standard-TTL, die hier für alle Einträge gilt.
- Zeile 2: Der «Reverse Lookup» soll mit dieser Datei für das Netz `192.168.1.0` ermöglicht werden. Da die Zone hier `1.168.192.in-addr.arpa` heisst, will man dies natürlich nicht an die Rechnernamen anhängen, deshalb sind diese alle komplett mit Domain und abschliessendem `.` eingetragen. Der Rest entspricht dem, was im vorangegangenen Beispiel für `welt.all`, bereits beschrieben wurde.
- Zeile 3-7: Siehe vorangegangenes Beispiel für `welt.all`.
- Zeile 9: Diese Zeile gibt auch hier wieder den Nameserver an, der für diese Zone zuständig ist, diesmal wird aber der Name komplett mit Domain und abschliessendem `.` hier eingetragen.
- Zeile 11-13: Das sind die Pointer-Records, die zu einer IP-Adresse auf den zugehörigen Rechnernamen zeigen. Hier steht am Anfang der Zeile nur die letzte Stelle der IP-Adresse, ohne abschliessenden `.` Wird jetzt die Zone daran angehängt und man denkt sich das `.in-addr.arpa` weg, hat man die komplette IP-Adresse in umgekehrter Reihenfolge. Zonentransfers zwischen den verschiedenen Versionen von BIND sollten normalerweise kein Problem darstellen.

## 8.5 TFTP für Linux

Der Menüpunkt «TFTP Server» bringt uns zu einer einfachen Konfiguration des FTP-Servers:



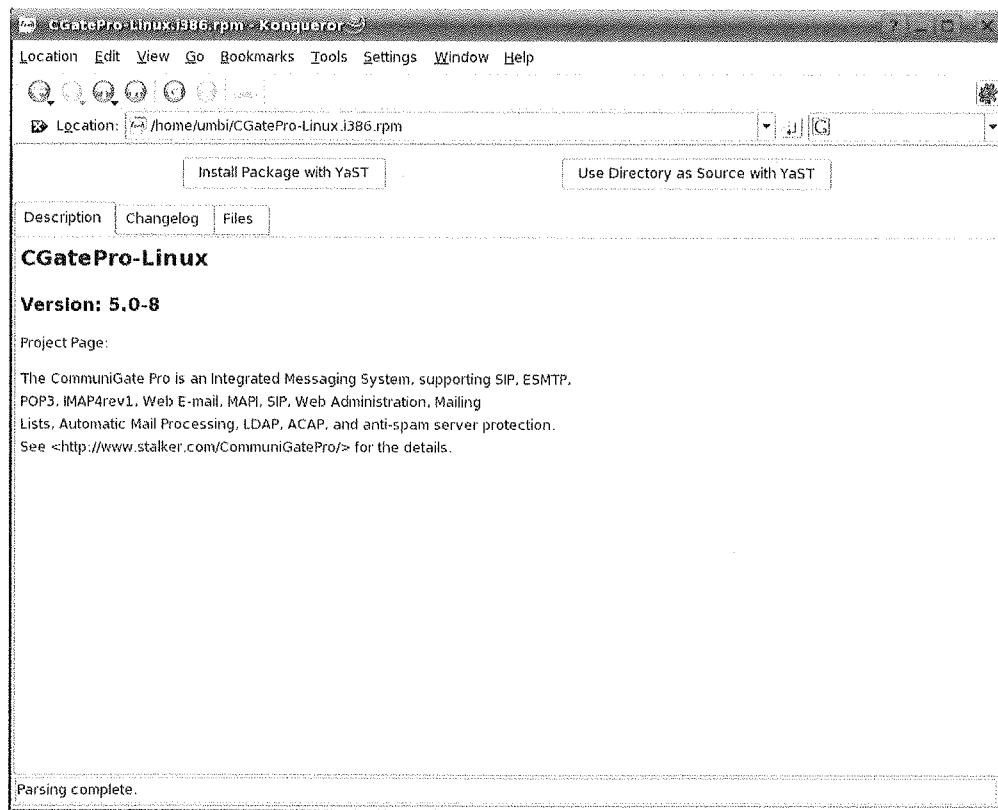
Der Server ist auf «enable» zu setzen, der Pfad weist den Weg zum Default-Verzeichnis.

Das default-Verzeichnis ist das Verzeichnis, wo die Benutzer einloggen und Dateien up- oder downloaden können. Die Benutzer, Passwörter sowie die Benutzerberechtigungen werden über das Betriebssystem (Linux-User erfassen) gesetzt.

## 8.6 SMTP und Postfach für Linux mit Stalker CommuniGate

Unter <http://www.stalker.com/content/download.htm> kann die Software «CommuniGate» Mailserver gedownloadet werden. Diese Software kann gratis evaluiert werden (ohne Funktionseinschränkung) und lässt sich via Web-GUI konfigurieren.

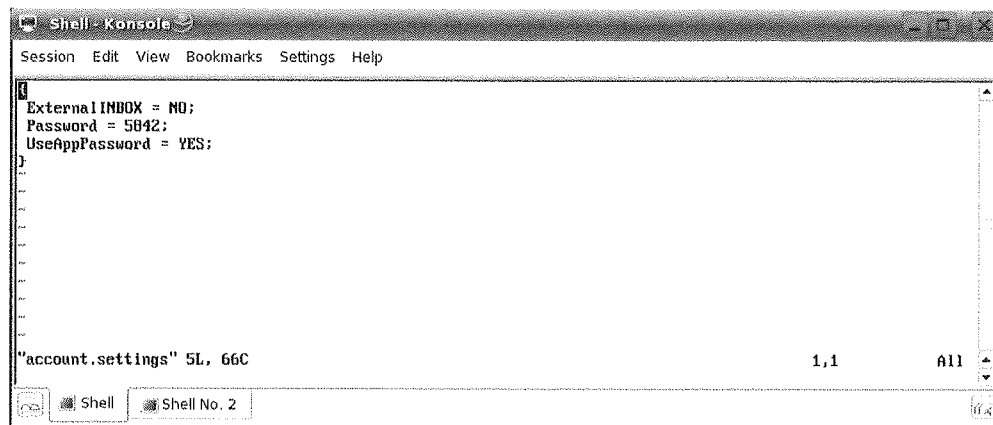
Die Installation auf Linux erfolgt per RPM-File (automatische Installation). Sobald der Download beendet wird, kann mit dem «Konqueror» File Explorer das File angeklickt werden «Install with YaST», woraufhin die Installation selbstständig durchgeführt wird.



Danach mithilfe der Dokumentation unter <http://www.communiGate.com/CommuniGatePro/Install.html#Linux> die Konfiguration vornehmen: In der Linux «shell» mit dem Befehl «chkconfig postfix off» den Mailserver (SMTP) ausschalten, da dieser sonst den Port 25 blockiert. Mit «ps -ef | grep postfix» die Prozesse des Mailserver suchen und mit «kill <PID>» abschalten.

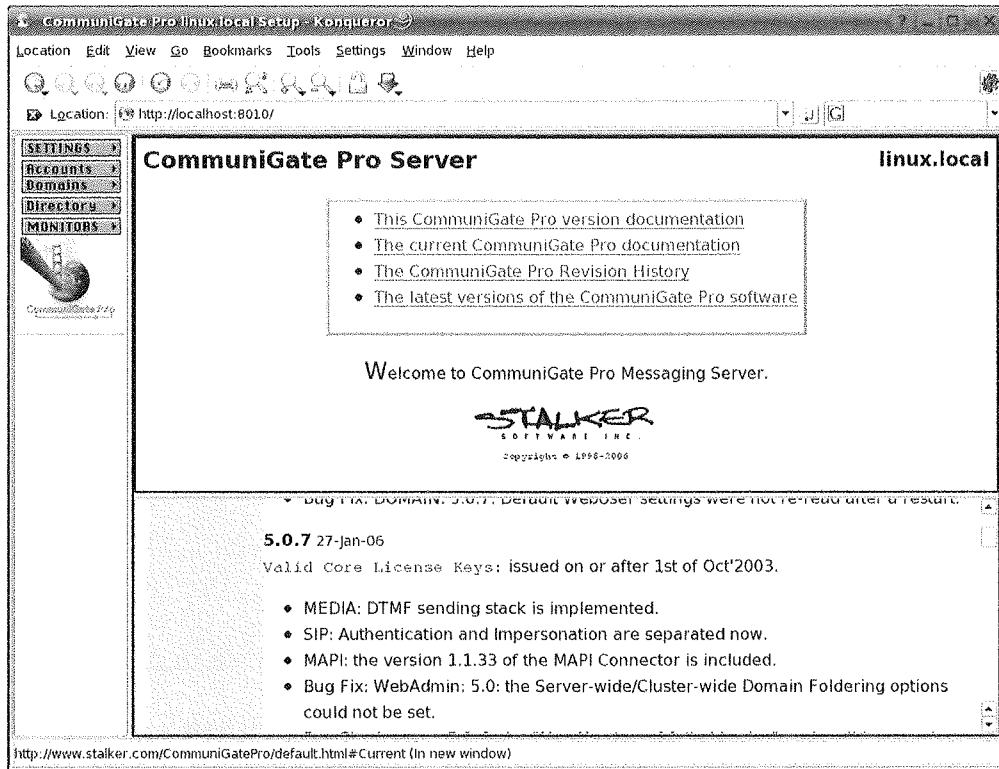
CommuniGate starten mit dem Befehl «./etc/rc.d/CommuniGate start».

Um mit der Konfiguration fortfahren zu können, benötigt man das Initial-Passwort. Dieses findet man unter `/var/CommuniGate/Accounts/postmaster.macnt`. Öffnen der Datei «account.settings» mit einem Editor und notieren des Passworts bei «Password = <zahl>».

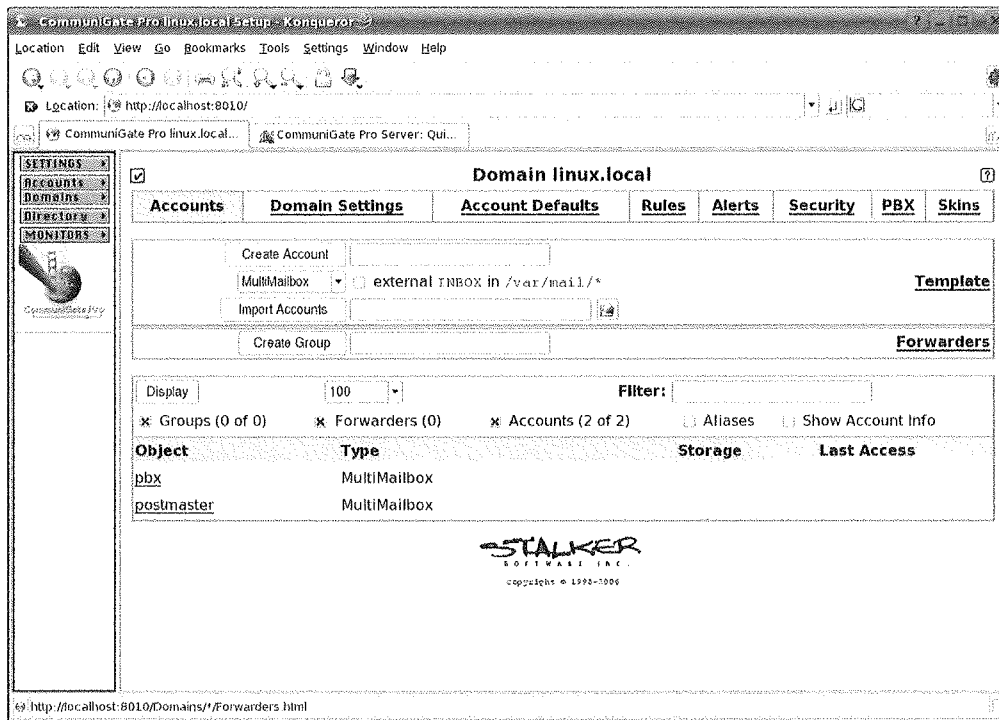


Unter «<http://localhost:8010>» mit dem Benutzernamen «postmaster» und dem notierten Passwort einloggen.



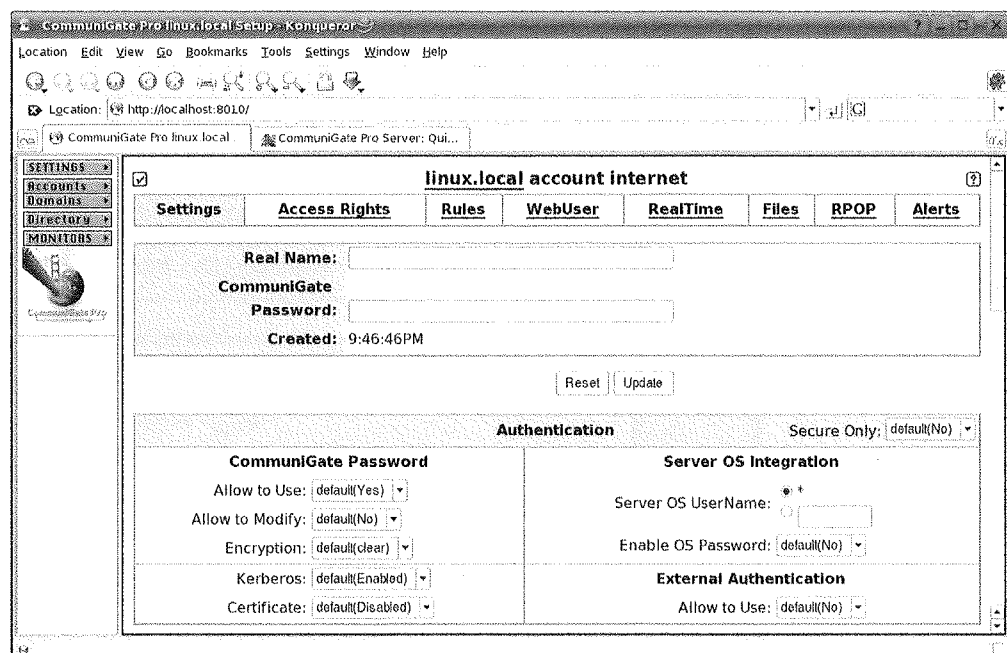


«Accounts» wählen:



Mit »Create Account“ wird der eingetragene Account erstellt. Dieser Account kann nachfolgend mit dem entsprechenden Benutzerkonto auf dem Betriebssystem verknüpft werden; muss aber nicht zwingend und kann völlig separat im CommuniGate verwaltet werden.

Folgende Einstellungen können vorgenommen werden:



Real Name: der Vor- und Nachname des Benutzers

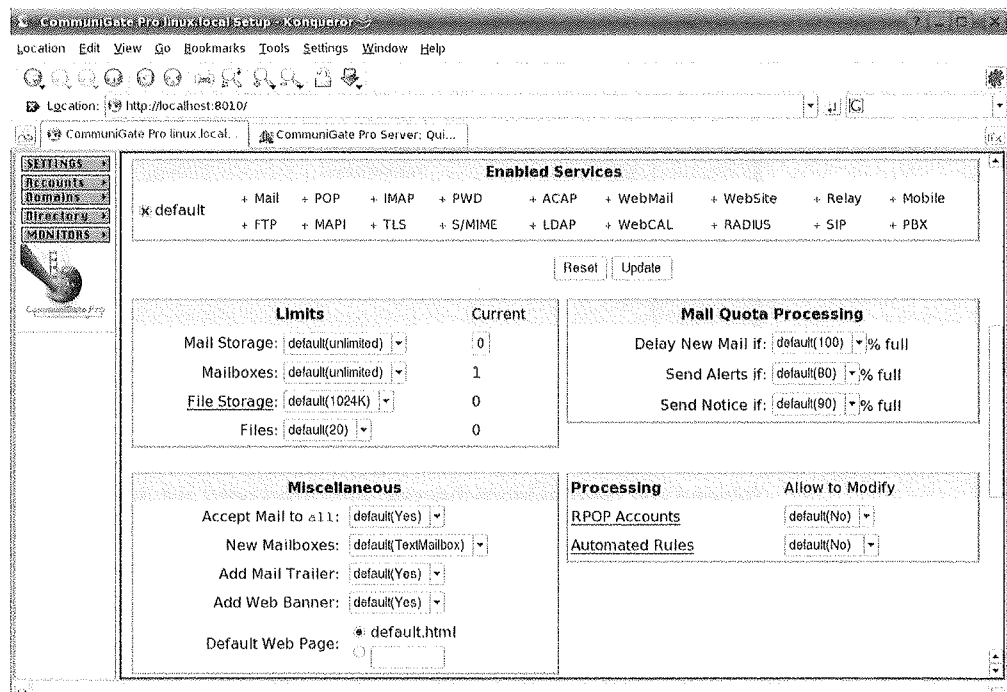
CommuniGate Password: Passwort des Benutzerkontos

CommuniGate Password: Einstellungen zum Passwort des Benutzers

Server OS Integration: Verknüpfung mit Benutzerkonto auf Betriebssystem-Ebene (\* bedeutet, dass der Benutzername auf dem Betriebssystem gleich ist wie der CommuniGate Benutzername). Hier wird auch eingestellt, ob das Passwort des Betriebssystems gelten soll.

External Authentication: weitere externe Authentisierung (unabhängig vom Betriebssystem)

Weitere Einstellungen:



Enabled Services: zeigt die Dienste an, die für diesen Benutzer aktiviert sind (Informationsübersicht, Anpassung/Einstellung erfolgt in anderen Menüpunkten).

Limits: Beschränkungen für das Benutzerkonto (Platz in der Mailbox etc.)

Mail Quota Processing: Nachrichten an den Benutzer, falls Platz aufgebraucht wird

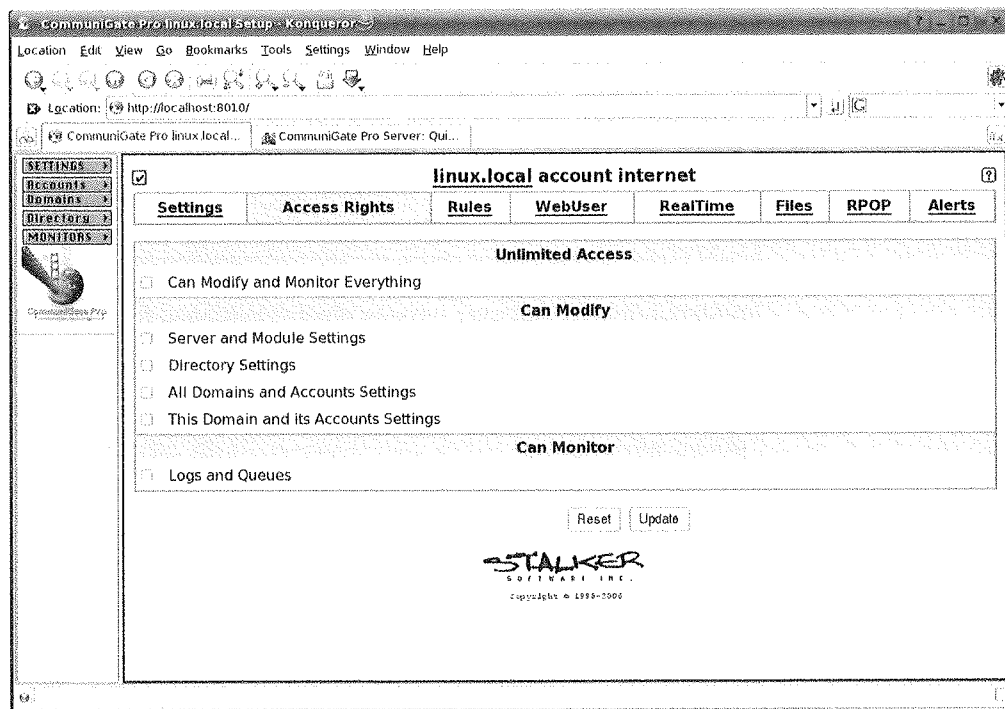
Miscellaneous: Diverse Einstellungen (default belassen)

Processing: Einstellungen, ob der Benutzer POP-Konten und Regeln bearbeiten kann

Weitere Einstellungen:

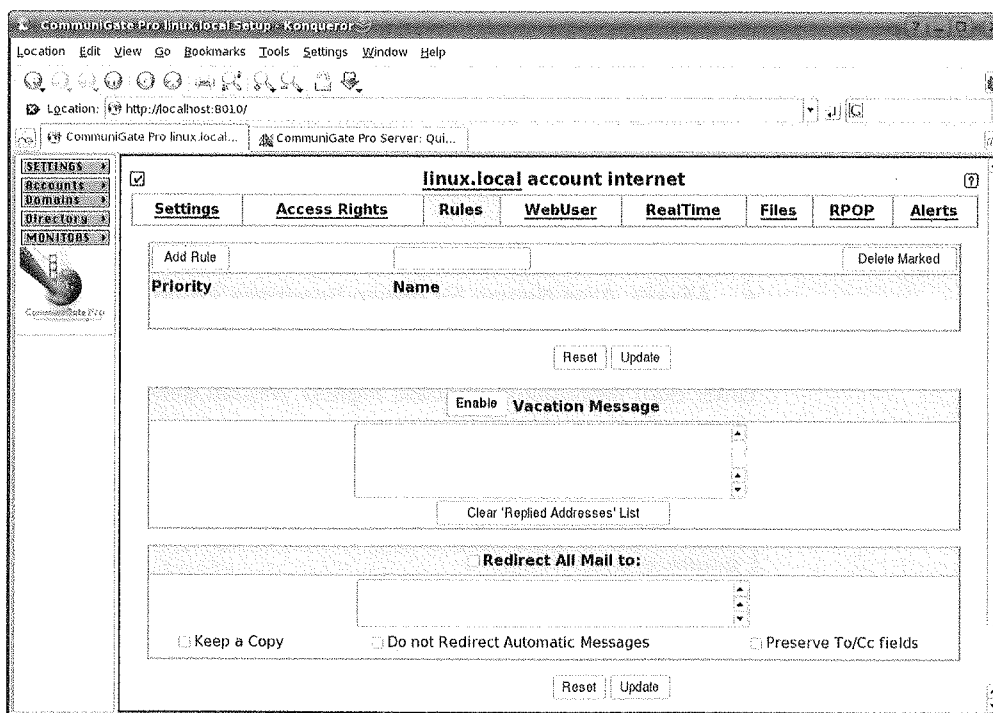
- Aliases: Alias-Namen für diesen Benutzeraccount
- Mailing Lists: Mailinglisten, die unter diesem Benutzerkonto laufen sollen

Unter dem Menüpunkt «Access Rights» werden generelle Zugriffsberechtigungen vergeben:

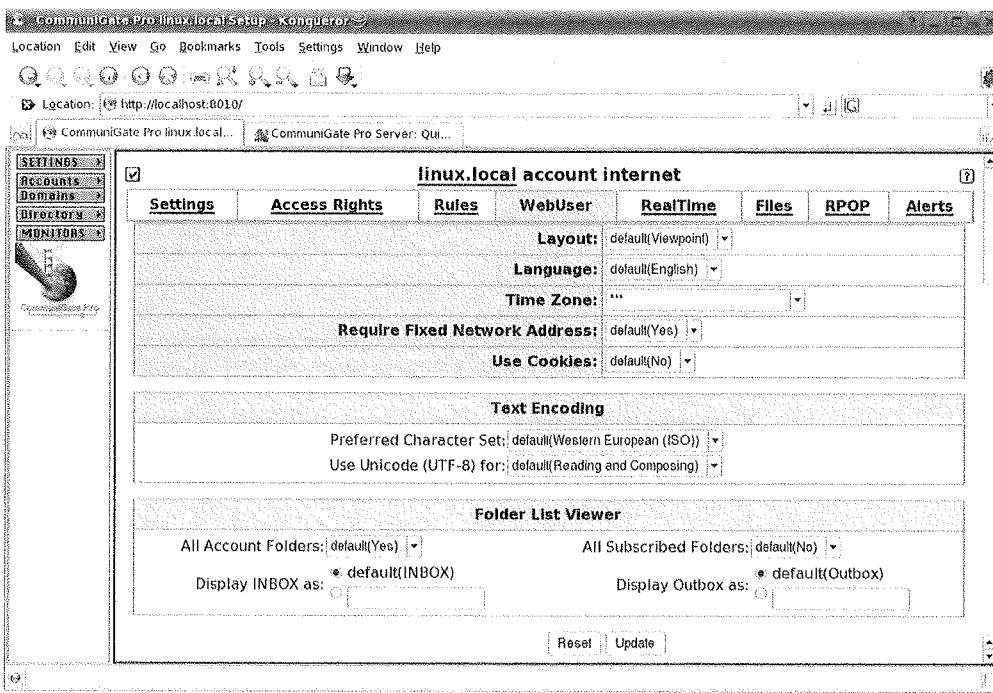


Ein normaler Benutzer soll am besten nichts einstellen können (keine Checkbox aktivieren).

Unter dem Menüpunkt «Rules» können Bearbeitungsregeln für eingehende Nachrichten eingerichtet werden (inklusive «out of office» Meldungen und «redirect» in andere Mailboxen/Postfächer):

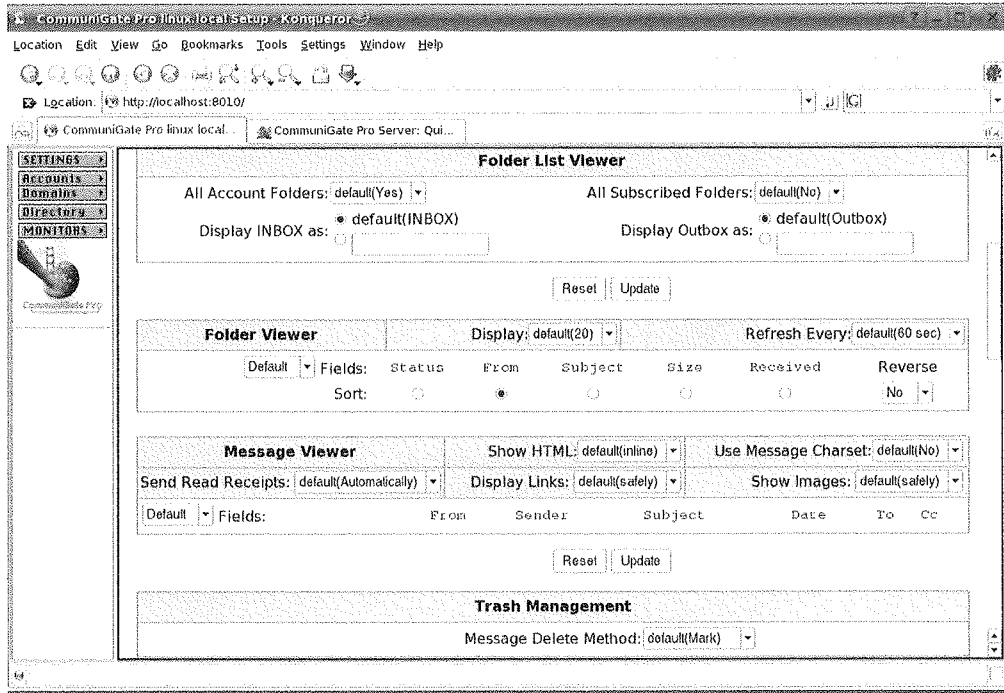


Im Menüpunkt «WebUser» werden die Einstellungen für das Webmail vorgenommen:



Die Einstellungen werden soweit nötig angepasst (Layout des Web-GUI, Language=Sprache, Time Zone = Zeitzone, Require Fixed Network Address = ob der Benutzer eine fixe IP-Adresse für den Zugriff benötigt, Use Cookies = Einsatz von Cookies).

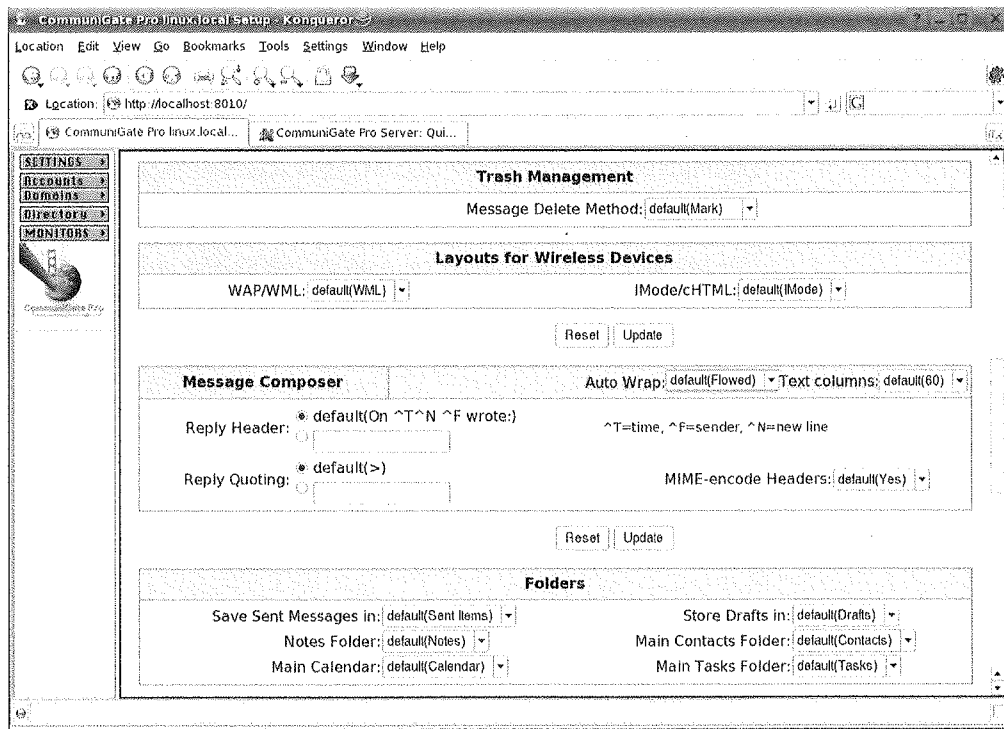
Text Encoding wird ebenfalls in der default-Einstellung belassen.



Folder List Viewer auf default lassen

Folder Viewer auf default lassen

Message Viewer wenn nötig anpassen, sonst default lassen

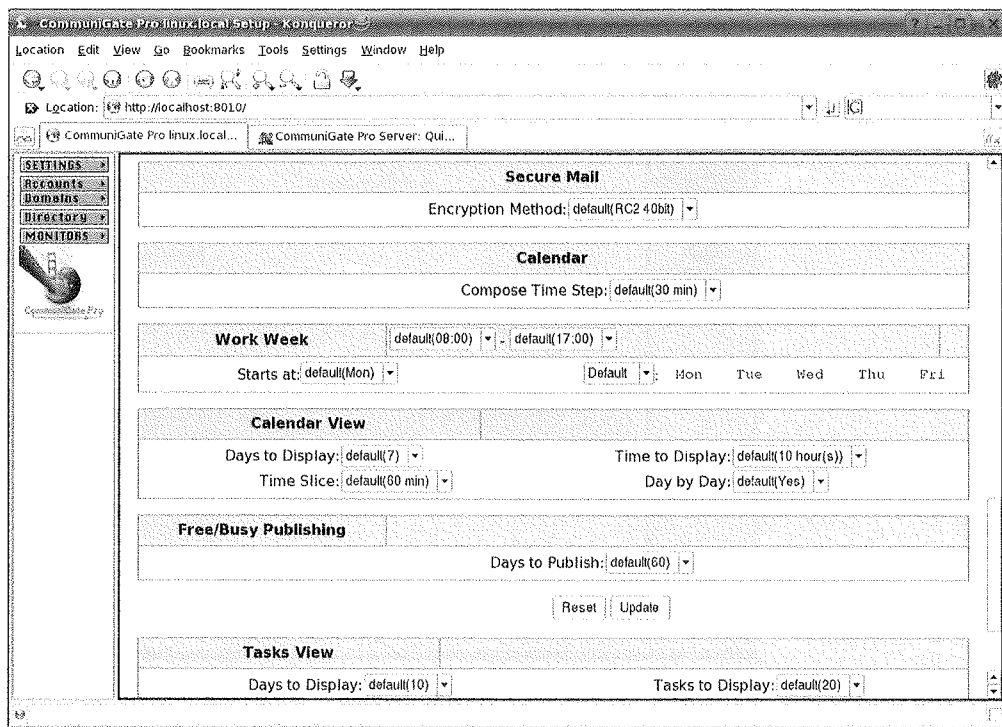


Trash Management: Möglichkeiten zum Löschen von E-Mails durch den Benutzer

Layout for Wireless Device: Aussehen der Webseite bei Zugriff mit mobilen Geräten (Handy, WAP etc.)

Message Composer: Verhalten des Editors für neue Nachrichten

Folders: Default-Einstellungen für die Verzeichnisse des Postfachs



Secure Mail: Konfiguration der Verschlüsselungsmethode für verschlüsselte E-Mails

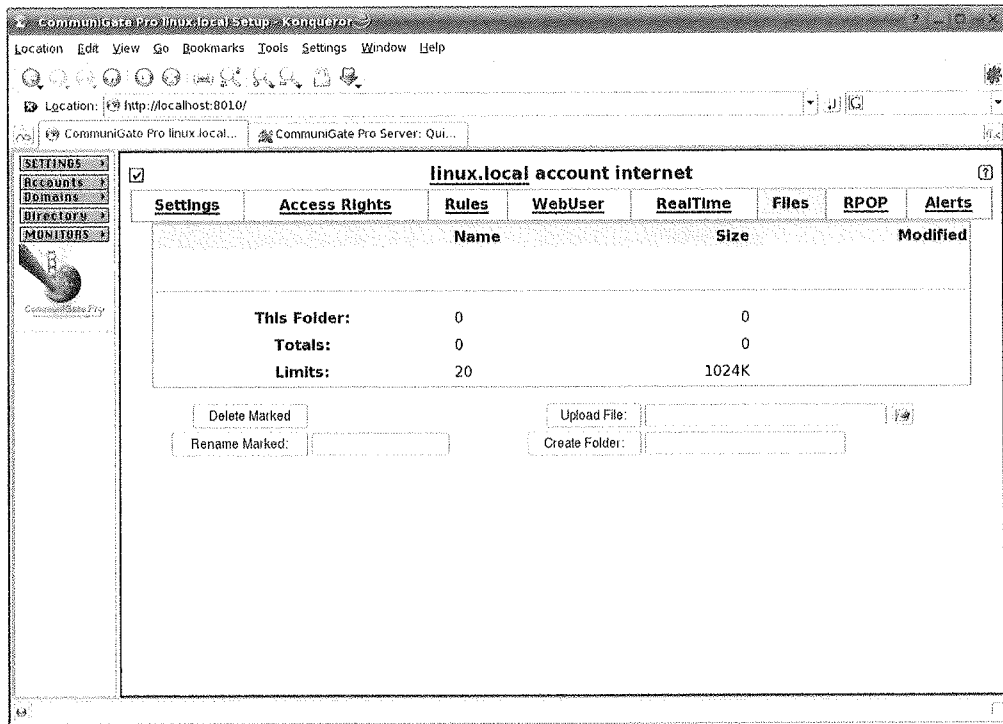
Calendar: Konfiguration der Kalendereinstellungen innerhalb des Postfachs (Webmail), inklusiv «Work Week», «Calendar View» und «Free/Busy Publishing» = Ansicht der Besetzt-/Frei-Zeit für andere Benutzer oder Besucher

Task View: Einstellungen für die ToDo-Tasks

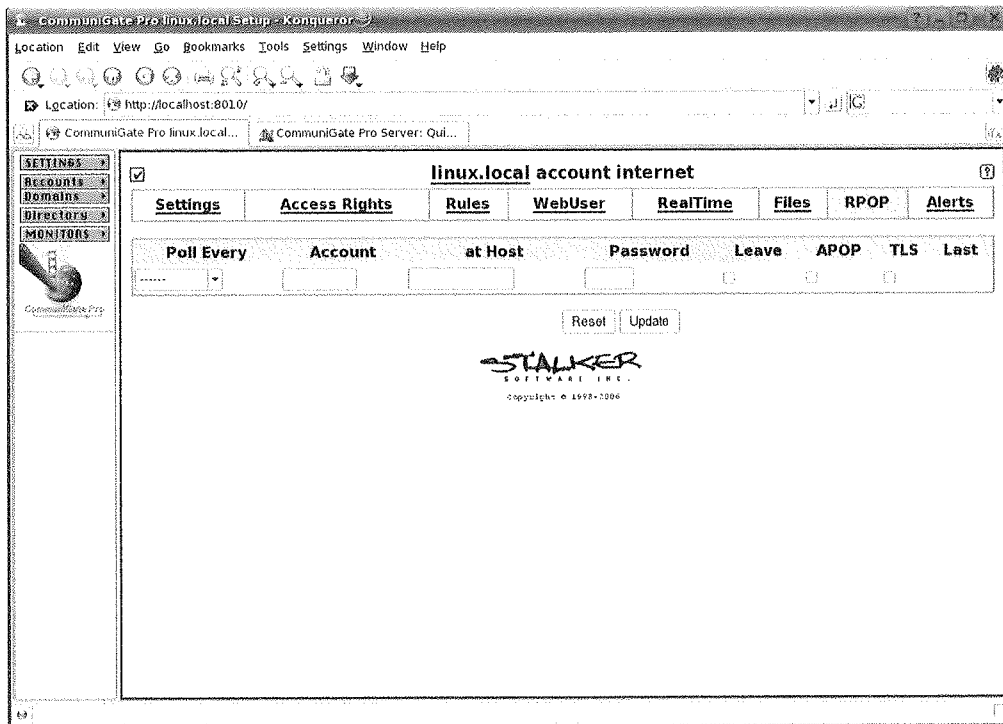
Die Funktionen «Kalender» und «Tasks» wurden anderen gängigen E-Mail-Programmen nachempfunden und sind nur über das Webmail zugänglich.

Der Menüpunkt «RealTime» ermöglicht die Konfiguration von Konferenzen etc. und wird in diesem Modul nicht benötigt.

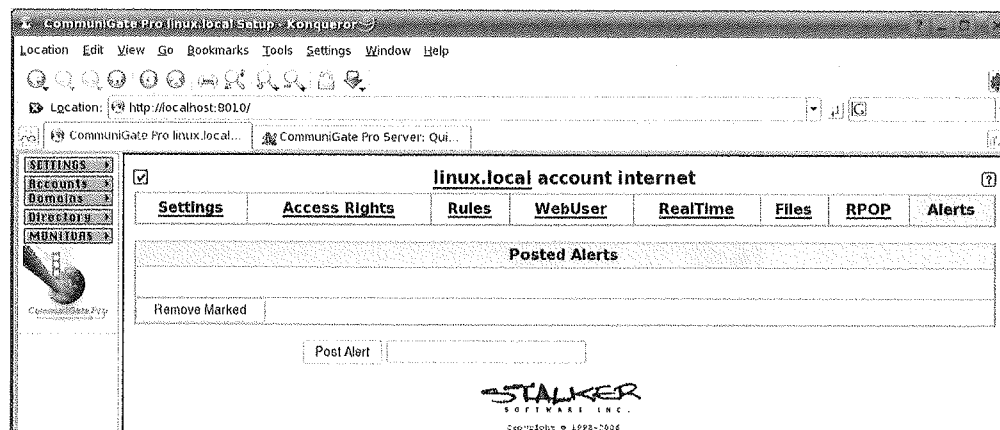
Im Menüpunkt «Files» ist eine Statistik über die Benutzung der Postfächer des Benutzers ersichtlich:



Im Menüpunkt «RPOP» kann eingestellt werden, ob der CommuniGate Server auf anderen E-Mail-Konten mit POP die E-Mails abfragen und im Postfach des Benutzers anzeigen soll:

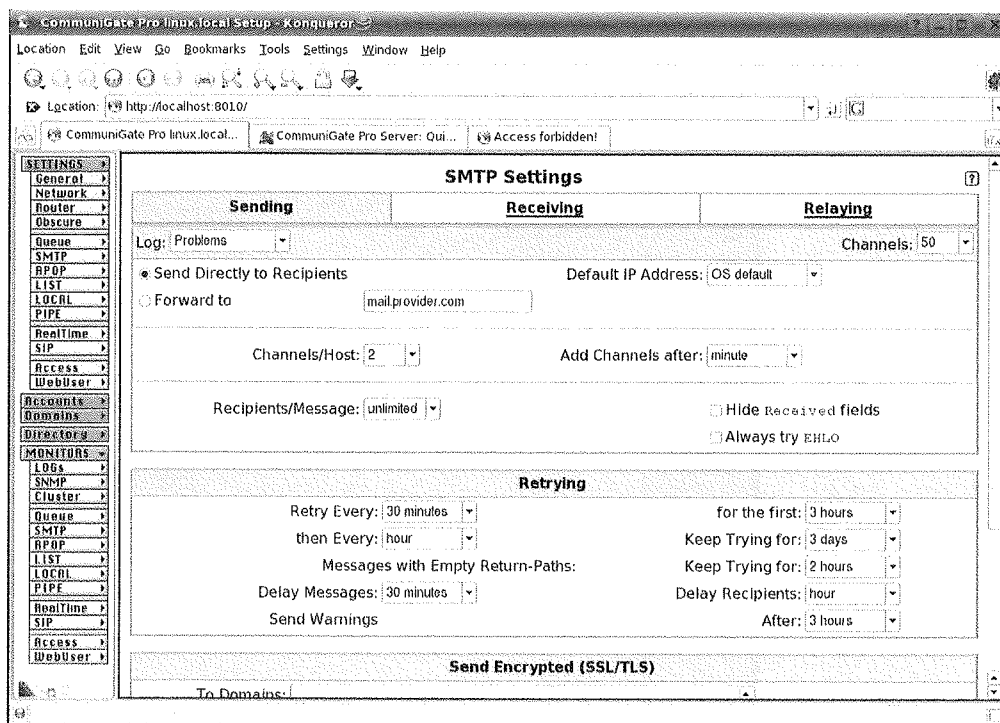


Unter «Alerts» können schliesslich eine oder mehrere Meldungen für den Benutzer erfasst werden (z. B. «bitte alte E-Mails löschen, da Quota bald erreicht wird» etc.):



Nachdem die benötigten Benutzer-Postfächer eingerichtet wurden, muss noch das Passwort des «postmaster» geändert werden, indem auf das entsprechende Konto geklickt wird unter «Accounts» und das «CommuniGate Passwort» geändert wird und mit «Update» bestätigt.

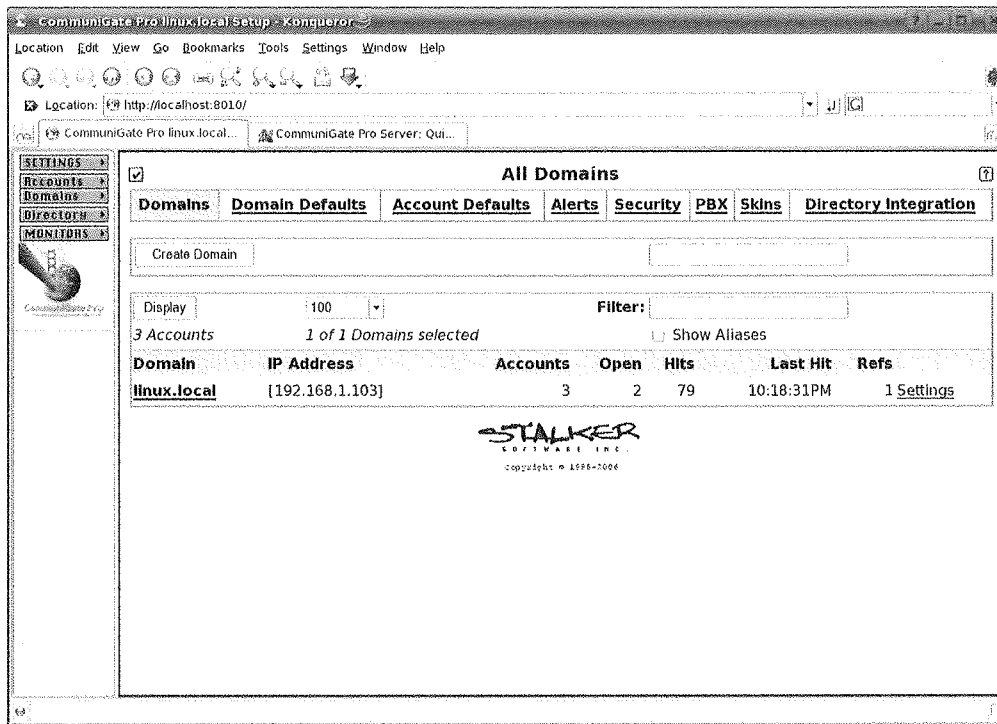
Die Einstellungen für SMTP werden unter «Settings», «SMTP» vorgenommen:



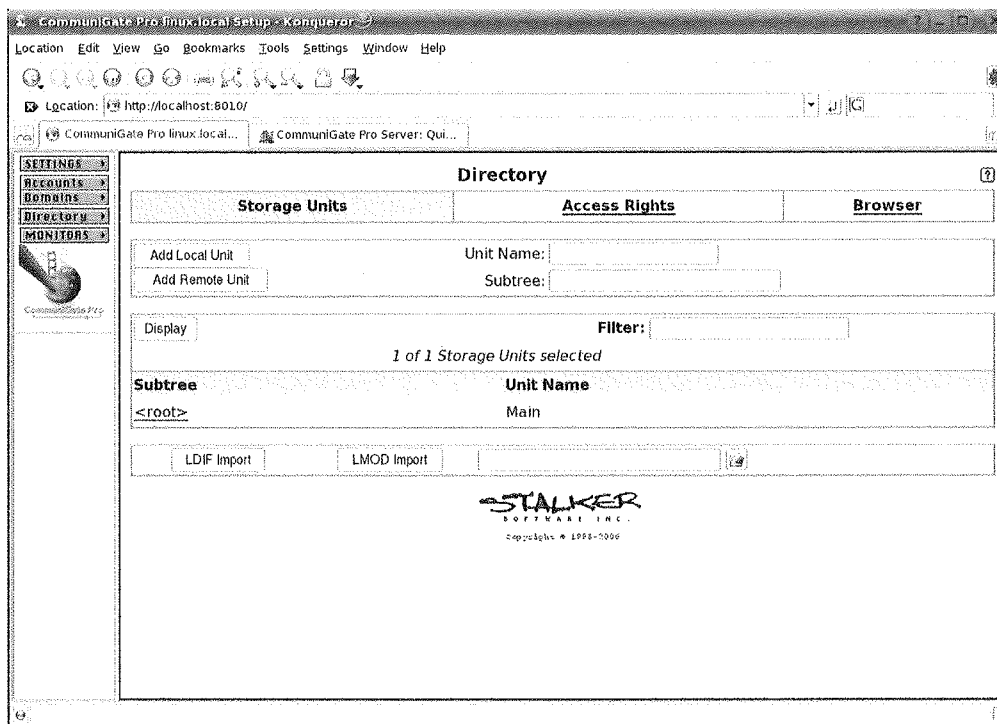
Die Einstellungen unter «Sending», «Receiving» und «Relaying» können grundsätzlich so belassen werden, da diese defaultmässig eine grundlegende Sicherheit bieten.



Unter dem linken Menüpunkt «Domains» können auch zusätzliche Domains konfiguriert werden (mit eigenen Benutzerkontos und Postfächern):



Unter «Directory» können externe Verzeichnisdienste (LDAP basierend) an CommuniGate angeschlossen werden:



Unter dem Menüpunkt «MONITORS» können schliesslich die verschiedenen Ereignisse des Mailserver überwacht werden.

Unter <http://www.communiGate.com/CommuniGatePro/default.html> können weitere detaillierte Anleitungen für die Konfiguration eingesehen werden.

## Repetitionsfragen

- 
- 22 Worauf muss geachtet werden, wenn die Firewall des Betriebssystems ausgeschaltet wird?
- 
- 26 Weshalb empfiehlt sich die Verwendung von Gruppenberechtigungen?
- 
- 4 Wie nennt sich das verschlüsselte Protokoll eines Webserver und wie funktioniert es?
- 
- 9 Was ist ein Zonentransfer in einem DNS-Server?
- 
- 14 Worauf ist beim Betrieb eines FTP-Servers zu achten?
- 
- 19 Welches sind die Probleme beim Betreiben eines Mailserver im Internet?
-

## 9 Log-Services und Sicherungsprozeduren gemäss Lösungsentwurf aufsetzen

---

Die Konfiguration des Logging der einzelnen Dienste ist in den Installationshinweisen beschrieben. Folgende Beschreibungen geben Hinweise auf die grundsätzlichen Aspekte, die beim Logging zu beachten sind.

### 9.1 Logfile-Rotation

---

Durch die Rotation der Logfiles (z. B. täglich, oder auch stündlich, je nach Bedarf) wird sichergestellt, dass ein Logfile einerseits nicht übermässig gross wird, andererseits kann so schnell nachvollzogen werden, zu welchem Zeitpunkt Einträge gemacht wurden, um Probleme aufzuspüren. Die meisten Dienste erlauben es, solche Konfigurationen vorzunehmen. Falls dies nicht möglich ist, erlauben spezielle Logserver-Dienste eine Rotation der Logfiles.

Das am meisten benutzte Logging-Protokoll ist «syslog», es bestehen verschiedene Server-Implementationen davon.

### 9.2 Externer Logserver

---

Ein externer Logserver erlaubt es, Logfiles zentral aufzubewahren und zu archivieren. Zudem können so auf einfache Weise zusammenhängende Applikationen und Systeme «debugged» werden, da alle Log-Informationen zentral zur Verfügung stehen. Ein Problem, das vom Internetserver verursacht wird, und auf anderen Servern ebenfalls zu Problemen führt, kann so schnell nachvollzogen werden, indem die Meldungen des einzelnen Zeitpunkts in den verschiedenen Systemen gesucht werden.

Es ist deshalb wichtig, dass alle Systeme eine synchronisierte Zeitangabe benutzen (dazu wird oft das Network Time Protocol NTP eingesetzt). Ansonsten werden die Ereignisse zwar zentral gelogged, unterscheiden sich aber, da die Systeme nicht die synchrone Zeit haben. Zusammenhängende Ereignisse können so nur viel schwieriger nachvollzogen werden.

Ein weiterer Vorteil des zentralen Logserver ist der Schutz der Logfiles vor Manipulation. Wird der Internetserver angegriffen und versucht der Angreifer, die Logfiles zu ändern (um Spuren des Angriffs zu verwischen), so kann er im Normalfall nicht auf den Logserver zugreifen. Der Logserver kann die Log-Dateien auch zentral sichern, da alle auf dem gleichen System vorhanden sind und so auch weniger Speicherplatz auf dem Internetserver benötigen.

### 9.3 Logfile-Analyse

---

Statistiken aus Logfiles? Wer Logfiles genau analysiert, kommt schnell zu überraschenden Ergebnissen über die eigenen Besucher, wo Sie herkommen, und was sie auf Ihren Seiten finden wollen.

Jeder Hit wird aufgezeichnet. Dabei werden nicht nur die IP des Besuchers gespeichert, nein sogar der verwendete Browser kann ermittelt werden. Die Logfiles sind die Speicherplätze für diese Daten. In ihnen wird jede Anfrage an den Server protokolliert und abgespeichert. Wer auf diese Statistiken Zugriff hat, kann sehr viel über die Entwicklung des Besucherstroms, und sogar die Zielgruppe ermitteln.

Die Logfiles sind meistens im selben Verzeichnis wie der ROOT selbst. Sie sind einfache Textdateien und können mit jedem Editor ausgelesen werden.

Jeder Aufruf an den Server wird protokolliert. Es kann so bestimmt werden, von wo der Besucher kam, welchen Browser er verwendet, und wie viele Seiten er angeschaut hat. Jeder Aufruf steht in der Logfile in einer eigenen Zeile. Eine Zeile könnte z. B. so aussehen:

```
192.168.156.36 - [20/Jan/2005:19:35:09 +0100] «GET / HTTP/1.1» 200 25641 www.dev-  
mag.net «http://www.devmag.net/» «Mozilla/4.0 (compatible; MSIE 5.5; Windows ME;  
DigExt)»
```

Diese Zeile beschreibt einen kompletten Aufruf der Seite. Der Zeilen-Inhalt scheint eher ungeordnet, doch besteht er aus einer festen Struktur. Der erste Teil ist die IP-Adresse des Rechners, welcher den Aufruf getätigt hat. Die IP-Adresse ist eine wandelnde Nummer. Bei jeder neuen Internetverbindung wird dem Computer vom jeweiligen Internetanbieter eine neue IP-Adresse aus einem Pool gegeben. Diese IP-Adresse ist innerhalb dieser Session einmalig. Diese einmalige Nummer erlaubt die Kommunikation zwischen den verschiedenen Rechnern. Auf diese Angabe folgt ein Bindestrich. Nach diesem werden nähere Informationen zum Aufruf der Seite gemacht. Zunächst kommt das Datum und die Uhrzeit, es steht in []-Klammern. Es ist in dem amerikanischen Standard angegeben Tag/Monat/Jahr. Getrennt von einem Doppelpunkt kommt die genaue Angabe des Zeitpunkt des Aufrufs. Diese Angabe ist im GMT Zeitformat. Bei dem Beispiel kommt der Besucher aus dem Raum, in dem die MEZ gilt, deshalb muss eine Stunde addiert werden, dies geschieht durch das +0100. Die konkrete Zeit des Aufrufs war also 20:35:09 Uhr. Zur Sommerzeit in unseren «Gefilden» beträgt die Zeitverschiebung +0200, also zwei Stunden.

Die nächste Angabe spezifiziert den Aufruf. Die Methode GET legt fest, dass die Daten vom Server an den Client gesendet wurden, nach dieser Angabe steht das Protokoll, mit welchem die Daten kodiert werden. Es ist hier das HTTP Protokoll. Es kann sein, dass als Methode in den Logfiles auch ein HEAD auftaucht. Diese Methode wird vor allem von Suchmaschinen verwendet, die dadurch nur Daten zu der angeforderten Datei erhalten. Dies kann das letzte Änderungsdatum des Dokuments sein. Mit diesem Datum wird dann abgewägt, ob die Seite neu indexiert wird. Nach der Angabe der Methode und des Protokolls folgt der Rückgabecode des Servers. Ist der Seitenaufruf geglückt, dann wird als Rückgabecode 200 zurückgegeben.

#### Weitere Rückgabecodes

- 200 OK: Der Request wurde erfolgreich durchgeführt.
- 204 No Content: Das Dokument, welches angefordert wird, enthält keine Daten.
- 206 Partial Content: Die Übertragung wurde unterbrochen. Dies kann vom Browser aus geschehen, oder bei einem Update der Seite.
- 300 Multiple Choices: Es gibt mehrere (ähnliche) Dateien. Der Server kann die Datei nicht eindeutig ermitteln, und bietet mehrere Auswahlmöglichkeiten.
- 301 Moved Permanently: Die Datei wurde an einen anderen Ort verschoben.
- 304 Not Modified: Die Datei wird komplett aus dem Cache (server- und/oder clientseitig) geladen.
- 400 Bad Request: Der Webserver «versteht» die Anfrage nicht.
- 401 Unauthorized: Sie sind nicht autorisiert, diesen Bereich zu betreten.
- 403 Forbidden: Der Zugriff auf die angeforderte Datei wird verweigert.
- 404 Not Found: Die Datei wurde nicht gefunden (ist nicht vorhanden), oder der URL wurde falsch eingegeben.
- 500 Internal Server Error: Ein unbekannter Server-Fehler ist aufgetreten. Oftmals entstehen diese durch falsche Anwendung von .htaccess-Dateien, oder durch Fehler im CGI.
- 503 Service Unavailable: Der Server kann die Anfrage zeitweilig nicht bearbeiten, z. B. bei Wartungsarbeiten.

Zurück zu der Logfile-Analyse:

Auf den Rückgabecode folgt eine Zahl. Sie gibt die genau übertragene Datenmenge in Bytes an. Diese Zahl entspricht also der Dateigrösse. Danach folgt der URL zu dem Dokument, welches aufgerufen wurde. Der URL weist auf den Root; es war also ein direkter Request. Auf diesen URL folgt der URL der Seite, auf welcher sich der Besucher zuletzt befand. Bei einer direkten Anfrage entfällt diese Angabe.

Bei einer indirekten Anfrage kommt man z. B. über einen Link einer anderen Seite zu der Seite, hier steht dann der URL der Seite, von welcher man auf die andere Seite gekommen ist. Diese Seite bezeichnet man als Referer-Seite.

Die folgenden Angabe geben nähere Informationen zu dem Client, bzw. zu dem System von welchem der Aufruf getätigt worden ist. Diese Angaben erstrecken sich von dem verwendeten Browser bis zu dem Betriebssystem. In dem Beispiel verwendet der Besucher, leicht zu erkennen, den Internet Explorer in der Version 5.5. Zudem arbeitet er mit Windows ME als Betriebssystem. Kommt der Request von einem Spider, oder von einem Robot, dann steht hier der Name des jeweiligen Spiders oder Robots.

Da jeder Hit nach diesem oder einem ähnlichem Muster aufgebaut ist, wird auch die Analyse fast zu einem Kinderspiel. Es gibt Programme bzw. Skripte, die jeden Hit auslesen, und in ihre Bestandteile auseinandernehmen, und dann in einer hübschen, übersichtlichen Statistik wieder zusammensetzen. Komplexere Statistiksysteme ermitteln zudem oftmals von dem Besucher über JavaScript weitere Daten, wie z. B. die Bildschirmauflösung oder ähnlich. Somit lässt sich leicht ermitteln, für welche Besuchergruppe eine Seite optimiert werden sollte.

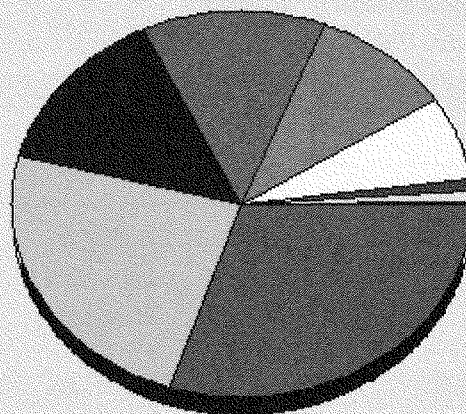
Ein kostenloses Skript welches die Logfiles analysiert ist Webalizer: [www.webalizer.org](http://www.webalizer.org)

### Beispiele von Webalizer-Reports

Monats-Statistik für Maerz 2004		
Summe Anfragen		313
Summe Dateien		236
Summe Seiten		195
Summe Besuche		83
Summe kb		5784
Summe unterschiedlicher Rechner (IP-Adressen)		73
Summe unterschiedlicher URLs		49
	<b>Schnitt</b>	<b>Maximum</b>
Anfragen pro Stunde	1	20
Anfragen pro Tag	44	81
Dateien pro Tag	33	60
Seiten pro Tag	27	66
Besuche pro Tag	11	17
kb pro Tag	826	1842
Anfragen nach Status-Code		
Code 200 - OK		236
Code 206 - Unvollständiger Inhalt		4
Code 301 - Seite dauerhaft an anderer Stelle		2
Code 304 - Seite nicht geändert		23
Code 404 - Seite nicht gefunden!		48

5	9	2.88%	9	3.81%	494	8.53%	1	1.20%	212.204.49.194
6	9	2.88%	7	2.97%	219	3.79%	1	1.20%	ferret.poly.edu
7	8	2.56%	4	1.69%	145	2.50%	4	4.82%	gerhard5.bis.uni-oldenburg.de
8	8	2.56%	8	3.39%	108	1.87%	1	1.20%	p3e9e92b9.dip0.t-ipconnect.de
9	8	2.56%	4	1.69%	15	0.26%	0	0.00%	slovo-out.yandex.ru
10	7	2.24%	0	0.00%	2	0.04%	0	0.00%	lj1154.inktomisearch.com
11	6	1.92%	4	1.69%	21	0.37%	0	0.00%	crawl8.googlebot.com
12	6	1.92%	5	2.12%	248	4.29%	1	1.20%	hg-73155.euv-frankfurt-o.de
13	5	1.60%	5	2.12%	248	4.28%	1	1.20%	213.181.81.140
14	5	1.60%	1	0.42%	4	0.07%	2	2.41%	crawler2.googlebot.com
15	4	1.28%	4	1.69%	86	1.48%	2	2.41%	216.185.57.134
16	4	1.28%	4	1.69%	85	1.48%	2	2.41%	216.185.57.226
17	4	1.28%	4	1.69%	41	0.71%	1	1.20%	pd9e1e7e5.dip.t-dialin.net
18	4	1.28%	4	1.69%	47	0.81%	1	1.20%	pd9e9d364.dip0.t-ipconnect.de
19	4	1.28%	4	1.69%	41	0.71%	1	1.20%	pd9ef1f3e.dip.t-dialin.net
20	4	1.28%	3	1.27%	9	0.15%	1	1.20%	wfp2.almaden.ibm.com
21	3	0.96%	3	1.27%	12	0.20%	0	0.00%	pd9e8e14b.dip.t-dialin.net
22	2	0.64%	2	0.85%	41	0.71%	1	1.20%	l.kpmg.de
23	2	0.64%	2	0.85%	41	0.71%	1	1.20%	194.242.42.20
24	2	0.64%	2	0.85%	41	0.71%	1	1.20%	194.76.232.148
25	2	0.64%	2	0.85%	34	0.60%	1	1.20%	216.185.57.106
26	2	0.64%	2	0.85%	48	0.82%	1	1.20%	216.185.57.110
27	2	0.64%	1	0.42%	4	0.07%	1	1.20%	216.185.57.146
28	2	0.64%	1	0.42%	4	0.07%	1	1.20%	217.160.250.226
29	2	0.64%	1	0.42%	2	0.03%	1	1.20%	64.242.88.50
30	2	0.64%	2	0.85%	41	0.71%	1	1.20%	ab-42041.euv-frankfurt-o.de

Anfragen aus Laendern in Monat Maerz 2004



USA-Univers./Schulen (30%)  
 Deutschland (24%)  
 Firmen (COM) (14%)  
 Unbekannte Adressen (13%)  
 Netzwerke (NET) (10%)  
 Russische Foederation (7%)  
 Japan (1%)  
 Litauen (1%)

## 9.4 Sicherungsprozeduren und Back-up

---

Der Internetserver ist in regelmässigen Abständen zu sichern (Back-up). Dazu werden entweder die zu sichernden Daten in regelmässigen Abständen auf ein anderes System kopiert, oder eine automatisierte Back-up-Applikation führt die Sicherung selbstständig durch. Dabei ist es wichtig, dass die Sicherungsprozedur nicht während der Hauptverkehrszeiten des Internetservers durchgeführt werden (da oft Ressourcen auf dem System benötigt und z. T. die Applikationen während der Sicherung abgeschaltet werden müssen, damit die Datenintegrität gewährleistet ist). Idealerweise erfolgt sie spät abends oder nachts, damit länger dauernde Sicherungsprozeduren genügend Zeit zum Durchlaufen haben.

Back-up-Prozeduren sollten grundsätzlich zur Zeit der tiefsten Last auf dem System durchgeführt werden. Falls das System während dieser Zeit nicht überwacht wird (z. B. nachts), so ist am morgen genügend Zeit für Notfallmassnahmen einzuplanen.

### Repetitionsfragen

---

23 Was ist beim Protokollieren (logging) speziell zu beachten?

---

27 Die Datensicherung/Back-up sollte zu welcher Zeit NICHT durchgeführt werden?

---

## 10 Fremde Ressourcen anbinden

---

Bei der Verbindung des Internetserver mit fremden Ressourcen/Diensten auf anderen Servern sind die folgenden wichtigen Punkte zu beachten:

- Erreichbarkeit der Server und Authentifizierung/Autorisierung des Internetserver (d. h., darf der Internetserver überhaupt Anfragen an den fremden Server stellen, z. B. wenn er nicht innerhalb der gleichen Domain ist und somit nicht erkannt wird). Zudem werden für Authentifizierungsanfragen (Windows mit NTLM, LDAP-Requests) und Datenbankverbindungen oft proprietäre Protokolle und Ports verwendet. Diese müssen auf einer dazwischen liegenden Firewall entsprechend freigeschaltet werden. Solche Verbindungen verursachen oft grosse Risiken, weshalb die Anbindung an fremde Ressourcen im Voraus abgeklärt werden muss, um nicht gegen vorhandene Sicherheitsregelungen zu verstossen.
- Sicherheit/Vertraulichkeit der Verbindungen: Da über Anbindungen an Fremdsysteme oft wichtige/vertrauliche Daten ausgetauscht werden, sind solche Verbindungen mit entsprechenden Sicherheitsmassnahmen aufzusetzen

### 10.1 Authentifizierungsserver

---

Oft wird die Benutzerverwaltung nicht auf dem Internetserver, sondern auf einem dafür eingerichteten Server vorgenommen. Typischerweise kommen folgende Authentifizierungsdienste zum Einsatz:

- Microsoft Active Directory
- Lightweight Directory Access Protocol (LDAP)

Beide funktionieren grundsätzlich über Port 389, wobei Active Directory oft noch proprietäre Windows-Authentifizierung unterstützt und deshalb weitere Ports benötigt (137–139, 445).

Für den Internetserver lohnt sich die Installation eines separaten Verzeichnisdienstes (in der DMZ), da externe Benutzer auf einem internen Verzeichnisdienst ein zusätzliches Risiko darstellen.

### 10.2 Datenbanken

---

Moderne Datenbanken werden heute über TCP/IP verbunden. Somit gehorchen die Datenbankverbindungen oft ähnlichen Regeln wie andere TCP/IP-Verbindungen (Routing, DNS-Auflösung). Oft tauschen Datenbank-Managementsysteme (DBMS) Steuerkommandos mit den Applikationen aus, welche in extrem kleinen Paketen versendet werden. Dies kann bei verschlüsselten Verbindungen zu Verzögerungsproblemen führen; die Performance kann dadurch gestört werden.



### 10.3 Sicherheitsaspekte bei fremden Ressourcen

---

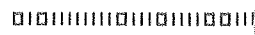
Die Anbindung von Verzeichnisdiensten oder Datenbanken an den Internetserver ist mit zusätzlichen Risiken verbunden. Ein Angreifer kann den Internetserver kompromittieren und erhält so möglicherweise Zugriff auf Benutzer- oder Kundendaten sowie weitere Firmendaten in der Datenbank.

Um diese Risiken zu verringern, empfiehlt sich die Installation von entsprechenden Sicherheitsmassnahmen zwischen Internetserver und Datenbank (Intrusion Detection System, Reverse Proxy/Web Entry Server, Firewall). Die Verbindungen sind vorzugsweise zu verschlüsseln (wenn möglich) und detailliert zu loggen, um jederzeit nachvollziehen zu können, welche Zugriffe (durch welche Benutzer) vorgenommen wurden.

#### Repetitionsfrage

---

- 5 Soll für den Internetserver ein zentraler Authentifizierungsdienst (z. B. Active Directory im LAN) benutzt werden oder ein dedizierter Dienst (z. B. in der DMZ)?
-





## Einleitung, Lernziele und Schlüsselbegriffe

---

### Einleitung

---

Im Teil D geht es um den Betrieb und die Wartung des Internetservers sowie um die Abnahme des Internetservers durch den Auftraggeber. Für die Wartung werden die Anforderungen festgelegt und verschiedene Möglichkeiten zur Überwachung des Internetservers aufgezeigt. Die Systemabnahme wird zusammen mit dem Auftraggeber durchgeführt und in einem Abnahmeprotokoll dokumentiert. Das Abnahmeprotokoll bildet somit den Abschluss des Projekts Internetserver.

### Lernziele und Lernschritte

---

Lernziele	Lernschritte
<input type="checkbox"/> Kennt mindestens eine Testmöglichkeit und ein Testprogramm für die wichtigen Anforderungen.	<ul style="list-style-type: none"> <li>• Welche Testmöglichkeiten und -methoden werden für einen Internetserver eingesetzt?</li> <li>• Welche Testprogramme stehen für Internetserver zur Verfügung?</li> </ul>
<input type="checkbox"/> Ist in der Lage, die Messergebnisse bez. Belastungsprofil, Datenvolumen, erforderlicher Dienste, zu integrierender Applikationen in einem Abnahmeprotokoll zusammenzufassen.	<ul style="list-style-type: none"> <li>• Wie ist ein Abnahmeprotokoll zu strukturieren?</li> <li>• Welche Informationen werden im Abnahmeprotokoll aufgenommen und beschrieben?</li> <li>• Wer stellt das Abnahmeprotokoll aus und unterschreibt es?</li> </ul>

### Schlüsselbegriffe

---

Update, Upgrade, Dienstprogramm, Monitoring, Prozess

## 11 Anforderungen an die Wartung definieren

---

Nachdem der Internetserver vollständig aufgesetzt ist, müssen noch (innerhalb der Betriebsdokumentation) die Anforderungen an die Wartung des Servers festgehalten werden.

### 11.1 Betrieb des Internetservers

---

Der Internetserver wird typischerweise während 24h betrieben. Dies hat entsprechend Einfluss auf die Wartung des Systems, da ein Ausfall grundsätzlich bedeutet, dass die Dienstleistungen des Servers nicht zur Verfügung stehen. In einem Service Level Agreement mit dem Kunden ist deshalb festzuhalten, wann die Verfügbarkeit garantiert wird und wie lange ein Ausfall höchstens dauern darf.

#### 11.1.1 Updates und Upgrades

---

Um Hard- und Software-Updates (Patches) und -Upgrades (Versionsänderungen) durchzuführen, müssen die Dienste für kurze Zeit ausser Betrieb genommen werden. Solche Ausfallzeiten sind vorzeitig an alle betroffenen Anwender zu kommunizieren, damit frühzeitig Massnahmen zur Vermeidung von Problemen oder Ausfällen getroffen werden können.

Wichtig ist auch, ein Ausweichkonzept vorzubereiten, um reagieren zu können, falls der Update/Upgrade nicht erfolgreich beendet werden kann. Dann wird entweder ein Rollback (zurück auf funktionierende Version) oder eine Neuinstallation bzw. ein Ausweichen auf ein Ersatzsystem durchgeführt.

### 11.2 Dienstprogramme zur Systemüberwachung in SuSE Linux

---

In diesem Kapitel werden verschiedene Programme und Mechanismen vorgestellt, mit denen Sie den Zustand Ihres Systems untersuchen können. Zudem werden einige für die tägliche Arbeit nützliche Dienstprogramme sowie deren wichtigste Optionen beschrieben. Für die vorgestellten Befehle werden jeweils beispielhafte Ausgaben dargestellt. Darin ist die erste Zeile der Befehl selbst (nach einem Dollarzeichen als Eingabeaufforderung). Auslassungen sind durch eckige Klammern ([ . . . ]) gekennzeichnet und lange Zeilen werden, falls erforderlich, umgebrochen. Umbrüche langer Zeilen sind durch einen umgekehrten Schrägstrich (\) gekennzeichnet.

```
$ command -x -y
```

```
output line 1
```

```
output line 3 is annoyingly long, so long that \
```

```
we have to break it
```

```
output line 3
```

```
[...]
```

```
output line 99
```

Damit möglichst viele Dienstprogramme erwähnt werden können, sind die Beschreibungen kurz gehalten. Weitere Informationen zu allen Befehlen finden Sie auf den ent-

sprechenden Manualpages. Die meisten Befehle verstehen auch die Option `--help`, mit der Sie eine kurze Liste der verfügbaren Parameter anzeigen können.

### 11.2.1 Liste der geöffneten Dateien: `lsdf`

Um eine Liste aller Dateien anzuzeigen, die für den Prozess mit der Prozess-ID `PID` geöffnet sind, verwenden Sie `-p`. Um beispielsweise alle von der aktuellen Shell verwendeten Dateien anzuzeigen, geben Sie Folgendes ein:

```
$ lsdf -p $$

COMMAND PID USER FD TYPE DEVICE SIZE NODE NAME
zsh 4694 jj cwd DIR 0,18 144 25487368 /suse/jj/t
(totan:/real-home/jj)
zsh 4694 jj rtd DIR 3,2 608 2 /
zsh 4694 jj mem REG 3,2 11648 20610
/usr/lib/zsh/4.2.0/zsh/rlimits.so
[...]
zsh 4694 jj mem REG 3,2 13647 10891 /lib/libdl.so.2
zsh 4694 jj mem REG 3,2 1349081 10908 /lib/tls/libc.so.6
zsh 4694 jj mem REG 3,2 56 12410
/usr/lib/locale/de_DE.utf8/LC_TELEPHONE
[...]
zsh 4694 jj mem REG 3,2 59 14393
/usr/lib/locale/en_US/LC_NUMERIC
zsh 4694 jj mem REG 3,2 56444 20598
/usr/lib/zsh/4.2.0/zsh/computil.so
zsh 4694 jj 0u CHR 136,48 50 /dev/pts/48
zsh 4694 jj 10u CHR 136,48 50 /dev/pts/48
```

Es wurde die spezielle Shell-Variablen `$$` verwendet, deren Wert die Prozess-ID der Shell ist. Wird der Befehl `lsdf` ohne Parameter eingegeben, werden alle aktuell geöffneten Dateien angezeigt. Da dies in der Regel recht viele sind, wird dieser Befehl selten verwendet. Die Liste der Dateien kann jedoch mit Suchfunktionen kombiniert werden, um sinnvolle Listen zu generieren. Beispiel: Liste aller verwendeten zeichenorientierten Geräte:

```
$ lsdf | grep CHR

sshd 4685 root mem CHR 1,5 45833 /dev/zero
sshd 4693 jj mem CHR 1,5 45833 /dev/zero
zsh 4694 jj 0u CHR 136,48 50 /dev/pts/48
X 6476 root mem CHR 1,1 38042 /dev/mem
```

```
ls -l 13478 jj 0u CHR 136,48 50 /dev/pts/48
ls -l 13478 jj 2u CHR 136,48 50 /dev/pts/48
grep 13480 jj 2u CHR 136,48 50 /dev/pts/48
```

### 11.2.2 Liste der Benutzer bzw. Prozesse, die auf Dateien zugreifen: fuser

---

Es kann hilfreich sein zu ermitteln, welche Prozesse oder Benutzer aktuell auf bestimmte Dateien zugreifen. Angenommen, Sie möchten ein Dateisystem unmounten, das unter `/mnt` gemountet ist. `umount` gibt «device is busy» zurück. Mit dem Befehl `fuser` können Sie anschließend ermitteln, welche Prozesse auf das Gerät zugreifen:

```
$ fuser -v /mnt/*

USER PID ACCESS COMMAND
/mnt/notes.txt jj 26597 f.... less
```

Nach dem Beenden des Prozesses `less`, der auf einem anderen Terminal ausgeführt wurde, kann das Unmounten des Dateisystems erfolgreich ausgeführt werden.

### 11.2.3 Dateieigenschaften: stat

---

Mit dem Befehl `stat` zeigen Sie die Eigenschaften einer Datei an:

```
$ stat xml-doc.txt

File: `xml-doc.txt'

Size: 632 Blocks: 8 IO Block: 4096 regular file

Device: eh/14d Inode: 5938009 Links: 1

Access: (0644/-rw-r--r--)  Uid: (11994/  jj)  Gid: ( 50/  suse)

Access: 2004-04-27 20:08:58.000000000 +0200
Modify: 2003-06-03 15:29:34.000000000 +0200
Change: 2003-07-23 17:48:27.000000000 +0200
```

Mit dem Parameter `--filesystem` werden Eigenschaften des Dateisystems angezeigt, in dem sich die angegebene Datei befindet:

```
$ stat . --filesystem

File: "." ID: 0 Namelen: 255 Type: ext2/ext3

Blocks: Total: 19347388 Free: 17831731 Available: 16848938 Size: 4096

Inodes: Total: 9830400 Free: 9663967
```

Wenn Sie die z-Shell (`zsh`) verwenden, müssen Sie `/usr/bin/stat` eingeben, da die z-Shell einen in die Shell integrierten `stat`-Befehl mit unterschiedlichen Optionen und einem anderen Ausgabeformat hat:

```
% type
stat stat is a shell builtin
```

```
% stat .  
  
device 769  
  
inode 4554808  
  
mode 16877  
  
nlink 12  
  
uid 11994  
  
gid 50  
  
rdev 0  
  
size 4096  
  
atime 1091536882  
  
mtime 1091535740  
  
ctime 1091535740  
  
blksize 4096  
  
blocks 8  
  
link
```

#### 11.2.4 Prozesse: top

---

Mit dem Befehl `top`, das für «Table of Processes» (Tabelle der Prozesse) steht, wird eine Liste der Prozesse angezeigt, die alle zwei Sekunden aktualisiert wird. Das Programm wird mit der Taste `Q` beendet. Mit der Option `-n 1` wird das Programm nach einmaliger Anzeige der Prozessliste beendet. Im Folgenden finden Sie ein Beispiel für die Ausgabe des Befehls `top -n 1`:

```
top - 14:19:53 up 62 days, 3:35, 14 users, load average: 0.01, 0.02, 0.00  
Tasks: 102 total, 7 running, 93 sleeping, 0 stopped, 2 zombie  
Cpu(s): 0.3% user, 0.1% system, 0.0% nice, 99.6% idle  
Mem: 514736k total, 497232k used, 17504k free, 56024k buffers  
Swap: 1794736k total, 104544k used, 1690192k free, 235872k cached  
  
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ Command  
1426 root 15 0 116m 41m 18m S 1.0 8.2 82:30.34 X  
20836 jj 15 0 820 820 612 R 1.0 0.2 0:00.03 top  
1 root 15 0 100 96 72 S 0.0 0.0 0:08.43 init  
2 root 15 0 0 0 0 S 0.0 0.0 0:04.96 keventd  
5 root 15 0 0 0 0 S 0.0 0.0 0:00.71 bdflush  
  
[...]
```



```
1362 root 15 0 488 452 404 S 0.0 0.1 0:00.02 nscd
1379 root 18 0 56 4 4 S 0.0 0.0 0:00.01 mingetty
1380 root 18 0 56 4 4 S 0.0 0.0 0:00.01 mingetty
```

Wenn Sie die Taste `F` drücken, während `top` aktiv ist, wird ein Menü geöffnet, in dem das Format der Ausgabe umfassend bearbeitet werden kann. Um nur die Prozesse eines bestimmten Benutzers zu überwachen, kann der Parameter `-U UID` verwendet werden. Ersetzen Sie `UID` durch die Benutzer-ID des Benutzers. Der Befehl `top -U $(id -u Benutzername)` gibt die UID des Benutzers auf Basis des Benutzernamens zurück und zeigt dessen Prozesse an.

### 11.2.5 Prozessliste: `ps`

---

Mit dem Befehl `ps` wird eine Liste von Prozessen generiert. Wenn die Option `r` hinzugefügt wird, werden nur Prozesse aufgelistet, die aktuell CPU-Zeit in Anspruch nehmen:

```
$ ps r
PID TTY STAT TIME COMMAND
22163 pts/7 R 0:01 -zsh
3396 pts/3 R 0:03 emacs new-makedoc.txt
20027 pts/7 R 0:25 emacs xml/common/utilities.xml
20974 pts/7 R 0:01 emacs jj.xml
27454 pts/7 R 0:00 ps r
```

Dieser Parameter muss ohne Minuszeichen angegeben werden. Die verschiedenen Parameter werden manchmal mit und manchmal ohne Minuszeichen angegeben. Die Manualpage wirkt auf potenzielle Benutzer häufig abschreckend. Glücklicherweise gibt es den Befehl `ps --help`, mit dem eine kurze Hilfeseite angezeigt werden kann. Um zu prüfen, wie viele `emacs`-Prozesse ausgeführt werden, geben Sie Folgendes ein:

```
$ ps x | grep emacs
1288 ? S 0:07 emacs
3396 pts/3 S 0:04 emacs new-makedoc.txt
3475 ? S 0:03 emacs .Xresources
20027 pts/7 S 0:40 emacs xml/common/utilities.xml
20974 pts/7 S 0:02 emacs jj.xml

$ pidof emacs
20974 20027 3475 3396 1288
```

Mit dem Parameter `-p` werden die Prozesse anhand ihrer Prozess-ID ausgewählt:

```
$ ps www -p $(pidof xterm)
PID TTY STAT TIME COMMAND
9025 ? S 0:01 xterm -g 100x45+0+200
```

```
29854 ? S 0:21 xterm -g 100x75+20+0 -fn \  
-B&H-Lucida  
writer-Medium-R-Normal-Sans-12-120-75-75-M-70-iso10646-1  
4378 ? S 0:01 xterm -bg MistyRose1 -T root -n root -e su -l  
25543 ? S 0:02 xterm -g 100x45+0+200  
19930 ? S 0:13 xterm -bg LightCyan  
21686 ? S 0:04 xterm -g 100x45+0+200 -fn \  
lucidasanstypewriter-12  
26547 ? S 0:00 xterm -g 100x45+0+200
```

Sie können die Prozessliste entsprechend Ihren Anforderungen formatieren. Mit der Option `-L` wird eine Liste aller Schlüsselwörter zurückgegeben. Geben Sie den folgenden Befehl ein, um eine nach Speichernutzung aller Prozesse sortierte Liste zu erhalten:

```
$ ps ax --format pid,rss,cmd --sort rss
```

```
PID RSS CMD
```

```
2 0 [ksoftirqd/0]
```

```
3 0 [events/0]
```

```
17 0 [kblockd/0]
```

```
[...]
```

```
10164 5260 xterm
```

```
31110 5300 xterm
```

```
17010 5356 xterm
```

```
3896 29292 /usr/X11R6/bin/X -nolisten tcp -br vt7 -auth  
/var/lib/xdm/authdir/au
```

## 11.2.6 Prozessbaum: pstree

---

Mit dem Befehl `pstree` wird einer Liste der Prozesse in Form einer Baumstruktur generiert:

```
$ pstree

init--atd
|-3*[automount]
|-bdflush
|-cron
[...]
|-usb-storage-1
|-2*[xterm---su---zsh]
|-xterm---zsh---ssh
`-zsh---startx---xinit4--X
  `-ctwm+-xclock
    |-xload
    `-xosview.bin
```

Mit dem Parameter `-p` werden die Namen durch die jeweiligen Prozess-IDs ergänzt. Damit auch die Befehlszeilen angezeigt werden, verwenden Sie den Parameter `-a`:

```
$ pstree -pa

init,1
|-atd,1255
[...]
`-zsh,1404
  `-startx,1407 /usr/X11R6/bin/startx
    `-xinit4,1419 /suse/jj/.xinitrc [...]
      |-X,1426 :0 -auth /suse/jj/.Xauthority
        `-ctwm,1440
          |-xclock,1449 -d -geometry -0+0 -bg grey
          |-xload,1450 -scale 2
          `-xosview.bin,1451 +net -bat +net
```

### 11.2.7 Wer macht was: w

---

Mit dem Befehl `w` ermitteln Sie, wer beim System angemeldet ist und was die einzelnen Benutzer gerade machen. Beispiel:

```
$ w
15:17:26 up 62 days, 4:33, 14 users, load average: 0.00, 0.04, 0.01
USER TTY LOGIN@ IDLE JCPU PCPU WHAT
jj pts/0 30Mar04 4days 0.50s 0.54s xterm -e su -l
jj pts/3 23Mar04 3:28m 3.21s 0.50s -zsh
[...]
jj pts/7 07Apr04 0.00s 9.02s 0.01s w
[...]
jj pts/14 12:49 37:34 0.20s 0.13s ssh totan
```

Die letzte Zeile verrät, dass der Benutzer `jj` eine SSH-Verbindung zum Computer `totan` aufgebaut hat. Wenn sich Benutzer von entfernten Systemen angemeldet haben, können Sie mit dem Parameter `-f` anzeigen lassen, von welchen Computern aus diese Verbindungen aufgebaut wurden.

### 11.2.8 Speichernutzung: free

---

Die Nutzung des Arbeitsspeichers (RAM) wird mit dem Dienstprogramm `free` überprüft. Es werden Details zum freien und zum verwendeten Speicher (sowie zu den Auslagerungsbereichen) angezeigt:

```
$ free
total used free shared buffers cached
Mem: 514736 273964 240772 0 35920 42328
-/+ buffers/cache: 195716 319020
Swap: 1794736 104096 1690640
```

Mit `-m` erfolgen alle Angaben in MB:

```
$ free -m
total used free shared buffers cached
Mem: 502 267 235 0 35 41
-/+ buffers/cache: 191 311
Swap: 1752 101 1651
```

Die wirklich wichtigen Informationen sind in der folgenden Zeile enthalten:

```
-/+ buffers/cache: 191 311
```

Hier wird der von den Puffern und Cache-Speichern genutzte Arbeitsspeicher berechnet. Der Parameter `-d N` gewährleistet, dass die Anzeigen alle `N` Sekunden aktualisiert wird. So wird die Anzeige mit `free -d 1.5` beispielsweise alle 1,5 Sekunden aktualisiert.

### 11.2.9 Dateisysteme und ihre Nutzung: `mount`, `df` und `du`

---

Mit dem Befehl `mount` können Sie anzeigen, welches Dateisystem (Gerät und Typ) an welchem Mountpunkt gemountet ist:

```
$ mount

/dev/hdb2 on / type ext2 (rw)

proc on /proc type proc (rw)

shmfs on /dev/shm type shm (rw)

automount(pid1012) on /suse type autofs \
(rw,fd=5,pgrp=1012,minproto=2,maxproto=3)

totan:/real-home/jj on /suse/jj type nfs \
(rw,nosuid,rsize=8192,wsiz=8192,hard,intr,nolock,addr=10.10.0.1)
```

Die Gesamtnutzung der Dateisysteme kann mit dem Befehl `df` ermittelt werden. Der Parameter `-h` (oder `--human-readable`) übersetzt die Ausgabe in ein für normale Benutzer verständliches Format.

```
$ df -h

Filesystem Size Used Avail Use% Mounted on
/dev/hdb2 7.4G 5.1G 2.0G 73% /
/dev/hda1 74G 5.8G 65G 9% /data
shmfs 252M 0 252M 0% /dev/shm
totan:/real-home/jj 350G 324G 27G 93% /suse/jj
```

Benutzer des NFS-Dateiservers `totan` sollten ihre Home-Verzeichnisse umgehend bereinigen. Die Gesamtgröße aller Dateien in einem bestimmten Verzeichnis und dessen Unterzeichnissen lässt sich mit dem Befehl `du` ermitteln. Der Parameter `-s` unterdrückt die Ausgabe der detaillierten Informationen. `-h` übersetzt die Daten wieder in ein verständliches Format. Mit dem Befehl

```
$ du -sh ~

361M /suse/jj
```

können Sie feststellen, wie viel Platz Ihr eigenes Home-Verzeichnis belegt.

### 11.2.10 Das Dateisystem `/proc`

---

Das Dateisystem `/proc` ist ein Pseudo-Dateisystem, in dem der Kernel wichtige Daten in Form von virtuellen Dateien speichert. Der CPU-Typ kann beispielsweise mit dem folgenden Befehl abgerufen werden:

```
$ cat /proc/cpuinfo

processor : 0

vendor_id : AuthenticAMD

cpu family : 6

model : 8

model name : AMD Athlon(tm) XP 2400+

stepping : 1

cpu MHz : 2009.343

cache size : 256 KB

fdiv_bug : no

[...]
```

Die Zuordnung und Verwendung der Interrupts kann mit dem folgenden Befehl ermittelt werden:

```
$ cat /proc/interrupts

CPU0

0: 537544462 XT-PIC timer

1: 820082 XT-PIC keyboard

2: 0 XT-PIC cascade

8: 2 XT-PIC rtc

9: 0 XT-PIC acpi

10: 13970 XT-PIC usb-uhci, usb-uhci

NMI: 0

LOC: 0

ERR: 0

MIS: 0
```

Einige wichtige Dateien und die enthaltenen Informationen sind:

`/proc/devices`

verfügbare Geräte

`/proc/modules`

geladene Kernel-Module

`/proc/cmdline`

Kernel-Befehlszeile

```
/proc/meminfo
```

detaillierte Informationen zur Arbeitsspeichernutzung

```
/proc/config.gz
```

gzip-komprimierte Konfigurationsdatei des aktuell aktivierten Kernels

Weitere Informationen finden Sie in der Textdatei `/usr/src/linux/Documentation/filesystems/proc.txt`. Informationen zu aktuell laufenden Prozessen befinden sich in den `/proc/NNN`-Verzeichnissen, wobei NNN für die Prozess-ID (PID) des jeweiligen Prozesses steht. Mit `/proc/self/` können die zum aktiven Prozess gehörenden Eigenschaften abgerufen werden:

```
$ ls -l /proc/self
```

```
lrwxrwxrwx 1 root root 64 Apr 29 13:52 /proc/self -> 27585
```

```
$ ls -l /proc/self/
```

```
total 0
```

```
dr-xr-xr-x 2 jj suse 0 Apr 29 13:52 attr
```

```
-r--r--r-- 1 jj suse 0 Apr 29 13:52 cmdline
```

```
-r----- 1 jj suse 0 Apr 29 13:52 environ
```

```
lrwxrwxrwx 1 jj suse 0 Apr 29 13:52 exe -> /bin/ls
```

```
dr-x----- 2 jj suse 0 Apr 29 13:52 fd
```

```
-r--r--r-- 1 jj suse 0 Apr 29 13:52 mounts
```

```
lrwxrwxrwx 1 jj suse 0 Apr 29 13:52 root -> /
```

```
-r--r--r-- 1 jj suse 0 Apr 29 13:52 stat
```

```
dr-xr-xr-x 3 jj suse 0 Apr 29 13:52 task
```

```
-r--r--r-- 1 jj suse 0 Apr 29 13:52 wchan
```

Die Adresszuordnung der Programmdateien und Bibliotheken befindet sich in der Datei

```
maps:
```

```
$ cat /proc/self/maps
```

```
08048000-0804c000 r-xp 00000000 03:02 22890 /bin/cat
```

```
0804c000-0804d000 rw-p 00003000 03:02 22890 /bin/cat
```

```
40017000-40018000 rw-p 40017000 00:00 0
```

```
4002b000-40135000 r-xp 00000000 03:02 10908 /lib/tls/libc.so.6
```

```
4013d000-40141000 rw-p 4013d000 00:00 0
```

```
bffffe00-c0000000 rw-p bffffe00 00:00 0
```

```
fffffe00-ffffff00 ---p 00000000 00:00 0
```

### 11.2.11 vmstat, iostat und mpstat

---

Das Dienstprogramm `vmstat` fasst Statistiken zum virtuellen Arbeitsspeicher zusammen. Es liest die Dateien `/proc/meminfo`, `/proc/stat` und `/proc/*/stat` aus. Mit diesem Programm können Engpässe der Systemleistung ermittelt werden. Der Befehl `iostat` fasst Statistiken zur CPU sowie zu Ein- und Ausgaben für Geräte und Partitionen zusammen. Die angezeigten Informationen stammen aus den Dateien `/proc/stat` und `/proc/partitions`. Mithilfe der Ausgabe kann die Ein- und Ausgabelast zwischen den Festplatten optimiert werden. Der Befehl `mpstat` fasst CPU-bezogene Statistiken zusammen.

### 11.2.12 procinfo

---

Wichtige Informationen zum Dateisystem `/proc` werden mit dem Befehl `procinfo` zusammengefasst:

```
$ procinfo

Linux 2.6.4-54.5-default (geeko@buildhost) (gcc 3.3.3 ) #1 1CPU
[roth.suse.de]

Memory: Total Used Free Shared Buffers
Mem: 516696 513200 3496 0 43284

Swap: 530136 1352 528784

Bootup: Wed Jul 7 14:29:08 2004 Load average: 0.07 0.04 0.01 1/126 5302
user : 2:42:28.08 1.3% page in : 0
nice : 0:31:57.13 0.2% page out: 0
system: 0:38:32.23 0.3% swap in : 0
idle : 3d 19:26:05.93 97.7% swap out: 0
uptime: 4d 0:22:25.84 context :207939498
irq 0: 776561217 timer irq 8: 2 rtc
irq 1: 276048 i8042 irq 9: 24300 VIA8233
irq 2: 0 cascade [4] irq 11: 38610118 acpi, eth0, uhci_hcd
irq 3: 3 irq 12: 3435071 i8042
```

Verwenden Sie den Parameter `-a`, wenn Sie alle Informationen sehen möchten. Der Parameter `-nN` aktualisiert die Informationen alle `N` Sekunden. Beenden Sie in diesem Fall das Programm mit der Taste `Q`. Standardmässig werden die kumulativen Werte angezeigt. Mit dem Parameter `-d` werden die Einzelwerte generiert. `procinfo -dn5` zeigt die Werte an, die sich in den letzten fünf Sekunden geändert haben:

```
Memory: Total Used Free Shared Buffers Cached
Mem: 0 2 -2 0 0 0

Swap: 0 0 0

Bootup: Wed Feb 25 09:44:17 2004 Load average: 0.00 0.00 0.00 1/106 31902
```



```
user : 0:00:00.02 0.4% page in : 0 disk 1: 0r 0w
nice : 0:00:00.00 0.0% page out: 0 disk 2: 0r 0w
system: 0:00:00.00 0.0% swap in : 0 disk 3: 0r 0w
idle : 0:00:04.99 99.6% swap out: 0 disk 4: 0r 0w
uptime: 64d 3:59:12.62 context : 1087
irq 0: 501 timer irq 10: 0 usb-uhci, usb-uhci
irq 1: 1 keyboard irq 11: 32 ehci_hcd, usb-uhci,
irq 2: 0 cascade [4] irq 12: 132 PS/2 Mouse
```

### 11.2.13 Systemaufrufe eines aktiven Programms: strace

---

Mit dem Dienstprogramm `strace` können Sie alle Systemaufrufe eines aktuell ausgeführten Prozesses verfolgen. Geben Sie den Befehl wie üblich ein und fügen Sie am Zeilenanfang `strace` hinzu:

```
$ strace ls
execve("/bin/ls", ["ls"], [/* 88 vars */]) = 0
uname({sys="Linux", node="edison", ...}) = 0
brk(0) = 0x805b000
old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) \
= 0x40017000
open("/etc/ld.so.preload", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=76333, ...}) = 0
old_mmap(NULL, 76333, PROT_READ, MAP_PRIVATE, 3, 0) = 0x40018000
[...]
ioctl(1, SNDCTL_TMR_TIMEBASE or TCGETS, {B38400 opost isig icanon echo ...}) = 0
fstat64(3, {st_mode=S_IFDIR|0755, st_size=144, ...}) = 0
fcntl64(3, F_SETFD, FD_CLOEXEC) = 0
getdents64(3, /* 0 entries */, 4096) = 0
close(3) = 0
fstat64(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 48), ...}) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) \
```

```
= 0x40018000

write(1, "ltrace-ls.txt myfile.txt strac"... , 41) = 41

munmap(0x40018000, 4096) = 0

exit_group(0) = ?
```

Um beispielsweise alle Versuche, eine bestimmte Datei zu öffnen, zu verfolgen, geben Sie Folgendes ein:

```
$ strace -e open ls myfile.txt

open("/etc/ld.so.preload", O_RDONLY) = -1 ENOENT (No such file or direc-
tory)

open("/etc/ld.so.cache", O_RDONLY) = 3

open("/lib/libattr.so.1", O_RDONLY) = 3

open("/proc/mounts", O_RDONLY) = 3

[...]

open("/proc/filesystems", O_RDONLY) = 3

open("/proc/self/attr/current", O_RDONLY) = 4
```

Um alle untergeordneten Prozesse zu verfolgen, verwenden Sie den Parameter `-f`. Das Verhalten und das Ausgabeformat von `strace` können weitgehend gesteuert werden. Weitere Informationen erhalten Sie durch die Eingabe von `man strace`.

#### 11.2.14 Erforderliche Bibliothek angeben: `ldd`

Mit dem Befehl `ldd` können Sie ermitteln, welche Bibliothek die als Argument angegebene dynamische Programmdatei laden würde:

```
$ ldd /bin/ls

linux-gate.so.1 => (0xffffe000)

librt.so.1 => /lib/tls/librt.so.1 (0x4002b000)

libacl.so.1 => /lib/libacl.so.1 (0x40033000)

libselinux.so.1 => /lib/libselinux.so.1 (0x40039000)

/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)

libattr.so.1 => /lib/libattr.so.1 (0x4016d000)
```

Statische Binärdateien benötigen keine dynamische Bibliotheken:

```
$ ldd /bin/sash

not a dynamic executable

$ file /bin/sash

/bin/sash: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), \for
GNU/Linux 2.2.5, statically linked, stripped
```

## Repetitionsfrage

---

10

Was ist beim betreiben eines Internetservers wichtig?

---

## 12 Systemabnahme und Betriebsübergabe planen

---

Wenn der Internetserver fertig aufgesetzt und konfiguriert ist und alle Tests erfolgreich durchlaufen sind, kann die Abnahme geplant und durchgeführt werden. Nach erfolgreicher Abnahme wird der Internetserver vom Administrator betrieben bzw. an den Kunden übergeben.

### 12.1 Abnahmeprotokoll vorbereiten und prüfen

---

Im Abnahmeprotokoll sind die Punkte aufzuführen, die bei der Abgabe des Systems zu prüfen sind. Dadurch wird sichergestellt, dass der Internetserver den im Pflichtenheft festgehaltenen Anforderungen genügt. Im Wesentlichen beinhaltet das Abnahmeprotokoll die bereits durchgeführten Testfälle; oft im reduzierteren Umfang.

### 12.2 Systemabnahme bzw. -abgabe durchführen

---

Zusammen mit dem Abnehmer wird das System gemäss Protokoll abgenommen. Die durchzuführenden Schritte und Ergebnisse der Tests werden nochmals dokumentiert und vom Abnehmer bestätigt (Visum).

Das Abnahmeprotokoll beweist die korrekte Funktionalität des Systems und ist die Grundlage für die Verrechnung der Leistungen. Gleichzeitig bestätigt es, dass die Abnahme erfolgt ist und die Verantwortung für den Betrieb des Systems ab diesem Zeitpunkt an den Abnehmer übergeht.

### 12.3 Abnahmeprotokoll

---

Ein beispielhaftes Abnahmeprotokoll:

- Titel: «Abnahmeprotokoll Internet-Server für Kunde XYZ AG»
- Datum, Autor/Ersteller
- Weitere Referenzen, z. B. Auftragsnummer, Projekt-Identifikation
- Empfänger mit Name und Adresse
- Übergabeobjekte: Internetserver, weitere Hardware, Back-up-Bänder/CD, Software-Originalversionen, Lizenzvereinbarungen, Verträge
- Dokumentationen: Handbücher, Betriebsanleitungen, Dokumentation zu Konfigurationen
- Durchgeführte Tests und Dokumentation/Testergebnisse
  - Installation und Konfiguration von Anwendungen/Diensten
  - Testergebnisse der verschiedenen Funktionen der Anwendungen/Dienste
  - Testprotokoll, durch den Kunden-Testbeauftragten unterschrieben (falls Tests mit Kunde durchgeführt wurden)
- Offene Mängel, idealerweise mit Datum bis zur Behebung
- Datum, Name, Unterschrift des Autors und des Abnehmers (Kunde)

### Repetitionsfrage

---

## Teil E Anhang

---

## Gesamtzusammenfassung

---

Das Buch vermittelt die notwendigen Kompetenzen für das Modul 239 «Internetserver realisieren». Es werden die Grundlagen der Handlungsziele erklärt und die Installation eines Internetservers gemäss den Vorgaben der Handlungsziele durchgeführt und mit Beispielen versehen. Als Grundlage wird das Open Source Betriebssystem SuSE Linux, Version 10, verwendet. Die Komponenten des Internetservers bestehen gesamthaft aus frei erhältlichen Software-Paketen. Die Bezugsquellen für zusätzliche Software werden jeweils am entsprechenden Ort aufgeführt.

Im Teil A werden mögliche Vorgehensmethoden für die Erarbeitung der Anforderungen an einen Internetserver vorgestellt. Anschliessend werden Internetserver und deren Einsatzgebiete näher erläutert. Anschliessend folgt eine Vorstellung der Dienste eines Internetservers und der dazu notwendigen Systeme und Software. Die für den Internetserver benötigten Protokolle sowie Architekturen und Bauweisen werden ebenfalls näher beschrieben und mit Beispielen versehen.

Im nächsten Teil (Teil B) wird die Konzeption und Dimensionierung durchgeführt. Dazu muss zuerst die IST-Situation analysiert werden. Insbesondere werden die benötigten Dienste für den Internetserver festgelegt. Danach folgt eine Beschreibung der angebotenen bzw. vom Auftraggeber geforderten Dienstleistungen, die der Internetserver anbieten soll. Darunter versteht man beispielsweise Informationen (statisch) wie Produktkataloge und Preislisten, Bestellungen bzw. E-Shop-Systeme für den Einkauf via Internet. Anschliessend wird der SOLL-Zustand festgelegt. Dabei kann es nötig sein, bestehende Applikationen für den Einsatz im Internet «fit» zu machen, d. h., möglicherweise sind Schnittstellen zusätzlich nötig. Allgemeine Sicherheitsrichtlinien, insbesondere Anforderungen bezüglich Datenschutz, werden ebenfalls betrachtet. Eine wichtige Anforderung beim Internetserver ist die Verfügbarkeit der Dienste (da der Internetserver ständig im Internet verfügbar ist) sowie das Datenvolumen, welches zwischen Benutzer und Internetserver übertragen wird. Die Benutzerprofile und die externen Zugriffe sind ebenfalls Bestandteil der Kapitel im zweiten Teil des Lehrmittels. Nachdem alle Anforderungen erhoben wurden, wird die Lösung (Hard- und Software, Konfiguration) entworfen und dokumentiert. Dabei werden auch Namenskonventionen und Standardeinstellungen angesprochen. Die Vorbereitung der System- und Sicherheitstests (Testfälle, Dokumentation des Testablaufs) sind ebenfalls Bestandteil der Planung.

Im Teil C wird detailliert auf die einzelnen Installationsschritte und insbesondere die Konfiguration der einzelnen Dienste eingegangen. Schritt für Schritt wird die Grundkonfiguration des Betriebssystems und anschliessend die detaillierte Konfiguration der Dienste Webserver, Mailserver, DNS, FTP-/Fileserver aufgezeigt. Optionen und Varianten zur Konfiguration der Dienste werden im Detail erklärt, damit die Besonderheiten der einzelnen Dienste dem Leser bekannt sind. Anschliessend werden die Log-Dienste (Logfiles) und Sicherungsprozeduren für Back-ups beschrieben. Speziell wird auf die Analyse der Logfiles mit entsprechenden Hilfsmitteln (Tools) eingegangen. Die Anbindung von fremden Ressourcen wie Authentifizierungsserver (z. B. LDAP oder Active Directory) und Datenbanken wird ebenfalls angesprochen.

Im letzten Teil (Teil D) geht es um den Betrieb und die Wartung des Internetservers sowie um die Abnahme des Internetservers durch den Auftraggeber. Für die Wartung werden die Anforderungen festgelegt und verschiedene Möglichkeiten zur Überwachung des Internetservers aufgezeigt. Die Systemabnahme wird zusammen mit dem Auftraggeber durchgeführt und in einem Abnahmeprotokoll dokumentiert. Das Abnahmeprotokoll bildet somit den Abschluss des Projekts Internetserver.

## Antworten zu den Repetitionsfragen

- 
- 1 Seite 14 Information, Planen, Entscheiden, Realisieren, Kontrollieren, Abschliessen
- 
- 2 Seite 27 Betriebssystem und Server-Software für die einzelnen Dienste
- 
- 3 Seite 42 Vor allem das/die Server-Systeme selbst, Netzwerkanschlüsse und der physische Aufbewahrungsort (Serverschrank/-rack, Rechenzentrum)
- 
- 4 Seite 100 Das verschlüsselte Protokoll heisst SSL (Secure Sockets Layer) oder in der moderneren Variante TLS (Transaction Layer Security). Es funktioniert auf dem Prinzip der asymmetrischen Verschlüsselung, d. h., zum Austausch der Schlüssel werden digitale/elektronische Zertifikate verwendet. Der Server stellt sein öffentliches Zertifikat dem anfragenden Client zur Verfügung, womit der Schlüssel geheim ausgetauscht werden kann.
- 
- 5 Seite 107 Beim Betrieb eines zentralen, internen Verzeichnisdienstes besteht das Problem, dass vertrauliche Daten nach draussen gelangen können. Auf einem dedizierten Authentifizierungsserver können nur die Benutzer des Internetservers (separat) erfasst werden, getrennt von den internen Benutzern. Der Nachteil von verschiedenen Authentifizierungsservern ist die Synchronisation oder das Führen mehrerer Benutzer-Identitäten (für einen internen Benutzer, der auch von extern zugreifen muss).
- 
- 6 Seite 14 Projektmanagement ist die Verwaltung des Projekts (Planung, Kontrolle der einzelnen Tätigkeiten). Projektarbeit ist die Durchführung der Tasks (Arbeitseinheiten) der einzelnen Projektphasen.
- 
- 7 Seite 27 Das Protokoll bzw. die Protokollfamilie TCP/IP
- 
- 8 Seite 42 Es sind Installationsabhängigkeiten zu beachten, damit die Software miteinander betrieben werden kann und keine Konflikte bei der Installation auftreten.
- 
- 9 Seite 100 Der Zonentransfer bezeichnet den Austausch/Bekanntgabe der DNS-Information eines Servers, die bei einem DNS-Server gespeichert/verwaltet wird. Damit synchronisieren sich die einzelnen DNS-Server untereinander.
- 
- 10 Seite 125 Einerseits die ständige Aktualisierung und der Schutz vor sich ändernden Gefahren (Viren, Angriffe von aussen; Patches der Hersteller), sowie die ständige Überwachung (Monitoring) der «lebenswichtigen» Anzeichen des Internetservers wie die Prozessor- und Memory-Auslastung sowie Warnmeldungen des Systems und der einzelnen Dienste (mit Hilfe von Dienstprogrammen sichtbar)
- 
- 11 Seite 18 Ein Internetserver wird im Intranet oder als Webserver eingesetzt; mit verschiedenen Zusatzdiensten.
- 
- 12 Seite 27 Beides wird bei einem Internet Service Provider betrieben. Beim hosting wird (ein Teil eines) Server gemietet (Konfiguration und Betrieb grundsätzlich durch Provider); beim housing wird ein (eigener/gemieteter) Server beim Provider betrieben (Konfiguration und Betrieb grundsätzlich durch Abnehmer/Kunde).

- 
- 13** Seite 46 Funktionstests bei den Anwendungen, technische Tests im umliegenden Netzwerk und am Server selbst (Penetration-Test), Verfügbarkeits-/Last-/Stress-Test, allgemeine Tests der Sicherheitsanforderungen
- 
- 14** Seite 100 Beim Betrieb eines Fileservers im Internet (FTP-Server) ist darauf zu achten, dass keine rechtswidrigen Inhalte von (unbekannten) Benutzern auf dem Server gespeichert oder ausgetauscht werden. Der FTP-Server muss vor Missbrauch geschützt werden, da der Betreiber für den Inhalt haftbar ist.
- 
- 15** Seite 126 Mit dem Abnahmeprotokoll wird die Übergabe des fertig konfigurierten Internetservers vom Auftragnehmer an den Auftraggeber dokumentiert. Gleichzeitig dient das Abnahmeprotokoll der Verifikation allfälliger Mängel oder Anforderungen, die noch nicht zufriedenstellend erfüllt wurden. Das Abnahmeprotokoll markiert den Schlusspunkt des Projekts Internetserver.
- 
- 16** Seite 18 Ein Internetserver stellt neben dem Webdienst (WWW-Server, Webserver) zusätzlich noch Namensauflösung (DNS), Fileserver (FTP) und E-Mail-Dienste (SMTP, Postfach) zur Verfügung.
- 
- 17** Seite 33 Anzahl der aktuellen Kunden und der Zielgruppe des Internetservers, Angebotene Anwendungen; aktuelle Serverumgebung, Netzwerk und Systeme und Anwendungen
- 
- 18** Seite 46 Betriebs-, Benutzer- und Installationsdokumentation
- 
- 19** Seite 100 Die zwei hauptsächlichen Probleme beim Betreiben eines Mailserver sind Spam (unverlangte E-Mail-Werbepostungen) und der Missbrauch des Mailservers, um bösartige Software (Viren) oder Spam (Relaying) zu versenden. Die Probleme können mit entsprechenden Sicherheitsmassnahmen mehrheitlich in den Griff bekommen werden.
- 
- 20** Seite 27 Physischer Server, Netzwerkkomponenten (Router, Switch) und Sicherheitskomponenten (Firewall); nach Bedarf zusammenschliessen mehrerer physischer Server zu Clustern
- 
- 21** Seite 37 Applikationen, die «web-enabled» werden sollen; Datenvolumen und Verfügbarkeitsanforderungen, Benutzerprofile und -berechtigungen, Datenschutzerfordernungen aus Gesetzen (für Personendaten)
- 
- 22** Seite 100 Bei ausgeschalteter Firewall muss unbedingt eine zusätzliche (dedizierte) Firewall für den Schutz des Internetservers sorgen. Generell empfiehlt es sich, für den Schutz eine separate Hardware (= Firewall) zu betreiben, da bei einem Fehler in der internen Firewall der Internetserver sofort exponiert im Internet steht.
- 
- 23** Seite 105 Eine einheitliche, synchronisierte Zeitangabe der Log-Einträge hilft beim auffinden von voneinander abhängigen Einträgen. Dies ist vor allem beim Betrieb eines zentralen Logservers wichtig.
- 
- 24** Seite 27 Die Anzahl der angebotenen Anwendungen und die Anzahl der (gleichzeitigen) Benutzer des Systems



- 
- 25** Seite 37      Der Datenschutz bezweckt den Schutz der Privatsphäre durch Schutz der Personendaten; bei der Datensicherheit werden die Anforderungen an Vertraulichkeit, Integrität und Verfügbarkeit an die Daten generell gemeint.
- 
- 26** Seite 100      Anstelle der Berechtigung von einzelnen Benutzern, Dateien oder Prozessen empfiehlt es sich, diese in Gruppen zu verwalten. Die Berechtigung kann somit der Gruppe erteilt werden, was die Verwaltung der Berechtigungen (Autorisierungsverwaltung) extrem vereinfacht. Die Dokumentation wird dadurch ebenfalls leichter verständlich.
- 
- 27** Seite 105      Da Back-ups ressourcen-intensiv sind, sollten diese nachts/während minimalem Verkehrsaufkommen durchgeführt werden, um das System nicht zusätzlich zu belasten.

## Glossar

---

### A

<b>ACL</b>	Access Control List; Zugriffsliste der Berechtigungen († Autorisation) der Benutzer
<b>AD</b>	Active Directory; Variante von † LDAP von Microsoft
<b>allow</b>	Erlauben; Bezeichnung für eine Aktion der Firewall
<b>Apache</b>	Weit verbreiteter Webserver (open source)
<b>Applikation</b>	Anwendung
<b>Authentifikation</b>	Beweis der Identität (z. B. durch Passwort)
<b>Autorisation</b>	Berechtigung; eine Aktion durchzuführen (z. B. lesen, schreiben etc.)
<b>AV</b>	Antivirus

---

### B

<b>Benutzerprofil</b>	Identität eines Benutzers inklusiv dessen Berechtigungen und Rollen
<b>Binärdatei</b>	Datei bestehend aus maschinenlesbaren (binären) Zeichen; für Menschen ohne Hilfsmittel nicht lesbar
<b>BIND</b>	Berkeley Internet Name Domain; open-source Software Paket für Namensauflösung († DNS)
<b>Browser</b>	Anwendung zur Betrachtung von HTML-Webseiten
<b>BSD</b>	Berkeley Standard Distribution; Variante von Unix (FreeBSD, NetBSD etc.)

---

### C

<b>Cache</b>	Zwischenspeicher; oft werden Webseiten in einem † Proxy zwischengespeichert
<b>CGI</b>	Common Gateway Interface
<b>Client</b>	Anwendungsprogramm, das mit einem Serverprogramm (Serverdienst, Service, Server) kommuniziert; oder Hardware, die mit Server-Hardware kommuniziert
<b>CRUD</b>	Create Read Update Delete; Grundlegende Datenbankoperationen, die auch zur Berechtigung von Benutzern in Anwendungen verwendet werden († Autorisation)

---

### D

<b>Datenschutz</b>	Schutz der Privatsphäre durch Datenschutzgesetze; nicht zu verwechseln mit Datensicherheit
<b>Datensicherheit</b>	Gewährleistung der Anforderungen Vertraulichkeit, Integrität und Verfügbarkeit gegenüber von Systemen (Hardware) und Anwendungen (Software) sowie Daten
<b>Datensicherung</b>	Sicherung der Daten zur Gewährleistung von Integrität und Verfügbarkeit
<b>DAV</b>	Distributed Authoring and Versioning, auch WebDAV; offener Standard zur Bereitstellung von Daten im Internet als «Online-Festplatte»

<b>DBMS</b>	DataBase Management System; Schnittstellensoftware zwischen Datenbank und Applikation; zuständig für die Verwaltung der Daten auf der Datenbank
<b>Dedicated, dediziert</b>	Speziell dazu ausgelegt, eine besondere Aufgabe zu erfüllen (z. B. dedizierter Webserver), d. h., das System ist für eine genau definierte Aufgabe bestimmt (Wortstamm bedeutet: Widmung)
<b>deny</b>	Ablehnen; Bezeichnung für eine Aktion der Firewall
<b>Deziiert</b>	Bestimmt, entschieden, energisch (oft verwechselt mit † Dedicated, dediziert)
<b>Dienst</b>	Softwareprogramm, das bestimmte (oft kleine) Aufgaben erfüllt, z. B. horchen auf Verbindungsanfragen für Webserver; auch † Service oder daemon bezeichnet
<b>Dienstprogramm</b>	Anwendungen zur Analyse, Überwachung oder Verwaltung eines Systems (Support-Programme für die Erledigung der o.g. Aufgaben)
<b>Direktive</b>	Anweisung; bezeichnet Anweisung im Konfigurationsfile des Apache Webservers
<b>Directory</b>	Verzeichnis; grundsätzlich verwendet zur Referenzierung von Benutzer-Verzeichnissen
<b>DMZ</b>	Demilitarisierte Zone; separates Netzwerk zwischen Internet und internem Netz (Intranet, LAN); durch eine Firewall oder Paketfilter geschützt
<b>DNS</b>	Domain Name Service; Dienst und Protokoll zur Auflösung von Domain-Namen
<b>Domäne</b>	Siehe «Domain»
<b>Domain</b>	Zusammenhängender Teilbereich einer hierarchischen Struktur in DNS (Baumstruktur), d. h., www.sun.com ist eine subdomäne von sun.com, welches eine Subdomäne von .com ist
<b>Downtime</b>	Zeit, während der ein Dienst/Server nicht zur Verfügung steht (wegen Ausfall, Wartung etc.)
<b>Dualboot</b>	Möglichkeit, zwei oder mehrere Betriebssysteme auf einem physischen System zu starten
<b>Dual-homed</b>	Bezeichnet ein System mit zwei oder mehreren (multi-homed) Schnittstellen; im Normalfall sind damit Netzwerkschnittstellen (Netzwerk-Interfaces) gemeint; ein solches System kann somit mehrere IP-Adressen haben

---

## E

<b>E-Mail</b>	Elektronische Nachrichten
<b>entry</b>	Eintritt oder oft im Zusammenhang mit «Eintrag» verwendet, womit oft ein Eintrag in einem Konfigurationsfile gemeint ist
<b>Evaluation</b>	Erhebung, Überprüfung, Vergleich
<b>Ethernet</b>	Übertragungsprotokoll der OSI Schicht 2 für verkabelte Verbindungen (LAN); das kabellose Äquivalent ist Wireless LAN
<b>Extranet</b>	Teil des Intranet, der über externe Zugänge erreichbar ist

---

## F

<b>Fileserver</b>	Server, der Daten/Dateien verwaltet und zur Verfügung stellt (oft auch für FTP-Server verwendet)
<b>Firewall</b>	System, welches vertrauenswürdige von nicht-vertrauenswürdigen Netzwerken trennt und mithilfe von Regeln konfiguriert ist (welche Verbindungen erlaubt sind und welche nicht)
<b>FTP</b>	File Transfer Protocol; Protokoll zur Dateiübertragung

---

**H**

<b>Hit</b>	Schlag; oft verwendet als Bezeichnung für die Anzahl der sog. «page-hits» auf einer Website, d. h., wie oft eine Website abgerufen wurde
<b>Host</b>	Wirt; Bezeichnung für ein System, welches Anwendungen beherbergt und oft Clients bedient; ein Internetserver ist ein typischer Host
<b>Hosting</b>	Betriebung eines Servers durch einen Provider (es wird nur der Dienst, z. B. Webserver, gemietet und nicht selber durch den Auftraggeber betrieben)
<b>Housing</b>	Betriebung eines Servers durch den Auftraggeber, wobei der Server bei einem Provider «housed» wird, d. h. dort Unterschlupf in Form von Energie, physischem Schutz vor Zerstörung, Diebstahl, bekommt.
<b>HTTP</b>	Hypertext Transfer Protocol; Protokoll der OSI-Schicht 7 zur Übertragung von HTML-Dokumenten und zur Verfolgung von Hyperlinks
<b>HTTPS</b>	HTTP-Secure; gesicherte Variante des HTTP-Protokolls, welche mit SSL oder TLS gesichert wird (verschlüsselte und authentifizierte Verbindungen)
<b>Hub</b>	Hardware zur physischen Erweiterung eines Netzwerks mit mehreren Schnittstellen; vergleichbar mit einer Mehrfach-Elektro-Steckdose; ohne Intelligenz

---

**I**

<b>I/O</b>	Input/Output; oft als Bezeichnung für die Anzahl eingehender und ausgehender Anfragen eines Systems
<b>Identifikation</b>	Eindeutige Bezeichnung eines Subjekts; oft der Benutzername oder User-ID
<b>IIS</b>	Internet Information Server; Webserver der Firma Microsoft
<b>IMAP</b>	Internet Message Access Protocol; Protokoll zur Verwaltung eines Postfachs, ohne die E-Mail-Nachrichten auf den Client zu laden
<b>Internet</b>	Bezeichnung für das weltweite Netzwerk von Computern mit einer grossen Anzahl verschiedener Protokolle
<b>Intranet</b>	Bezeichnung für ein Netzwerk, welches auf der gleichen Technologie wie das Internet (TCP/IP) basiert, aber nur einem geschlossenen Benutzerkreis (Unternehmens-intern) zur Verfügung steht
<b>IPERKA</b>	Projektarbeitsmethode: Information, Planen, Entscheiden, Realisieren, Kontrollieren, Abschliessen

---

**L**

<b>LAN</b>	Local Area Network; lokales Netzwerk innerhalb eines Gebäude; basiert auf verschiedenen Protokollen (oft Ethernet und TCP/IP)
<b>LDAP</b>	Lightweight Directory Access Protocol; offenes Protokoll für den gleichnamigen Verzeichnisdienst
<b>Link</b>	Abkürzung für Hyperlink; ein Verweis auf ein anderes HTML-Dokument
<b>Linux</b>	Freies Betriebssystem auf Basis von Unix
<b>Listen</b>	Hören, Horchen; Bezeichnung für die Tätigkeit eines Dienstes an einem TCP- oder UDP-Port (daemon, listener)

<b>Loadbalancer</b>	System, welches die Last gleichmässig auf dahinter liegende Systeme verteilt; oft verwendet für Internetserver mit hohem Verkehrsaufkommen und in Clustern
<b>Log, Logfile</b>	Datei, welche Anfragen und Tätigkeiten von Anwendungen protokolliert
<b>Log-in</b>	Tätigkeit, um als Benutzer in eine Anwendung einzusteigen (einloggen)
<b>Logserver</b>	Server (physisch/software), welcher (zentral) verschiedene Logdateien (Protokolle) führt

---

## M

<b>MAN</b>	Metropolitan Area Network; erweitertes LAN innerhalb eines geografischen Gebiet, oft einer Stadt
<b>Monitoring</b>	Überwachung des Systems bezüglich Erfüllung der Anforderungen (Systemüberwachung), z. B. Unterbrüche, Fehler etc.
<b>MTA</b>	Mail Transfer Agent; auch SMTP-Server, Anwendung, die für die Weiterleitung von E-Mails besorgt ist
<b>MX</b>	Mail Exchanger; Eintrag im DNS-Server, der den Mailserver für eine Domain kennzeichnet

---

## N

<b>Nameserver</b>	Bezeichnung für DNS-Server (Auflösung von Domainnamen in IP-Adressen)
<b>NIS</b>	Network Information Service; proprietärer Verzeichnisdienst von Solaris
<b>notify</b>	Meldung; oftmals als Konfigurationsoption oder Befehl verstanden, damit eine Anwendung bestimmte Ereignisse meldet (ins Logfile schreibt oder an den Administrator meldet via E-Mail etc.)
<b>NTLM</b>	NT LAN Manager; Authentifizierungsschema, welches in Windows-Systemen Verwendung findet
<b>OSS</b>	Open Source Software; frei verfügbare Software (quelloffen)
<b>OSI</b>	Open Systems Interconnection; Architektur zur Bezeichnung von sieben Schichten eines Netzwerks

---

## P

<b>Paket</b>	Einheit von Daten, die über eine Netzwerkverbindung versendet werden; beinhaltet Header (mit Adressinformationen) und body oder payload (mit den eigentlichen Daten)
<b>Paketfilter</b>	Rudimentäre Firewall, welche Pakete nach den Kriterien SOURCE (Absender), TO (Empfänger) und PORT (für das verwendete Protokoll, z. B. 80 für http) filtert
<b>Pentest</b>	Penetration Test; Testmöglichkeit für Internetserver
<b>Perl</b>	Script-ähnliche Programmiersprache mit vielen Möglichkeiten
<b>PHP</b>	PHP Hypertext Preprocessor; ursprünglich für Personal Homepage Tools; Skriptsprache für Webseiten
<b>POP</b>	Post Office Protocol; Protokoll zum Abrufen von E-Mails aus einem Postfach
<b>Postfach</b>	Elektronischer Briefkasten, der mit POP oder IMAP-Protokollen abgerufen werden kann
<b>Printserver</b>	System, welches Druckaufträge verwaltet und ausführt

<b>Proxy</b>	Dienstprogramm (oder System), welches Datenverkehr vermittelt; oft zur Beschleunigung des Datenverkehrs und zur Erhöhung der Sicherheit eingesetzt
<b>Prozess</b>	Ausführendes Programm innerhalb eines Betriebssystems (arbeitendes Programm = Prozess); ein oder mehrere Prozesse können eine Anwendung/Applikation bilden

---

## R

<b>RADIUS</b>	Remote Authentication Dial-In User Service; Client-Server Protokoll, das zur Authentisierung und Autorisierung von Einwahl-Verbindungen verwendet wird
<b>RBAC</b>	Role Based Access Control; Architektur für rollen-basierte Autorisierung (Berechtigung), d. h., ein Benutzer ist einer Rolle zugewiesen; die Berechtigung wird für die Rolle und nicht für den Benutzer erteilt
<b>Replikation</b>	Vermehrung; wird im Zusammenhang mit der mehrfachen Führung von Daten verwendet (replizierte Datensammlungen), um Zugriffsgeschwindigkeit hoch zu halten
<b>Ressource</b>	Quelle; der Begriff wird oft im Zusammenhang mit Leistung verwendet (wie viele Ressourcen benötigt der Webserver = wie viel Leistung in RAM, Harddisk-Speicher etc. wird benötigt)
<b>RFC</b>	Request for comment; technische und organisatorische Dokumente, die im Wesentlichen alles rund ums Internet spezifizieren; nachzulesen bei <a href="http://www.rfc.org">www.rfc.org</a>
<b>Root</b>	Wurzel; Bezeichnung für zuoberst liegendes Verzeichnis oder Administrator-Benutzer in UNIX-Betriebssystemen
<b>Root DNS Server</b>	Der oberste hierarchische DNS-Server des Internet; insgesamt gibt es 13 root-Server Instanzen, welche die oberste Instanz für DNS-Anfragen definieren
<b>Root-Server</b>	Internetserver, der in der Kontrolle des Kunden liegt (root=Administrator), d. h. ein gemieteter Server bei einem Internet Service Provider, auf den der Kunde administrativen Zugriff hat
<b>Router</b>	System, welches Pakete zwischen Netzwerken nach bestimmten Regeln (routen = Wege) weiter sendet oder ablehnt (Grundlage für Paketfilter)

---

## S

<b>sendmail</b>	Bezeichnung für einen Mail Transfer Agent
<b>Server</b>	System, welches einen † Client bedient
<b>Service</b>	Dienst (Dienstprogramm), welches Anfragen von Clients (Anwenderprogrammen) beantwortet
<b>SEUSAG</b>	Systemplanungsmethode: Systemgrenzen, Einflussgrößen, Untersysteme, Schnittstellen, Analysieren, Gemeinsamkeiten
<b>Sicherheit</b>	Bezeichnung der Grundmerkmale Vertraulichkeit, Integrität und Verfügbarkeit
<b>SMTP</b>	Simple Mail Transfer Protocol; Protokoll für E-Mail-Übertragung
<b>SNMP</b>	Simple Network Management Protocol; Protokoll zur Verwaltung und Überwachung von Netzwerk-Geräten
<b>SOA</b>	Start of Authority; Bezeichnung eines Eintrags für den DNS-Server
<b>Solaris</b>	Betriebssystem von SUN
<b>ssh</b>	Secure shell; Protokoll für sichere telnet-Verbindungen

<b>SSL</b>	Secure Socket Layer; Protokoll der OSI-Schicht 5 zur Verschlüsselung darüber liegender Verbindungen (prinzipiell für https verwendet, aber auch für POP3, IMAP einzusetzen)
<b>SSL-Accelerator</b>	Hardware, welche in einem Server die Ver- und Entschlüsselung vornimmt, um den Hauptprozessor zu entlasten
<b>Stresstest</b>	Testmöglichkeit für Internetserver, wobei eine grosse Belastung simuliert wird (viele Benutzer, hohe Anzahl Benutzeranfragen etc.)
<b>SUN</b>	Firmennamen eines System- und Betriebssystemherstellers
<b>Switch</b>	Netzwerk-Komponente, welche auf OSI-Schicht 2 funktioniert und (Ethernet)-Pakete vermittelt

---

**T**

<b>TCP/IP</b>	Protokollfamilie, die als Grundlage für das Internet verwendet wird; besteht hauptsächlich aus den Protokollen TCP (OSI Schicht 4) und IP (OSI Schicht 3)
<b>Timeout</b>	Bezeichnung der Zeiteinheit, nach der eine Aktion ausläuft und ungültig wird (z. B. 2-minütiger timeout beim Aufruf einer Website, wenn nach 2 Minuten noch keine Antwort kommt, wird der Aufruf abgebrochen)
<b>TLS</b>	Transaction Layer Security; standardisierte Variante von SSL (aber nicht kompatibel zu SSL)
<b>Traffic</b>	Verkehr (gemeint ist damit Datenverkehr)
<b>Transaktion</b>	Abgeschlossene Einheit einer Menge von Aktionen (Beispiel einer Transaktion: Bestellung einer CD in einem E-Business-Shop)

---

**U**

<b>Unix</b>	Betriebssystem, welches ursprünglich in den 70er-Jahren von Bell Laboratories entwickelt wurde und nun als Basis für eine ganze Familie von Betriebssystemen (u. a. Linux, Solaris, BSD etc.) dient
<b>UPS</b>	Uninterruptible Power Supply: Unterbrechungsfreie Stromversorgung
<b>Update</b>	Aktualisierung einer Anwendung/Programm zur Fehlerbehebung oder -korrektur bzw. um kleinere Funktionalitäten zu liefern (Versionsnummer ändert nicht an erster Stelle)
<b>Upgrade</b>	Versionssprung auf eine neue Version bzw. Installation einer höherwertigen Version (Versionsnummer ändert an erster Stelle, Bezeichnung z. B. «Office Basic» zu «Office Professional»)
<b>Uptime</b>	Zeitdauer, während der ein System erreichbar ist
<b>URL</b>	Uniform Resource Locator, technische Bezeichnung für Webadresse wie z. B. www.tagi.ch
<b>User</b>	Benutzer
<b>USV</b>	Unterbrechungsfreie Stromversorgung, engl. UPS uninterruptible power supply

---

**V**

<b>Virtual Host</b>	Bezeichnung für einen virtuellen Webserver (bzw. Domain), welcher auf einem physischen Server betrieben wird (d. h. unter einer IP-Adresse sind mehrere Domains erreichbar, welche als virtual hosts auf dem Webserver konfiguriert wurden)
<b>VLAN</b>	Virtual LAN; nicht zu verwechseln mit WLAN (Wireless LAN); Möglichkeit zur Trennung von Netzwerken auf OSI Schicht 2

---

**W**

<b>WAN</b>	Wide Area Network; Bezeichnung für ein globales Netzwerk, welches sich über geografische Grenzen erstreckt; oft auch als Synonym für das Internet verwendet
<b>Web</b>	Abkürzung für World Wide Web; der «sichtbare» Teil des Internets welcher auf dem Protokoll http basiert
<b>web-enabled</b>	Bezeichnung für eine Anwendung, die «web-befähigt» wurde; d. h., die Anwendung kann auch mit dem http-Protokoll angesprochen werden und ist somit auf einem Webserver betreibbar
<b>Webserver</b>	System, welches http-Anfragen beantwortet und Webseiten verwaltet und zur Verfügung stellt
<b>Windows</b>	Betriebssystem der Firma Microsoft
<b>WLAN</b>	Wireless LAN; LAN über Luft (ohne Kabel); basiert auf der Protokollfamilie IEEE 802.11 (im Vergleich dazu Ethernet (kabel-basiert): IEEE 802.16)
<b>WWW</b>	World Wide Web

---

**Z**

<b>Zone</b>	Eintrag im Konfigurationsfile eines DNS-Server
-------------	--



## Stichwortverzeichnis

### A

Abnahmeprotokoll	126
Access Control Lists	37
ACL	37
Administration	37
Anbieter	27
Anforderungen	12
Anwendungen	16
Apache Webserver	54
Apache-Module	69
Applikatorische Tests	43
Architektur	25
Ausgangslage	11
Authentifizierungsserver	106

### B

Back-up	105
Benutzerdokumentation	46
Benutzerprofile	36
Betrieb des Internetserver	111
Betriebsdokumentation	46
Betriebssystem für den Internetserver	39
Betriebsübergabe	126

### C

CommuniGate	7, 53, 89, 90, 91, 92, 97, 98, 99
CRUD	36

### D

Datenbanken	106
Datenschutz	34
DNS für Linux	77
DNS-Server	23
Dokumentation	5, 7, 37, 43, 45, 63, 64, 70, 73, 74, 90
Dynamisch	16

### E

Einsatzgebiet	15
Ethernet	24, 133, 134, 137, 138

Externe Zugriffe	37
Externer Logserver	101

### F

Fileserver	17
Fremde Ressourcen	106
FTP-Server	23

### H

Harddisk	21, 27
Hardware	7, 19, 20, 21, 22, 24, 25, 32, 38, 39, 40, 43, 45, 66, 132, 134, 137
http	7, 15, 16, 23, 24, 25, 31, 34, 42, 44, 49, 54, 55, 56, 58, 61, 64, 68, 69, 70, 72, 73, 74, 75, 76, 81, 89, 90, 99, 135, 138

### I

Installationsabhängigkeiten	40
Installationsdokumentation	45
Internetdienste	19
Internetprotokolle	23
Internetserver	5, 6, 7, 12, 15, 16, 18, 20, 21, 22, 25, 26, 27, 31, 32, 34, 35, 36, 38, 39, 40, 41, 42, 43, 44, 45, 46, 48, 49, 52, 81, 101, 105, 106, 107, 111, 126, 134, 135, 136, 137
IPERKA	12
Ist-Situation	31

### K

Know-how	38
Kompetenz	6

### L

Lasttest	45
Layer 1	24
Layer 2	24
Layer 3	24
Layer 4	24
Layer 5	24
Layer 6	25
Layer 7	25
Links	7
Log-Services	101

**M**

Modul 123	7
Modul 127	7
Modul 239	5

**N**

Namen	41
Nameserver BIND	82
Netzwerk	21

**P**

Personendaten	34
Postfach	89
Projekt	11
Projektarbeit	12
Projektmanagement	12
Prozessor	20, 36

**R**

RAM	20, 38, 63, 118
-----	-----------------

**S**

Serverumgebung	32
SEUSAG	13
Sicherheit gewährleisten	39
Sicherheitsrichtlinien	34
Sicherheitstests	44

Sicherungsprozeduren	101, 105
SMTP	89
Software	21
Soll-Zustand	34
Standardeinstellungen	42
SuSe Linux 10 OSS	49
Systemabnahme	126
Systemtest	5, 43
Systemüberwachung	111

**T**

Technische Tests	43
Technische/Methodologische Voraussetzungen	6
Traffic	35
Transaktions-orientiert	16

**V**

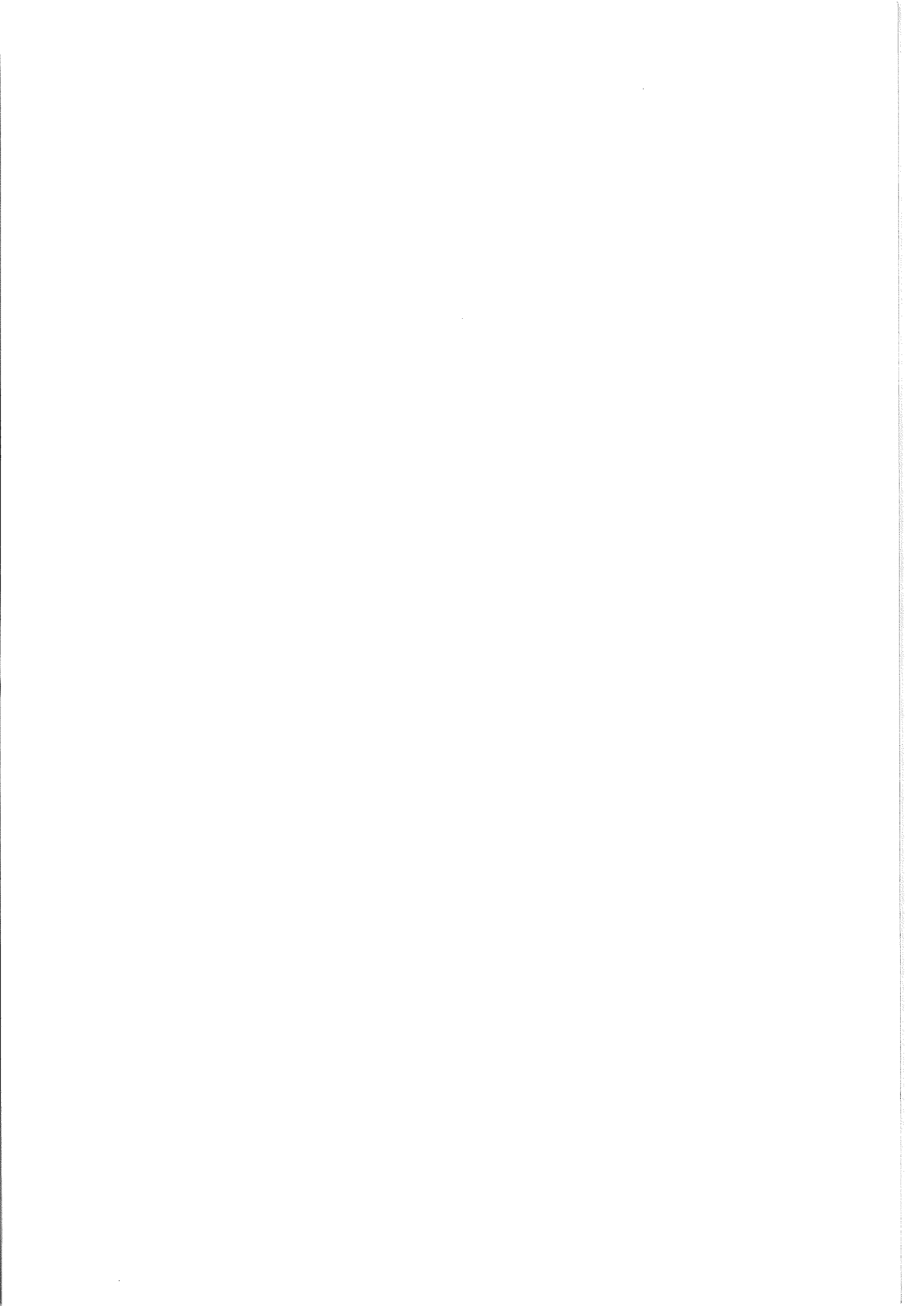
Virtuelle Hosts	66
Volumen	35

**W**

Wartung	111
Webserver	22

**Z**

Zonendateien	86
Zugriffskontrolle	37





**Umberto Annino, Dipl. Wirtschaftsinformatiker**

Nach der KV-Lehre zum Informatik-Anwendungsentwickler arbeitete er für verschiedene Unternehmen im IT-Bereich. Dabei lag der Schwerpunkt auf Data Warehousing und Reportingssystemen im Bankenbereich. Nebenbei absolvierte er die Ausbildung zum Informatiker mit Fachausweis und anschließend zum diplomierten Informatiker. Heute arbeitet er in einer Informatik-Sicherheitsfirma in Zug im Bereich Netzwerk- und Applikationssicherheitssysteme und schliesst gerade das FH-Nachdiplomstudium in Integrativem Qualitätsmanagement ab. Er unterrichtet an verschiedenen Schulen Fächer der Informatik- und Informationssicherheit und ist bei der WISS seit 2001 als Dozent für verschiedene Module des Bereichs «Security» tätig.

Dieses Lehrbuch vermittelt die Grundlagen für die Planung, das Aufsetzen und den Betrieb eines Internet-servers. Der behandelte Stoff umfasst die Inhalte, die für das Modul 239 «Internetserver in Betrieb nehmen» der Informatik-Grundausbildung vorgegeben sind.

Das Kernthema bildet der Internetserver mit den Diensten Web (HTTP), E-Mail (SMTP, Postfach), File-server (FTP) und Domain Service (DNS).

Im ersten Teil werden die Grundlagen eines Internet-servers beschrieben sowie das Vorgehen für die Informationsbeschaffung (Planung) dargelegt. Im zweiten Teil steht die Erhebung der IST- und Planung der SOLL-Situation im Vordergrund. Im dritten Teil wird die Realisierung des Internetservers mit beispielhaften Konfigurationen und detaillierten Erklärungen der Konfigurationmöglichkeiten vorgenommen. Der letzte Teil fokussiert auf das Testen und die Betriebsübergabe inklusiv Dokumentation und Sicherheitsanforderungen an den Internetserver.

Das Lehrmittel richtet sich an Lehrlinge und Schülerinnen einer Informatik-Erstausbildung. Es kann aber auch in der Erwachsenenbildung eingesetzt werden, beispielsweise für Berufsumsteiger oder in der Weiterbildung.