

4 Einen Webserver einrichten

Bis hierher haben wir einiges über die Hintergründe und das Einrichten von TCP/IP kennen gelernt. Außerdem haben wir uns mit der Hardware, dem Begriff des Servers sowie der Strukturierung eines Intranets und seinen Diensten beschäftigt. Nun wird es um den ersten zentralen Server im Rahmen vieler Intranets gehen: Wir wollen die Einrichtung, die Administration und den Betrieb eines Webserver behandeln. Hinzu kommen allgemeine Hintergrundfragen zur Arbeitsweise eines Webserver und die speziellen Anforderungen, wenn Sie Ihren Webserver im Internet verfügbar machen wollen.

4.1 Um was es geht

Wir werden in diesem Kapitel die Bereitstellung eines Webserver in einem Intranet in Angriff nehmen. Die Kenntnis um das WWW möchte ich voraussetzen und nur einige Grundlagen des WWW anreißen. Wir werden konkret zwei Webserver einrichten.

Sambar¹ ist ein einfach einzurichtender Webserver unter Windows und mittlerweile auch Linux, der zudem diverse weitere Features bereitstellt. Ich halte ihn aus didaktischer Sicht zum Einstieg in den Webserverbetrieb für sehr gut geeignet. Außerdem ist er für den praktischen Betrieb in einem Intranet sehr brauchbar, denn in Sambar sind zusätzlich ein FTP-Server sowie diverse Proxys (Server für den Übergang zum Internet) integriert. Letzteres qualifiziert ihn als Basisprogramm für einige weitere Kapitel (und Sie sparen sich ein neues Programm). Und schließlich ist er – bis auf wenige Ausnahmen – kostenlos einzusetzen. Dazu ist Sambar in Relation zu seiner Leistungsfähigkeit einfach zu installieren, sehr komfortabel zu konfigurieren und in diver-

Sambar

1. Nicht mit Samba ohne r zu verwechseln – das ist ein Programm zum direkten Zusammenspiel zwischen Linux und Windows.

sen Syntaxelementen mit Apache kompatibel. Im Grunde kann man Sambar nur vorwerfen, dass das Webinterface bei einer deutschen Einstellung einen ziemlichen Sprachmischmasch zwischen Deutsch und Englisch produziert.

Apache Der zweite Webserver, den wir behandeln wollen (und müssen!), ist Apache. Diese Server ist im Internet das Maß aller Dinge. Er ist der mit Abstand am häufigsten eingesetzte Webserver und äußerst leistungsfähig. Wir wollen ihn sowohl unter Linux als auch Windows aufsetzen. Bezüglich Apache werden wir uns aber kurz fassen, denn er ist dermaßen leistungsfähig und damit auch komplex, dass man ihn entweder richtig (und dann nicht einfach auf ein paar Seiten) behandeln sollte oder ihn nur mit der Installation und dem Betrieb in einer Grundkonfiguration vorstellen kann. Sie werden aber aufgrund der Ausführungen zu Sambar viele Details in Apache wiedererkennen.

Alternativen Als Alternativen zu den beiden Hauptwebservern in diesem Buch kommen kurz der Xitami-Webserver zur Sprache sowie die Webserver PWS (Personal Web Server) und IIS (Internet Information Server), die in die meisten Windows-Varianten integriert sind.

Bei der Behandlung der Webserver werden wir eingehen auf das Bereitstellen von Daten über den Webserver, die Arbeitsweise des Webserver im Hintergrund, die tatsächliche Dateistruktur und die via HTTP-Zugriff sichtbare Dateistruktur. Außerdem behandeln wir Details zum Umgang mit serverseitigen Dateien (HTML, Skripte, ...), den Umgang mit dem Clientprogramm (Browser), den Zugriff von außerhalb auf den Server und erste Sicherheitsfragen.

Sicherheit Auch wenn es Sie im ersten Moment vielleicht etwas enttäuschen wird, möchte ich bereits am Anfang eine Einschränkung deutlich machen: Zu einem richtig sicheren und störungsfreien Betrieb eines Webserver, der zum Internet offen ist, gehört mehr als das, was wir hier innerhalb dieses Buchs ansprechen können. Das gilt auch für die anderen Server, die wir aufsetzen werden. Sie lernen hier die Grundlagen, auf denen diese Server aufsetzen, und gegebenenfalls können Sie die detaillierten Sicherheitsfragen später erarbeiten (was didaktisch auch erst nach den Grundlagen erfolgen sollte).

4.2 Das WWW

Wenn Sie in Ihrem Intranet einen Webserver aufsetzen und darüber Informationen bereitstellen, erzeugen Sie quasi ein kleines WWW. Das WWW ist ja ein Internetdienst, der wie ein (logisches) Spinnennetz Informationen miteinander verknüpft und sie Besuchern bereitstellt.

Es gibt diverse Techniken und Protokolle, die den Grundaufbau des WWW bilden. Zuerst sind die Webserver zu nennen, die die Daten bereitstellen. Dann ist ein Client von Nöten, damit ein Serverdienst überhaupt genutzt werden kann. Diese WWW-Clients sind die Webbrowser. Nun fehlen noch zwei grundsätzliche Elemente:

Aufbau des WWW

- Protokolle, die klären, wie die zu übertragenden Daten zu verstehen sind.
- Sprachen und Techniken, die die Darstellung von Webseiten beschreiben bzw. das gesamte Prozedere programmierbar machen.

4.2.1 Protokolle

Die Datenübertragung im Internet und in einem Intranet ist über die TCP/IP-Familie geregelt. Dadurch ist zwar dafür gesorgt, dass Daten von Rechner A zu Rechner B übermittelt werden, aber noch nicht festgelegt, wie die ankommenden Daten zu verstehen sind. Wenn Sie einen Brief nach China schicken wollen, ist das leicht möglich – vorausgesetzt Sie kennen die Zieladresse und werfen den entsprechend korrekt beschrifteten Brief in einen Briefkasten. Die international zusammenarbeitenden Postorganisationen sorgen (meistens) für die korrekte Zustellung. Ebenso kann Ihnen Ihr Partner antworten, wenn Sie Ihren Absender auf dem Brief notiert haben. Aber angenommen, Ihr Partner ist ein Chinese – versteht er dann auch, was Sie ihm geschrieben haben? Und verstehen Sie seine Antwort (unter der Annahme, dass Sie kein Chinese sind)? Wenn ich davon ausgehe, dass Sie kein Mandarin können und Ihr Kommunikationspartner kein Deutsch, werden Sie zwar Daten austauschen können, jedoch kein Wort verstehen. Sie sollten sich auf eine Sprache einigen, die Sie beide verstehen. Das entspricht dann einem Anwendungsprotokoll.

Auf der obersten Ebene des Schichtenmodells zur Netzwerkkommunikation (gleich ob OSI oder DoD) regeln Anwendungsprotokolle, wie Daten zu modulieren und zu verstehen sind, damit die Kommunikationspartner damit klarkommen. Für das WWW ist vor allem das Protokoll HTTP (HyperText Transfer Protocol) entscheidend. Das ergänzende Protokoll HTTPS (S ist die Abkürzung für Secure) dient zum Aufbau von sicheren Verbindungen und wird meist in Zusammenhang mit Verschlüsselungsverfahren verwendet. HTTP arbeitet standardmäßig auf Port 80 und ist im RFC 1945 (alter Standard) bzw. 2068 (neu) spezifiziert.

Anwendungsprotokolle

4.2.2 Sprachen und Techniken

Sprachen und Techniken

Im WWW gibt es eine ganze Reihe von Sprachen bzw. Techniken, aus denen Webseiten aufgebaut werden. Deren Unterteilung kann man verschiedenartig angehen. Ich möchte sie in vier Kategorien einteilen:

- HTML bzw. XHTML
- Textdarstellung außer HTML
- Programmier- bzw. Skriptsprachen samt Datenbanktechnik
- Multimediatechniken

HTML (HyperText Markup Language) bzw. dessen syntaktisch strengerer Ableger XHTML (Extended HyperText Markup Language) ist die Grundlage jeder Webseite. Diese Beschreibungssprache verwendet jede Webseite, zumindest zum Erzeugen eines Grundgerüsts, in dem sich dann verschiedenste andere Techniken verankern und worüber auch Aktionen auf dem Server ausgelöst werden. Eine HTML-Seite wird von einem Browser beim Webserver angefordert, von diesem ungeprüft an den Client geschickt und beim Client interpretiert. Ein Browser kann aber auch reinen Text darstellen (um sozusagen das untere Ende der Leistungsfähigkeit darzustellen) oder auch Grafiken mit bestimmten Formaten (etwa GIF, JPEG oder PNG). Da sich in eine Webseite die unterschiedlichsten Ergänzungsformate integrieren lassen, können jene Webseiten erzeugt werden, die man heute in all ihrer Pracht kennt (angefangen mit einfachen Bildern über Stilvorlagen bis hin zu multimedialen Elementen wie Animationen).

Clientseitige Programmierung

Im Rahmen einer Webseite kann auch programmiert werden. Dazu gibt es einmal die Skriptsprachen wie JavaScript oder VBScript, welche den Browser selbst steuern. Die Skriptbefehle lassen sich sowohl in eine HTML-Seite integrieren als auch als externe Dateien hinzubinden und werden dann vom Browser umgesetzt. Oder es werden kleine Programme in eine Webseite integriert (etwa Java-Applets oder ActiveX-Controls), die auf dem Clientrechner ausgeführt werden und nicht mehr den Beschränkungen des Browsers unterliegen². Diese Techniken fasst man unter dem Begriff *clientseitige* Programmierung zusammen.

Serverseitige Programmierung

Für viele Fälle genügen die reine Darstellung von Webseiten und die Programmierung auf Clientseite nicht. Gerade bei der Interaktion mit dem Besucher oder dem Speichern von Daten hat man mit rein clientseitigen Techniken wenig Chancen. Aber auch der Webserver kann programmiert werden. Die serverseitigen Programmieretechniken

2. Wohl aber den Sicherheitseinstellungen – im Fall von Java-Applets (nicht bei ActiveX-Controls).

sind sehr gefragt, denn Gästebücher, Zugriffszähler, Onlineshops oder Auswertungen von Besucherbefragungen sind nur dann möglich, wenn man auf dem Serverrechner gezielte Aktionen ausführen kann.

Wie auf einem Clientrechner kann man auf dem Serverrechner mit Skript- bzw. Programmiertechnologien arbeiten. Sehr populär ist derzeit PHP, aber auch Active Server Pages (ASP), Java Server Pages (JSP), Server Side Includes (SSI), Java-Servlets oder Perl zählen zu diesem Kreis. Historisch wäre auch noch das Common Gateway Interface CGI zu erwähnen, was aber einen Standard bezeichnet und keine konkrete Sprache. Manche Server haben von Hause aus die Fähigkeit zum Umgang mit einigen dieser Techniken integriert, andere können sie in Form von Modulen hinzufügen (wobei dem Webserver in seinen Konfigurationseinstellungen mitgeteilt werden muss, wo sich die fraglichen Module befinden). Sehr wichtig ist bei der *serverseitigen* Programmierung auch, dass man viele Webserver direkt oder meist indirekt über eine der angesprochenen serverseitigen Programmiermodule mit einer Datenbank (etwa MySQL) erweitern kann.

Grundsätzlich muss man sich als Betreiber eines Webserverns und beim Umgang mit serverseitigen Programmier Techniken zumindest grob im Klaren sein, wie diese Techniken arbeiten. Da gibt es zum einen die Techniken, die als eigenständiges Programm auf dem Serverrechner aufgerufen werden. Etwa Java-Servlets: Sie werden im Rahmen der auf dem Serverrechner installierten, virtuellen Java-Maschine (JVM) ausgeführt. Ähnlich funktioniert es bei EXE-Dateien, die ebenfalls per Aufruf über den Webserver ausgeführt werden. Häufiger werden aber auch auf dem Server Skripttechniken eingesetzt. Dabei wird ein Skript auf dem Server aufgerufen und dieses vom Server bzw. einem verfügbaren Zusatzmodul interpretiert.

Wenn ein Anwender von einem Webserver eine Datei anfordert und der Server nicht nur die Datei zum Client schicken, sondern darüber hinaus aktiv Aktionen ausführen soll, ist eine Kennzeichnung notwendig, die anzeigt, wie diese Dateien zu erkennen sind und wie der Webserver damit umzugehen hat. Diese Kennzeichnung erfolgt – wie allgemein üblich – über bestimmte Dateierweiterungen. Historisch wird von vielen Webservern oft die Erweiterung *.CGI* akzeptiert. Weitere Erweiterungen wie *.PL* (für Perl) oder *.PHP* (für PHP) hängen davon ab, welche Techniken der Webserver unterstützt. Bei SSI bekommen solche Dateien meist die Dateierweiterungen *.SHTM* oder *.SHTML* (das lässt sich aber in vielen Webservern konfigurieren und wir werden es in der Praxis auch sehen).

Server-Parsing

Oft werden nun serverseitige Skripte in einer Klartextdatei mit reinem Text gemischt, der vom Server nicht ausgeführt werden soll (als reiner Text ist in diesem Zusammenhang auch HTML zu verstehen –

das sind Anweisungen, die nur der Client interpretiert). Wird vom Besucher eine Datei angefordert, die so gekennzeichnet ist, dass darin Befehle für den Webserver enthalten sind, wird er sie nicht ungeprüft zum Client schicken. Der Server wird sie untersuchen (parsen) und überprüfen, ob darin Befehle sind, die er versteht. Das wird so bei SSI, JSP oder ASP gemacht. Diese Techniken basieren auf der Grundlage des *Server-Parsing*. Bei gewöhnlichen Webseiten erhält der Server eine Anforderung für eine Webseite und sendet diese ohne Überprüfung an den Client zurück. Wenn eine Datei jedoch Server-Parsing fordert, analysiert der Server die Klartextdatei vor der Rücksendung und sucht nach serverseitigen Befehlen. Erkennt der Webserver einen solchen Befehl, führt er die entsprechende Aktion aus.

4.3 Sambar

Starten wir jetzt mit der Einrichtung von Sambar. Sie können das Programm aus dem Internet unter <http://www.sambar.com> laden. Die Setup-Dateien sind etwa 5,7 MByte groß. Mittlerweile ist dieser Server sowohl unter Windows als auch unter Linux verfügbar. Als Setup-Programm für Windows erhalten Sie eine *.exe*-Datei, die Sie zur Installation direkt ausführen können. Für die Linux-Version erhalten Sie ein *.tar.gz*-Archiv. Wir werden Sambar in der Version 6 als Referenz verwenden. Beachten Sie, dass sich Vorgängerversionen teilweise im Webinterface und teilweise auch im Leistungsumfang unterscheiden. Sambar wird uns durch mehrere Kapitel des Buchs verfolgen, denn einzelne Features von Sambar können hervorragend als Basis mehrerer Kapitel dienen. Allerdings ist Sambar mittlerweile stark im Leistungsumfang angewachsen. Er ist nicht nur ein Webserver, sondern kann auch als FTP-Server (siehe Kapitel 5), Proxy für verschiedene Dienste (siehe Kapitel 10) und noch diverse weitere Dinge wie E-Mail-Server verwendet werden. Wir behandeln nur einen Teil der Funktionen von Sambar und konzentrieren uns in diesem Kapitel auf die reine Webserver-Funktionalität.

4.3.1 Sambar installieren

Unter Windows

Die Installation von Sambar ist unter Windows denkbar einfach. Sie entspricht der üblichen Installation von Windows-Programmen und sollte keinerlei Fragen oder gar Schwierigkeiten aufwerfen. Die Datei *Setup.exe* löst den unter Windows üblichen Installationsprozess aus. Der Server kann in ein beliebiges Laufwerk und Verzeichnis außer dem Stammverzeichnis eines Laufwerks installiert werden. Die Setup-Routine erzeugt alle notwendigen Verzeichnisstrukturen und Einträge.

Unter Linux ist die Sache noch einfacher – Sie brauchen bloß das Archiv zu entpacken (etwa mit dem Programm *ark*). Speichern Sie es dann, etwa in das Verzeichnis */usr/local* oder auch in Ihr Home-Verzeichnis. Hierbei benötigen Sie – je nach Zielverzeichnis – unter Umständen Root-Rechte. Es werden in etwa die gleichen Unterverzeichnisse angelegt, die die Setup-Routine unter Windows erzeugt³. Sambar ist in beiden Welten sofort startklar.

Unter Linux

4.3.2 Sambar starten und grundlegend konfigurieren

Die Installation von Sambar führt bereits eine Grundeinstellung durch, die zumindest für einen ersten Test (unter Umständen sogar für einfache Praxisfälle) bereits ausreicht.

Der erste Start

Nach der Installation von Sambar unter Windows finden Sie gewöhnlich einen Eintrag im Menü *Programme* bzw. ein Icon auf dem Desktop, über das Sie Sambar starten können. Nach dem Start wird in der Taskleiste ein Symbol sichtbar, das die Aktivität des Servers anzeigt und über das ein Monitorfenster geöffnet werden kann. Über das Kontextmenü auf dem Icon kann der Server überdies neu gestartet oder heruntergefahren werden.

Unter Windows

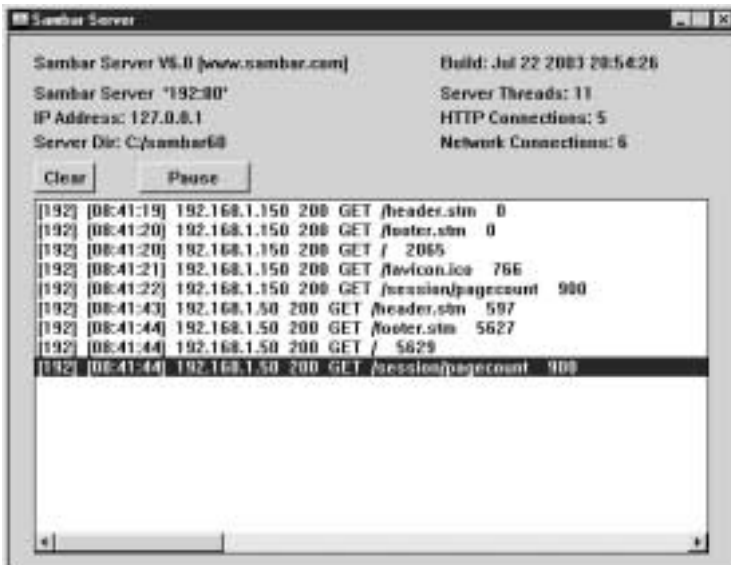


Abb. 4-1

Der Monitor von Sambar unter Windows

3. Es gibt allerdings ein paar Unterschiede, was auch aufgrund der Plattform verständlich ist.

Mehr sieht man erst einmal nicht von dem Server. Er läuft als Hintergrundprozess ohne grafische Aktivität (das haben wir im letzten Kapitel ja besprochen). Aber der Browser auf dem gleichen Rechner ist nach dem Start sofort in der Lage, mit diesem lokalen Server zu kommunizieren (über den URL *localhost*). Andere Clientrechner im Intranet müssen unter Umständen noch entsprechend eingerichtet werden (entweder Sie haben das bereits gemacht oder wir werden gleich noch dazu kommen).

Unter Linux

Wenn Sie Sambar unter Linux installiert haben, ist der Start nicht ganz so komfortabel. Sie müssen innerhalb einer Shell im Installationsverzeichnis (etwa */usr/local/sambar*) in das *bin*-Verzeichnis wechseln und dort *./server* eingeben. Ein paar kurze Meldungen in der Shell erscheinen, und das war es.

Abb. 4-2

Linux-typisch gibt es nach dem erfolgreichen Start des Servers nur ein paar kurze Textmeldungen.



```

rj@p2: - Befehlsfenster 2 - Konsole <2>
Strg  Befehlsfenster  Ansicht  Leerzeichen  Einstellungen  Hilfe
p2:/usr/local/sambar60b31/bin # ./server
Sambar Server 06.00 running on 'p2' [port 80]
(c) Copyright Tod Sambar 1995-2003
All rights reserved.
Commercial distribution of this software prohibited.
mailto: tod@sambar.com for product information
  
```

Mehr gibt Sambar erst einmal nicht von sich. Aber der Aufruf über einen Browser funktioniert, und darum geht es ja.

Konfiguration

Über ein Webinterface

Die Konfiguration des Sambar-Servers funktioniert nicht über ein eigenes Dialogprogramm, sondern über ein HTML-Formular im Browser oder mit Hilfe eines gewöhnlichen Texteditors. Insbesondere erlaubt die Technik des HTML-Webinterfaces grundsätzlich auch eine Konfiguration des Servers von einem beliebigen anderen Rechner im Internet/Intranet aus. Sämtliche Einstellungen von Sambar liegen als Klartextdateien im Installationsverzeichnis des Servers vor, aber um etwas komfortabler an die wichtigsten Einstellungsmöglichkeiten von Sambar heranzukommen, starten Sie zuerst einen Browser auf dem Rechner, auf dem Sambar läuft und geben dort dann einfach *localhost* als

URL ein. Die Standardstartseite von Sambar wird angezeigt. Unmittelbar nach der Installation ist die Administration per Webinterface aus Sicherheitsgründen nur vom lokalen Rechner (und in einigen Sambar-Versionen von ein paar ausgewählten IP-Adressen im Intranet) aus möglich (Voreinstellung)! Das liegt daran, dass es in der Grundeinstellung kein Passwort für den Administrator gibt! Außerdem sollten Sie beachten, dass die Übermittlung von sensiblen Daten über die Webinterface-Administration ein Sicherheitsproblem sein kann, denn in der Regel (außer bei SSL-Verwendung) werden alle Eingaben, auch Passwörter, unverschlüsselt zwischen Client und Server übertragen! Das ist selbst in einem kleinen Intranet schon dann ein Problem, wenn man etwa ein nicht abgesichertes WLAN verwendet oder nicht allen Teilnehmern im Netz vollständig trauen kann.

Über den Link *System Administration* können Sie sich zum Konfigurieren des Servers einloggen. Da dies natürlich nicht jeder Besucher darf, muss man sich als Administrator einloggen. Dem Zugriff auf die Folgeseiten ist deshalb ein Anmeldedialog vorgeschaltet. Als Benutzername sollte zuerst *admin* und kein Passwort eingegeben werden (Voreinstellung von Sambar).

Der Link System
Administration



Abb. 4-3

Die Startseite von Sambar
mit Anmeldedialog –
hier die Linux-Variante

Auf der nachfolgenden Seite kann Sambar nun konfiguriert werden⁴. Die Konfiguration kann wie gesagt nur der Administrator übernehmen. Beachten Sie, dass Sambar Sie ausloggt, wenn Sie längere Zeit inaktiv sind, obwohl eine Konfigurationsseite möglicherweise noch

Änderungen werden erst
nach dem Neustart des
Servers aktiv.

angezeigt wird. Wenn Sie Änderungen der Einstellungen von Sambar durchführen, werden die meisten Modifikationen erst nach dem Neustart des Servers aktiv! Dazu können Sie in Windows mit der rechten Maustaste das Icon des Servers in der Taskleiste anklicken und *Restart* auswählen. Alternativ kann der Neustart aber auch über die Konfigurationsseiten von Sambar erfolgen. Dort finden Sie zumindest in der Linux-Variante einen Link zum Shutdown. Gestartet wird die Linux-Variante wieder aus der Shell, und im Grunde ist das auch die bessere Stelle zum Herunterfahren.

Abb. 4-4

Der Konfigurationsbereich
von Sambar



4.3.3 Sambar als Webserver konfigurieren

Für den Sambar-Server können nun verschiedene Einstellungen vorgenommen werden, von denen wir uns eine Auswahl (!) ansehen. Wir konfigurieren Sambar vor allem nur soweit, wie es seine Aufgabe als Webserver betrifft. Die Konfiguration der FTP- und Proxy-Features besprechen wir in den entsprechenden Server späteren Kapiteln des Buchs. Die restlichen Möglichkeiten von Sambar werden wir außen vor lassen.

4. Beachten Sie, dass fast alle Seiten des Webinterface in der Linux- und in der Windows-Variante gleich aussehen.

Den Server konfigurieren

Wenn Sie das Webinterface zur Administration aufrufen, gelangen Sie zur Hauptseite der Konfiguration. Die Benutzerführung von Sambar hat sich in der Version 6 bezüglich der Konfiguration gegenüber den Vorgängerversionen stark verändert. Ganz oben finden Sie nun die Links *System*, *HTTP*, *Sendmail*, *Server*, *Sicherheit*, *Dienste* und *Werkzeuge*. Je nachdem, welchen der Links Sie anklicken, bauen sich in der Zeile darunter neue Hyperlinks auf, die zu den konkreten Einstellungsmöglichkeiten führen.

Die Hauptseite zur Konfiguration

Unter *System-Konfigurieren* (bei deutscher Einstellung) finden Sie die wichtigsten Einstellungsmöglichkeiten zu allen verfügbaren Servern von Sambar. Schauen wir uns die wichtigsten davon an (diejenigen, die wir nicht ansprechen, können Sie in der Regel einfach in der Voreinstellung belassen).

Der Link System – Konfigurieren

Zuerst können Sie jeden der im Server angemeldeten Besucher als Administrator registrieren. Die grundsätzliche Anmeldung im System erfolgt an anderer Stelle – unter *User Management*. Meist behält man *admin* bei. Die darunter befindliche Eingabe spezifiziert die IP-Adressen, von wo aus eine Administration via Webinterface möglich ist (diese Sicherheitseinstellung hatten wir oben schon angesprochen). Je restriktiver die Festlegung erfolgt, desto sicherer ist es natürlich. Am strengsten ist der ausschließliche Zugang nur vom gleichen Rechner über den URL *localhost*.

Admin-Zugriff

Der Port des Servers ist für Webserver mit 80 vorbelegt und sollte normalerweise auch so beibehalten werden (80 ist der Standardport für Webserver – mehr zu Ports finden Sie in Kapitel 11). Wenn Sie mehr als einen Webserver auf einem Rechner betreiben wollen, können Sie hier den Port für Sambar verändern. Diese Situation ist nicht so ganz abwegig. Es könnte der Fall eintreten, dass Sie auf Ihrem Rechner zusätzlich einen weiteren Webserver laufen lassen wollen, weil dieser ein Feature mitbringt, das Ihnen Sambar nicht liefert, Sie sonst aber auf Sambar setzen wollen. Dann muss einer der beiden Webserver den Port 80 räumen.

Port 80

Die maximale Anzahl der Verbindungen, die der Webserver gleichzeitig bedienen kann, ist eine Einstellung, die Sie in der Regel nicht zu verändern brauchen. Auch die nachfolgenden Angaben zur Sicherheit etc. vernachlässigen wir hier aus Platzgründen. Vor allem können sie meist bei der Standardeinstellung bleiben. Dies gilt insbesondere für den rein lokalen Einsatz von Sambar im Intranet. Ebenso sollten Sie die Einstellungen, ob Sambar als HTTPS- und FTP-Server arbeiten soll, hier außer Acht lassen.

Weitere Einstellungen

Der Link HTTP

Unter *System* gibt es jetzt für uns zunächst nichts zu konfigurieren. Wenn Sie in der Hauptseite zur Konfiguration ganz oben den Link *HTTP* und dann *Konfigurieren* anklicken, kommen Sie zu weiteren Einstellungsmöglichkeiten, die für den Betrieb des Webserver von Bedeutung sind.

Die Konfigurationsdateien

Sie können sich mit den Links ganz oben die vollständigen Konfigurationsdateien von Sambar anzeigen lassen: Dies sind *config.ini*, *iconmap.ini*, *mime.ini* und *vhosts.ini*. Für eine intensivere Beschäftigung mit Sambar ist das später sicher sinnvoll. Am Anfang ist aber die Konfiguration über das Webinterface einfacher, zumal Sie hier nicht editieren können. Das machen Sie – wie bei allen Anzeigen von Konfigurationsdateien im Browser – gegebenenfalls mit einem Editor.

Defaultseiten

Zunächst wird Ihnen noch einmal der Port angezeigt, den Sie schon weiter oben gesehen haben. Wichtig sind die beiden Angaben darunter. Zuerst sehen Sie die Defaulteinstiegsseiten, wenn Besucher nur den URL des Servers ohne Dateispezifikation angeben. Die Defaultseiten sind normalerweise *index.html* oder *index.htm*. Sambar verwendet noch *index.stm* und *index.asp* als Defaultseiten (je nach Situation). Wenn ein Besucher keine konkrete Datei bei einem URL angibt (etwa *http://127.0.0.01/* bzw. *http://127.0.0.01/verzeichnis/*), wird stattdessen *http://127.0.0.01/index.html* bzw. *http://127.0.0.01/verzeichnis/index.html* angezeigt.

Das Verzeichnis für HTML-Dokumente

Besonders wichtig ist das öffentliche Stammverzeichnis für HTML-Dateien (*Documents Directory*). Das ist das Verzeichnis, in dem alle zu veröffentlichenden Dateien untergebracht werden. Unter Sambar heißt das defaultmäßig *docs* und befindet sich unmittelbar unterhalb des Installationsverzeichnisses.

Das Verzeichnis für CGI-Skripte

Wenn wir jetzt einige Felder nach unten wandern, finden Sie die Angabe des Verzeichnisses, in dem CGI-Skripte zu stehen haben. Ein üblicher Verzeichnisname dafür ist *cgi-bin* – ebenfalls als relativer URL vom Sambar-Installationsverzeichnis aus gesehen. Wenn nun ein unter Sambar unterstütztes Skript aufgerufen werden soll, muss es mit `http://[Host]/cgi-bin/[Script-Name]` (oder allgemein `http://[Host]/[Name CGI-Verzeichnis]/[Script-Name]`) spezifiziert werden. Das ist bei sämtlichen Aufrufarten von Skripten (etwa als SSI, als Link oder Eingabe in der Adresszeile des Browsers) zu berücksichtigen. Zudem kann man unten noch bestimmte Dateierweiterung als CGI-Skripte definieren, die unabhängig von ihrer Position im Dateisystem von Sambar als CGI-Skripte akzeptiert werden. Bleibt das Feld leer, werden nur CGI-Skripte in dem CGI-Verzeichnis ausgeführt. Ebenso legen Sie die Dateierweiterung für SSI fest.

4.3.4 Die Clients einrichten

Damit nun Clients in einem Intranet überhaupt auf Sambar oder ein anderes Serverprogramm im Intranet zugreifen können, müssen sie entsprechend konfiguriert werden. Es muss eingestellt werden, wann ein Zugriff auf das Internet (falls ein Übergang vorhanden ist) erfolgen soll und wann ein Webserver innerhalb des Intranets zu suchen ist. Dazu kommen in der Regel zwei Varianten zum Einsatz, die wir in Kapitel 10 genauer untersuchen werden:

1. Sie geben für Ihren Rechner ein so genanntes Gateway an, über das Sie global auf das Internet zugreifen, und spezifizieren, in welchen Fällen Sie nicht auf das Internet zugreifen, sondern im lokalen Netz bleiben.
2. Sie geben in einem Clientprogramm entweder allgemeine LAN-Einstellungen oder einen so genannten Proxy an, über den dieses spezielle Clientprogramm Zugang zum Internet bekommt, und spezifizieren die Situationen, bei denen Sie im lokalen Netzwerk bleiben.

Im Fall eines Webbrowsers wollen wir uns auf den Fall 2 beschränken. Sie finden in jedem modernen Browser einen Menüpunkt zu Programmeinstellungen und dort eine Stelle zum Einrichten des LANs bzw. der Verbindungsart zum Internet und/oder eines besagten Proxys. Exemplarisch wollen wir das Prozedere zum Einrichten anhand des Internet Explorer 6 unter Windows und des Konqueror unter Linux durchspielen. Die Übertragung auf andere Browser ist leicht nachzuvollziehen.

Im Internet Explorer finden Sie unter *Extras-Optionen* in der Kategorie *Verbindungen* die *LAN-Einstellungen*. Sie stellen dort entweder ein, dass Sie keinen Proxyserver für Ihr LAN verwenden oder aber zumindest für lokale Adressen das Internet umgehen (siehe Abb. 4–5).

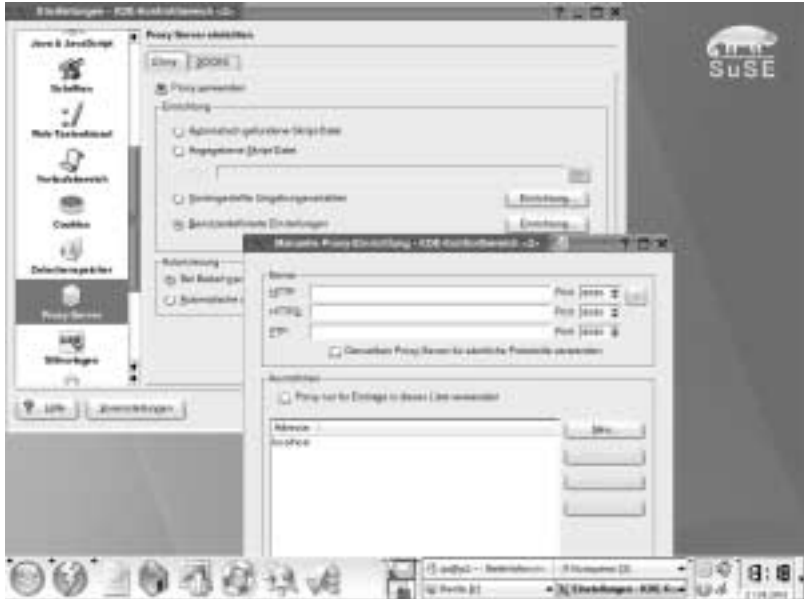
*LAN-Einstellung im
Internet Explorer*

Beim Linux-Browser Konqueror finden Sie die Verbindungseinstellungen unter *Einstellungen* und dort unter *Konqueror einrichten...* Dort gibt es eine Kategorie *Proxy Server*, wo Sie einstellen können, dass Sie für lokale Adressen nicht auf das Internet zugreifen.

*Verbindungseinstellungen
im Konqueror*

Sie sollten nun mit Ihrem Webbrowser auf den lokalen Sambar-Webserver zugreifen können.

Abb. 4-5
Einstellen der
Proxy-Funktionalität im
Internet Explorer



4.3.5 Verzeichnisstrukturen und Sicherheit

Wenn Sie einen Server betreiben, öffnen Sie immer ein Stück Ihres Eigentums (Ihren Rechner) für eine fremde Nutzung. Das heißt, Sie geben bei einem Webserver auch einen Teil Ihrer Festplatte frei.

Ein Webserver auf einem Rechner stellt immer eine bestimmte Verzeichnisstruktur bereit, die öffentlich zugänglich ist (im Fall von Sambar das Unterverzeichnis *docs*). Dabei bedeutet »öffentlich zugänglich«, dass entweder bestimmte Verzeichnisse zum Lesen, Schreiben oder auch zu beidem freigegeben sind. Oft findet man die Situation vor, dass eine Verzeichnisansicht verboten ist, aber Dateien aus eben diesem Verzeichnis abgerufen werden können. Dabei kann es sich zum Beispiel um HTML-Seiten handeln, die öffentlich zugänglich sein sollen, während die Verzeichnisansicht über eine Serverkonfiguration oder eine Defaultindexdatei unterbunden wurde. Ebenso können bestimmte Verzeichnisse zur Ausführung von aktiven Operationen auf der Serverplattform freigegeben werden. Im Fall von Sambar könnten das etwa CGI- oder ASP-Skripte im Unterverzeichnis *cgi-bin* sein.

Wenn Sie ein Serverprogramm installieren, werden innerhalb des Installationsverzeichnisses oder in einem außerhalb befindlichen Verzeichnis Verzeichnisstrukturen geschaffen, die einen solchen Zugang zu Ihrem Rechner freigeben. Wichtig für den Betreiber des Webserver ist, dass grundsätzlich sämtliche Zugriffe auf einen Serverrechner über das Serverprogramm auf dieses Verzeichnis und/oder ausgewählte

Unterverzeichnisse bzw. parallel befindliche Verzeichnisse beschränkt sind. Ist ein Zugriff auf oberhalb befindliche Verzeichnisse oder parallele, nicht freigegebene Verzeichnisse über das Serverprogramm möglich, kann man mit ziemlicher Sicherheit davon ausgehen, dass dies ein potenzieller Hackerzugriff ist, der Schwachstellen im Serverprogramm oder dem Gesamtsystem ausnutzt. Wird über einen Browser die Adresse eines Servers angegeben, darf der Anwender nur die freigegebenen Inhalte sehen und auf keinen Fall übergeordnete Zugriffe erhalten. Alle weiteren Verzeichnisangaben jenseits des Hosts im URL beginnen bei dem physikalischen Publikationsverzeichnis (nicht auf der Wurzel des Betriebssystems⁵). Das gilt auch (und erst recht) für FTP-Zugriffe, wie wir noch sehen werden.

Verdeutlichen wir diese Fragen anhand einiger Strukturen von Sambar unter Linux; wir gehen davon aus, dass das Programm in der Voreinstellung installiert wurde.

Das Installationsverzeichnis von Sambar sei in der Folge `/usr/local/sambar60` (Linux-Version). Darin befinden sich diverse Verzeichnisse.

Die Verzeichnisstruktur von Sambar

bin

Das Verzeichnis `bin` enthält die Programmdateien sowie einige Laufzeitbibliotheken. Gelingt es einem Hacker, die eigentliche Programmdatei von Sambar durch eine manipulierte Variante auszutauschen, ist das für den Betreiber eine absolute Katastrophe.

config

Das Verzeichnis `config` enthält die Konfigurationsdateien von Sambar in Form von Klartext. Schauen Sie einfach einmal in einzelne Dateien mit einem Editor hinein. Alle Konfigurationen, die Sie über das Webinterface vornehmen, werden hier gespeichert. Und noch diverse weitere Einstellungen, an die Sie über das Webinterface nicht direkt kommen (die aber selten angepasst werden müssen). Natürlich können Sie die Dateien per Klartexteditor verändern. Die Datei `config.ini` enthält die wichtigsten Einstellungen von Sambar. Darüber hinaus gibt es einzelne Dateien zu Datenbankeinstellungen, DHCP, DynDNS, Makros und vielem mehr. In dem Verzeichnis befindet sich u.a. auch eine Klartextdatei (`passwd`) mit den Passwörtern der eingerichteten Nutzer von Sambar (inklusive des Administrators). Zwar wird ein Passwort selbst verschlüsselt, aber es ist nur das kleine Hacker-1x1 notwendig, um dieses Problem zu beseitigen (darauf werde ich beim Thema Sicherheit

5. Für die Cracks: Sambar kennt auch virtuelle Verzeichnisse und virtuelle Hosts – wir werden in diesem Kapitel noch darauf eingehen.

noch eingehen). Ebenso sensibel sind die *ini*-Dateien, die die Sicherheitseinstellungen von Sambar betreffen. Sollte diese Datei via Browser, FTP- oder Remoteprogramm einem potenziellen Angreifer in die Hände fallen, wäre das gesamte Sicherheitskonzept von Sambar ausgehebelt. Es ist offensichtlich, dass das niemals der Fall sein darf und dieses Verzeichnis für den Zugriff von außen verborgen bleiben muss.

log

Sambar protokolliert in der Regel seinen Betrieb laufend. Welche Rechner haben wann zugegriffen, mit welchem Browser, was für ein Betriebssystem läuft, was wurde angefordert etc. Das Verzeichnis *log* beinhaltet solche und andere Protokollschritte in Form von Klartextdateien. Sowohl für eine Fehlersuche als auch für die Kontrolle der Besucher eines Webserver sind diese Dateien Gold wert. Auch sie sollten nur dem Administrator zugänglich sein.

docs

Das Verzeichnis *docs* ist das Verzeichnis, das für einen Zugriff über einen Browser freigegeben ist (wir haben das oben bei der Konfiguration gesehen). In diesem Verzeichnis und in den darin enthaltenen Unterverzeichnissen befinden sich alle Dateien, die der Webserver zurückliefert: Grafiken, HTML-Dateien, Java-Applets, Stylesheet-Dateien, Skriptdateien zur Ausführung auf dem Clientrechner etc. Außerdem gibt es noch Skripte, die aus einer Webseite heraus aufgerufen werden können – diese werden aber ausgeführt und nicht direkt als Dateien an den Aufrufer zurückgegeben. Innerhalb des *docs*-Verzeichnisses werden in der Regel für alle Personen, die ihre Daten über den Webserver veröffentlichen wollen, eigene Unterverzeichnisse angelegt. Ein Lesezugriff sollte daher auf das gesamte Verzeichnis samt enthaltenen Unterverzeichnissen möglich sein. Ein Schreibzugriff sollte aber nur berechtigten Benutzern gestattet sein. Das heißt, nur ein Eigentümer sollte jeweils Dateien in seinem Verzeichnis verändern dürfen – in der Praxis kommt hier meist FTP zum Einsatz.

Unterverzeichnisse für einzelne Nutzer

Über einen Webserver stellen normalerweise verschiedene Personen Webseiten und andere Dateien bereit. Denken Sie nur an Ihren Internetprovider, bei dem Sie Ihre Webseiten hosten. Die verschiedenen Nutzer erhalten dort jeweils innerhalb der Verzeichnisstruktur des Webserver ein eigenes Verzeichnis, auf das sie in der Regel unbeschränkten Zugriff haben⁶. Sie können – meist mit FTP – beliebige

Dateien auf ihren Weospace laden, Dateien löschen, umbenennen etc. In ihr eigenes Verzeichnis können diese Nutzer HTML-Dateien und Grafiken laden, aber auch Dateien, die der zugrunde liegende Webserver ausführen kann, etwa PHP-, Perl- oder ASP-Skripte.

cgi-bin und servlets

Wenn nun jemand über den Webserver auf eine ausführbare Datei zugreift und der Webserver sie einfach ausführt, könnte jedermann, der Weospace auf dem Webserverrechner hat, den Webserver beliebig programmieren und steuern. Das ist sehr oft nicht im Sinne des Administrators. Das Ausführen von serverseitigen Skripten oder Programmen kostet Prozessorzeit und damit auch Geld, belastet die Ressourcen und ist nicht zuletzt ein erhebliches Sicherheitsrisiko. Aus diesen Gründen wird sehr oft nur ein Verzeichnis innerhalb der Webserverstruktur zur Ausführung von serverseitigen Skripten bzw. Programmen freigegeben. Auf dieses haben dann nur ausgewählte Personen Zugriff, oder aber es werden interne Mechanismen für einen kontrollierten Zugriff vorangeschaltet. So ein Verzeichnis ist *cgi-bin*. Nur darin befindliche Skripte werden von Sambar auch ausgeführt (das wurde bereits angedeutet). Für den Spezialfall der Java-Servlets erfüllt das Verzeichnis *servlets* die gleiche Aufgabe. Skripte oder Programme, die sich in anderen Unterverzeichnissen des Installationsverzeichnisses befinden, werden von Sambar in der Grundeinstellung nicht ausgeführt. Da Sambar von Haus aus Perl unterstützt, finden Sie in *cgi-bin* als Muster einige Dateien mit der Erweiterung *.pl*.

4.3.6 Konkreter Zugriff auf Sambar von einem Client aus

Spiele wir den Zugriff auf den Sambar-Server über einen Client einmal durch und betrachten, was der Anwender sieht und wie er sich im System bewegt. Der Server soll in unserem Beispiel unter *192.168.1.233* erreichbar sein, und Sie geben einfach die Adresse von Ihrem Sambar-Host an.

*HTTP-Sicht via
Verzeichnissicht*

Bei der Browser-Eingabe *http://192.168.1.233* wird die Datei *index.stm* aus dem Verzeichnis *docs* angezeigt. Physisch ist das */usr/local/sambar60/docs/index.stm*. Bei *http://192.168.1.233/syshelp/quickst.htm* ist der tatsächliche Dateipfad */usr/local/sambar60/syshelp/quickst.htm*, den aber der Anwender natürlich nicht nachvollziehen kann und darf. Er sieht über den Browser nur die Struktur ab dem Wurzelverzeichnis von Sambar.

6. Auf unserem Server werden wir solche Nutzerbereiche später im Verzeichnis */usr/local/sambar60/docs* einrichten.

Abb. 4-6

Der User sieht die Adresse des Servers auf oberster Zugriffsebene.

**Abb. 4-7**

Die tatsächliche Verzeichnisstruktur zur Datei quickst.htm ab der Wurzel



4.3.7 Eigene Dateien bereitstellen

Sehen wir uns nun an, wie Sie über Sambar eigene Dateien zum Download bereitstellen. Das können HTML-Dateien, aber auch beliebige andere Dateien sein. Sämtliche Dateien, die Sie unter *docs* zur Verfügung stellen, können von jedem, der per HTTP Zugang zum Webserver hat, auf seinen Rechner übertragen werden. Später werden wir den Upload auf den Serverrechner per FTP erledigen, aber vorerst arbeiten wir einfach mit den Dateimanagern des Betriebssystems.

Sie können alle Dateien direkt in *docs* bereitstellen. Dann wären diese unmittelbar über die IP-Adresse des Sambar-Rechners und den Dateinamen erreichbar. Aber um die Sache besser zu strukturieren, legen wir unter *docs* ein Unterverzeichnis an und nennen es *felix*. Dort erzeugen wir eine einfache Webseite, die *Willkommen.html* heißen soll und nur etwas Text enthält.

Wenn Sie nun die IP-Adresse von unserem Rechner mit dem Sambar-Server angeben und den URL um den Verzeichnisnamen ergänzen, bekommen Sie den Inhalt des Verzeichnisses angezeigt (wobei die tatsächliche physikalische Struktur ab der Wurzel wie gesagt nicht sichtbar ist).

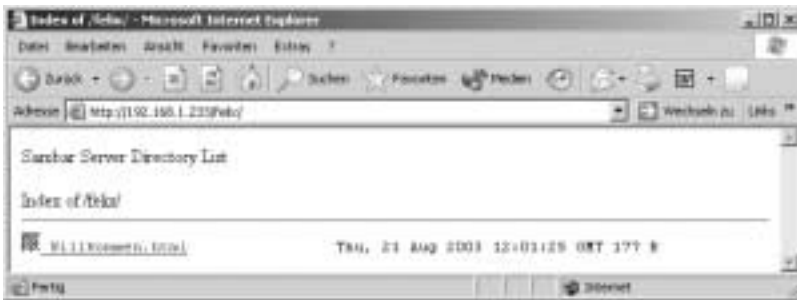


Abb. 4-8

Der Inhalt des
Unterverzeichnisses *felix*

Viele Webserver unterdrücken wie schon angedeutet eine solche Anzeige eines Verzeichnisinhalts, da sie in vielen Fällen nicht gewünscht ist und sogar ein Sicherheitsrisiko darstellt. Falls Ihr Webserver die Anzeige der Verzeichnisebene nicht automatisch unterdrückt, können Sie das auch über einen Trick selbst einstellen. Wenn Sie im Verzeichnis *felix* eine Datei *index.html* anlegen (diese kann leer sein), kann ein Besucher die Verzeichnisstruktur nicht mehr sehen. Auch wenn er keine Datei explizit angibt, würde stattdessen immer diese Defaultseite angezeigt.

Wenn Sie nun die Webseite direkt anklicken oder den vollständigen URL angeben, wird die Seite in Ihrem Browser angezeigt.



Abb. 4-9

Der vollständige URL zur
Webseite bewirkt die
Anzeige im Browser

Legen wir nun noch ein weiteres Unterverzeichnis von *docs* mit Namen *florian* an (parallel zu *felix*). Darin sollen sich eine Webseite befinden, die eine Grafik im gleichen Verzeichnis referenziert, und eine weitere Datei, mit der die meisten Browser nicht unmittelbar umgehen können – eine Datei mit der Erweiterung *.exe*.

Abb. 4–10

Die physische Struktur des zweiten Unterverzeichnisses auf dem Serverrechner

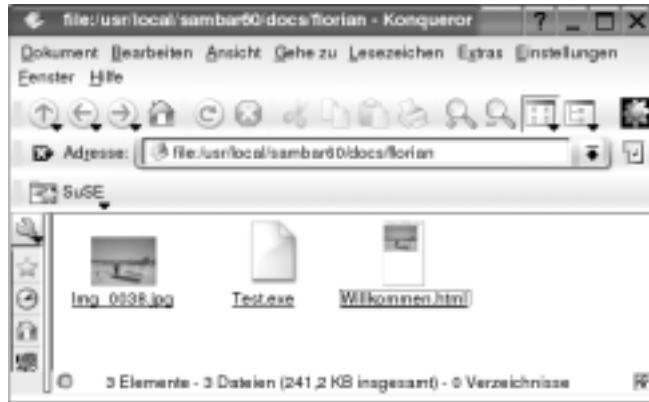
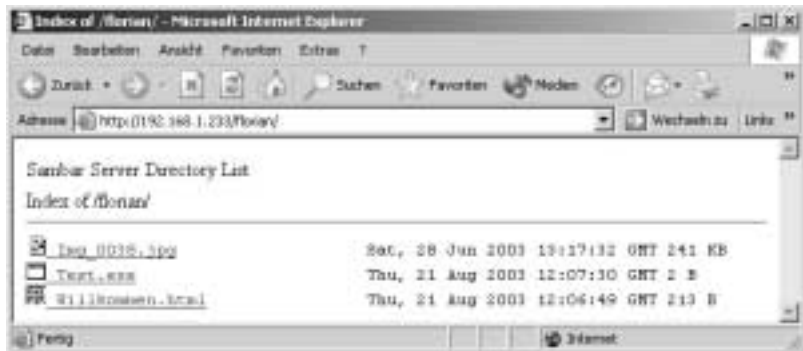


Abb. 4–11

Über die IP-Adresse und den Namen des Unterverzeichnisses bekommen Sie wieder die physikalische Struktur auf dem Serverrechner ab dem Dokumentenverzeichnis angezeigt



Rufen Sie wieder die Webseite in einem Browser auf. Sie bekommen sie samt der integrierten Grafik, die sich in der Webserverstruktur auf der gleichen Ebene befindet, angezeigt (siehe Abb. 4–12).

Unbekannte Dateitypen

Da Sie ja beliebige Dateien über einen Webserver zum Download bereitstellen können, kommt es natürlich auch vor, dass ein Browser nicht direkt mit einem geladenen Dateityp umgehen kann. In diesem Fall bietet er in der Regel die Möglichkeit an, die Datei auf dem lokalen System zu speichern. Wir haben dafür die Datei *Test.exe* in dem Verzeichnis *florian* angelegt. Wenn Sie diese anklicken oder deren URL eingeben, wird der Browser ein Fenster öffnen, in dem Sie die Spezifika für das Speichern angeben können. Beachten Sie aber, dass der Internet Explorer oftmals so konfiguriert ist, dass er automatisch versucht, eine Datei zu öffnen, statt eine Speicherung anzubieten. Auch bei diversen



Abb. 4-12

Die Webseite im
Unterverzeichnis florian

Dateitypen, bei denen es wenig Sinn hat oder eine automatische Öffnung gar gefährlich ist (etwa `.exe`). Solch leichtsinniges Verhalten sollten Sie dem Browser auf jeden Fall austreiben.

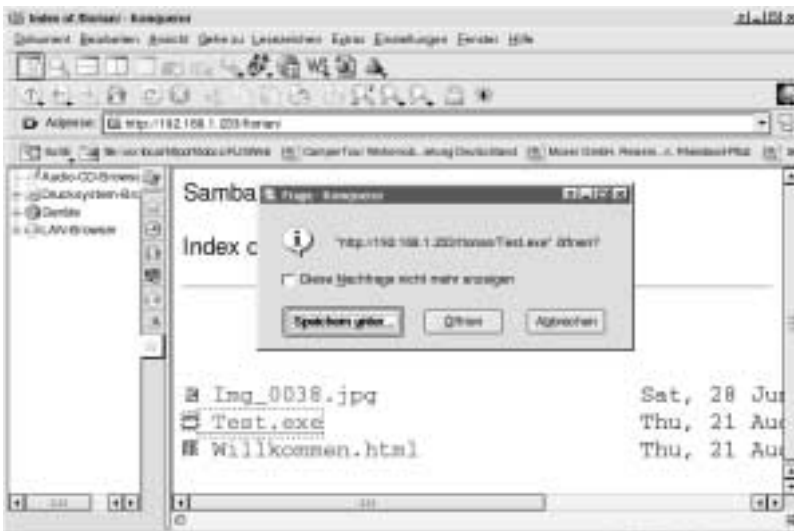


Abb. 4-13

Die Datei eines Typs, der
unbekannt bzw. nicht für
die Darstellung in einem
Webbrowser vorgesehen
ist, sollte man speichern
können.

4.3.8 Sambar absichern

Sicherheit von Sambar

Die Grundkonfigurationen von Sambar lassen den Webserver in weiten Bereichen offen wie ein Scheunentor. Zwar sollten die Beschränkungen der Zugriffe per HTTP auf das *docs*-Verzeichnis greifen und ein Administrator erst einmal nur vom lokalen Rechner aus einloggen können. Für den Betrieb in kleinen Intranets ist die Grundeinstellung sicher ausreichend. Jedoch gibt es diverse Sicherheitsprobleme, die vor allem dann kritisch werden, wenn Sie Ihren Sambar-Rechner im Internet zugänglich machen oder Sambar als Gateway zum Internet verwenden. In der Rubrik *Sicherheit* des Webinterfaces zur Administration können Sie Sambar in vielerlei Hinsicht absichern. Das Thema ist jedoch recht komplex und umfangreich, so dass wir – trotz der Länge dieses Abschnitts – nur einen Einstieg behandeln und vor allem in Kapitel 12 wieder darauf zurückkommen werden.

Benutzer und Gruppen

In Sambar können Sie zunächst einzelne Benutzer anlegen, die Sie darüber hinaus Gruppen zuordnen können. Diese Benutzer sind hauptsächlich in Hinsicht auf FTP (siehe Kapitel 5), auf eingeschränkten HTTP-Zugriff und auf einige weitere Konstellationen von Bedeutung. In gewissen Situationen wird Sambar also von einem Besucher eine Legitimation fordern; dies betrifft z.B. den Upload von Dateien, Zugriff auf bestimmte Verzeichnisse oder Dateien etc.

In der Rubrik *Sicherheit* finden Sie die Links *Benutzer* und *Gruppe* bzw. deren englisches Äquivalent.

Benutzer

Wenn Sie *Benutzer* anklicken, sehen Sie von vornherein drei Gruppen mit schon angelegten Benutzern:

- Die Gruppe *root* mit dem Administrator *admin*
- Die Gruppe *user* mit dem Benutzer *billy-bob*
- Die Gruppe *others* mit dem Benutzer *anonymous*

user

Der Defaultuser der Gruppe *user* hat erst einmal nur den Charakter eines Beispielnutzers. Dessen Profil können Sie für eigene Benutzer weitgehend übernehmen. Falls Sie echte Benutzer anlegen wollen, ist es sicher fraglich, ob einer davon den Spitznamen *billy-bob* haben möchte :-).

others

Der Nutzer der Gruppe *others* ist nicht ganz so zufällig gewählt. In Hinsicht auf FTP und das so genannte *Anonymous-FTP* haben sowohl die Wahl des Namens als auch dessen Rechte eine Bedeutung. Wir kommen auf diesen Gesichtspunkt im nächsten Kapitel zurück.

Der Nutzer der Gruppe *root* ist *admin*. Dieser ist uns ja schon am Anfang begegnet, denn *root* hat in der Grundeinstellung von Sambar die Administrationsrechte per Webinterface.

Wenn Sie nun das Symbol des Mülleimers ganz links neben einem Nutzernamen anklicken, wird der Nutzer gelöscht. Den *admin* können Sie allerdings nicht löschen.

Wenn Sie das *i*-Symbol oder den Namen anklicken, können Sie die Einstellungen für einen Nutzer sehen und verändern. Das sollten wir für *admin* auf jeden Fall durchspielen. Da er in der Grundeinstellung kein Passwort hat, liegt hier nämlich ein nicht tolerierbares Sicherheitsrisiko.

root

Nutzer löschen

Update User



Abb. 4-14

Anzeige und
Manipulation von
Benutzerdaten

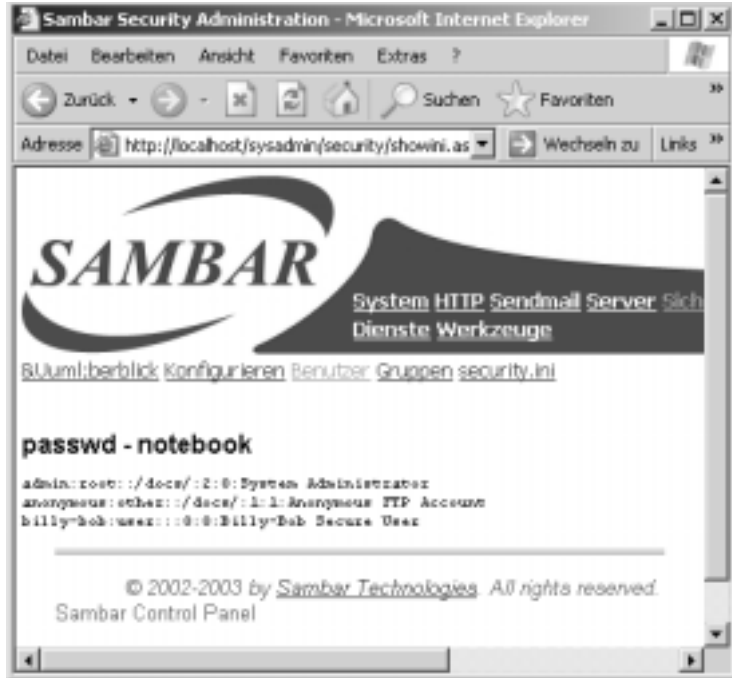
Sie sehen, dass Sie in der Folgeseite ein Passwort eingeben können. Vergeben Sie auf jeden Fall für den Administrator ein – sicheres – Passwort und drücken Sie *Update User*. Sie brauchen den Server in diesem Fall nicht neu zu starten. Die Änderungen wirken sich bei laufendem Betrieb aus. Die restlichen Angaben sind im Wesentlichen für FTP interessant und sollen in Kapitel 5 zur Sprache kommen.

Wenn Sie zurück zur allgemeinen Verwaltung der Benutzer gehen und auf der linken Seite oben die Rubrik *User Management* anschauen, sehen Sie dort vier Hyperlinks. *Login Performance* soll uns hier nicht weiter interessieren. Mit dem Link *passwd* erhalten Sie den Inhalt der Passwortdatei von Sambar.

User Management

Abb. 4-15

Der Inhalt der Datei
passwd



Über *Create User* können Sie einen neuen Benutzer anlegen. Das Webinterface sieht im Wesentlichen genau so aus, wie Sie es von der Manipulation eines bestehenden Benutzers kennen. Sie können (und müssen) nur zusätzlich den Namen für den User vergeben.

Der Link *Update Password* führt zu einer Beispielseite, die zeigt, wie legitimierte Benutzer im System ihr eigenes Passwort verändern können. Die referenzierte Webseite befindet sich im *sysadmin*-Ordner und sollte für eine individuelle Anwendung in das *docs*-Verzeichnis kopiert werden. Wenn Sie sich aber auf der Einstiegsseite von Sambar etwas umschauen, werden Sie auf der linken Seite unter *Pro Server Features* einen Link *User Desktop* finden (siehe Abb. 4-16).

Über diesen Link können sich legitimierte Benutzer zu einem Bereich anmelden, in dem sie gewisse Dinge tun dürfen. Unter anderem können Benutzer dort ihr Passwort ändern (siehe Abb. 4-17).

Gruppen

Gruppen haben für uns nicht die Bedeutung wie Benutzer, denn Sambar ordnet darunter im Wesentlichen nur Benutzer an. Allerdings können bei der Zugriffssteuerung Gruppen spezifiziert werden. Das erleichtert den Vorgang, da man nur wenige Angaben braucht, um mehrere Nutzer zu reglementieren. Das Anlegen einer Gruppe ist trivial. Sie geben einfach einen Namen an. Die Gruppe taucht danach unmittelbar bei der potenziellen Einordnung der einzelnen User im Listenfeld auf.



Abb. 4-16
Anmeldung zum User
Desktop

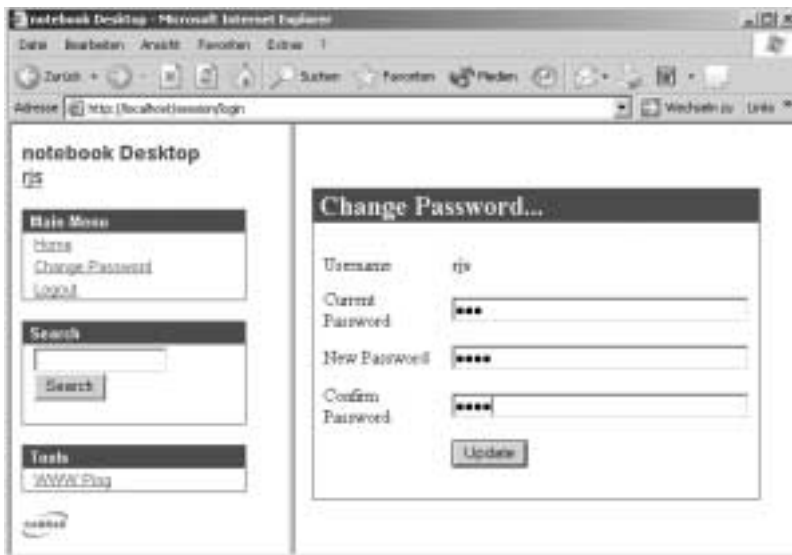


Abb. 4-17
Der User Desktop mit der
Änderung des Passwortes

Zugriffsregeln

Teils abhängig von den legitimierten Usern im System, teils unabhängig davon kann man in Sambar drei Sicherheitsmodelle für die Regulierung von Zugriffen feststellen:

Zugriffsregeln

- URL-basierte Zugriffskontrolle – über die Datei *security.ini*
- Dateibasierte Zugriffskontrolle – über die Datei(en) *.htaccess*
- Hostbasierte Zugriffskontrolle über die Dateien *security.ini* und/oder *.htaccess*

Sambar implementiert Zugriffskontrollmechanismen mit zwei Stufen der Authentifizierungsinformation. Die *security.ini*-Datei befindet sich im *config*-Verzeichnis und ist eine globale Zugriffskonfigurationsdatei. Die Datei *.htaccess* ist eine optionale Kontrolldatei für jedes Verzeichnis innerhalb der Sambar-Verzeichnisstruktur (sie kann also mehrfach vorkommen). Sie kann viel genauer die Zugriffe auf individuelle Verzeichnisse organisieren, als es in einer globalen Einstellung möglich ist.

Beide Verfahren können gemeinsam verwendet werden, aber das sollte mit Vorsicht geschehen, um keine Widersprüche zu erzeugen. Mit den *[restrict]*-Einstellungen in der Datei *security.ini* werden übrigens ausschließlich HTTP-Anfragen reglementiert, wohingegen die *.htaccess*-Restriktionen für sämtliche Arten des Serverzugriffs gelten.

Sämtliche Sicherheitseinstellungen, die Sie für Sambar treffen, sind unabhängig von den Sicherheitseinstellungen des Betriebssystems. Mit anderen Worten – ein Zugriff auf Sambar wird nicht vom Betriebssystem des Hosts kontrolliert.

URL-basierte
Zugriffskontrolle

Die URL-basierte Zugriffskontrolle schränkt Zugriffe auf den Server aufgrund der angefragten URL ein. Die Einstellungen in der Datei *security.ini* werden jedoch nur beim Start von Sambar ausgewertet. Die Folge ist, dass jede Änderung dort einen Neustart des Servers erzwingt, damit sie wirksam wird!

Dateibasierte
Zugriffskontrolle

Die dateibasierte Zugriffskontrolle erlaubt eine vielfältige Gestaltung von Zugriffsmechanismen. Das sind zum Beispiel die Angaben von Sicherheitsgebieten, hostbasierter Zugriff, User- und Gruppenzugriffskontrolllisten etc. Diese Sicherheitseinstellungen werden vom Server bei jedem Zugriff neu eingelesen. Soweit es möglich ist, sind die Syntax und die Regeln zur Spezifizierung der Zugriffe kompatibel mit dem Apache-Server, aber nicht alle der Apache-Möglichkeiten sind unter Sambar auch implementiert. Dennoch ist bereits das Regelwerk von Sambar sehr umfangreich.

Hostbasierte
Zugriffskontrolle

Die hostbasierte Zugriffskontrolle erlaubt verzeichnisbezogene Restriktionen, die auf der IP-Adresse des anfragenden Hosts beruhen. Dazu gibt es beispielsweise eine DENY-Liste mit auszuschließenden Adressen. Wenn etwa eine IP-Adresse in der »deny from«-Liste in der *.htaccess*-Datei enthalten ist, wird bei einem Zugriff von dieser Quelle auf Sambar eine Fehlermeldung »401 (Not Authorized)« angezeigt.

Die konkrete Absicherung

Die Modifizierung der Sicherheitsdateien von Sambar per Klartexteditor ist nicht unbedingt trivial, aber im Webinterface von Sambar können Sie relativ bequem die wichtigsten Einstellungen vornehmen. Beachten Sie, dass Eingaben in den Formularen in der Regel als Klartext zum Server geschickt werden und damit von einem potenziellen Angreifer zu lesen sind, wenn er das Netzwerk ausspionieren kann und Sie von einem entfernten Rechner zugreifen (wir kommen in den Kapiteln 11 und 12 darauf zurück).

Wenn Sie im Administrationsbereich den Link *Sicherheit* und dann *Konfigurieren* anklicken, gelangen Sie an einige zentrale Stellen zum Einrichten der Sicherheitseinstellungen von Sambar. Am Anfang sind die Administrator-IP und User-IP sicher am interessantesten. Damit können Sie aufgrund der IP-Adressen des zugreifenden Hosts den Login auf Sambar reglementieren. Nur von den angegebenen IP-Adressen aus können sich der root bzw. ein User einloggen, wenn dies gefordert wird (etwa zu Administration oder dem User Desktop).

Sicherheit – Konfigurieren



Abb. 4-18

Festlegung, von welchen IP-Adressen ein Administrator sich einloggen und ein normaler Besucher zugreifen kann

gleiche Abbildung wie 4-4?

Unter *Benutzer* und *Gruppen* kommen Sie zur Benutzer- bzw. Gruppenverwaltung, die wir gerade betrachtet haben.

Der Link *security.ini* führt zur direkten Manipulation der Datei *security.ini* und damit zur Festlegung diverser globaler Regeln, nach denen Sambar Zugriffe gewähren oder verweigern kann. Hier wird

Der Link *security.ini*

angegeben, welche User und Gruppen was dürfen, welche IP-Adressen welchen Zugriff haben etc. Dies ist mit Abstand die wichtigste Stelle zum globalen Absichern von Sambar. Schauen wir uns die wichtigsten Details in Bezug auf den Webserver an⁷. Als Ergänzung möchte ich aber auch auf die Onlinehilfe von Sambar und die Beschreibungen im Webinterface verweisen.

Um Beispiele beschreiben zu können, gilt in der Folge, dass der Sambar-Server meist unter der Adresse `192.168.1.233` zu erreichen ist. Alternativ wird gelegentlich ein zweiter Sambar-Server unter `192.168.1.31` verwendet.

Security Redirects In der Kategorie *Security Redirects* geben Sie in der linken Spalte einzelne Unterverzeichnisse oder auch Dateien auf dem Sambar-Server an (URI), welche an einen beliebigen URL umgeleitet werden sollen.

Wenn Sie etwa die Datei `/nix.html` in die linke Spalte notieren und in der zugehörigen rechten Spalte `http://192.168.1.31` angeben, wird die Benutzereingabe `http://192.168.1.233/nix.html` (die IP-Adresse des Sambar-Servers) auf `http://192.168.1.31` umgeleitet.

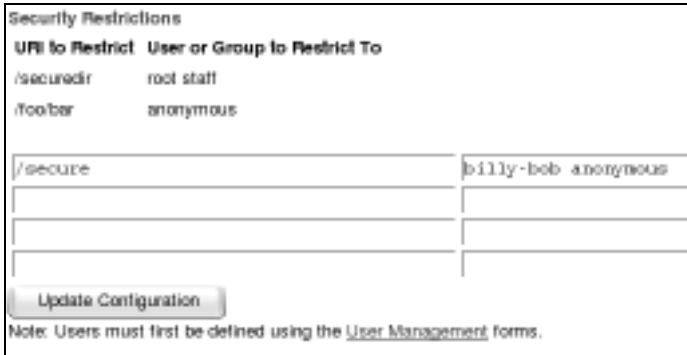
Abb. 4-19

Umleitung einzelner URIs
an andere Adressen

Security Redirects	
URI to Redirect	URL to Redirect To
/microsoft	http://www.microsoft.com/
/docs	http://techpub.sambar.com/
/homepage	http://www.sambar.com
/nix.html	http://192.168.1.31
/notebook	http://192.168.1.31
<input type="button" value="Update Configuration"/>	

Security Restrictions In der Kategorie *Security Restrictions* geben Sie in der linken Spalte ausgewählte URIs an, zu denen bestimmte User oder Gruppen (rechte Spalte) keinen Zugang erhalten sollen. Mehrere User oder Gruppen können für einen URI, durch Leerzeichen getrennt, notiert werden. Diese Benutzer oder Gruppen müssen vorher im *User Management* eingerichtet worden sein.

7. Sie finden hier auch Sicherheitseinstellungen zu den anderen Features von Sambar wie FTP oder E-Mail.

**Abb. 4-20**

Angabe von Strukturen, auf die bestimmte User und Gruppen keinen Zugang haben sollen

Wenn ein Besucher den entsprechenden URL eingibt, wird die gewünschte Seite bzw. die Defaultseite des Verzeichnisses nicht einfach angezeigt, sondern ein Anmeldedialog zwischengeschaltet, in dem der Besucher seinen Benutzernamen und das zugehörige Kennwort eingeben muss.

**Abb. 4-21**

Nur mit einer Anmeldung erreichbar

Ist er berechtigt (also nicht in der Liste für den URI ausgeschlossen), wird er weitergeleitet. Andernfalls erhält er eine entsprechende Fehlermeldung (Servercode 401).

**Abb. 4-22**

Zugang verweigert – Servercode 401

Security IP Restrictions

In der Kategorie *Security IP Restrictions* geben Sie URIs an, auf die nur von den rechts notierten IP-Adressen zugegriffen werden darf. Beim Zugriff von einer anderen Adresse erhält ein Besucher den Fehlercode 403.

Abb. 4-23

Einschränkung der IP-Adressen, von denen zugegriffen werden darf

Security Deny

In der Kategorie *Security Deny* gibt man IP-Adressen von Clients an, bei deren Zugriff Sambar auf den rechts notierten URL umleiten soll.

Abb. 4-24

Umleitung bei den angegebenen IP-Adressen – der Zugriff von 192.168.1.50 wird unmittelbar auf 192.168.1.233 umgeleitet.

Security HTTP Accept

Sehr wichtig ist die Kategorie *Security HTTP Accept*. Dort geben Sie an, von welchen Adressen aus Sambar Usern HTTP-Zugriffe gestatten soll. Doch Vorsicht – hier treffen Sie (aber nur scheinbar) auf eine Syntax, wie sie in Sambar an anderer Stelle (*.htaccess*) und auch unter Apache eingesetzt wird. Mit Klartextnotationen, die mit *Deny ...* und *Allow ...* beginnen, verbieten oder gestatten Sie in *.htaccess*-Dateien und auch unter Apache bestimmten Clients gezielt eine spezifische Nutzung von Zugriffen. Die Syntax erlaubt dort eine feine Spezifizierung, welche Clients von einer bestimmten Adresse oder einem Adressbereich Zugriff auf bestimmte Strukturen auf dem Server haben. Wenn Sie das Webinterface von Sambar anschauen, suggeriert die vorbelegte Beschreibung in der rechten Spalte, Sie könnten auch hier solche Klartextangaben machen, die eine Wirkung haben (siehe Abb. 4-25).

Die Spalte auf der rechten Seite ist jedoch ausschließlich eine *Beschreibung* dessen, was Sie auf der linken Seite über IP-Adressen bzw. IP-Adressbereiche spezifizieren. Ob da **Allow all clients* oder ***

IP Addresses to Accept	Description
*	Allow all clients
140.175*	Allow only corporate users

Abb. 4-25

IP-Adressen Zugriff gestatten oder verbieten – * gibt alle Zugriffsadressen frei.

Die Antwort ist 42 steht, ist vollkommen belanglos. Beide Angaben geben sämtliche IP-Adressen als Zugriffsbasis frei. Nur die linke Spalte zählt. Doch Vorsicht! Sambar fordert, dass *irgendetwas* bei der Beschreibung steht! Das Feld in der rechten Spalte darf nicht leer bleiben, wenn in der linken Spalte ein Wert steht. Sambar ignoriert die Angabe ansonsten.

IP Addresses to Accept	Description
*	Allow all clients
140.175*	Allow only corporate users
127.0.0.1	Localhost
192.168.1.233	Ein lokaler Rechner
192.168.1.25	Noch ein lokaler Rechner
192.168.1.58	Beide ist das Lokal voll

Abb. 4-26

Die Beschreibung muss sein, ist aber ohne Auswirkungen.

Das sind jetzt wahrscheinlich die wichtigsten Sicherheitseinstellungen in der Datei *security.ini* für dem Betrieb von Sambar als Webserver gewesen, wenn man das Thema nicht bis ins Detail verfolgen will. Die weiteren Einstellungsmöglichkeiten der Datei *security.ini* über das Webinterface müssen wir aus Platzgründen auslassen bzw. wir werden sie bei den jeweiligen Themen (FTP, Mail, Proxy) noch einmal berühren.

.htaccess

Wie schon erwähnt, können Sie unter Sambar für jedes Verzeichnis in der Sambar-Struktur individuelle Zugriffsrechte vergeben. Sie legen dazu in jedem Verzeichnis eine Klartextdatei mit Namen *.htaccess* an. Die notwendigen Details zur Syntax sprengen aber unseren Rahmen und werden nur skizziert. Nur soweit: Wir haben in dem letzten Konfigurationspunkt HTTP-Zugriffe global auf bestimmte IP-Adressen

Zugriffsrechte auf Verzeichnisse

bzw. IP-Bereiche festgelegt. Wenn Sie die verzeichnisspezifischen Zugriffsregeln über jeweils dort eingefügte *.htaccess*-Dateien anwenden und damit viel feiner Zugriffsregeln festlegen wollen, finden Sie dort vielfältig zu strukturierende Zugriffseinstellungen. Diese verwenden weitgehend die gleiche Syntax, wie sie auch unter Apache zum Einsatz kommt. Dort finden Sie dann wirklich Klartextformulierungen wie *allow from all*, um den Zugriff auf einzelne Verzeichnisse von allen Stellen aus zu gewähren, *allow from 192.164.170.128 193.160.** zur Beschränkung auf bestimmte Adressen und Bereiche oder auch *deny from all* zum vollständigen Ausschluss eines Zugriffs auf ein Verzeichnis. Das ist nicht so widersinnig, wie es zuerst scheint, denn man kann mit ergänzenden Regeln den Zugriff wieder für einzelne IP-Adressen oder IP-Bereiche freigeben. Die recht komplizierte Syntax gestattet auch über Angaben wie *order deny,allow* festzulegen, ob die *deny*-Direktiven vor den *allow*-Direktiven ausgewertet werden oder umgekehrt (*order allow,deny*).

4.3.9 Virtuelle Hosts

Mehrere Server auf einer
Machine

Ein interessantes Feature von Sambar sind so genannte virtuelle Hosts. Damit kann man auf einer Maschine mehrere Server betreiben, die unter der gleichen IP-Adresse, aber unterschiedlichen DNS-Namen erreichbar sind und unterschiedliche Inhalte anbieten. Ihr Internetprovider tut genau dies wahrscheinlich mit den vielen Webservern für seine Kunden. Ebenso ist es möglich, zwei verschiedene Hosts unter der gleichen IP-Adresse anzubieten, wobei die direkte Angabe der IP-Adresse zum einen und die Angabe von einem DNS-Namen zum anderen Host führt. Die Zuordnung hängt also explizit davon ab, was ein Anwender eingibt. Gibt ein Anwender die IP-Adresse an, wird er beispielsweise zum Host A geleitet. Gibt er den DNS-Namen an, gelangt er zum Host B. Obwohl auch Host B die gleiche IP-Adresse hat.

DNS-Namen auflösen

Der Knackpunkt steckt bereits in dem letzten Satz – um mit Sambar mehrere Hosts auf einem Rechner (sinnvoll) laufen zu lassen, brauchen Sie einen DNS-Server, der verschiedene DNS-Namen auf eine IP-Adresse auflöst, oder aber eine andere Zuordnung zwischen IP-Adresse und einem symbolischen Namen. Da Sambar gut mit der *hosts*-Datei zusammenarbeitet, tragen Sie beispielsweise auf jedem Client im Intranet darin die gewünschten DNS-Namen und unter Umständen jeweils die gleiche IP-Adresse ein und legen dann bei den DNS-Einstellungen aller beteiligten Clients fest, dass Sie die *hosts*-Datei verwenden sollen (siehe dazu Kapitel 2). Aber wie gesagt – Sie können auch einfach festlegen, dass die direkte Eingabe einer IP-Num-

mer zu einer anderen Adresse wie die Eingabe eines DNS-Namens führt (obwohl der gleiche Host dahinter liegt). Das Verfahren ist der Umleitung von URLs ähnlich, aber nicht ganz identisch.

So könnte eine sehr einfache *hosts*-Datei aussehen:

```
127.0.0.1    localhost
192.168.1.233 vhost.rjs.local
```

Hier würde der DNS-Name *vhost.rjs.local* auf die IP-Adresse *192.168.1.233* abgebildet. Alternativ kann natürlich auch Ihr DNS-Server den oder die DNS-Namen auf die gleiche IP-Adresse auflösen (siehe dazu Kapitel 11).

Einen virtuellen Host anlegen

Wir richten nun konkret einen virtuellen Host ein. Beachten Sie, dass dies natürlich nur Beispielangaben sind. Legen Sie zuerst ein Arbeitsverzeichnis für den virtuellen Host an. Etwa *vhosts/* (unter Linux) innerhalb des freigegebenen Verzeichnisses von Sambar (also */docs*). Legen Sie darin eine Datei *index.html* an. Gehen Sie nun zur Sambar-Adminstrationsseite und loggen Sie sich ein. Unter *HTTP* finden Sie den Hyperlink *virtuelle Hosts*. Wählen Sie dort *Create New Virtual Host*. Geben Sie in der folgenden Seite diese Daten ein:

- Name: *vhost.rjs.local*
- Documents Directory *vhosts/*

Die restlichen Einstellungen können Sie erst einmal wie vorgeschlagen beibehalten.

Abb. 4-27

Anlegen eines virtuellen Hosts

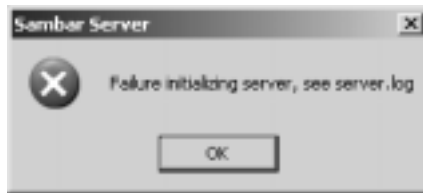
Klicken Sie dann auf die Schaltfläche *Create Virtual Host* und starten Sie den Server neu. In Ihrem Browser sollte nun unter *vhost.rjs.local* die Datei *index.html* aus dem Verzeichnis *e:\vhosts* angezeigt werden, wenn Sie von einem Rechner aus zugreifen, der den DNS-Namen auflösen kann.

4.3.10 Troubleshooting

Probleme beim Start von Sambar

Sambar ist im Grunde ein vollkommen problemloser und pflegeleichter Webserver. Dennoch kann es gerade beim Start gelegentlich zu Problemen kommen. Unter Windows äußert sich das dann meist mit einem Warnfenster.

Abb. 4-28
Sambar konnte nicht starten.



Der Hinweis in dem Warnfenster sagt auch genau das, was Sie in diesem Fall immer machen sollten. Schauen Sie sich die Datei *server.log* im *log*-Verzeichnis mit einem Editor an. Eine Logdatei könnte so aussehen:

Listing 4-1
Auszug aus einer Log-Datei von Sambar

```
[2003-08-22 09:30:52] INFO: Common Library loading messages for
  locale 'de_de' (D:/sambar60/messages/de_DE/messages.ini)
[2003-08-22 09:30:52] INFO: Common Library loading messages for
  locale 'en_us' (D:/sambar60/messages/en_US/messages.ini)
[2003-08-22 09:30:52] INFO: Common Library loading messages for
  locale 'po_br' (D:/sambar60/messages/po_BR/messages.ini)
[2003-08-22 09:30:52] INFO: Common library initialized...
[2003-08-22 09:30:53] INFO: Network IP Address is 192.168.1.31
[2003-08-22 09:30:53] ERROR: Unable to bind to port 80; the port
  is in use by another server.
[2003-08-22 09:30:53] ERROR: Failure starting the HTTP
  listener...
[2003-08-22 09:30:53] INFO: CRON Scheduler signaled to
  shutdown...
[2003-08-22 09:30:53] INFO: Waiting for 0 active threads to
  terminate...
[2003-08-22 09:30:53] INFO: Closing with NO OPEN THREADS...
[2003-08-22 09:30:53] INFO: Closing with 0 open connections...
[2003-08-22 09:30:54] INFO: Server library exited...
[2003-08-22 09:30:54] INFO: Common library exited...
```

Den kritischen Fehler in diesem Fall zeigt die im Listing fett gekennzeichnete Zeile an. Der Port 80 wird bereits von einem anderen Programm verwendet. In diesem Fall wäre es eine Lösung, den Port von Sambar zu ändern. Nur dummerweise kommt man nicht an das Webinterface zur Konfiguration, wenn Sambar nicht startet. Ihnen bleibt nur der Weg, die Änderung in der betroffenen Konfigurationsdatei (hier *config.ini*) selbst vorzunehmen. Suchen Sie in unserem Fall nach *HTTP Port* und ändern Sie dessen Wert.

Entsprechend verfahren Sie bei allen anderen Fehlern, wobei die meisten Startfehler von belegten Ports herrühren dürften. Die Datei *server.log* zeigt an, was los ist und in einer der Ini-Dateien (meist *config.ini*) suchen Sie die zugehörige Stelle und ändern sie ab (unter Umständen sogar nur auf einen temporären Wert, um den Server erst einmal starten zu können).

4.3.11 Der Webserver im Internet

Wenn Sie Sambar auf einem Rechner laufen lassen, der Kontakt zum Internet hat, kann man auch aus dem Internet auf Sambar zugreifen (Zugriffsbeschränkungen per Firewall, Filterung der IP-Adressen auf dem Server oder Ähnliches außer Acht gelassen). Es gibt keinen qualitativen Unterschied, ob ein Besucher aus dem Intranet oder dem Internet zugreift. Haben Sie gar eine statische IP-Adresse und auch einen offiziellen DNS-Namen, ist Sambar somit ein voll funktionstüchtiger Webserver, wie ihn ein Webpace-Provider auch betreibt. Nur wäre es vermessen, wenn Sie jetzt glauben, Sie könnten Sambar mit den geringen Einstellungen, die wir vorgenommen haben, sicher und stabil für den Interneteinsatz betreiben. Es ist ein erhebliches Sicherheitsrisiko mit dem Betrieb eines Webserver mit freiem Zugriff verbunden. Das fängt bei Sicherheitslücken in der Software an (obwohl Sambar ein sehr sicherer Server ist) und endet – vor allem – in der ungenügenden Absicherung bzw. Konfiguration für den rauen Internet-Alltag.

Und ich warne noch einmal: Wir haben – trotz des recht umfangreichen Sicherheitsabschnitts – Sambar nur so aufgesetzt, um ihn im kleinen Intranet zu betreiben oder damit zu experimentieren. Beachten Sie vor einer öffentlichen Bereitstellung auf jeden Fall Kapitel 11 und 12.

Achtung!

4.4 Apache

Apache ist nicht umsonst der mit Abstand am weitesten verbreitete Webserver im Internet. Er spielt sogar in einer anderen Liga als Sam-

bar, obgleich dieser mittlerweile an Funktionalität erheblich zugelegt hat. Die Hersteller von Sambar sagen auf ihrer Website selbst, dass Sambar nicht an die Features und Konfigurationsmöglichkeiten von Apache heranreicht (vor allem in Hinsicht auf Sicherheitseinstellungen, Performance-Optimierung und unterstützende Module). Aber dieses Mehr an Leistung macht insbesondere die Konfiguration von Apache nicht ganz unproblematisch, weshalb sich Sambar meines Erachtens als Einstiegsserver besser eignet (dort steht explizit eine einfache Installation, Einrichtung und Konfiguration im Vordergrund). Dennoch wollen wir Apache hier zumindest aufsetzen.

Bezug von Apache

Laden können Sie Apache von der Website <http://www.apache.org>. Wir schauen uns hier die Version 2 an. Die Linux-Version wird als gepacktes *.tar.gz*-Archiv vertrieben, worin sich die Quelltextdateien befinden. Die Windows-Version wird in *.msi*-Dateien ausgeliefert.

4.4.1 Installieren und starten

Die Installation von Apache ist unkompliziert. Zwar muss man unter Linux die Quelltextdateien zur Installation kompilieren, aber das sollte keinen Linux-Anwender schrecken (und es wird hier beschrieben).

Unter Linux

Zuerst extrahieren Sie das gepackte Archiv – am besten in ein temporäres Verzeichnis wie */tmp*. Wechseln Sie in das bei der Extraktion dort erzeugte Verzeichnis. Anschließend führen Sie den typischen Linux-»Dreiklang« *./configure – make – make install* zur Kompilierung und Installation aus.

Um etwa in das Verzeichnis */usr/local/apache2* zu installieren, geben Sie nacheinander Folgendes ein:

```
./configure --prefix=/usr/local/apache2
make
make install
```

Zwischen den einzelnen Schritten wird in der Shell eine große Anzahl von Meldungen ausgegeben, die Sie weitgehend ignorieren können (außer, es tritt ein Fehler auf). Sie müssen nur den Abschluss der einzelnen Schritte abwarten.

Anschließend starten Sie Apache wie folgt (siehe Abb. 4–29):

```
/usr/local/apache2/apachectl start
```

Unter Windows

Unter Windows rufen Sie einfach die *.msi*-Datei (hier *apache_2.0.43-win32-x86-no_ssl.msi*) auf. Der folgende Installationsprozess ist weitgehend selbsterklärend.



Abb. 4-29

Apache ist gestartet

Wenn Apache vollständig installiert ist, startet das Programm automatisch. Zudem haben Sie Einträge im Startmenü zum Starten und Stoppen des Servers zur Verfügung. Unter Windows finden Sie in der Taskleiste auch ein Icon, über das Sie einen Monitor aktivieren können.

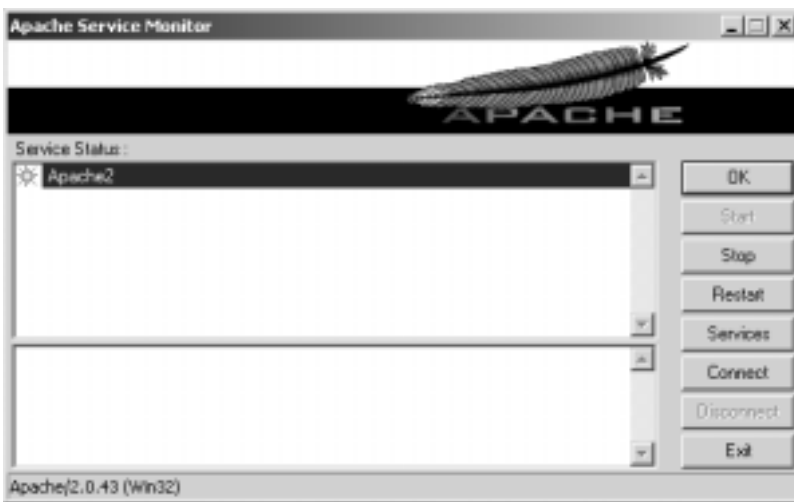


Abb. 4-30

Der Apache-Monitor zum Starten und Stoppen des Servers unter Windows

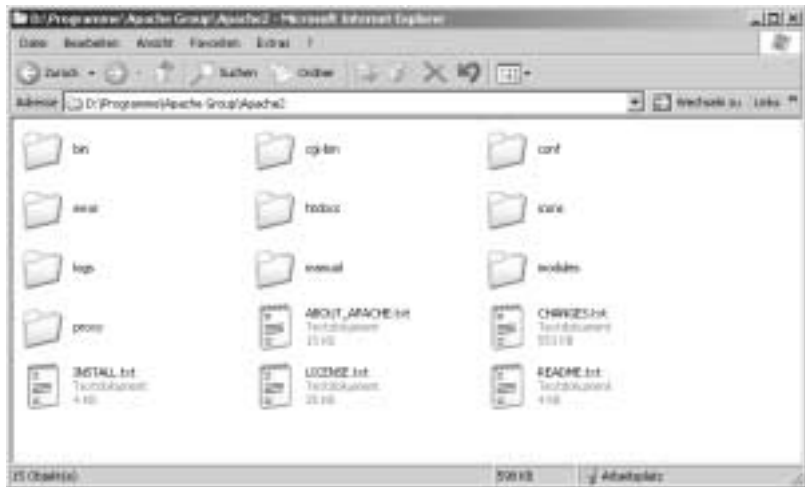
4.4.2 Die Verzeichnisstruktur von Apache

Die Verzeichnisstruktur von Apache ist der von Sambar nicht unähnlich. Zwar unterscheiden sich auch hier die Verzeichnisse, die unter Linux und Windows angelegt werden, aber wesentliche Strukturen sind in beiden Welten aus der Erfahrung mit Sambar wieder zu erkennen.

Abb. 4-31
Die Verzeichnisse
unter Linux



Abb. 4-32
Die Verzeichnisse
unter Windows



- In *bin* befinden sich die Programmdateien, Skripte und Laufzeitbibliotheken (je nach Version).
- Das Verzeichnis *cgi-bin* kennen wir ebenfalls von Sambar.
- Das Verzeichnis *logs* ebenso.
- In *conf* befinden sich die Konfigurationsdateien von Apache. Das entspricht *config* unter Sambar.
- Das Verzeichnis *htdocs* ist das Gegenstück zu *docs*. Hier werden die Dateien und Unterverzeichnisse abgelegt, die über den Server öffentlich gemacht werden sollen.
- Die weiteren Verzeichnisse wie *error* (Fehlermeldungen) oder *icons* (Icons) sind relativ selbsterklärend und müssen normalerweise nicht angepackt werden.

4.4.3 Apache konfigurieren

Die Konfiguration von Apache läuft im Wesentlichen über die Manipulation der Datei *httpd.conf* im Verzeichnis *conf*. Dies ist eine Klartextdatei, die Sie mit jedem Editor bearbeiten können. Leider besitzt Apache selbst kein direktes Interface zur komfortablen Nutzerführung. Zwar gibt es einige Tools, die bei der Konfiguration helfen, aber im Grunde kommen Sie nicht umhin, sich mit der Datei selbst zu beschäftigen. Wir wollen nur soweit die Syntax erklären, dass Sie selbst ein paar Experimente machen können, und ansonsten auf spezielle Apache-Literatur verweisen.

httpd.conf

Die Datei ist sehr gut dokumentiert (allerdings in Englisch). Wenn ein # am Beginn einer Zeile steht, ist das als Kommentar zu verstehen. Viele Befehlszeilen in der Datei sind auskommentiert und können einfach aktiv gesetzt werden, indem das #-Zeichen am Beginn der Zeile eliminiert wird. Da bereits unzählige Einstellungen optional in der Datei als Kommentar vorgesehen sind, wird die Datei allerdings recht groß und schwer lesbar.

Viele Angaben in der Datei *httpd.conf* werden Sie aufgrund der Ausführungen zu Sambar wieder erkennen. Wir schauen uns nur ein paar als Beispiele an, wobei wir hier die Windows-Variante als Basis verwenden:

- Die Angabe *ServerRoot »D:/Programme/Apache Group/Apache2«* gibt ein Installationsverzeichnis von Apache an.
- Die Angabe *DocumentRoot »D:/Programme/Apache Group/Apache2/htdocs«* ist das öffentlich freigegebene Dokumentenverzeichnis.
- Mit *Listen 80* geben Sie den Port an.
- Über *DirectoryIndex index.html index.html.var* spezifizieren Sie die Defaultseiten.
- Mit *AccessFileName .htaccess* wird der Name der Dateien angegeben, die die individuellen Zugriffsregeln für einzelne Verzeichnisse enthalten (das ist ja aus Sambar bekannt).

4.4.4 Der konkrete Betrieb

Für einfache Experimente genügt es, in das Verzeichnis *htdocs* Dateien zu kopieren und/oder dort Verzeichnisse für Unterstrukturen zu erzeugen. Allerdings ist Apache so nur für wirklich einfache Experimente zu nutzen und sollte nicht ins Internet gebracht werden. Um Apache wirklich produktiv einzusetzen, müssen Sie sich intensiv mit dem Programm auseinandersetzen und kommen wahrscheinlich nicht um weitere Literatur herum. Etwa »Apache Webserver 2.0 – Installation,

Konfiguration, Programmierung« vom Addison-Wesley-Verlag (ISBN: 3827320399) oder »Webserver betreiben – HTTP und Apache: Grundlagen, Konzepte und Lösungen« vom dpunkt-Verlag (ISBN: 3932588002).

4.5 Weitere Webserver

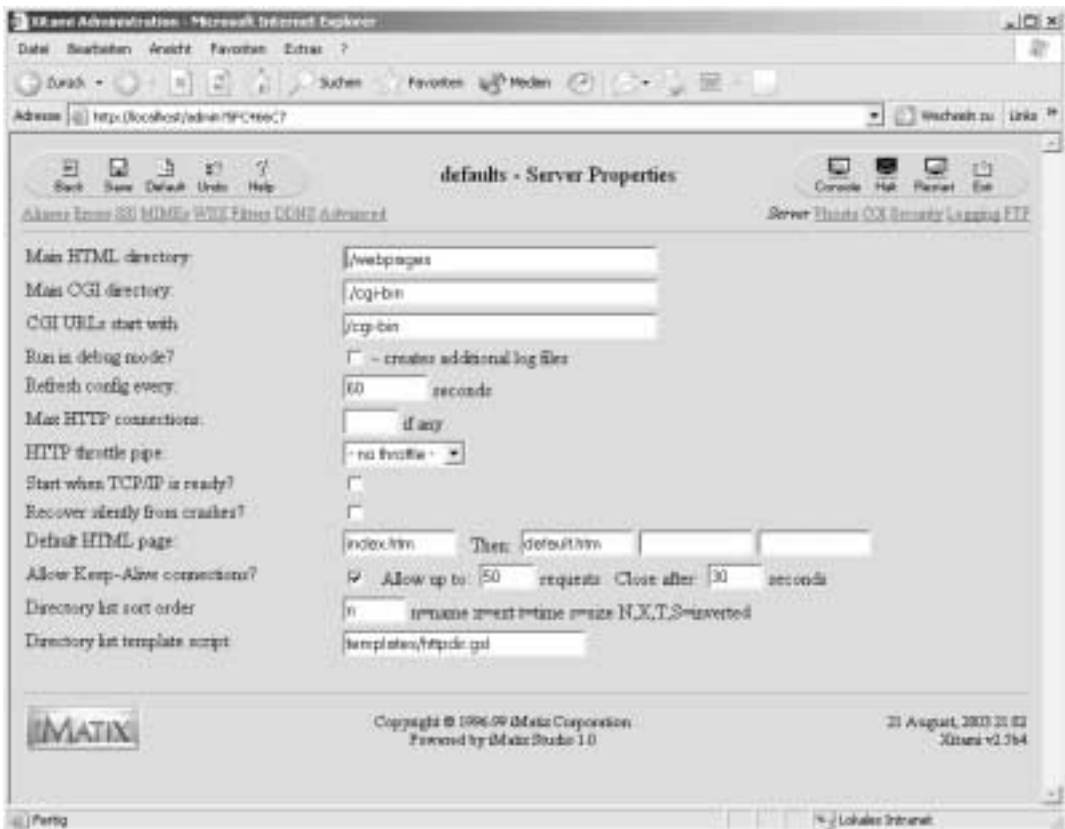
Xitami und PWS

Zu den beiden Hauptwebservern in diesem Buch gibt es diverse Alternativen. Als direkte Alternative zu Sambar unter Windows kann man den Xitami-Webserver sowie den in einigen Windows-Varianten integrierten Webserver PWS (Personal Web Server) von Microsoft sehen. Den Xitami-Webserver können Sie von der Webseite <http://www.imatix.com/html/xitami/index.htm> laden, während der PWS auf Ihrer Windows-Installations-CD oder bei Microsoft (<http://www.microsoft.com>) zu finden ist.

Abb. 4-33

Ein klares und übersichtliches Webinterface zeichnet die Konfiguration des Xitami-Webserver aus

Der Xitami-Webserver lässt sich extrem einfach installieren und bietet ein sehr komfortables Webinterface zu Konfiguration, das dem von Sambar nicht nachsteht.



In der gleichen Liga wie Apache spielt der mit dem PWS verwandte IIS (Internet Information Server) von Microsoft, den es aber nicht für Linux gibt. Beide Microsoft-Webserver verzahnen sich tief ins Betriebssystem bzw. in Microsoft-Technologien. Vorteile sind beispielsweise einfache Zugriffe auf Datenbanken und die Verwendungsmöglichkeit der hauseigenen .NET-Technik. Ebenso bringen die Server sowohl die Konfigurationsmöglichkeit über ein Webinterface als auch eine eigene Konfigurationssoftware mit. Nachteile dieser tief verzahnten Systeme sind erhebliche Anfälligkeiten in Hinblick auf Sicherheit sowie eine auf Dateiebene wenig transparente Datei-, Konfigurations- und Verwaltungsstruktur (keine einfachen Datei- und Klartextstrukturen). Außerdem ist man auf Windows als Server-Betriebssystem beschränkt und hat in Bezug auf Lizenzen bei diversen Konstellationen mit einigen Kosten zu rechnen.

4.6 Weiterführende Themen

- Wollen Sie einen Webserver über das Internet zugänglich machen, sollten Sie sich mit den Konfigurationsmöglichkeiten intensiv beschäftigen, Kapitel 11 und 12 beachten und vor allem Quellen und Dokumentationen dazu besorgen, wie der Server noch weiter abzusichern ist, als wir das getan haben. Aber es kommen auch bürokratische Probleme auf Sie zu. Sie müssen bestimmte Pflichtangaben veröffentlichen, Inhalte und Verlinkungen überwachen, Datenschutzbestimmungen einhalten und ähnliche Späßchen. Das kann schnell ein gefährliches Pflaster und/oder Fass ohne Boden werden.
- Das Anbieten von dynamischen Inhalten per ASP, Perl, PHP, Servlets/JSP sowie Datenbanken ist ein sehr spannendes Thema, zu dem Sambar und Apache die Features aus Sicht des Webserver bieten. Wenn Sie eine Datenbank im Zusammenhang mit Ihrem Webserver betreiben wollen, wäre MySQL eine gute Wahl. Im folgenden Abschnitt mit Quellen und weiteren Informationen finden Sie dazu gute Einstiegsmöglichkeiten.
- Der Umgang mit HTTPS und SSL ist ein lohnendes, aber nicht gerade einfaches Thema. Darüber lassen sich sichere, verschlüsselte Verbindungen zwischen dem Webserver und einem Client aufbauen. Sowohl Sambar als auch Apache unterstützen diese Techniken. In einem kleinen Intranet hat das Thema weniger Bedeutung als bei der Übermittlung von sensiblen Daten über das Internet. In Kapitel 11 und 12 kommen wir auf Verschlüsselung zurück.

- Sambar ist mittlerweile ein vollständiger E-Mail-Server. Dieses Thema wollen wir aber später (Kapitel 6) noch ansprechen und greifen da auch auf andere Programme zurück. Sowohl die Homepage von Sambar als auch die Onlinehilfe von Sambar bieten jedoch gute Informationen zum Einstieg, wenn Sie Sambar als E-Mail-Server aufsetzen wollen.
- Der Ausbau der virtuellen Hosts ist sicher auch ein Thema, dem Sie sich vertiefend widmen können. Auch hier bietet die Onlinehilfe von Sambar weitere Informationen.

4.7 Zusammenfassung

Wir haben uns in diesem Kapitel mit der Installation, Administration und dem Betrieb von Webservern beschäftigt. Dabei haben wir uns im Wesentlichen auf den Betrieb in einem abgesicherten Intranet beschränkt, aber auch wichtige Details zum Absichern eines Webserver berührt. Schwerpunkt der Praxis bildete das Programm Sambar, aber auch Apache wurde verwendet.

4.8 Quellen und weitere Informationen

- HTTP-Definition: <http://www.ietf.org/rfc/rfc2068.txt>
- Die Homepage von Sambar: <http://www.sambar.com/>
- Die Homepage von Microsoft: <http://www.microsoft.com>
- Die Homepage von Apache: <http://www.apache.org>
- Die Homepage von MySQL: <http://www.mysql.com> bzw. (auf Deutsch) <http://www.mysql.de>
- Die Homepage von Imatix: <http://www.imatix.com>
- Das W3C: <http://www.w3c.org>
- German Sambar Zone: <http://home.arcor.de/gwadro/>
- Newsgruppen zum Thema Webserver:
de.comm.infosystems.www.servers und
de.comp.os.unix.linux.misc

4.9 Übungen

- Stellen Sie Informationen über Sambar, Xitami und Apache bereit. Verwenden Sie verschiedene Verzeichnisse und bieten Sie die unterschiedlichsten Dateien zum Download an.
- Experimentieren Sie mit dem Xitami-Webserver und vergleichen Sie ihn mit Sambar.

- Sehen Sie sich die einzelnen Konfigurationsdateien von Sambar mit einem Editor an. Versuchen Sie, die einzelnen Anweisungen zu verstehen, und experimentieren Sie damit. Notieren Sie sich aber vor Änderungen die Originaleinstellungen. Wenn der Server nach einer Veränderung nicht mehr startet, setzen Sie diese einfach zurück.
- Analysieren Sie die Sicherheitsdateien von Sambar. Testen Sie die unterschiedlichen Einstellungen, die Sie entweder per Webinterface oder Klartexteditor setzen. Wenn Sie mehrere Rechner zu einem Intranet verbunden haben, unterbinden Sie Zugriffe von bestimmten IP-Adressen in Ihrem Intranet etc. und testen Sie die Wirkung.
- Schauen Sie sich das Log-Verzeichnis von Sambar, Apache und Xitami an. Öffnen Sie die einzelnen Log-Dateien mit einem Editor. Sie werden bei den Zugriffsprotokollierungen (bei allen drei Servern die Datei *access.log*) etwas der folgenden Art sehen (hier Xitami):

```
127.0.0.1 - - [21/Aug/2003:20:59:17 +0100]
"GET /images/im0096c.gif HTTP/1.1" 200 4763
"http://localhost" "Mozilla/4.0 (compatible; MSIE 6.0;
Windows NT 5.1)"
127.0.0.1 - rjs [21/Aug/2003:21:01:39 +0100]
"GET /admin/$back.gif HTTP/1.1" 200 239
"http://localhost/admin?5FC466C7" ""
...
127.0.0.1 - rjs [21/Aug/2003:21:02:13 +0100]
"POST /admin?5FC466C7 HTTP/1.1" 200 7997
"http://localhost/admin" ""
```

Versuchen Sie, die Dateien zu lesen. Sie sollten sowohl erkennen, wer auf den Server zugegriffen hat, von wo, wann und – je nach Webserver – mit welchem Browser und Betriebssystem. Die Log-Dateien geben interessante Aufschlüsse darüber, welche Daten zwischen Server und Client ausgetauscht werden und was ein Webserverbetreiber so alles über seine Besucher protokollieren kann. Mehr dazu finden Sie bei der Behandlung der tiefer gehenden TCP/IP-Details in Kapitel 11.

- Testen Sie die Beispiele, die in Sambar integriert sind.
- Untersuchen die weiter gehenden Möglichkeiten, die Ihnen Sambar bietet.
- Versuchen Sie, die Konfigurationsdatei von Apache zu verstehen.

4.10 Aufgaben

1. Welches Protokoll ist Basis für das WWW?
2. Was ist HTML?
3. Was ist der Standardport für einen Webserver?
4. Kann man auf einem Rechner mehrere Webserver gleichzeitig laufen lassen?
5. Was ist Server-Parsing?
6. Wie heißt das öffentliche Verzeichnis von Sambar?
7. Wie heißt das öffentliche Verzeichnis von Apache?
8. Was ist eine Defaultseite?