7 Dateiarchive per FTP bereitstellen

Der FTP-Dienst (**F**ile **T**ransfer **P**rotocol) dient dazu, Dateien zwischen zwei Rechnern auszutauschen. FTP gehört zu den klassischen Internet-Diensten und ist für jede Hard- und Softwareplattform verfügbar, über die ein Internetzugang möglich ist.

FTP-Server kann man einsetzen zum

- Bereitstellen von Dateien zum Fernladen (Download) durch Benutzer und
- Aufnehmen von Dateien zum Fernspeichern (Upload) durch Benutzer.

Bei FTP unterscheidet man zwischen anonymen Zugang und Zugang mit Benutzernamen und Passwort. In der SuSE-Grundinstallation kann sich jeder auf dem Linux-Server bekannte Nutzer per FTP mit seinem Home-Verzeichnis verbinden. Einen anonymen Zugang zum Fernladen und Fernspeichern muss man erst konfigurieren.

Da FTP-Benutzer sich frei im Verzeichnisbaum des Linux-Servers bewegen dürfen, entstehen Sicherheitsrisiken.

In diesem Kapitel wird es hauptsächlich darum gehen, den Server schrittweise sicherer zu machen.

Zuerst lernen Sie grundlegende Ideen zu FTP und zur sicheren FTP-Installation kennen:

- Zugänge für normale Benutzer,
- Zugänge für spezielle Benutzer,
- Zugänge für anonyme Benutzer.

Sie lernen Grundlagen von `wu.ftp` und von Konzepten kennen, wie man anonyme Benutzern und auch normalen Benutzern den Zugriff auf einen kleinen Ast des Dateibaumes beschränkt, ihnen aber dennoch grundlegende Dateibefehle zur Verfügung stellt.

Ferner lernen Sie Sicherheitskonzepte für lesenden FTP-Zugriff (Download) und schreibenden FTP-Zugriff (Upload) in ein besonderes Upload-Verzeichnis kennen. Wenn man anonymen Benutzern den Upload erlaubt, dann sollte man die gespeicherten Dateien erst nach einer Kontrolle durch Systemadministratoren auch zum Download bereitstellen, um Risiken durch Viren und unerwünschte Inhalte zu begrenzen.

7.1 Wann brauchen Sie einen eigenen FTP-Server?

In einem reinen Windows-Netz tauschen Anwender Daten am einfachsten über die Netzwerkumgebung aus. Auf freigegebene Ordner kann man über das Netz zugreifen (SMB-Protokoll). Mit dem Programm Samba (siehe Kapitel 9) kann man Linux-Server so ausrüsten, dass sie sich in dieses System integrieren.

Sind im Intranet verschiedenartige Betriebssysteme vorhanden, oder sollen Dateien auch über das Internet angeboten werden, so empfiehlt sich ein eigener FTP-Server.

7.2 So arbeitet ein FTP-Server

FTP arbeitet mit je einem Verbindungskanal zum Steuern der Übertragung und für die Übertragung selbst:

- Auf dem Kommandokanal wartet der FTP-Server auf Befehle.
- Die eigentlichen Daten versendet oder empfängt er dann über einen gesonderten Datenkanal.

Als Kommandos erwartet der Server Befehle, die üblichen Unix- oder DOS-Kommandos entsprechen. Darunter sind Befehle zum Arbeiten mit dem Verzeichnisbaum, aber auch spezielle Kommandos für die Datenübertragung. Da viele dieser Befehle ein intensives Zusammenspiel zwischen Server und Client erfordern, nutzt FTP zwei Kanäle. Die wichtigsten Kommandos sind:

Befehl	*Erläuterung*
`ls, dir`	Anzeige des Inhaltsverzeichnisses
`cd <Zielverzeichnis>`	Verzeichniswechsel auf dem Server
`lcd <Zielverzeichnis>`	Verzeichniswechsel auf dem Client
`ascii, asc`	ASCII-Übertragungsmodus einschalten
`binary`	Binären Übertragungsmodus einschalten

Befehl	*Erläuterung*
`get <Datei>`	Angegebene Datei vom Server laden.
`mget <Datei(en)>`	Mehrere Dateien vom Server holen, Wildcards * und ? erlaubt.
`put <Datei>`	Datei zum Server übertragen.
`put <Datei(en)>`	Mehrere Dateien zum Server übertragen, Wildcards * und ? erlaubt.
`quit`	Programm beenden.

Tabelle 7.1: FTP-Befehle und Erläuterungen

Die meisten Benutzer haben nur noch wenig direkt mit diesen Kommandos zu tun, da es für alle Betriebssysteme sehr komfortable FTP-Clients (z.B. WS_FTP) gibt, die sich wie der Windows-Dateimanager bedienen lassen. Im Hintergrund senden diese Programme die Standardbefehle an den Server.

Sehr achten sollte man immer auf den Übertragungsmodus. Im ASCII-Modus überträgt FTP die Dateien zeilenweise. Das Zeilenende erkennt das sendende System an den jeweiligen Endmarkierungen, das Zielsystem ergänzt die eigenen Endmarkierungen. Bei DOS/Windows endet eine Textzeile immer mit der Zeichenfolge `#10#13`, unter Linux nur mit der Zeichenfolge `#10`. Beim Mac ist es `#13`.

Kopiert man eine Textdatei binär zwischen verschiedenen Systemen, so stimmen diese Zeilenschaltungen nicht, ein Mac-Text z.B. besteht auf einem Linux-System nur aus einer einzigen Zeile. Besonders problematisch ist das beim Übertragen von Programmquelltext, der dann auf dem Zielsystem nicht funktionieren kann. Im ASCII-Modus setzt FTP die Zeilenschaltungen richtig um.

Binärdateien kopiert FTP immer unverändert.

7.3 FTP-Server einrichten und verwalten

In der Unix-Welt gibt es viele verschiedene FTP-Server Implementationen mit unterschiedlichen Konfigurationsmöglichkeiten und Sicherheitslevels. Standardmäßig installiert YaST einen sehr einfachen Server, den `in.FTP` mit dem `nkitb` aus der Serie `a` (Grundsystem).

Dieser FTP-Server dürfte auch bei Ihnen sofort funktionieren (siehe Kapitel 5).

Der FTP-Server braucht nur eine globale Konfigurationsdatei, die Datei */ETC/FTPUSERS*. Hier finden Sie eine Liste von Benutzern, die FTP nicht benutzen dürfen. In der Vorgabe sind dies die Standardbenutzer, die jeweils zu bestimmten Programmen gehören.

/etc/ftpusers

```
#
# ftpusers  This file describes the names of
# the users that may
#     _*NOT*_ log into the system via the FTP server.
#     This usually includes "root", "uucp", "news" and the
#     like, because those users have too much power to be
#     allowed to do "just" FTP...
#
adabas
amanda
at
bin
cyrus
daemon
dbmaker
db2fenc1
db2inst1
db2as
empress
fax
firewall
fnet
games
gdm
gnats
irc
informix
ixess
lnx
lp
man
mdom
mysql
named
news
nobody
nps
postfix
postgres
root
skyrix
```

```
uucp
virtuoso
yard
# End.
```

In dieser Datei sind diejenigen Benutzer eingetragen, die FTP nicht nutzen dürfen. Hierzu sollten alle systeminternen Benutzer wie `news` und `uucp` gehören, vor allem aber auch `root`. Mit dem Root-Account könnte man sonst per FTP alle Dateien auf dem gesamten System überschreiben oder löschen.

Will man einzelne Benutzer vom FTP-Zugang ausschließen, so nimmt man sie einfach in diese Datei mit auf.

Viele FTP-Server erlauben auch einen anonymen Zugriff von Benutzern, die keinen Account auf dem System besitzen. Für den anonymen Zugang werden üblicherweise die folgenden Daten benutzt:

Feld	*Inhalt*	*Erläuterung*
`Benutzername`	anonymous oder ftp	Wie oft habe ich mich da schon vertippt.
`Passwort`	beliebig	Üblich ist es hier, die eigene E-Mail-Adresse anzugeben; manche Systeme überprüfen die Gültigkeit.

Tabelle 7.2: Anonymer Zugriff von Benutzern

Dieser anonyme Zugriff wird auch oft aus einem Webbrowser heraus genutzt. Die meisten Browser übermitteln beim Zugriff auf FTP-Adressen automatisch Benutzernamen und Passwort für den anonymen Zugriff.

In der Standardkonfiguration ist der anonyme Zugriff gesperrt. Um ihn zu aktivieren, muss man ein Kommentarzeichen in der Datei `/etc/pam.d/ftpd` entfernen:

```
#%PAM-1.0

# Uncomment this to achieve what used to be ftpd -A.
# auth       required     /lib/security/pam_listfile.so
# item=user sense=allow file=/etc/ftpchroot onerr=fail

auth     required       /lib/security/pam_listfile.so
➥ item=user sense=deny file=/etc/ftpusers onerr=succeed
# Uncomment the following line for anonymous ftp.
auth     sufficient     /lib/security/pam_ftp.so
```

```
auth     required       /lib/security/pam_unix.so
auth     required       /lib/security/pam_shells.so
account  required       /lib/security/pam_unix.so
password required       /lib/security/pam_unix.so
session  required       /lib/security/pam_unix.so
```

Die Datei führt die Module auf, die für die unterschiedlichen Arten der Authentizierung zuständig sind. Vor der hervorgehobenen Zeile steht ursprünglich ein #-Zeichen, das diese Zeile deaktiviert. Entfernen Sie das Zeichen und speichern Sie die Datei, um anonymen FTP-Zugriff zu erlauben.

Der FTP-Server stellt anonymen Benutzern eine sog. Changed-Root-Umgebung (`chroot`) zur Verfügung, die aber noch nicht konfiguriert ist.

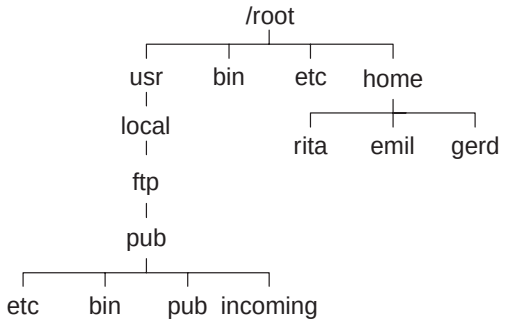

Abbildung 7.1: Changed-Root

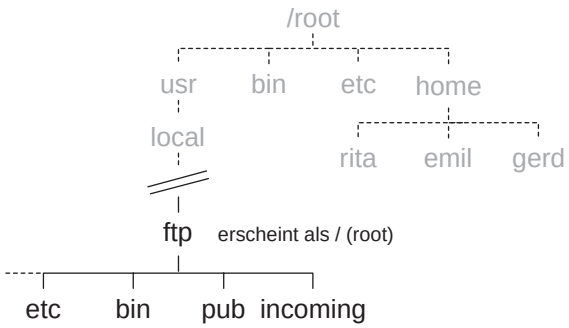

Abbildung 7.2: Dateisystem aus Sicht eines anonymen FTP-Nutzers

Changed-Root-Umgebungen geben Benutzern keinen Zugriff auf das gesamte Dateisystem, sondern nur auf einen Teil davon. Sie geben Benutzern ein verändertes Wurzelverzeichnis (Changed-Root). Hier in der Installation ist das

der Pfad /usr/local/ftp, das Home-Verzeichnis des Benutzers FTP. Für anonyme Benutzer ist das die Wurzel des Verzeichnisbaumes, den sie sehen können. Dieses System kann Sicherheitsrisiken vermindern.

Das veränderte Wurzelverzeichnis nimmt anonymen Benutzern jeglichen Zugriff auf Standardbefehle wie z.B. ls, die außerhalb des zulässigen Verzeichnisbaumes liegen. Somit können anonyme Benutzer sich zwar anmelden, mehr aber nicht.

Damit Befehle wie ls Benutzern auch hier wieder zur Verfügung stehen, muss man einige Standarddateien im Verzeichnis /usr/local/ftp zur Verfügung stellen. Das ist aufwendig, da man sehr auf die Rechte achten muss und die Programme auch nicht einfach an die Standardbibliotheken herankommen. Am einfachsten installiert man das Paket ftpdir aus der Serie n nach, das genau die benötigten Dateien und Verzeichnisse enthält. Auf dem FTP-Server finden Sie die Datei ftpdir.rpm im Verzeichnis n1.

Nach der Anmeldung sehen anonyme Benutzer jetzt eine Vielzahl von Ordnern, wie hier z.B. bei einem Zugriff aus dem Internet Explorer heraus:

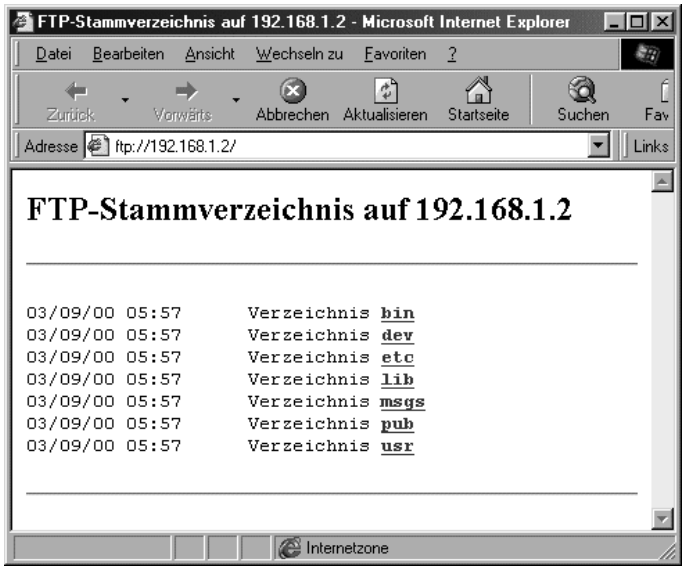

Abbildung 7.3: Anonymer Zugriff mit dem Internet Explorer

Die meisten der aufgeführten Ordner hängen mit der veränderten Umgebung zusammen. Für den Datenaustausch ist der Ordner pub zuständig. Hier können Systemverwalter Dateien zum Download anbieten. Damit der User ftp die Dateien lesen kann, müssen die Eigentumsverhältnisse und die Dateirechte passend eingestellt sein. Mit

```
chmod 444 *
```

ist man da auf der sicheren Seite.

SuSE hat voreingestellt, dass anonyme Benutzer aus diesem Ordner zwar Dateien lesen, dort aber keine Dateien ablegen können. Das dient wieder der Sicherheit. Will man anonymen Benutzern erlauben, Dateien auf dem FTP-Server abzulegen, so sollte man neben `pub` einen Ordner `incoming` einrichten und für diesen die Dateirechte passend setzen.

Die scheinbar einfachste Möglichkeit wäre, für den Ordner alles freizugeben:

```
chmod 777 incoming
```

Dann könnten anonyme Benutzer dort Dateien ablegen und alle dort abgelegten Dateien auch wieder laden. Das ist riskant, da anonyme Benutzer hier auch unerwünschte Inhalte und virenverseuchte Dateien ablegen können.

Üblich ist es daher, mit

```
chmod 733 incoming
```

die Rechte so einzustellen, dass anonyme Benutzer dort Dateien ablegen, aber kein Inhaltsverzeichnis dieses Ordners abrufen können.

7.4 Zugriffssteuerung mit wu.ftp

Das Prinzip der Changed-Root-Umgebung ist eine feine Sache und auch für eingetragene Benutzer wünschenswert. Dazu müssen Sie an drei Stellen Veränderungen vornehmen:

- Den bisherigen FTP-Server `in.ftpd` durch den `wu.ftpd` ersetzen,
- Festlegen, für welche Benutzergruppen Sie diese Veränderung umsetzen wollen und
- die Einträge der Home-Verzeichnisse der Benutzer verändern.

Will man Changed-Root-Umgebungen für normale Benutzer aktivieren, so sollte man statt des installierten FTP-Servers `in.FTP` einen konfigurierbaren Server einrichten, z.B. den `wu.ftp`. Dieser Server befindet sich bei SuSE im Paket `wuftpd` der Serie `n` bzw. in der Datei `wuftpd.rpm` im Verzeichnis `n1` auf dem FTP-Server.

Der bisherige Server braucht man dazu nicht zu entfernen.

Wenn zwei FTP-Server installiert sind, müssen Sie festlegen, welcher davon zukünftig starten soll. Da meist der Superdämon `inetd` die Standarddienste aufrufen, müssen Sie dessen Konfigurationsdatei `/etc/inetd.conf` editieren, wobei SuSE schon viel vorbereitet hat:

`/etc/inetd.conf` (Auszug ab Zeile 22):

```
# These are standard services.
#
# ftp  stream  tcp  nowait  root  /usr/sbin/tcpd  wu.ftpd -a
# ftp  stream  tcp  nowait  root  /usr/sbin/tcpd  proftpd
ftp  stream  tcp  nowait  root  /usr/sbin/tcpd  in.ftpd
#
```

In der Datei sind schon Einträge für alle von SuSE gelieferten FTP-Server vorhanden. Sie müssen nur die Zeile `in.ftp` auskommentieren, indem Sie das Kommentar-Zeichen # voranstellen und die `wu.ftp` Zeile aktivieren, indem Sie das #-Zeichen und das dann führende Leerzeichen entfernen.

`/etc/inetd.conf` (veränderte Version ab Zeile 22):

```
# These are standard services.
#
ftp  stream tcp  nowait  root  /usr/sbin/tcpd  wu.ftpd-2.6 -a
# ftp  stream  tcp  nowait  root  /usr/sbin/tcpd  proftpd
# ftp  stream  tcp  nowait  root  /usr/sbin/tcpd  in.ftpd
#
```

SuSE liefert zwei Versionen des wu.ftpd. Die ältere Version steht unter dem Namen `wu.ftpd` zur Verfügung, die neuere unter `wu.ftpd-2.6`. Da die ältere Version fehlerhaft ist, sollten Sie wie hier im Listing angegeben auf die neue Version umstellen.

Nun muss der Superdämon noch die veränderte Konfiguration erfahren:

```
/sbin/init.d/inetd reload
```

Den nächsten FTP-Zugriff bedient nun wu.ftpd. Man wird hierbei keinen Unterschied feststellen. Lediglich Systemverwalter sehen, dass der wu.ftpd nun jede Datenübertragung in der Datei `/var/log/xferlog` protokolliert und jede Anmeldung in der `/var/log/messages` einträgt. Schon dies erhöht die Systemsicherheit.

Für das eigentliche Ziel, möglichst hohe Sicherheit, muss man die Konfigurations-Datei `/etc/ftpaccess` erweitern. Die installierte Version dieser Datei ist zu knapp gehalten.

Sie müssen die Konfigurationsdatei deutlich erweitern, wenn Sie die Zugriffsrechte so einstellen wollen, dass ein sicherer anonymer Zugriff möglich wird.

SuSE hat dafür unter `/usr/share/doc/packages/wuftpd/ftpaccess.anonymous` eine umfangreichere Datei abgelegt, mit der Sie die `/etc/ftpaccess` ersetzen sollten.

```
cp  /usr/share/doc/packages/wuftpd/ftpaccess.anonymous
/etc/ftpaccess
```

Nach diesem Kopiervorgang erlaubt auch der wu.ftpd den Zugriff anonymer Benutzer. Im folgenden Listing sind die Stellen, die Sie aus Sicherheitsgründen noch ändern sollten, fett hervorgehoben. Direkt danach lesen Sie die Erläuterungen dazu:

`/etc/ftpaccess`:

```
# email of the responsible person for the %E-cookie
email ftp-admin@localhost

#
# if you specify a list of hosts for the "local"
# class, only those
# hosts will be allowed to login as
# "real". All other hosts can
# only login as "anonymous".
#
class   local   real                    *
class   remote  guest,anonymous         *

readme  README* login
readme  README* cwd=*

# limit of 20 connections
limit   local   20   Any    /usr/local/ftp/msgs/msg.dead
limit   remote  20   Any    /usr/local/ftp/msgs/msg.dead

#
# output /usr/local/ftp/msgs/welcome.msg on login
# and all ".message" files in subdirectories
#
banner  /usr/local/ftp/msgs/welcome.msg
message .message                                cwd=*

#message /msgs/welcome.msg                      login
#message /usr/local/ftp/msgs/welcome.msg        login   local
#message /msgs/welcome.msg                      login   remote

# do not check password for anonymous logins
#passwd-check rfc822 warn
passwd-check none
```

```
# allow compression/tar for all users
compress        yes             local remote
tar             yes             local remote

# log all transfers
#log commands real
log transfers anonymous,real inbound,outbound

#shutdown /etc/shutmsg

# do not give those files. do not give
# "core"-files in any directory.
noretrieve /etc/passwd /etc/group core .notar
noretrieve /usr/local/ftp/incoming

# do not allow these commands for anonymous users
chmod         no        anonymous
delete        no        anonymous
overwrite     no        anonymous
rename        no        anonymous
umask         no        anonymous

#
# !! see documentation how to setup
# uploads for anonymous users !!
#
# specify the upload directory information
upload  /usr/local/ftp  *             no     nobody  nogroup
➥ 0000 nodirs
upload  /usr/local/ftp  /bin          no
upload  /usr/local/ftp  /etc          no
upload  /usr/local/ftp  /incoming     yes
➥ root    daemon  0600 nodirs

# path-filter...
#path-filter  anonymous  /msgs/pathmsg
#^[-A-Za-z0-9_\.]*$  ^\.  ^-
#path-filter  guest      /msgs/pathmsg
#^[-A-Za-z0-9_\.]*$  ^\.  ^-

# specify which group of users will be treated as "guests".
guestgroup users
```

In der Konfigurationsdatei sind gegenüber der Vorlage drei wichtige Details geändert. Bei der Rechtevergabe steht ursprünglich:

```
# do not allow these commands for anonymous users
chmod         no        guest,anonymous
delete        no        guest,anonymous
overwrite     no        guest,anonymous
rename        no        guest,anonymous
umask         no        guest,anonymous
```

Damit verbieten Sie sowohl für den anonymen Benutzer, als auch für den mit der Changed-Root-Umgebung explizit die angegebenen Operationen. Beide Gruppen dürfen also hier weder Dateien löschen (delete) noch Dateien überschreiben (overwrite).

Wenn Sie Ihren bekannten Benutzern vertrauen, dann können Sie durch Entfernen von `guest` diesen Rechte wieder einräumen. Im Extremfall geben Sie also an:

```
# do not allow these commands for anonymous users
chmod         no        anonymous
delete        no        anonymous
overwrite     no        anonymous
rename        no        anonymous
umask         no        anonymous
```

Damit können dann normale FTP-Nutzer Dateien löschen, überschreiben oder die Dateirechte ändern.

Die Zeile

```
upload  /usr/local/ftp  /incoming     yes    root    daemon
➥ 0600 nodirs
```

ist in der Vorlage auskommentiert, um keine Uploads zuzulassen. Es muss nur das # Zeichen am Zeilenanfang entfernt werden, um das zu ändern.

Die Einstellung `yes` erlaubt anonyme Uploads. Die Eigentümer der abgelegten Dateien sind `root` und die `daemon` Gruppe.

Die Dateirechte setzt der FTP auf `0600` (nur der Eigentümer darf Lesen und Schreiben) und anonyme Benutzer dürfen durch die Einstellung `nodirs` keine Unterverzeichnissen anlegen.

Wichtig ist die letzte Zeile, die ursprünglich auch auskommentiert war. Benutzer, die der angegebenen Gruppe `users` angehören, normalerweise also alle, haben keinen freien FTP-Zugriff mehr, sondern bekommen nur noch eine Changed-Root-Umgebung. Dies ist die sicherste Einstellung.

Damit Anwender die Changed-Root-Umgebung nutzen können, muss man noch zwei Einstellungen ändern.

Wenn die Home-Verzeichnisse der Benutzer alle im Ordner `/home` liegen, ist das Home-Verzeichnis des Benutzers `test` also `/home/test`.

Für die geänderte Umgebung muss man hier `/home/./test` einstellen, um dem FTP-Server deutlich zu machen, dass er für diesen Benutzer die Changed-Root-Umgebung aktivieren muss. Geben Sie dazu

```
usermod -d /home/./test  test
```

ein. Usermod mit dem Parameter `-d` ändert das Home-Verzeichnis des angegebenen Benutzers.

Der Punkt im Pfad ist die Grenze, die Benutzer beim Verzeichniswechsel nicht überschreiten dürfen, ihr Rootverzeichnis ist `/home`.

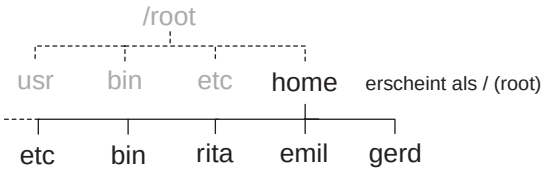

Abbildung 7.4: Dateisystem aus Sicht autorisierter FTP-Nutzer

Nach diesen Beschränkungen der Benutzer auf einen kleinen Ast des Dateibaumes fehlen den Benutzern noch die Standardordner mit wichtigen Unix-Befehlen wie `ls`. Die hatten Sie schon einmal unter `/usr/local/ftp` angelegt. Kopieren Sie sie mit

```
cp -a /usr/local/ftp/*  /home
```

an die richtige Stelle.

Damit Sie nicht für jeden Benutzer einzeln diese Standardverzeichnisse mit den Befehlen einrichten müssen, sollten Sie `/home` als Verzeichniswurzel nehmen und nicht `/home/test`.

Danach ist die Konfiguration abgeschlossen und zukünftig können User das Verzeichnis `/home` bei FTP nicht mehr verlassen.

FTP kennt dann drei Benutzer-Gruppen:

- Anonyme, hier wird das Verzeichnis `/usr/local/ftp` als Rootverzeichnis eingestellt.
- Mitglieder der Gruppe `users` mit dem Rootverzeichnis `/home`.
- Bekannte Nutzer, die nicht der Gruppe `users` angehören behalten vollen Zugriff auf das Dateisystem.

Damit ist eine grundlegende Sicherung des FTP erreicht. Für weitere Sicherheitsmerkmale befragen Sie bitte die Manpages zu `ftpaccess` und `wu.ftpd`.

7.5 Zugriffe protokollieren und auswerten

Zugriffe auf allgemein zugängliche Dienste sollten Systemverwalter immer kontrollieren, insbesondere wenn sie auch anonyme Benutzer zulassen. Ansonsten besteht die Gefahr, dass sich die Speicher mit illegaler oder unerwünschter Software füllen.

In der bisherigen Konfiguration hält wu.ftpd wenig Informationen fest. In der Datei `/var/log/messages` protokolliert er die Zugriffe auf den Server:

```
Apr 17 16:47:17 boss wu.ftpd-2.6[927]: connect from
➥ 192.168.1.40 (192.168.1.40)
```

In der Datei `/var/log/xferlog` speichert wu.ftpd folgendermaßen, welche Dateien anonyme Benutzer übertragen:

```
Mon Apr 17 16:49:47 2000 1 192.168.1.40 382942
➥ /usr/local/ftp/incoming/stgb.html b _ i a guest@unknown ftp
➥ 0 * c
```

Die folgende Veränderung der Datei `/etc/ftpaccess` bewirkt, dass wu.ftpd für alle Benutzergruppen Details sehr ausführlich protokolliert:

`/etc/ftpaccess` (Auszug ab Zeile 39):

```
#log all transfers
log commands anonymous,real,guest
log transfers anonymous,real,guest inbound,outbound
```

So protokolliert der wu.ftpd alle Kommandos und jede Datei-Übertragung für alle drei Nutzergruppen. Das Protokoll können Sie recht einfach auswerten.

```
Apr 17 16:41:46 boss ftpd[886]: USER adams
Apr 17 16:41:46 boss ftpd[886]: PASS password
Apr 17 16:41:46 boss ftpd[886]: failed login from 192.168.1.40
➥ [192.168.1.40]
```

Die Benutzeranmeldung ist gescheitert, Benutzername oder Passwort sind falsch.

```
Apr 17 16:59:40 boss wu.ftpd-2.6[943]: connect from
➥ 192.168.1.40 (192.168.1.40)
Apr 17 16:59:41 boss ftpd[943]: USER adams
Apr 17 16:59:41 boss ftpd[943]: PASS password
Apr 17 16:59:41 boss ftpd[943]: PWD
Apr 17 16:59:41 boss ftpd[943]: SYST
Apr 17 16:59:41 boss ftpd[943]: PORT
Apr 17 16:59:41 boss ftpd[943]: LIST
```

Diese Benutzeranmeldung ist erfolgreich. Anschließend hat der FTP-Client das aktuelle Verzeichnis (PWD) und den Verzeichnisinhalt (LIST) abgefragt.

```
Apr 17 16:59:47 boss ftpd[943]: CDUP
Apr 17 16:59:47 boss ftpd[943]: PWD
Apr 17 16:59:47 boss ftpd[943]: PORT
Apr 17 16:59:47 boss ftpd[943]: LIST
```

Ein Verzeichniswechsel (CDUP) auf die nächsthöhere Verzeichnisebene.

Eine erfolgreiche Datenübertragung hinterlässt in der `/var/log/messages` einen Eintrag wie:

```
Apr 17 17:07:22 boss ftpd[943]: TYPE Image
Apr 17 17:07:22 boss ftpd[943]: PORT
Apr 17 17:07:22 boss ftpd[943]: STOR stgb.html
Apr 17 17:07:23 boss ftpd[943]: PWD
Apr 17 17:07:23 boss ftpd[943]: TYPE ASCII
Apr 17 17:07:23 boss ftpd[943]: PORT
Apr 17 17:07:23 boss ftpd[943]: LIST
```

Die gleiche Datenübertragung ergibt in der `/var/log/xferlog` den folgenden Eintrag:

```
Mon Apr 17 17:07:23 2000 1 192.168.1.40 382942
➥ /home/adams/stgb.html b _ i g adams ftp 0 * c
```

Blicken Sie regelmäßig in die Protokolldateien und achten Sie vor allem auf die Häufung von Login-Fehlern, die von Hack-Versuchen herrühren könnten. Achten Sie auch darauf, was Benutzer mit vollem Dateizugriff auf Ihrem System treiben. Zugriffe dieser Benutzer auf Systemdateien sollten Sie dazu veranlassen, diese auf eine Changed-Root-Umgebung zu beschränken.

Wer einen Upload-Ordner für anonyme Benutzer anbietet, sollte dort abgelegte Dateien regelmäßig prüfen und gegebenenfalls zum Download anbieten.

7.6 Statistische Auswertung mit Webalizer

Das Programm Webalizer haben Sie bereits im Kapitel 6 kennengelernt. Es dient, wie auch der Name schon sagt, ursprünglich zur Auswertung von Logdateien von Webservern.

Die aktuelle Version des Programms kann auch Informationen aus der Datei `/etc/xferlog` auszuwerten. Auf gut besuchten Servern ist das sicherlich eine große Hilfe.

Die folgende Beschreibung geht davon aus, dass Sie die FTP-Statistik zusätzlich zu einer eventuell vorhandenen Web-Statistik pflegen möchten.

Sie müssen zuerst ein Verzeichnis einrichten, in dem Webalizer die FTP-Statisktik ablegen kann. Eine Möglichkeit wäre `/usr/local/httpd/htdocs/ftpalizer`:

```
mkdir  /usr/local/httpd/htdocs/ftpalizer
```

Nun müssen Sie eine zweite Konfigurationsdatei erzeugen, die für die Analyse der FTP Logdatei angepasst ist. Sie können dazu einfach die vorhandene Datei kopieren, z.B. als `ftpalizer.conf`:

```
cp  /etc/webalizer.conf  /etc/ftpalizer.conf
```

Damit der Webalizer auch mit der Datei *xferlog* richtig umgehen kann, müssen Sie diese Datei ein bisschen anpassen. Am wichtigsten ist dabei die Einstellung, die dem Webalizer mitteilt, dass es sich um eine Logdatei des FTP-Servers und nicht eine des Web-Servers handelt.

`/etc/ftpalizer.conf` (Auszug ab Zeile 26)

```
# LogFile defines the web server log file to use.
# If not specified here or on on
# the command line, input will default to STDIN.

LogFile        /var/log/xferlog

# LogType defines the log type being processed.
# Normally, the Webalizer
# expects a CLF or Combined web server log as input.
# Using this option, you can process ftp logs as well
# (xferlog as produced by wu-ftp and others).
# Values can be 'web' or 'ftp', with 'web' the default.

LogType ftp
```

```
# OutputDir is where you want to put the output files.
# This should be a full path name, however relative ones
# might work as well.
# If no output directory is specified, the current directory
# will be used.

OutputDir       /usr/local/httpd/htdocs/ftpalizer
```

Nun können Sie den Webalizer starten und ihm die eben erstellte Konfigurationsdatei konkret über den Parameter `-c` angeben.

```
webalizer  -c  /etc/ftpalizer.conf
```

Auch diesen Programmaufruf können Sie natürlich in die Cron-Tab von root mit aufnehmen und damit die Auswertung tagesaktuell halten.