

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 den Systemadministrator auch zum Download bereitstellen. Das vermeidet Risiken durch Viren und unerwünschte Inhalte.

7.1 Wann brauchen Sie einen eigenen FTP-Server?

In einem reinen Windows-Netz erfolgt der Datenaustausch am einfachsten über die Netzwerkumgebung. Auf freigegebene Ordner kann man dann über das Netz zugreifen (SMB-Protokoll). Über das Programm Samba (siehe Kapitel 9) kann man auch den Linux-Server so ausrüsten, dass er sich in dieses System integriert.

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 zwei Verbindungskanälen.

- 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 Linux- 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:

<i>Befehl</i>	<i>Erläuterung</i>
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

<i>Befehl</i>	<i>Erläuterung</i>
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 Zielenende 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 dann, wenn man Programmquelltexte überträgt, die dann auf dem Zielsystem nicht funktionieren. Im ASCII-Modus setzt FTP die Zeilenschaltungen richtig um.

Binärdateien werden immer unverändert kopiert.

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 verfügt nur über eine einzige globale Konfigurationsdatei:

/etc/ftusers

```
#
# ftusers 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:

<i>Feld</i>	<i>Inhalt</i>	<i>Erläuterung</i>
Benutzername	<code>anonymous</code> oder <code>ftp</code>	Wie oft habe ich mich da schon ver tippt.
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 Authentifizierung 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.

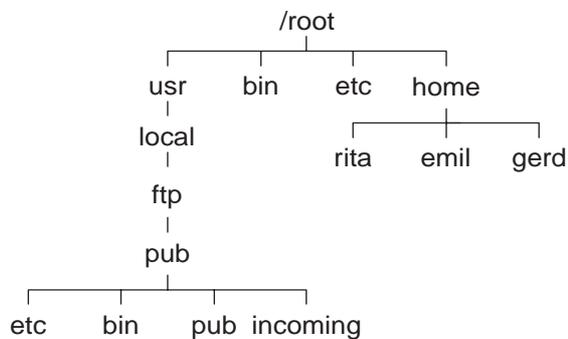


Abbildung 7.1: Changed-Root

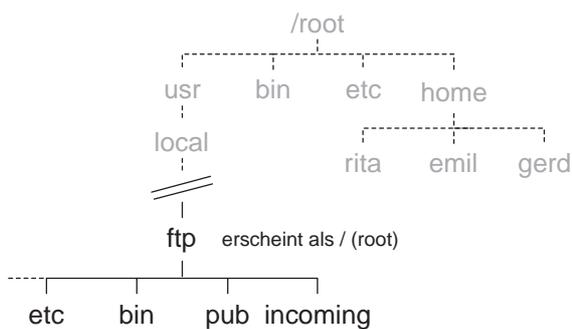


Abbildung 7.2: Dateisystem aus Sicht eines anonymen FTP-Nutzers

Changed-Root-Umgebungen geben Benutzern keinen Zugriff auf das gesamte Dateisystem, sondern nur auf einen Teil davon. Sie geben dem Benutzer 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 `ftplib` aus der Serie `n` nach, das genau die benötigten Dateien und Verzeichnisse enthält.

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

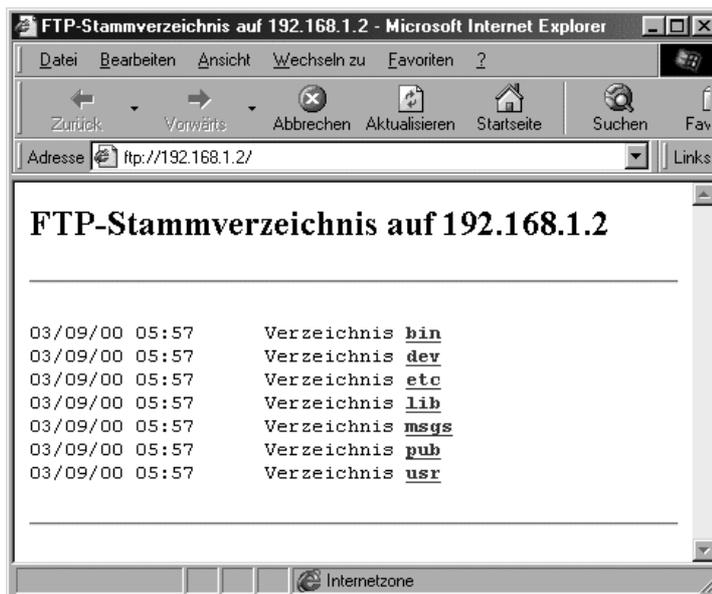


Abbildung 7.3: Anonymer Zugriff mit dem Internet Explorer

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.

Nach der Voreinstellung können anonyme Benutzer aus diesem Ordner zwar Dateien lesen, dort aber keine Dateien ablegen. 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 Eintragung für das Home-Verzeichnis bei den Benutzern 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`.

Der bisherige Server muss dazu nicht entfernt werden.

Wenn zwei FTP-Server installiert sind, müssen Sie festlegen, welcher davon zukünftig gestartet werden soll. Die Standarddienste werden in der Regel vom Superdämon `inetd` aufgerufen, also 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 FTP-Server vorhanden, die SuSE mitliefert. Sie müssen nur die Zeile `in.ftpd` auskommentieren, indem Sie das Kommentar-Zeichen `#` voranstellen und die `wu.ftpd` 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 gleich auf die neue Version umstellen, so wie hier angegeben.

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 System-sicherheit.

Für das eigentliche Ziel, möglichst hohe Sicherheit, muss man die Konfigurations-Datei `/etc/ftppaccess` erweitern. Die installierte Version dieser Datei ist zu knapp gehalten. SuSE hat aber unter `/usr/doc/packages/wuftpd/ftppaccess.anonymous` eine umfangreichere Datei abgelegt, mit der Sie die `/etc/ftppaccess` ersetzen sollten.

```
cp /usr/doc/packages/wuftpd/ftppaccess.anonymous
/etc/ftppaccess
```

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/ftppass:`

```
# 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/messages/msg.dead
limit remote 20 Any /usr/local/ftp/messages/msg.dead

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

#message /messages/welcome.msg login
#message /usr/local/ftp/messages/welcome.msg login local
#message /messages/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
```

Dies verbietet sowohl für den anonymen Benutzer, als auch für den mit der Changed-Root-Umgebung explizit Operationen. Wenn Sie Ihren bekannten Benutzern vertrauen, dann können Sie durch Entfernen von `guest` diese Rechte für Nutzer mit Changed-Root-Umgebung wieder vergeben. Im Extremfall also zu:

```
# 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.

Die Einstellung `yes` legt fest, dass anonyme Uploads erlaubt sind. Die Eigentümer der abgelegten Dateien sind `root` und die `daemon` Gruppe. Die Dateirechte werden auf `0600` gesetzt 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 die Changed-Root-Umgebung genutzt werden kann, sind noch zwei Einstellungen zu 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 hier `/home/./test` eingestellt werden, indem man

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

eingibt. `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`.

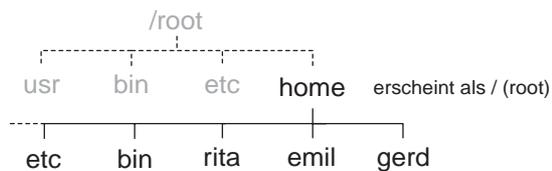


Abbildung 7.4: Dateisystem aus Sicht autorisierter FTP-Nutzer

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`, hier wird `/home` als Rootverzeichnis eingestellt.
- Bekannte Nutzer, die nicht der Gruppe `users` angehören; diese behalten vollen Zugriff auf das Dateisystem.

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

7.5 Zugriffe protokollieren und auswerten

Zugriffe auf allgemein zugängliche Dienste sollten Sie immer kontrollieren, insbesondere wenn Sie auch den anonymen Benutzer zulassen. Ansonsten besteht die Gefahr, dass Ihr Computer mit illegaler oder unerwünschter Software gefüllt wird.

Mit der bisherigen Konfiguration hält der `wu.ftp` 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 er die Dateiübertragungen für den anonymen Benutzer folgendermaßen:

```
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
```

Mit der folgenden Veränderung der Datei `/etc/ftppaccess` können Sie erreichen, dass `wu.ftp` für alle Benutzergruppen Details sehr ausführlich protokolliert:

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

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

So protokolliert der `wu.ftp` alle Kommandos und Datei-Übertragungen 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 das 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 die Benutzer mit vollem Dateizugriff auf Ihrem System so treiben. Zugriffe dieser Benutzer auf Systemdateien sollten Sie dazu veranlassen, diesen auf eine Changed-Root-Umgebung zu beschränken.

Wenn Sie einen Upload-Ordner für den anonymen Benutzer anbieten, dann wäre es zumindest unhöflich, dort abgelegte Dateien nicht regelmäßig zu prüfen und dann gegebenenfalls zum Download anzubieten.

