Aus dem Nähkästchen geplaudert: Pro-FTPD

# Transferwissen

Dateien weitergeben ist in jeder Hinsicht das ureigene Ziel freier Software – der Protokollklassiker FTP (File Transfer Protocol) löst diese Aufgabe sehr effizient. Bevor ein Admin diesen Server einrichtet, sollte er aber einiges über die Hintergründe wissen. Marc André Selig

**Fürs Verteilen** von Dateien über das Internet gibt es zahlreiche Protokolle, aber nicht jedes eignet sich für jeden Zweck. Die recht beliebten Peer-to-Peer-Dienste (Gnutella, E-Donkey und Verwandte) wurden für Szenarien geschaffen, in denen jeder Teilnehmer Files anbieten und empfangen möchte. Beim Verbreiten freier Software gilt eine andere Prämisse: Relativ wenige Anbieter – die Entwickler – stehen einer großen Schar Interessenten gegenüber. Die bekanntesten Protokolle für diese Konstellation sind HTTP und FTP.

Für einfache Aufgaben ist HTTP die nahe liegende Variante: Apache ist fast überall vorhanden, auch die Clientsoftware ist auf jedem System installiert und obendrein denkbar einfach zu bedienen. HTTP lässt aber einige Features vermissen, die Anwender zum Beispiel bei großen Dateien und wackligen Verbindungen benötigen.

## Abgebrochene Übertragung fortsetzen

Am bedeutsamsten ist wohl das Verhalten bei abgebrochenen Übertragungen. Zwar enthält die Spezifikation für HTTP 1.1 die Funktionalität, eine Übertragung auf halbem Wege fortzusetzen. Viele Browser können damit allerdings noch nicht umgehen und verschenken somit wertvolle Bandbreite und Zeit. Obendrein ist der Betrieb eines Webservers trotz aller modernen Software relativ Ressourcen-intensiv.

Die meisten dieser Schwierigkeiten treten bei FTP [1] gar nicht auf. Dabei handelt es sich beim File Transfer Protocol nicht etwa um eine spätere Erweiterung oder Nachbesserung von HTTP, sondern um ein gänzlich anderes und außerdem wesentlich älteres Protokoll.

## Installation

Daher gehört für viele Webserver ein FTP-Daemon zum notwendigen Beiwerk. Wer eine solche Software installieren möchte, hat zunächst die Qual der Wahl. Der Kasten „Wichtige FTP-Daemons“ gibt eine Übersicht der verbreiteten FTP-Server unter Linux. Dieser Artikel beschreibt den leistungsfähigen, sicheren und obendrein leicht erlernbaren Pro-FTPD.

Die Mühe der Installation nehmen Ihnen viele aktuelle Distributionen ab – sie liefern die Pakete bereits mit. Falls nicht, müssen Sie selbst Hand anlegen und zunächst die Quelltexte und deren digitale Signaturen von [4] beziehen. Die Signaturen sollten Sie unbedingt prüfen: Listing 1 zeigt die einzelnen Schritte. Zuerst kontrollieren Sie mit »md5sum« die MD5-Prüfsummen. Damit ist sichergestellt, dass das Paket unversehrt ist und beim Download keine zufälligen Fehler auftraten.

Der wichtigere Test betrifft die GPG-Signatur. Im Gegensatz zur MD5-Summe prüft sie die Authentizität. Das Problem hierbei: Sie wissen in der Regel nicht, dass der zugehörige öffentliche Schlüssel tatsächlich dem Pro-FTPD-Entwickler TJ Saunders gehört. Es gibt keine Garantie, dass der Key korrekt ist. Dennoch ist eine korrekte Signatur ein weiterer Vertrauen bildender Schritt.

Das darauf folgende Übersetzen des Quellcodes und Installieren der erzeugten Dateien entspricht der üblichen Vorgehensweise unter Linux. Listing 2 zeigt die einzelnen Schritte. Interessant sind vor allem die Configure-Optionen in Zeile 3.

## Benutzer und Verzeichnisse

Die Installation ist damit aber noch nicht abgeschlossen. Prüfen Sie als Nächstes, ob geeignete Benutzer für den Daemon existieren. Pro-FTPD verwendet in seiner typischen Betriebsart gleich drei Nutzernamen, jeweils mit einer zugeordneten Gruppe: Der größte Teil des Codes läuft als Benutzer »nobody« mit der Gruppe »nogroup«. Für manche Aktionen benötigt der Daemon allerdings

Root-Rechte, mit denen Sie ihn auch starten müssen. Für die häufig genutzte anonyme Anmeldung (Anonymous FTP) verwendet Pro-FTP einen Gast-Account. Richten Sie dafür den User »ftp« mit der Gruppe »ftp« ein.

Dieser anonyme Nutzer benötigt ein Homeverzeichnis, in das Sie die öffentlich zugänglichen Dateien schreiben. Zu beachten ist dabei: Der Daemon sollte diese Files lesen können, sie sollten ihm aber nicht gehören. Pro-FTPD kann zwar per Konfiguration einen Schreibschutz erzwingen, es ist aber dennoch sehr zu empfehlen, sich an diese Sicherheitsregel zu halten.

Gegebenenfalls richten Sie zusätzlich ein Upload-Verzeichnis ein. Es heißt auf den meisten Servern »/incoming« oder »/pub/incoming«. Die Zugriffsrechte sollten so beschaffen sein, dass FTP zwar in dieses Directory schreiben darf, es jedoch nicht mehr lesen kann. Andernfalls bestünde die Gefahr, dass Ihr System von Crackern als temporäre Ablage im Netz missbraucht wird.

Welche Files bei solchen Gelegenheiten ohne Ihr Wissen auf Ihrem Server landen und welche – auch juristischen – Gefahren das nach sich zieht, lässt sich kaum abschätzen [8]. Die Verzeichnisstruktur »~ftp« könnte im Ergebnis so aussehen wie in Listing 3 gezeigt.

**Listing 1: Signaturen prüfen**

```
01 $ md5sum -c proftpd-1.2.9.tar.bz2.md5
02 proftpd-1.2.9.tar.bz2: OK
03 $ gpg proftpd-1.2.9.tar.bz2.asc
04 gpg: Signature made Fri Oct 31 09:39:42 2003 CET
   using DSA key ID A511976A
05 gpg: Good signature from "TJ Saunders
   <tj@castaglia.org>"
06 gpg: WARNING: This key is not certified with a
   trusted signature!
07 gpg:          There is no indication that the
   signature belongs to the owner.
08 Primary key fingerprint: 697E 684D 1668 D696 8428
   405C B78E 893F A511 976A
09 mas@ishi:/tmp>
```

**Listing 2: Pro-FTPD installieren**

```
01 $ bzip2 -cd proftpd-1.2.9.tar.bz2 | tar xf -
02 $ cd proftpd-1.2.9
03 $ ./configure --sysconfdir=/etc/proftpd --
   localstatedir=/var/run
04 [...]
05 $ make
06 [...]
07 $ su
08 Password:
09 # make install
```

**Listing 3: FTP-Homedir**

```
01 # chmod 733 ~ftp/incoming
02 # ls -la ~ftp
03 total 20
04 drwxr-xr-x   4 root   root   4096 Jan 18 18:58 .
05 drwxr-xr-x   4 root   root   4096 Jan 18 18:55 ..
06 drwx-wx-wx   2 root   root   4096 Jan 18 18:55
   incoming
07 drwxr-xr-x   2 root   root   4096 Jan 18 18:55 pub
08 -rw-r--r--   1 root   root     90 Jan 18 18:58
   welcome.msg
09 #
```

**Listing 4: Startskript**

```
01 # cp proftpd /etc/init.d/
02 # chown root:root /etc/init.d/proftpd
03 # chmod 744 /etc/init.d/proftpd
04 # cd /etc/init.d/rc3.d
05 # ln -s /etc/init.d/proftpd S85proftpd
06 # cd ../rc5.d
07 # ln -s /etc/init.d/proftpd S85proftpd
```

## Konfiguration und Aktivierung

Als nächsten Schritt prüfen Sie die vorgeschlagene Konfigurationsdatei »/etc/proftpd/proftpd.conf«. Pro-FTPD 1.2.9 installiert eine sinnvolle Minimalversion, sofern nicht bereits – aus einer früheren Installation – eine Konfigurationsdatei existiert. Diese Datei orientiert sich syntaktisch an Apache und will möglichst intuitiv verwendbar sein. Das Beispiel enthält aber nur Anweisungen für anonyme Downloads: Möchten Sie Uploads erlauben, sind einige Einträge zu ergänzen. Ein erweitertes Beispiel für »proftpd.conf« einschließlich Up- und Downloads finden Sie auf dem FTP-Server des Linux-Magazins [9].

Soll Ihr FTP-Server beim Systemstart automatisch hochfahren, haben Sie auf Unix-Systemen die Wahl zwischen zwei Varianten: Entweder wird der Prozess nur bei Bedarf von einem zentralen Verteiler namens »inetd« oder »xinetd« gestartet oder er läuft ständig im Hintergrund und kümmert sich selbsttätig um ankommende Verbindungen.

Die erste Variante hat eine frühere Folge des Admin-Workshops [6] beschrieben. Dieses Mal ist die zweite an der Reihe: der eigenständige Daemon. Er beansprucht zwar ununterbrochen Ressourcen, bietet dem Anwender dafür aber eine schnellere Reaktionszeit.

Ein Linux-System startet seine Serverdienste (und sonstige Systembestandteile) in der Regel über so genannte Init-Skripte. Diese Skripte werten ihre Kommandozeile aus: Mit dem Parameter »start« starten sie den Server-Dienst, mit »stop« beenden sie ihn. Ein »restart« sorgt für einen Neustart des Dienstes – das ist sinnvoll, um eine neue Konfiguration einlesen zu lassen.

Ein Beispiel für ein passendes Init-Skript steht unter [9] bereit. Kopieren Sie das File beispielsweise nach »/etc/init.d/proftpd«. Von dort aus erstellen Sie einen symbolischen Link in das eigentliche Startverzeichnis – auf einem typischen Linux-System sind dazu die Be-

**Wichtige FTP-Daemons**

Eine Vielzahl von FTP-Daemons ringt um die Gunst der Admins. Die folgende Liste führt vier etablierte und bekannte Vertreter dieser Servergattung auf.

**BSD-FTPD:** Der Urvater vieler Linux-FTP-Server war gleichzeitig Vorbild für zahlreiche andere Programmierprojekte. Auf kleinen Systemen leistet er heute noch gute Dienste und ist daher Bestandteil mancher Linux-Distribution. Komplexe Aufgaben oder intensive Arbeitsbelastung meistert er aber weniger gut.

**WU-FTPD:** Unter den derzeit noch benutzten Implementierungen gleicht der FTP-Daemon der Washington University [3] am ehesten dem BSD-FTPD. Der WU-FTPD ist sehr frei konfigurierbar und fügt sich trotzdem nahtlos in ein Unix-System ein. Eine Reihe in den letzten Jahren entdeckter Sicherheitslücken sollte mittlerweile behoben sein.

**NC-FTPD:** Dieser kommerzielle Server [2] ist nur für Privatanwender und Universitäten kostenlos erhältlich. Den Quellcode bekommt niemand zu Gesicht. Der Daemon erfreut sich dennoch einiger Beliebtheit, da er sich automatisch installiert. Zudem ist er selbst bei vielen gleichzeitigen Sessions sehr schnell.

**Pro-FTPD:** Wie NC-FTPD kann auch der in diesem Artikel vorgestellte Pro-FTPD [4] als eigenständiger Server laufen und somit Geschwindigkeitsvorteile gegenüber den Konkurrenten erzielen: Wenn er eine neue Verbindung annimmt, muss das Serversystem nicht erst einen neuen Prozess starten. Wahlweise lässt sich dieser Daemon aber auch aus »inetd« heraus aufrufen. Pro-FTPD ist Open Source, einfach zu konfigurieren und sinnvoll abzusichern.

fehle aus Listing 4 nötig. Je nach Distribution kann Zeile 4 statt »/etc/init.d/rc3.d« auch »/etc/rc3.d« lauten.
In einem Linux-System ist ein so genanntes Sequencer-Skript dafür zuständig, die einzelnen Dienste in der richtigen Reihenfolge zu starten. Die Reihenfolge ergibt sich aus dem Dateinamen: Das führende S bedeutet „Dienst starten“, danach folgt eine Nummer (Reihenfolge), die restlichen Buchstaben benennen den zu startenden Dienst.

## Fertig!

Starten Sie den FTP-Server als Root mit »/etc/init.d/proftpd start« – das ist nur einmal erforderlich, nach jedem Reboot wird ihn das Sequencer-Skript des Systems automatisch aufrufen.
Wenn alles ohne Probleme funktioniert hat, steht Ihnen nun ein eigener FTP-Server unter der URL [ftp://localhost] zur Verfügung. Prüfen Sie die Funktionalität vom eigenen Rechner aus, aber auch von anderen Systemen. So stellen Sie sicher, dass die korrekten Firewall-Regeln aktiv sind. Als Nächstes bevölkern Sie das Heimatverzeichnis »~ftp/pub« noch mit den öffentlich zugänglichen Dateien.
Für den Einsatz auf kritischen Systemen gibt es noch eine ganze Reihe Tipps, um das System besser zu schützen. Zu empfehlen ist besonders die ausführliche Dokumentation auf [4]. *(fjl)*

### Infos

[1] RFC 959: [ftp://ftp.rfc-editor.org/in-notes/rfc959.txt]
[2] NC-FTPD: [http://www.ncftpd.com/ncftpd/]
[3] WU-FTPD: [http://www.wu-ftpd.org]
[4] Pro-FTPD: [http://proftpd.linux.co.uk]
[5] Quellcode des Pro-FTPD: [ftp://ftp.proftpd.org/distrib/source/]
[6] Marc André Selig, „Finger-Server einrichten und nutzen“: Linux-Magazin 11/03, S. 58
[7] Frank Bernard, „Lebendes Relikt – Sicherheitsprobleme beim FTP-Protokoll“: Linux-Magazin 06/02, S. 54
[8] Fred Andresen, „Haftung für Homepage-Inhalte“: Linux-Magazin 12/02, S. 59
[9] Konfigurationsdatei und Init-Skript: [ftp://ftp.linux-magazin.de/pub/magazin/2004/03/Admin-Workshop/]

## Das FTP-Protokoll

Für Browser-Benutzer sieht eine FTP-Verbindung nicht anders aus als HTTP. Die Technik dahinter unterscheidet sich allerdings grundlegend.

### Benutzernamen rund um FTP

Jede FTP-Sitzung beginnt mit einer Anmeldung mit Benutzername und Passwort. Diese Informationen überträgt das Protokoll unverschlüsselt im Klartext. Es empfiehlt sich daher, den gewohnten Login nicht für FTP zu verwenden. Zudem wäre es unpraktisch, wenn der Admin jedem Gast erst einmal einen Account zuweisen müsste, nur damit er irgendwelche Dateien erhalten oder senden kann.
Per Konvention gibt es daher in praktisch jedem FTP-Server einen besonderen Gast-Account namens »anonymous« oder »ftp«. Als Passwort akzeptiert der Server jede beliebige E-Mail-Adresse, auch den String »*Username*@« interpretiert er im Sinne von „Username auf diesem Rechner“.

### Ports

Für jeden Netzwerkdienst sind ein oder mehrere Ports vorgesehen. Beispielsweise sagt man: HTTP verwendet den TCP-Port 80. Damit ist gemeint: Der Server hält den Port 80 ständig offen und wartet auf eingehende Verbindungen. Der Client baut eine Verbindung von einem beliebigen Port auf seinem Computer zum Port 80 auf dem Server auf. Viele Protokolle verwenden so eine einfache Zuordnung.
Bei FTP ist der Ablauf leider nicht ganz so einfach. Wie Abbildung 1 zeigt, unterscheidet das Protokoll zwischen einer Kontrollverbindung (für die Befehle) und einer Datenverbindung für die angeforderten Files. Die Datenverbindung ist auch für Verzeichnislistings und Ähnliches zuständig.
Die Kontrollverbindung entspricht dem klassischen Client-Server-Modell, sie verwendet den TCP-Port 21. Wie die Datenverbindung behandelt wird, liegt im Ermessen des Clients. Er kann eine aktive Verbindung anfordern. In diesem Modus öffnet der Client einen beliebigen Port auf dem eigenen System und teilt diese Portnummer – per Kontrollkanal – dem Server mit. Der Server baut daraufhin aktiv eine TCP-Verbindung von Port 20 ausgehend zu dem angegebenen Port auf dem Client auf. Die Richtung des Verbindungsaufbaus ist also gerade umgekehrt als erwartet! Auch dass der Quell- und nicht nicht Zielport festgelegt ist, weicht von den üblichen Abläufen ab.
FTP-Clients an der Kommandozeile verwenden meist den aktiven Modus. So genannte passive Verbindungen sind üblicherweise eine Domäne der Webbrowser. Mozilla, Opera oder auch der Internet Explorer fordern beim Server einen weiteren Port für die Datenverbindung an. Der Server teilt die Portnummer mit, daraufhin öffnet der Client eine zweite Verbindung von einem beliebigen eigenen Port zum angegebenen Port auf dem Server. Die Kontrollverbindung bleibt in jedem Fall parallel zur Datenverbindung bestehen.

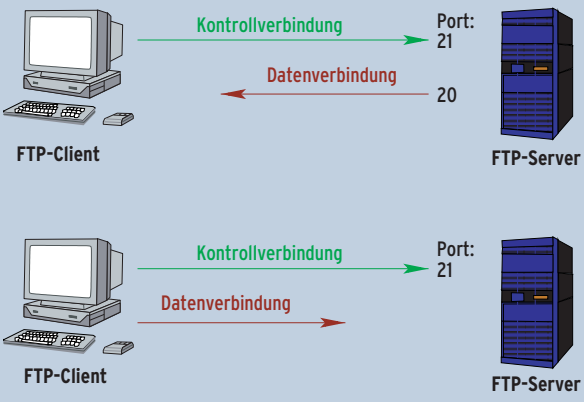


**Abbildung 1: Das FTP-Protokoll verwendet eine Kontroll- und eine oder mehrere Datenverbindungen. Im aktiven Modus (oben) öffnet der Server den Datenkanal, im passiven Modus (unten) der Client.**

### Firewall-Regeln beachten

Diese ungewöhnliche Architektur ist bei der Installation eines FTP-Servers zu berücksichtigen. Viele Distributionen richten nämlich ungefragt einen IP-Filter ein, der die zusätzlichen Ports schlicht sperrt. Lediglich die aktive Datenverbindung ist in üblichen Voreinstellungen erlaubt, da sie vom Server ausgeht. Die aktive Datenverbindung nützt aber nichts, wenn gar keine Kontrollverbindung zustande kommt. Daher sollten angehende FTP-Admins eventuell die Firewall-Konfiguration [7] aktualisieren und zusätzliche Filterregeln einrichten. Für einen FTP-Server auf einem 2.4er Kernel lauten diese:

```
iptables -A INPUT  -p tcp   ➥
  --sport 1024:  --dport 21   -j ACCEPT
iptables -A INPUT  -m state ➥
  --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state ➥
  --state ESTABLISHED,RELATED -j ACCEPT
```