

Ftpserver unter VNUML

von
Christian Oellig
Iwan Diel

Einführung

Bei der Suche nach einem geeigneten Ftp Server für unser Vnuml Projekt, sind wir auf eine Vielzahl potenzieller Kandidaten gestoßen. Doch der Ftp Server der für uns in Frage kam, sollte gewisse Kriterien erfüllen, die da wären:

- einfache Konfiguration und Bedienung
- geringe Größe
- umfangreiche Optionen.

Nach langem Hin und Her ist unsere Wahl auf Pure-Ftpd gefallen. Dieser Server erfüllt voll und ganz unsere Ansprüche, er ist klein, überschaubar was Konfiguration und Bedienung angeht und durch die Fülle an Funktionsanwendungen, für unsere Zwecke sehr gut geeignet. Insbesondere die eigentlich sehr einfache und (relativ) übersichtliche Konfiguration (via Schalter beim Aufruf) macht den Pure-Ftpd attraktiv.

Installation

Es gibt mehrere Möglichkeiten sich den Ftp Server runterzuladen. Entweder man bezieht das Programm direkt von der offiziellen Homepage <http://www.pureftpd.org/> oder benutzt apt-get. Wir haben uns für die zwei Variante entschieden.

Zunächst starten wir unser Vnuml Scenario

```
linux:/ # vnumlparser.pl -t xxxxxxxxxx.xml -vB
```

anschließend loggen wir uns mittels ssh auf die viltuelle Maschine ein *ssh uml1* z.B. Jetzt mit Hilfe von apt-get Pure-Ftpd runterladen auf die virtuelle Maschine, nicht vergessen zunächst nat und ipforwarding zu aktivieren, dies geht folgendermaßen

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE  
sysctl net.ipv4.conf.all.forwarding=1
```

die Befehle müssen global ausgeführt werden und nicht auf der virtullen Maschine selbst. Nun den Server runterladen

```
uml1:~# apt-get install pure-ftpd
```

Der Server ist nun auf der virtuelle Maschine installiert und kann benutzt werden.

Konfiguration und Start

Es gibt zwei verschiedene Möglichkeiten den Server zu starten, einmal mit Hilfe von inetd, wählen wir diese Art, wird der Server automatisch gestartet wenn das Vnuml Scenario hochfährt. Die jeweiligen Parameter, die der Ftp Server braucht um eine bestimmte Funktionalität zu erfüllen, werden in der */etc/inetd.conf* festgelegt.

Mit der zweiten Variante lassen wir den Server als daemon im Hintergrund laufen, d.h. wir starten ihn per Hand nachdem das Einloggen auf der virtuellen Maschine erfolgt ist.

In dieser Anleitung befassen wir uns nur mit der zweiten Methode, da es sich mit dieser viel flexibler arbeiten lässt.

Hier ein kleines Beispiel wie so eine Konfiguration des Servers aussehen könnte:

- Der FTP-Server soll als Daemon im Hintergrund starten **-B** und alle Benutzer in ihrem Home-Verzeichnis einsperren (chrooten)**-A**. Root ist allerdings vom chroot ausgenommen.
- Da bei FTP das Passwort im Klartext übertragen wird und es von jedem mit gelesen werden kann, verbieten wir es root sich per FTP einzuloggen. Mit **-u 1** legen wir fest, dass sich kein User mit einer UserID (uid) kleiner als 1 anmelden darf, root hat immer uid 0. Es ist ratsam auch Serverdiensten, deren uid normalerweise unter 100 liegt, den Login zu untersagen **-u 100**.
- Desweiteren soll der Server nur IPv4 benutzen **-4**, da wir noch kein IPv6 von unserem Provider bekommen. IPv6 kann mit **-6** aktivieren werden.
- Außerdem wollen wir auch Clients unterstützen, die sich nicht so ganz ans Protokoll halten. **-b** für "broken client support".
- Maximal sollen sich 20 Clients anmelden können **-c 20**,
- welche nach 5 Minuten Idle (Untätigkeit) getrennt werden sollen **-I 5**.
- Pro Client/IP lassen wir 8 Verbindungen zu **-C 8**.
- Damit der Server nicht eines Tages stehen bleibt, weil die Festplatten voll sind, vergeben wir ein globales Speicherlimit, sodass Pure-Ftpd die Festplatten maximal zu 90% füllen kann **-k 90**.
- Als letztes richten wir es noch so ein, dass ein Upload erst verfügbar ist, wenn er erfolgreich beendet wurde **-O**. So wird z.B. die neue Version der Webseite erst verfügbar wenn diese komplett auf dem Server vorhanden ist.

Als ersten Test können wir jetzt schon den FTP-Server mit unserer Grundkonfiguration starten:

```
uml1:~> pure-ftpd -B -A -4 -k 90 \  
-u 100 -c 20 -C 8 -I 5 -b -O
```

Pure-Ftpd ist jetzt auf allen Netzwerk-Interfaces verfügbar und wir können uns mit unseren normalen Benutzerdaten einloggen. Dies war nur ein kleines Beispiel um zu zeigen wie vielfältig die Funktionalität dieses Ftp Servers ist. Eine ausführliche Beschreibung aller Schalter findet man, wenn man den Befehl

```
uml1:~/# man pure-ftpd
```

eingibt.

Um den Ftp Server zu beenden muss der Prozess getötet werden. Mit ps -e wird uns die Nummer des Servers gezeigt wie hier z.B.

```
275 vc/0    00:00:00 getty  
276 vc/1    00:00:00 getty  
277 vc/2    00:00:00 getty  
278 tts/0   00:00:00 getty  
280 ?       00:00:00 sshd  
283 pts/0   00:00:00 bash  
345 ?       00:00:00 pure-ftpd  
531 pts/0   00:00:00 ps
```

Wir sehen das der Server die Nummer 345 hat, mit *kill 345* beenden wir den Server.

Virtual-Users

Ein weiteres Feature, das die Arbeit mit Pure-Ftpd erleichtert, ist die Möglichkeit auf einfachem Wege Virtual-Users anzulegen. Der Einsatz von Virtual-Users bietet dem Administrator einen einfachen Weg, viele User mit individuellen Einstellungen zu verwalten, ohne für jeden User einen eigenen System-Account anlegen zu müssen. Außerdem ist es möglich, pro User eine eigene UP/Download Ratio einzustellen, Logins nur von bestimmten IPs aus zuzulassen oder die Bandbreite zu limitieren. Mit System-Accounts sind diese individuellen Einstellungen nicht möglich.

Zum Start des Server benutzen wir nun die selben Schalter wie im obigen Beispiel und fügen weitere hinzu

- Zum einen **-l**, um Pure-Ftpd die Datenbank mit den Benutzerdaten anzugeben.
- Außerdem noch **-j**, wodurch die Home-Verzeichnisse beim ersten Login automatisch angelegt werden.
- und mit **-E** verbieten wir Anonymous Ftp

Der Aufruf sieht dann folgendermaßen aus (Achtung: immer darauf achten ob der Pfad zur Datenbank richtig ist)

```
uml1:/# pure-ftpd -B -A -4 -k 90 -u 100 -c 20 -C 8 -I 5 -b -E -j -l puredb:/etc/pureftpd.pdb
```

Nun läuft der Server mit der dran gehängten Datenbank, jetzt können wir Virtual-Users einrichten. Um Virtual-Users benutzen zu können, benötigt man mindestens einen System-Account. Als Benutzernamen verwenden wir *ftpuser* mit der Gruppe *ftpgroup*.

```
uml1:/# groupadd ftpgroup
uml1:/# useradd -g ftpgroup -d /dev/null -s /etc ftpusers
```

Als nächstes weisen wir dem System-Account ein Home Verzeichnis zu, dies haben wir zwar bereits gemacht, aber um sicher zu stellen, dass das Verzeichnis über ausreichend Speicherkapazität verfügt wird ein Anderes erstellt und dem System-User zugeteilt.

```
uml1:/# mkdir /home/ftpusers
uml1:/# chown ftpusers:ftpgroup /home/ftpusers
```

Nachdem die Rahmenbedingungen geschaffen sind, können wir jetzt den ersten User anlegen. Dazu benutzen wir *pure-pw* mit der Option *useradd*.

```
uml1:/# pure-pw useradd testuser -u ftpusers -d /home/ftpusers/testuser
```

Als Passwort verwenden wir *xxxx*.

Pure-pw speichert die Einstellungen in */etc/pureftpd.passwd*. Um auf die Daten später zugreifen zu können müssen die Einstellungen erst verfügbar gemacht werden, dies geschieht über den Befehl

```
uml1:/# pure-pw mkdb
```

Nach diesem Aufruf ist der User in der Datenbank registriert.

Virtual-User verwalten

Der Befehl *pure-pw* übernimmt auch die Verwaltung von Virtual-User Accounts und zwar mit den Optionen *usermod*, *userdel*, *passwd*, *list* und *show*, die selbst erklärend sein dürften. So benutzen wir z.B.

```
uml1:~/# pure-pw list
testuser      /home/ftpusers/testuser/./
```

und sehen, dass unser User erfolgreich erstellt wurde. Desweiteren erkennen wir, dass der User in seinem Home-Verzeichnis eingesperrt (chrooted) ist, dies wird gekennzeichnet mit *./* am Ende der Ausgabe.

Mit dem Befehl *pure-pw show testuser* bekommt man alle Details zu dem Account *testuser*

```
uml1:~/# pure-pw show testuser

Login           : testuser
Password        : $1$XfNnr7g0$rrUeKfAHaB8GyoW15tj7a/
UID             : 1002 (ftpusers)
GID             : 1002 (ftpgroup)
Directory       : /home/ftpusers/testuser/./
Full name       :
Download bandwidth : 0 Kb (unlimited)
Upload bandwidth : 0 Kb (unlimited)
Max files       : 0 (unlimited)
Max size        : 0 Mb (unlimited)
Ratio           : 0:0 (unlimited:unlimited)
Allowed local IPs :
Denied local IPs :
Allowed client IPs :
Denied client Ips :
Time restrictions : 0000-0000 (unlimited)
Max sim sessions : 0 (unlimited)
```

Wie wir an der Ausgabe schön sehen können hat dieser User keinerlei Einschränkungen. Hier kommt die Option *usermod* ins Spiel. Damit können wir User spezifisch Limits erteilen.

Wir können nun für unseren User *testuser* Limits setzen wie z.B.:

- das er 100 mb Speicherplatz zur Verfügung hat **-N 100**
- das die downloadrate 30 Kb/s **-t 30**
- und die uploadrate 50 Kb/s **-T 50** beträgt
- wir erlauben ihm max 10 Session zu öffnen **-y 10**
- und zu guter letzt verbieten wir dem User sich zwischen 5 und 6 Uhr morgens einzuloggen **-z 0500-0600**

Der Aufruf sieht dann folgendermaßen aus

```
uml1:/# pure-pw usermod testuser -N 100 -t 30 -T 50 -y 10 -z 0500-0600
```

Erneut `pure-pw show testuser` eingeben und die Limits für diesen User werden sichtbar

```
Login           : testuser
Password        : $1$XfNnr7g0$rrUeKfAHaB8GyoW15tj7a/
UID             : 1002 (ftpusers)
GID             : 1002 (ftpgroup)
Directory       : /home/ftpusers/testuser/./
Full name       :
Download bandwidth : 30 Kb (enabled)
Upload  bandwidth : 50 Kb (enabled)
Max files       : 0 (unlimited)
Max size        : 100 Mb (enabled)
Ratio           : 0:0 (unlimited:unlimited)
Allowed local IPs :
Denied local IPs :
Allowed client IPs :
Denied client IPs :
Time restrictions : 0500-0600 (enabled)
Max sim sessions : 10 (enabled)
```

Weitere User spezifische Parameter sind in der Manuel aufgeführt.

Einloggen auf dem Ftp Server

Zum Einloggen öffnen wir ein weiteres Konsolenfenster, und rufen mit `ftp` den ftp-Client auf. Das Einloggen erfolgt über die IP-Adresse der virtuellen Maschine auf dem der Ftp Server läuft, in unserem Falle wäre das

```
ftp> open 192.168.1.146
Connected to 192.168.1.146.
220----- Herzlich willkommen auf Pure-FTPd -----
220-Sie sind Benutzer 1 von 20 erlaubten.
220-Lokale Zeit: 00:34. Serverport: 21.
220-Dies ist ein privates System - Keine anonyme Anmeldung m lich.
220 Sie werden nach 5 Minuten Inaktivit getrennt.
Name (192.168.1.146:Linux): testuser
331 Benutzer testuser OK. Pa ort erforderlich.
Password:
230-Benutzer testuser hat Gruppenzugang zu: 1002
230 OK. Aktuelles Verzeichnis ist /.
Remote system type is UNIX.
Using binary mode to transfer files.
```

Die Anmeldung am Server war erfolgreich. Um den Überblick über die User zu behalten kann man mit `pure-ftpwho` genau sehen wer online ist und wer was macht. Das sieht dann in etwa so aus.

uml1:/# pure-ftpwho

PID	Login	For/Spd	What	File/IP
632	testuser	00:00	IDLE	
"	"	"	"	-> 192.168.1.145

Quellen

- [1] <http://www.pureftpd.org>
- [2] <http://www.pureftpd.org/documentation.shtml>