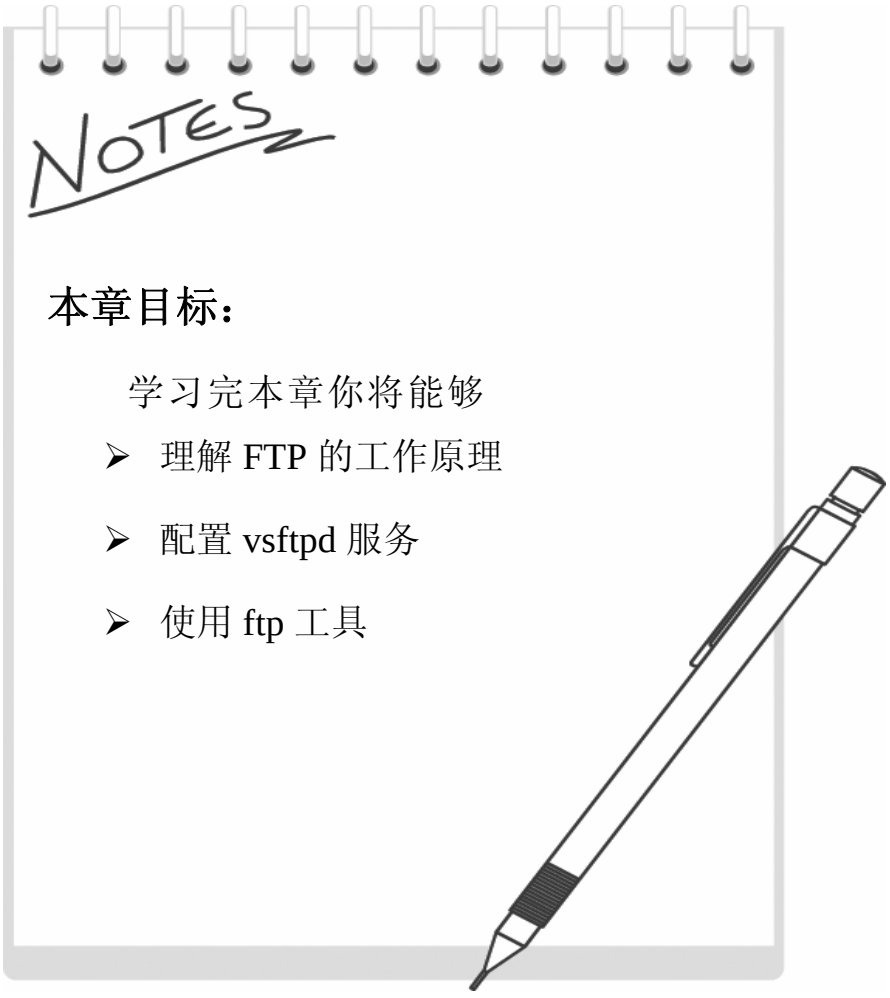
第七章 FTP 服务器的配置

FTP 的出现是为了解决文件和软件传输的问题，从而方便地从其它计算机系统获得资源。FTP 已经成为 Internet 上文件共享的一个标准，FTP 服务器中的文件按目录结构进行组织，用户通过网络与服务器的连接。查看或下载需要的文件。本章介绍有关 FTP 服务器方面的知识。



本章目标：

学习完本章你将能够

- 理解 FTP 的工作原理
- 配置 vsftpd 服务
- 使用 ftp 工具

1. FTP 协议分析

FTP 是 File Transfer Protocol（文件传输协议）的缩写，它采用两个 TCP 连接来传输一个文件：

（1）当 FTP 服务器启动后，服务器以被动方式打开众所周知的用于 FTP 的端口（21），等待客户的连接。客户则以主动方式打开 TCP 端口 21，来建立连接，该连接用于控制客户端与服务器端的命令传输；该连接将命令从客户传给服务器，并传回服务器的应答。由于命令通常是由用户键入的，所以 IP 对控制连接的服务类型就是“最大限度地减小迟延”。

（2）当一个文件在客户与服务器之间传输时，就创建一个数据连接，该连接使用服务器端的 20 端口。由于该连接用于传输目的，所以 IP 对数据连接的服务特点就是“最大限度提高吞吐量”。

图 7-1 描述了客户与服务器以及它们之间的连接情况。

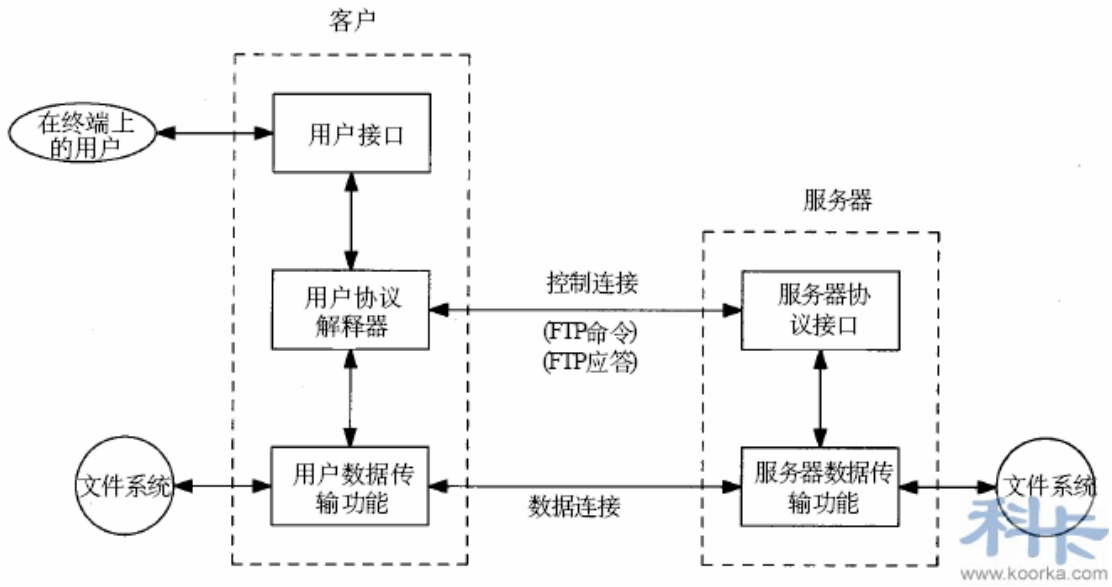


图 7-1

从图中可以看出，交互式用户通常不处理在控制连接中转换的命令和应答。这些细节均由两个协议解释器来完成。标有“用户接口”的方框功能是按用户所需提供各种交互界面（全屏幕菜单选择，逐行输入命令，等等），并把它们转换成在控制连接上发送的 FTP 命令。

类似地，从控制连接上传回的服务器应答也被转换成用户所需的交互格式。

从图中还可以看出，正是这两个协议解释器根据需要激活文件传送功能。

1.1 数据表示

FTP 协议规范提供了控制文件传送与存储的多种选择。在以下四个方面中每一个方面都必须作出一个选择。

1. 文件类型

(a) ASCII 码文件类型（默认选择）：文本文件以 NVT ASCII 码形式在数据连接中传输。这要求发方将本地文本文件转换成 NVT ASCII 码形式，而接收方则将 NVT ASCII 码再还原成本地文本文件。

其中，用 NVT ASCII 码传输的每行都带有一个回车，而后是一个换行。这意味着收方必须扫描每个字节，查找 C R、L F 对。

(b) EBCDIC 文件类型：该文本文件传输方式要求两端都是 EBCDIC 系统。

(c) 图像文件类型（也称为二进制文件类型）：数据发送呈现为一个连续的比特流。通常用于传输二进制文件。

(d) 本地文件类型：该方式在具有不同字节大小的主机间传输二进制文件。每一字节的比特数由发方规定。对使用 8 bit 字节的系统来说，本地文件以 8 bit 字节传输就等同于图像文件传输。

2. 格式控制

该选项只对 ASCII 和 EBCDIC 文件类型有效。

(a) 非打印（默认选择）：文件中不含有垂直格式信息。

(b) 远程登录格式：控制文件含有向打印机解释的远程登录垂直格式控制。

(c) Fortran 回车控制：每行首字符是 Fortran 格式控制符。

3. 结构

（a） 文件结构（默认选择）：文件被认为是一个连续的字节流。不存在内部的文件结构。

（b） 记录结构该结构：只用于文本文件（ASCII 或 EBCDIC）。

（c） 页结构：每页都带有页号发送，以便收方能随机地存储各页。

4. 传输方式

它规定文件在数据连接中如何传输。

（a） 流方式（默认选择）：文件以字节流的形式传输。对于文件结构，发方在文件尾提示关闭数据连接。对于记录结构，有专用的两字节序列码标志记录结束和文件结束。

（b） 块方式文件以一系列块来传输，每块前面都带有一个或多个首部字节。

（c） 压缩方式一个简单的全长编码压缩方法，压缩连续出现的相同字节。在文本文件中常用来压缩空白串，在二进制文件中常用来压缩 0 字节（这种方式很少使用，也不受支持。现在有一些更好的文件压缩方法来支持 FTP）。

通常由 Linux/Unix 实现的 FTP 客户和服务器把我们的选择限制如下：

- 类型：ASCII 或图像。
- 格式控制：只允许非打印。
- 结构：只允许文件结构。
- 传输方式：只允许流方式。

这就限制我们只能取一、两种方式： ASCII 或图像（二进制）。

1.2 FTP 命令

下面是使用 ftp 命令登录 FTP 服务器的过程：

```
[root@koorka ~]# ftp ftp.kernel.org
Connected to zeus-pub.kernel.org.
220 Welcome to ftp.kernel.org.
530 Please login with USER and PASS.
530 Please login with USER and PASS.
KERBEROS_V4 rejected as an authentication type
Name (ftp.kernel.org:root): ftp
331 Please specify the password.
Password:
……
230-
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

下面是使用 telnet 命令登录 FTP 服务器的过程：

```
[root@koorka ~]# telnet ftp.kernel.org 21
Trying 204.152.191.37...
Connected to ftp.kernel.org (204.152.191.37).
Escape character is '^]'.
220 Welcome to ftp.kernel.org.
user ftp
331 Please specify the password.
```

pass b@

230- Welcome to the

……

230-

230 Login successful.

从上面两个例子可以看出，当客户端与 FTP 服务器交互时需要发送一些 FTP 命令。而使用 FTP 工具时这些命令由 FTP 工具来发送，使用 Telnet 时，命令由用户来法送。

命令和应答在客户和服务器的控制连接上以 NVT ASCII 码形式传送。这就要求在每行结尾都要返回 CR、LF 对（也就是每个命令或每个应答）。

这些命令都是 3 或 4 个字节的大写 A S C I I 字符，其中一些带选项参数。从客户向服务器发送的 FTP 命令超过 30 种。图 7-2 给出了一些常用命令：

命　令	说　明
ABOR	放弃先前的FTP命令和数据传输
LIST *filelist*	列表显示文件或目录
PASS *password*	服务器上的口令
PORT *n1,n2,n3,n4,n5,n6*	客户端IP地址（*n1.n2.n3.n4*）和端口（*n5*×256+*n6*）
QUIT	从服务器注销
RETR *filename*	检索（取）一个文件
STOR *filename*	存储（放）一个文件
SYST	服务器返回系统类型
TYPE *type*	说明文件类型：A表示ASCII码，I表示图像
USER *username*	服务器上用户名

图 7-2

1.3 FTP 应答

从前面的例子可以看到，当客户端发送一个 FTP 命令时，服务器端总会有应答，应答都是 ASCII 码形式的 3 位数字，并跟有报文选项。其原因是软件系统需要根据数字代码来决定如何应答，而选项串是面向人工处理的。由于客户通常都要输出数字应答和报文串，一个可交互的用户可以通过阅读报文串（而不必记忆所有数字回答代码的含义）来确定应答的含义。

应答 3 位码中每一位数字都有不同的含义。图 7-3 给出了应答代码第 1 位和第 2 位的含义：

应答	说　　明
1yz	肯定预备应答。它仅仅是在发送另一个命令前期待另一个应答时启动
2yz	肯定完成应答。一个新命令可以发送
3yz	肯定中介应答。该命令已被接受，但另一个命令必须被发送
4yz	暂态否定完成应答。请求的动作没有发生，但差错状态是暂时的，所以命令可以过后再发
5yz	永久性否定完成应答。命令不被接受，并且不再重试
x0z	语法错误
x1z	信息
x2z	连接。应答指控制或数据连接
x3z	鉴别和记帐。应答用于注册或记帐命令
x4z	未指明
x5z	文件系统状态

图 7-3

第 3 位数字给出差错报文的附加含义。例如，这里是一些典型的应答，都带有一个可能的报文串。

- 125 数据连接已经打开；传输开始。
- 200 就绪命令。
- 214 帮助报文（面向用户）。
- 331 用户名就绪，要求输入口令。
- 425 不能打开数据连接。
- 452 错写文件。
- 500 语法错误（未认可的命令）。
- 501 语法错误（无效参数）。
- 502 未实现的 MODE (方式命令)类型。

通常每个 FTP 命令都产生一行回答。例如，QUIT 命令可以产生如下应答：

221 Goodbye.

如果需要产生一条多行应答，第 1 行在 3 位数字应答代码之后包含一个连字号，而不是空格，最后一行包含相同的 3 位数字应答代码，后跟一个空格符。例如，使用 Telnet 登录之后，HELP 命令可以产生如下应答：

```
……
230 Login successful.
help
214-The following commands are recognized.
 ABOR ACCT ALLO APPE CDUP CWD  DELE EPRT EPSV FEAT HELP LIST MDTM MKD MODE NLST NOOP OPTS PASS PASV PORT PWD  QUIT REIN REST RETR RMD  RNFR RNTO SITE SIZE SMNT STAT STOR STOU STRU SYST TYPE USER XCUP XCWD XMKD XPWD XRMD
214 Help OK.
```

该命令列出的就是 FTP 的命令。

1.4 连接管理

数据连接有以下三大用途：

1) 从客户向服务器发送一个文件。

2) 从服务器向客户发送一个文件。

3) 从服务器向客户发送文件或目录列表。

FTP 服务器把文件列表从数据连接上发回，而不是控制连接上的多行应答。这就避免了行的有限性对目录大小的限制，而且更易于客户将目录列表以文件形式保存，而不是把列表显示在终端上。

控制连接一直保持到客户-服务器连接的全过程，但数据连接可以根据需要随时来，随时走。那么需要怎样为数据连接选端口号，以及谁来负责主动打开和被动打开？

首先，前面说过通用传输方式（Linux/Unix 环境下唯一的传输方式）是流方式，并且文件结尾是以关闭数据连接为标志。这意味着对每一个文件传输或目录列表来说都要建立一个全新的数据连接。其一般过程如下：

1) 由于是客户发出命令要求建立数据连接，所以数据连接是在客户的控制下建立的。

2) 客户通常在客户端主机上为所在数据连接端选择一个临时端口号。客户从该端口发布一个被动的打开。

3) 客户使用 PORT 命令从控制连接上把端口号发向服务器。

4) 服务器在控制连接上接收端口号，并向客户端主机上的端口发布一个主动的打开。服务器的数据连接端一直使用端口 20。

图 7-4 给出了第 3 步执行时的连接状态。假设客户用于控制连接的临时端口是 1173，客户用于数据连接的临时端口是 1174。客户发出的命令是 PORT 命令，其参数是 6 个 ASCII 中的十进制数字，它们之间由逗点隔开。前面 4 个数字指明客户上的 IP 地址，服务器将向它发出主动打开（本例中是 140.252.13.34），而后两位指明 16bit 端口地址。由于 16bit 端口地址是从这两个数字中得来，所以其值在本例中就是 4×256+150 = 1174。

图 7-5 给出了服务器向客户所在数据连接端发布主动打开时的连接状态。服务器的端点是端口 20。

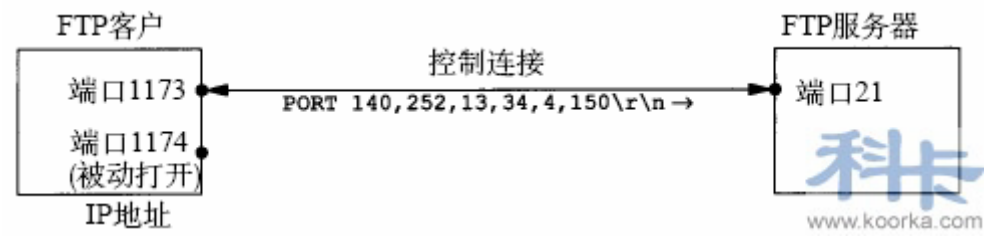


图 7-4

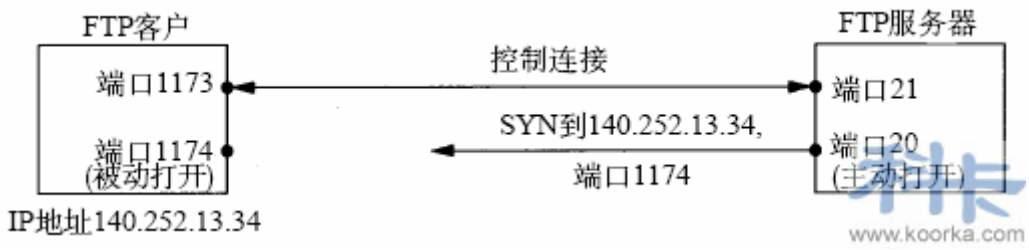


图 7-5

服务器总是执行数据连接的主动打开。通常服务器也执行数据连接的主动关闭，除非当客户向服务器发送流形式的文件时，需要客户来关闭连接（它给服务器一个文件结束的通知）。

客户也有可能不发出 PORT 命令，而由服务器向正被客户使用的同一个端口号发出主动打开，来结束控制连接。这是可行的，因为服务器面向这两个连接的端口号是不同的：一个是 20，另一个是 21。不过，下节我们将看到为什么现有实现通常不这样做。

2. vsFTPD 的配置

在 Linux 系统下，vsFTPd 是一款在 Linux 发行版中最受推崇的 FTP 服务器程序；特点是小巧轻快，安全易用。

目前在开源操作系统中常用的 FTPD 软件除 VSFTPD 外，主要有 ProFTPD、PureFTPd 和 wu-ftpd 等；至于哪个 FTP 服务器软件更好，哪个是你最熟悉的，哪个就是最好的。

2.1 安装 vsFTPD

在 RedHat Enterprise Linux 系统中，可以使用 rpm 包。如果要使用 RPM 包，只需要查询 vsftpd 是否安装即可：

```
[root@koorka ~]# rpm -q vsftpd
vsftpd-2.0.1-5.EL4.3
```

为了使大家在任何发行版本上都能随意配置，这里介绍从源代码来安装：

（1）获取源代码，并解压：

```
#wget -P /usr/src ftp://vsftpd.beasts.org/users/cevans/vsftpd-2.0.5.tar.gz
[root@koorka src]#cd /usr/src
[root@koorka src]# tar -zxvf vsftpd-2.0.5.tar.gz
```

（2）编译安装：

```
[root@koorka src]#cd vsftpd-2.0.5
[root@koorka vsftpd-2.0.5]# make
[root@koorka vsftpd-2.0.5]#make install
```

（3）确认 nobody 用户是否存在：

```
[root@koorka vsftpd-2.0.5]# id nobody
uid=99(nobody) gid=99(nobody) groups=99(nobody)
```

如果不存在，创建该用户：

```
[root@koorka vsftpd-2.0.5]#useradd –M nobody
```

（4）确认/usr/share/empty 目录是否存在：

```
[root@koorka vsftpd-2.0.5]# file /usr/share/empty
/usr/share/empty: directory
```

如果不存在，创建该目录：

```
[root@koorka vsftpd-2.0.5]#mkdir -p /usr/share/empty
```

（5）确认 ftp 用户是否存在，该用户将用于 ftp 的匿名访问：

```
[root@koorka vsftpd-2.0.5]# id ftp
uid=14(ftp) gid=50(ftp) groups=50(ftp)
```

如果不存在，创建用户：

```
[root@koorka vsftpd-2.0.5]#useradd –d /var/ftp –s /sbin/nologin ftp
[root@koorka vsftpd-2.0.5]#chown root.root /var/ftp
[root@koorka vsftpd-2.0.5]## chmod og-w /var/ftp
```

其主目录将用作匿名访问的“/”目录。

（6）安装软件：

```
[root@koorka vsftpd-2.0.5]#make install
```

（7）测试

为了测试安装是否成功，编辑配置文件/etc/vsftpd.conf 文件，文件内只需要如下两行：

```
listen=YES
anonymous_enable=YES
```

然后启动 vsftpd 服务：

```
[root@koorka vsftpd-2.0.5]# /usr/local/sbin/vsftpd /etc/vsftpd.conf &
```

然后使用 ftp 命令登录。如果成功，表示服务器安装已经成功。

（8）停止 vsftpd 服务

```
[root@koorka vsftpd-2.0.5]# pkill vsftpd
```

2.2 配置匿名 FTP 服务器

所谓匿名 ftp，就是用户不需要真实的用户名和密码就可以访问 FTP 服务器。是很多 FTP 下载服务器所采用的方式。

2.2.1 匿名 ftp 用户和用户组

当我们访问各大 FTP 下载服务器时，可能从来不去想我们以什么身份登录的，如果他的 FTP 允许匿名登录的话；例如我们在浏览器上打入；

ftp://ftp.kernel.org 或 ftp://ftp:ftp@ftp.kernel.org

我们会发现上面的两行最终都能访问，而且显示的结果也完全一样，最终都跳到 ftp://ftp.kernel.org 地址；那我们访问这个 FTP 时，是不是有用户和密码呢？是的，也是需要的，只是在服务器端允许匿名访问，而匿名访问的用户名和密码都是 ftp，只是我们因为匿名访问，没有感觉到他有用户名和密码罢了。第二个地址就是以 ftp 用户，密码也是 ftp 来访问 ftp://ftp.kernel.org；

如果以 ftp 命令连接 ftp.kernel.org 时，我们会发现需要输入用户 ftp，密码以@开头的任意字符才能访问；

在 FTP 服务器中，匿名用户的用户名是 ftp，密码任意；这个用户可以在您的操作系统中的 /etc/passwd 中能找得到；可能有类似下面的一行；

ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin

2.2.2 配置和测试

（1）配置 vsftpd.conf

典型的匿名访问的 ftp 配置文件/etc/vsftpd.conf（如果是使用 rpm 包安装，应该是/etc/vsftpd/vsftpd.conf）的内容如下：

```
listen=YES   #采用独立服务模式提供服务
anonymous_enable=YES #允许匿名访问
anon_upload_enable=YES #允许上传文件
write_enable=YES
xferlog_enable=YES   #启用日志记录
```

```
vsftpd_log_file=/var/log/vsftpd.log #日志文件的存储位置
anon_max_rate=15000 #15K     #限定最大上传下载带宽为 15K
max_clients=100     #最多能同时有 100 个用户连接
max_per_ip=2   #每个 IP 地址最多允许两个连接
banner_file=/etc/vsftpd/banner_file #欢迎信息
ftp_username=ftp   #使用匿名用户登录时，该用户的主目录就是 ftp 的“/”
```

（2）创建上传目录

```
[root@koorka ~]# mkdir /var/ftp/uploads
[root@koorka ~]# chown ftp.ftp /var/ftp/uploads
```

（3）创建欢迎信息文件/etc/vsftpd/banner_file，文件的内容如下：

```
Welcome to koorka.com!
```

（4）启动 FTP 服务

```
root@koorka ~]#/usr/local/sbin/vsftpd /etc/vsftpd.conf &
```

（5）测试。

```
[root@koorka ~]# ftp localhost
Name (localhost:root): ftp
Password:
ftp>cd uploads
ftp>put install.log
```

2.3 配置实名 FTP

实名 ftp 通常用于 Web 网站的页面上传。为了统一管理，我们把相关的配置文件都存储在/etc/vsftpd/目录下（包括 vsftpd.conf，在启动服务时指定配置文件路径即可）。

（1）创建配置文件/etc/vsftpd/vsftpd.conf，典型的 vsftpd.conf 文件如下：

```
listen=YES
pam_service_name=vsftpd      #指定用户的认证配置文件
anonymous_enable=NO          #关闭匿名 FTP 功能
local_enable=YES
chroot_list_enable=YES       #当用户登录时，只允许用户在主目录内活动
chroot_list_file=/etc/vsftpd/chroot_list      #受限定用户的用户列表文件
write_enable=YES             #允许用户上传文件
local_umask=022              #local_umask 决定了用户上传文件的默认权限
xferlog_enable=YES
vsftpd_log_file=/var/log/vsftpd.log
banner_file=/etc/vsftpd/banner_file
```

（2）创建 vsftpd 的用户认证配置文件/etc/pam.d/vsftpd，文件内容如下：

```
#%PAM-1.0
auth       required     pam_listfile.so item=user sense=deny file=/etc/vsftpd/ftpusers onerr=succeed
auth       required     pam_stack.so service=system-auth
auth       required     pam_shells.so
account    required     pam_stack.so service=system-auth
session    required     pam_stack.so service=system-auth
session    required     pam_loginuid.so
```

其中 file 参数指定的是拒绝访问该服务的用户列表。

（3）创建拒绝访问 FTP 服务的用户列表，典型情况下，ftpuser（文件名由步骤 2 中的 file 参数指定）的内容如下：

```
# Users that are not allowed to login via ftp
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
news
uucp
operator
games
nobody
```

文件中列出的是不允许访问该 ftp 服务器的用户，换句话讲，你不想让某个用户使用该 FTP 服务器，就可以将该用户列入该文件中。

（4）创建受限定活动范围的用户列表/etc/vsftpd/chroot_list。该文件中一行一个用户，需要限制活动范围的用户就添加到此文件中。

如果系统用户较多，而允许使用 FTP 服务的用户不多，可以在/etc/vsftpd/vsftpd.conf 文件中添加如下两行：

```
userlist_enable=YES          #启用用户列表
userlist_deny=NO
userlist_file=/etc/vsftpd/user_list  #允许使用 FTP 的用户列表文件
```

然后将允许访问 FTP 服务的用户添加于/etc/vsftpd/user_list 文件中。

2.4 vsftpd.conf 的常用参数

下面是配置文件的常用参数及其说明：

2.4.1 布尔选项

下边是布尔选项的列表。 一个布尔选项的值可以被设为 YES 或 NO

allow_anon_ssl

只有在 ssl_enable 被激活时才有用。如果设为 YES， 匿名用户将被允许使用安全的 SSL 联接。默认： NO

anon_mkdir_write_enable

如果设为 YES， 匿名用户将允许在某些情况下创建目录。这需要激活 write_enable 选项， 并且匿名 ftp 用户需要对父目录有写权限。默认：NO

anon_other_write_enable

如果设为 YES， 匿名用户将拥有除 上载， 和创建目录 外更多的权限， 比如 删除和重命名。通常不建议这么做，但完整的配置文件是包括这一选项的。默认：NO

anon_upload_enable

如果设为 YES，匿名用户在某些情况下允许上载文件。这需要将 write_enable 选项激活，并且匿名用户应当对对应目录有写权限。默认：NO

anon_world_readable_only

启用时，将只允许匿名用户下载具有全球读权限的文件。这将意味着 ftp 用户可以拥有自己的文件，特别是前边提到的上载的文件。默认：YES

anonymous_enable

用于控制是否允许匿名用户登录。 如果激活，ftp 和 anonymous 都将被视为匿名用户登录。默认：YES

ascii_download_enable

如果被激活，下载时将使用 ASCII 模式进行数据传输。默认：NO

ascii_upload_enable

如果被激活，上载时将使用 ASCII 模式进行数据传输。默认：NO

async_abor_enable

如果被激活，一个特别的 FTP 命令"async ABOR"将被激活。只有某些 FTP 客户端需要使用这一特性。另外，这个特性并不是很好控制，因此默认没有启用。不幸的是，如果没有启用这个特性，某些 FTP 客户端在取消一个传输时就会挂起，因此，您可能希望启用它。默认：NO

background

如果被激活，并且 vsftpd 以"listen"模式启动，vsftpd 将会 background 监听进程。即 control will immediately be returned to the shell which launched vsftpd。默认：NO

check_shell

注意!这个选项只对构建时加入 non-PAM 参数的 vsftpd 有效。如果令其失效，vsftpd 将不会检查有效用户的用于本地登录的/etc/shells。默认：YES

chmod_enable

如果被激活，将允许使用 SITE CHMOD 命令。注意!这只对本地用户有效。匿名用户从不允许使用 SITE CHMOD。默认：YES

chown_uploads

如果被激活，所有匿名上载的文件的宿主将会调整为 chown_username 中指定的用户。这样就便于管理，特别是从安全的角度考虑。默认：NO

chroot_list_enable

如果被激活，您需要提供一个需要将其限制于其家目录中的本地用户列表。如果将 chroot_local_user 设为 YES 则意义稍有不同。在此情况下，此列表变成不需将用户限制于其家目录的用户的列表。默认情况下，这个列表文件是/etc/vsftpd。chroot_list，但可以通过 chroot_list_file 选项来设定。默认：NO

chroot_local_user

如果设为 YES，本地用户，在登录后将(默认)被限制在其家目录中。警告：此选项有安全隐患，特别是在用户拥有上载权限，或可以 shell 访问的时候。如果您不清楚后果，请不要启用它。注意，这些安全隐患并不是 vsftpd 所特有的。所有的提供将本地用户进行目录限制的 FTP 守护进程有存在这种隐患。默认：NO

connect_from_port_20

用于控制在服务器端，是否使用端口 20(ftp-data)进行数据联接。基于安全的考虑，有些客户端需要这样做。相反，禁用这个选项，可以使 vsftpd 以较少特权运行。默认：NO。

deny_email_enable

如果激活，您应当提供一个禁止匿名用户用做密码的 e-mail 地址列表。默认情

况下，这个列表文件为/etc/vsftpd。banned_emails，当然，您可以通过 banned_email_file 选项指定。默认：NO

dirlist_enable

如果设为 NO，所有的目录列取命令都将被禁止。默认：YES

dirmessage_enable

如果启用，当用户首次进入一个新目录时，FTP 服务器将会显示欢迎信息。默认情况下，是扫描目录下的。message 文件获取的，当然，您也可以通过 message_file 选项设定。默认：NO

download_enable

如果设为 NO，所有的下载请求都将被拒绝。默认：YES

dual_log_enable

如果启用，将生成两个相似的日志文件，默认在/var/log/xferlog 和/var/log/vsftpd。log 目录下。前者是 wu-ftpd 类型的传输日志，可以用于标准工具分析。后者是 vsftpd 自己类型的日志。默认：NO

force_dot_files

如果激活，以“.”开始的文件和目录在目录列取的时候将会被显示，即使客户端没有使用"a"标识。这不包括"."和".."目录。默认：NO

force_local_data_ssl

只有在 ssl_enable 被激活时才能使用。如果被激活，则所有的非匿名用户登录时都被强制使用安全 SSL 联接来传送接收数据。默认：YES

force_local_logins_ssl

只有在 ssl_enable 被激活时才能使用。如果被激活，则所有的非匿名用户登录时都被强制使用安全 SSL 联接来传送密码。默认：YES

guest_enable

如果启用，所有非匿名用户都将以"guest"身份登录。guest 通过 guest_username 设定，来映射到一个指定用户。默认：NO

hide_ids

如果启用，所有目录中的用户和组信息列取时都将显示为"ftp"。默认：NO

listen

如果启用，vsftpd 将以独立模式运行。这就意味着 vsftpd 不能由类 inetd 来启动。vsftpd 应当直接执行。由 vsftpd 自身监听和处理联接请求。默认：NO

listen_ipv6

如 listen 参数，所不同的是，vsftpd 将对 IPv6 接口进行监听，而不是 IPv4 接口。此参数和 listen 参数相互独立。默认：NO

local_enable

用于控制是否允许本地登录。如果启用，/etc/passwd 中的普通帐号即可用于登录。默认：NO

log_ftp_protocol

如果启用，假若选项 xferlog_std_format 没有启用，所有的 FTP 请求和应答都会被记录。此选项对将对调试很有用。默认：NO

ls_recurse_enable

如果启用，此设置将允许用户使用"ls -R"。这有点安全威胁，因为在大型站点的根目录下进行 ls -R 将会消耗很多资源。默认：NO

no_anon_password

如果启用，匿名用户登录将不再需要密码，可以直接登录。默认：NO

no_log_lock

如果启用，在写日志文件时，将会阻止 vsftpd 使用文件锁定。这个选项通常不会启用。它的存在是为了处理操作系统的一个 bug，如 Solaris/Veritas 文件系统组合某些情况下试图锁定日志文件的现象。默认：NO

one_process_model

如果你使用 Linux2.4 内核，您就可以使用一个不同的安全模式，它只允许每个联接使用一个进程。这有一点小小的安全问题，但是提高了性能。如果您不清楚后果，或者您的站点要承受大量的并发用户联接时，请不要启用此选项。默认：NO

pasv_enable

如果数据传输时，您不允许使用 PASV 模式，则将此选项设为 NO。默认：YES

pasv_promiscuous

如果您要禁用 PASV 安全检查，将此选项设置为 YES。该检查用于确保数据传输联接与控制联接源于同一 IP 地址。如果不清楚后果，请不要启用此选项!此选项只有在某些使用安全隧道的方案中才能正常使用，或者需要 FXP 的支持。默认：NO

port_enable

如果您不允许使用端口模式获取数据联接，将此选项设置为 NO。默认：YES

port_promiscuous

如果您想禁用端口安全检查，将此选项设置为 YES。此检查用于确认出站的数据只流向客户端。搞清楚后果前，不要启用此选项！默认：NO

run_as_launching_user

如果您希望可以由用户来启动 vsftpd，将此选项设置为 YES。当不能使用 root 登录时，这通常很有用。严重警告：搞清楚后果前，不要启用此选项，随意的启用此选项将会导致非常严重的安全问题。特别是 vsftpd 没有/不能使用目录限制技术来限制文件访问时(甚至 vsftpd 是由 root 启动的)。一个愚蠢的替代方法是将选项 deny_file 设为{/*，*..*}，但是其可靠性并不能和限制目录相比，甚至不在一个等级上。如果启用此选项，应当限制其它选项的使用。例如，非匿名登录，上载文件宿主转换，使用源自端口 20 的联接和低于 1024 的端口不会工作。其它一些选项也可能受到影响。默认值：NO

secure_email_list_enable

如果您要为匿名用户指定一个做为密码的邮件地址列表，将此选项设置为 YES。在不需要创建虚拟用户的情况下，构建一个低安全性访问控制很有用。如果启用，匿名用户只有使用在 email_password_file 中指定的邮件地址做为密码，才能登录。文件格式是每行一个密码，没有空格。默认文件名是/etc/vsftpd。email_passwords。默认：NO

session_support

此选项用于控制 vsftpd 是否为登录保持会话。如果保持会话，vsftpd 将会尝试和更新 utmp 和 wtmp。如果使用 PAM 认证，同时还会打开 pam_session，直至登出。如果不需要保持登录会话，或许您希望禁用此选项，以使得 vsftpd 占用更少的进程和/或更少的特权。注意-utmp 和 wtmp 只有在启用 PAM 的情况下才支持。默认：NO

setproctitle_enable

如果启用，vsftpd 将会尝试在系统进程列表中显示会话状态信息。也就是说，进程报告将会显示每个 vsftpd 会话在做什么(闲置，下载等等)。出于安全的考虑，您可能需要将其关闭。默认：NO

ssl_enable

如果启用此选项，并在编译时加入 OpenSSL 支持，vsftpd 将支持通过 SSL 进行安全联接。此选项用于控制联接(包括登录)以及数据联接。您可能同时需要支持 SSL 的客户端。注意!!小心启用此选项。仅在需要时才启用。vsftpd 对使用 OpenSSL 库的安全性不做任何担保。启用此选项，就意味着您相信所安装的 OpenSSL 库的安全性。默认：NO

ssl_sslv2

只有激活 ssl_enable 选项时才有效。如果启用，此选项将允许使用 SSLv2 协议

进行联接。TLSv1 仍为首选联接。默认：NO

ssl_sslv3

只有激活 ssl_enable 选项时才有效。如果启用，此选项将允许使用 SSLv3 协议进行联接。TLSv1 仍为首选联接。默认：NO

ssl_tlsv1

只有激活 ssl_enable 选项时才有效。如果启用，此选项将允许使用 TLSv1 协议进行联接。TLSv1 仍为首选联接。默认：YES

syslog_enable

如果启用，任何本来应该输出到/var/log/vsftpd/vsftpd。log 的日志，将会输出到系统日志中。记录由 FTPD 完成。默认：NO

tcp_wrappers

如果启用，并且在编译 vsftpd 时加入了对 TCP_Wrappers 的支持，则连入请求转由 TCP_Wrappers 完成访问控制。另外，这是基于每个 IP 的配置机制。如果 tcp_wrappers 设置了 VSFTPD_LOAD_CONF 环境变量，则 vsftpd 会话将会试图加载在此变量中指定的 vsftpd 配置文件。默认：NO

text_userdb_names

默认情况下，目录列取时在用户和组字段显示的是数字 ID。如果启用此选项，则可以得到文本名称。基于性能的考虑，默认情况下关闭此选项。默认：NO

tilde_user_enable

如果启用，vsftpd 将试图解析类似~chris/pics 的路径名，即跟着用户名的波型号。注意，vsftpd 有时会一直解析~和~/(这里，~被解析称为初始登录路径)。~user 则只有在可以找到包含闲置目录的/etc/passwd 文件时才被解析。默认值：NO

use_localtime

如果启用，vsftpd 在列取目录时，将显示您本地时区的时间。默认显示为 GMT。由 MDTMFTP 命令返回的时间同样也受此选项的影响。默认：NO

use_sendfile

一个内部设定，用于测试在您的平台上使用 sendfile()系统的性能。默认：YES

userlist_deny

此选项只有在激活 userlist_enable 时才会有效。如果您将此选项设置为 NO，则只有在 userlist_file 文件中明确指定的用户才能登录系统。当登录被拒绝时，拒绝发生在被寻问命令之前。默认：YES

userlist_enable

如果启用，vsftpd 将会从 userlist_file 选项指定的文件中加载一个用户名列表。如果用户试图使用列表中指定的名称登录，那么他们将在寻问密码前被拒绝。这有助于阻止明文传送密码。详见 userlist_deny。默认：NO

virtual_use_local_privs

如果启用，虚拟用户将拥有同本地用户一样的权限。默认情况下，虚拟用户同匿名用户权限相同，这倾向于更多限制(特别是在写权限上)。默认：NO

write_enable

用于控制是否允许 FTP 命令更改文件系统。这些命令是：STOR，DELE，RNFR，RNTO，MKD，RMD，APPE 和 SITE。默认：NO

xferlog_enable

如果启用，将会维护一个日志文件，用于详细记录上载和下载。默认情况下，这个日志文件是/var/log/vsftpd.log。但是也可以通过配置文件中的 vsftpd_log_file 选项来指定。默认：NO

xferlog_std_format

如果启用，传输日志文件将以标准 xferlog 的格式书写，如同 wu-ftpd 一样。这可以用于重新使用传输统计生成器。然而，默认格式更注重可读性。此格式的日志文件默认为/var/log/xferlog，但是您也可以通过 xferlog_file 选项来设定。默认：NO

2.4.2 数字选项

下边是数字选项的列表。数字选项必须设置一个非负的整数。为了便于 umask 选项，同样也支持八进制数字。八进制数字首位应为 0。

accept_timeout

超时，以秒计，用于远程客户端以 PASV 模式建立数据联接。默认：60

anon_max_rate

允许的最大数据传输速率，单位 b/s，用于匿名客户端。默认：0(无限制)

anon_umask

用于设定匿名用户建立文件时的 umask 值。注意!如果您要指定一个八进制的数字，首位应当是"0"，否则将视作 10 进制数字。默认：077

connect_timeout

超时，单位秒，用于响应 PORT 方式的数据联接。默认：60

data_connection_timeout

超时，单位秒，用于设定空闲的数据连接所允许的最大时长。如果触发超时，则远程客户端将被断开。默认：300

file_open_mode

用于设定创建上载文件的权限。mask 的优先级高于这个设定。如果想允许上载的文件可以执行，将此值修改为 0777。默认：0666

ftp_data_port

FTP PORT 方式的数据联接端口。(需要激活 connect_from_port_20 选项)

默认：20

idle_session_timeout

超时，单位秒，远程客户端的最大 FTP 命令间隔。如果超时被触发，远程客户端将被断开。默认：300

listen_port

如果 vsftpd 以独立模式启动，此端口将会监听 FTP 连入请求。默认：21

local_max_rate

允许的最大数据传输速率，单位 b/s，用于限制本地授权用户。默认：0(无限制)

local_umask

用于设定本地用户上载文件的 umask 值。注意!如果您要指定一个八进制的数字，首位应当是"0"，否则将视作 10 进制数字。默认：077

max_clients

如果 vsftpd 以独立模式启动，此选项用于设定最大客户端联接数。超过部分将获得错误信息。默认：0(无限制)

max_per_ip

如果 vsftpd 以独立模式启动，此选项用于设定源于同一网络地址的最大联接数。超过部分将获得错误信息。默认：0(无限制)

pasv_max_port

为 PASV 方式数据联接指派的最大端口。基于安全性考虑，可以把端口范围指定在一样较小的范围内。默认：0(可以使用任意端口)

pasv_min_port

为 PASV 方式数据联接指派的最小端口。基于安全性考虑，可以把端口范围指定在一样较小的范围内。默认：0(可以使用任意端口)

trans_chunk_size

您可能不想修改这个设置，如果有带宽限制，可以尝试将此值设置为 8192。

默认：0(让 vsftpd 自己选择一个更合理的设置)

2.4.3 字符选项

下边是字符选项列表

anon_root

此选项声明，匿名用户在登录后将被转向一个指定目录(译者注：默认根目录)。失败时将被忽略。默认：(无)

banned_email_file

此选项用于指定包含不允许用作匿名用户登录密码的电子邮件地址列表的文件。使用此选项需要启用 deny_email_enable 选项。默认：/etc/vsftpd.banned_emails

banner_file

此选项用于指定包含用户登录时显示欢迎标识的文件。设置此选项，将取代 ftpd_banner 选项指定的欢迎标识。默认：(无)

chown_username

用于指定匿名用户上载文件的宿主。此选项只有在 chown_uploads 选项设定后才会有效。默认；root

chroot_list_file

此选项用于指定包含被限制在家目录中用户列表的文件。使用此选项，需启用 chroot_list_enable。如果启用了 chroot_local_user 选项，此文件所包含的则为不会被限制在家目录中的用户列表。默认：/etc/vsftpd.chroot_list

cmds_allowed

此选项指定允许使用的 FTP 命令(登录以后。以及登录前的 USER，PASS 和 QUIT)，以逗号分割。其它命令将被拒绝使用。这对于锁定一个 FTP 服务器非常有效。例如：mds_allowed=PASV，RETR，QUIT

默认：(无)

deny_file

此选项用于设定拒绝访问的文件类型(和目录名等)。此设定并不是对文件进行隐藏，但是您不能对其操作(下载，更换目录，以及其它操作)。此选项非常简单，不能用于严格的访问控制--文件系统的优先级要高一些。然而，此选项对于某些虚拟用户的设定非常有效。特别是在一个文件可以通过各种名称访问时(可能时通过符号联接或者硬联接)，应当注意拒绝所有的访问方法。与 hide_file 中给出名称匹配的文件会被拒绝访问。注意 vsftpd 只支持正则表达式匹配的部分功能。正因为如此，您需要尽可能的对此选项的设置进行测试。同时基于安全性考虑，建议您使用文件系统自身的访问控制。例如：deny_file={*.mp3，*.mov，.private}

默认：(无)

dsa_cert_file

此选项用于指定用于 SSL 加密联接的 DSA 证书的位置。

默认：(无-使用 RSA 证书)

email_password_file

此选项用于提供启用 secure_email_list_enable 选项，所需要的可替代文件。

默认：/etc/vsftpd.email_passwords

ftp_username

用于处理匿名 FTP 的用户名。此用的家目录即为匿名发 FTP 的根目录。

默认：ftp

ftpd_banner

用于替换首次连入 vsftpd 时显示的欢迎标识字符串。

默认：(无-显示默认 vsftpd 标识)

guest_username

参阅布尔选项 guest_enable。此选项用于将 guest 用户映射到一个真实用户上。

默认：ftp

hide_file

此选项用于设定列取目录时，要隐藏的文件类型(以及目录等)。尽管隐藏了，知道其宿主的客户端仍然能对文件/目录等有完全访问权限。与名称 hide_file 中包含的字符串匹配的项都将隐藏。注意 vsftpd 只支持正则表达式匹配的部分功能。例如：hide_file={*.mp3，.hidden，hide*，h?}

默认：(无)

listen_address

如果 vsftpd 以独立模式运行，此设定用于修改默认(所有本地接口)监听地址。格式为数字 IP 地址。默认：(无)

listen_address6

如 listen_address，不过应该指定为 IPv6 监听器指定默认监听地址。格式为标准 IPv6 地址格式。默认：(无)

local_root

本选项用于指定本地用户(即，非匿名用户)登录后将会转向的目录。失败时将被忽略。默认：(无)

message_file 此选项用于指定进入新目录时要查询的文件名。这个文件的内容为显示给远程用户的欢迎信息。使用此选项，需要启用 dirmessage_enable 选项。

默认：.message

nopriv_user

用于指定一个用户，当 vsftpd 要切换到无权限状态时，使用此用户。注意这最好是一个专用用户，而不是用户 nobody。在大多数机器上，用户 nobody 被用于大量重要的事情。默认：nobody

pam_service_name

用于指定 PAM 服务的名称。默认：ftp

pasv_address

此选项为 vsftpd 指定一个 IP 地址，用作对 PASV 命令的响应。IP 地址应该为数字模式。默认：(无-即地址从连入的联接套接字中获取)

rsa_cert_file

此选项用于指定 SSL 加密联接所用 RSA 证书的位置。

默认：/usr/share/ssl/certs/vsftpd.pem

secure_chroot_dir

此选项用于指定一个空目录。并且 ftp 用户不应对此目录有写权限。当 vsftpd 不需要访问文件系统是，此此目录做为一个限制目录，将用户限制在此目录中。

默认：/usr/share/empty

ssl_ciphers

此选项用于选择 vsftpd 允许使用哪些 SSL 加密算法来用于 SSL 加密联接。更多信息参阅 ciphers 的联机手册。注意这样可以有效的防止对某些发现漏洞的算法进行恶意的远程攻击。默认：DES-CBC3-SHA

user_config_dir

此选项用于定义用户个人配置文件所在的目录。使用非常简单，一个例子即可

说明。如果您将 user_config_dir 设置为/etc/vsftpd_user_conf 并以用户"chris"登录，那么 vsftpd 将对此用户使用文件/etc/vsftpd_user_conf/chris 中的设定。此文件的格式在联机手册中有详细说明。请注意，不是每个设定都能影响用户的。例如，有些设定只在用户会话开始时起作用。这包括 listen_address，banner_file，max_per_ip，max_clients，xferlog_file，等等。默认：(无)

user_sub_token

此选项需要和虚拟用户联合使用。其将依据一个模板为每个虚拟用户创建家目录。例如，如果真实用户的家目录由选项 guest_username 指定为/home/virtual/$USER，并且将 user_sub_token 设定为$USER，当虚拟用户 fred 登入后，将会进入(限制)目录/home/virtual/fred。如果 local_root 中包含了 user_sub_token 此选项也会起作用。

默认：(无)

userlist_file

此选项用于指定启用 userlist_enable 选项后需要加载文件的名称。

默认：/etc/vsftpd.user_list

vsftpd_log_file

此选项用于指定写入 vsftpd 格式日志的文件。如果启用了 xferlog_enable，而没有设定 xferlog_std_format 的话，日志将只会写入此文件。另为，如果设置了 dual_log_enable 的话，日志同样会写入此文件。更复杂一点，如果您设置了 syslog_enable 的话，输出将不会写入此文件，而是写入系统日志文件。

默认：/var/log/vsftpd.log

xferlog_file

此选项用于指定写入 wu-ftpd 样式日志的文件名。只有在 xferlog_enable 和 xferlog_std_format 做了相应设定，才会记录此传输日志。另外，如果您设置了 dual_log_enable 选项，也会记录此日志。

默认：/var/log/xferlog