

# GNU/Linux FTP



# GNU/Linux FTP

FTP 是 File Transfer Protocol（文件传输协议）的缩写。也是一个古老服务之一。

在 Unix/Linux 系统中常用的免费 FTP 服务器软件主要是 VSFTP。



# GNU/Linux FTP

## FTP 工作模式

主动模式 ( Active FTP ) :

在主动模式下，FTP 客户端随机开启一个大于 1024 的端口  $N$  向服务器的 21 号端口发起连接，然后开放  $N+1$  号端口进行监听，并向服务器发出 PORT  $N+1$  命令。服务器接收到命令后，会用其本地的 FTP 数据端口（通常是 20）来连接客户端指定的端口  $N+1$ ，进行数据传输。

# GNU/Linux FTP

## FTP 工作模式

被动模式 ( Passive FTP ) :

在被动模式下，FTP 客户端随机开启一个大于 1024 的端口 N 向服务器的 21 号端口发起连接，同时会开启 N+1 号端口。然后向服务器发送 PASV 命令，通知服务器自己处于被动模式。服务器收到命令后，会开放一个大于 1024 的端口 P 进行监听，然后用 PORT P 命令通知客户端，自己的数据端口是 P。客户端收到命令后，会通过 N+1 号端口后连接服务器的端口 P，然后在两个端口之间进行数据传输。

# GNU/Linux FTP

安装 VSFTP

```
#yum install vsftpd -y
```

启动 VSFTP

```
#systemctl start vsftpd
```

```
#systemctl enable vsftpd
```



# GNU/Linux FTP

VSFTP 配置文件及所在路径

```
#cd /etc/vsftpd
```

```
#ls -l vsftpd.conf
```

解读 vsftpd 配置文件



# GNU/Linux FTP

## 1. 实现匿名账户使用 VSFTP( 仅下载 )

1) 配置文件不需要修改

2) 启动 vsftpd

3) 客户端安装 ftp 客户端程序

```
#yum install ftp -y
```

4) 客户端使用匿名账户登陆

匿名账户 :ftp 密码 :随意



# GNU/Linux FTP

## 2. 实现实体账户使用 VSFTP

1) 配置文件不需要修改

2) 启动 vsftpd

3) 客户端安装 ftp 客户端程序

```
#yum install ftp -y
```

4) 客户端使用 ftp 服务器上的实体账户登陆

实体账户 :snow 密码 :123456





# GNU/Linux FTP

## 3. 开启匿名上传及建立目录机制

1) 修改 vsftpd.conf 配置文件

将 29 行注释取消

`anon_upload_enable = YES`

将 33 行注释取消

`anon_mkdir_wirte_enable = YES`

2) 启动 vsftpd



# GNU/Linux FTP

## 3. 开启匿名上传及建立目录机制

3) 修改 /var/ftp/pub 目录权限，使 ftp 匿名账户具备写权限

```
#chown -R ftp.ftp pub
```

## 4) 客户端测试



# GNU/Linux FTP

## 4. 开启匿名账户修改、删除权限

1) 修改 vsftpd.conf 配置文件  
于 33 行下增加

```
anon_other_wirte_enable = YES
```

2) 启动 vsftpd



# GNU/Linux FTP

## 4. 开启匿名账户修改、删除权限

3) 修改 /var/ftp/pub 目录权限，使 ftp 匿名账户具备写权限

```
#chown -R ftp.ftp pub
```

## 4) 客户端测试



# GNU/Linux FTP

## 5. FTP 其他限制

1) 限定最大并发连接数为 100

`max_clients=100`

2) 限定每客户端最多同时可以发起的链接个数

`max_per_ip=1`



# GNU/Linux FTP

## 5. FTP 其他限制

3) 限定匿名用户最大下载速率（单位：字节）

`anon_max_rate=81920`

4) 设定有效实体账户最大下载速率

`local_max_rate=102400`



# GNU/Linux FTP

## 5. FTP 其他限制

5) 设定 pasv 模式使用端口的范围

pasv\_min\_port=50000

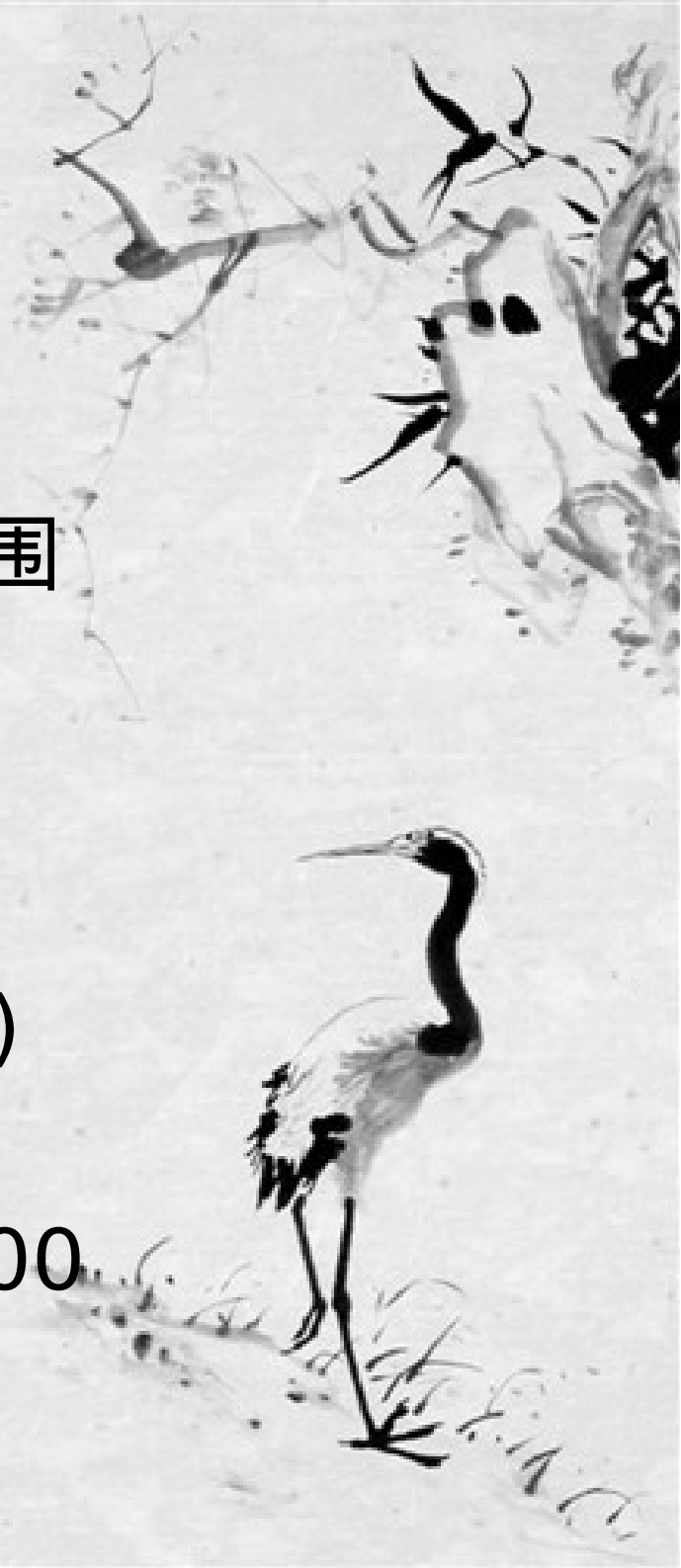
pasv\_max\_port=60000

6) 设置最大超时时间 ( 单位 : 秒 )

idle\_session\_timeout=120

data\_connection\_timeout=300

connection\_timeout=60



# GNU/Linux FTP

## 6. ftp 模板所在位置

```
#cd /usr/share/doc/vsftpd-*/EXAMPLE/
```





# GNU/Linux FTP

## 7. 实现 chroot

1) 编辑 vsftpd.conf

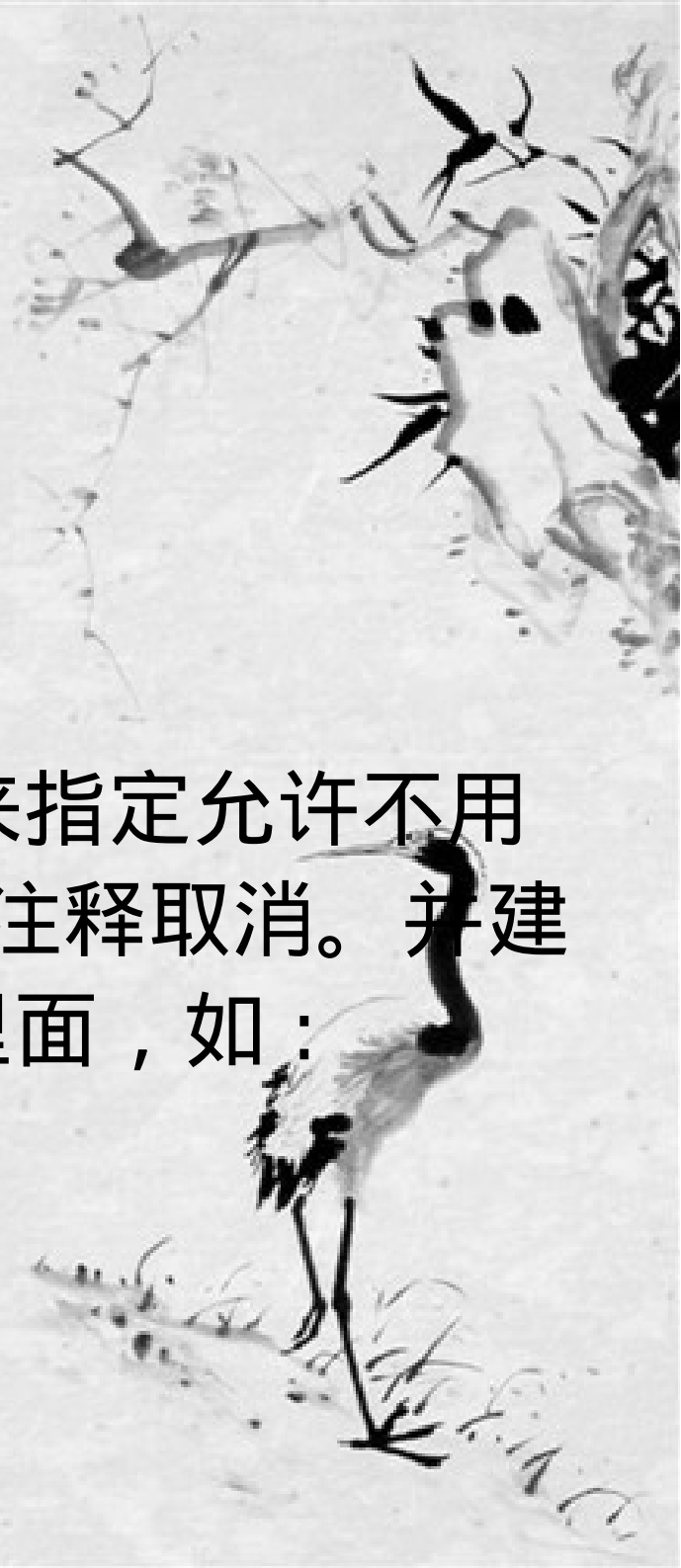
开启 100 行注释

如果打算使用 chroot\_list 列表来指定允许不用 chroot 的用户请将 101 及 102 行注释取消。并建立 chroot\_list 文件，将账户写入里面，如：

```
#cat /etc/vsftpd/chroot_list
```

```
snow
```

```
chuai
```



# GNU/Linux FTP

## 7. 实现 chroot

### 1) 编辑 vsftpd.conf

值得注意的是 chroot\_list 文件里的账户是不被 chroot 的



# GNU/Linux FTP

## 7. 实现 chroot

chroot 出现错误如：

500 OOPS: vsftpd: refusing to run with  
writable root inside chroot()

Login failed.

421 Service not available, remote server  
has closed connection



# GNU/Linux FTP

## 7. 实现 chroot

chroot 出现错误解决方法

在 vsftpd.conf 中增加

`allow_writeable_chroot=YES`



# GNU/Linux FTP

## 8. 禁止账户登陆

```
#cd /etc/vsftpd
```

```
#ls -l ftpusers ← 在此文件中的账户无法登陆  
ftp
```

```
#ls -l user_list ← 有 vsftpd.conf 中的  
userlist_enable 语句控制
```



# GNU/Linux FTP

## 8. 禁止账户登陆

当 `userlist_enable=NO` 时 `user_list` 列表不生效，列表内和列表外的用户都可登录

当 `userlist_enable=YES` 及 `userlist_deny=YES` 同时写到配置中时 `user_list` 列表内的用户不可以登录，列表外用户可登录

当 `userlist_enable=YES` 及 `userlist_deny=NO` 同时写到配置中时 `user_list` 只允许列表以外的用户登录，并且必需是以命令行的方式

# GNU/Linux FTP

## 9. FTP 虚拟账户

VSFTP 一个称为安全的保证是采用了虚拟用户的认证方式，它靠对 `/etc/pam.d/` 目录下指定的一个认证文件对用户进行认证，认证成功后再把虚拟用户映射为本地用户，该本地用户由服务器配置文件里的语句 `ftp_username` 的值指定。

使用虚拟用户认证，则原有系统账户将不再可用，可以把原系统账户加入到虚拟用户列表中。

# GNU/Linux FTP

## 10. FTP 虚拟账户实现

### 1) 建立虚拟账户文件

```
#vim ~/ftputers.txt
```

snow ← 账户

123456 ← 密码

lisa ← 账户

654321 ← 密码

.....





# GNU/Linux FTP

## 10. FTP 虚拟账户实现

### 2) 生成账户的数据库

```
#db_load -T -t hash -f ~/ftpusers.txt  
/etc/vsftpd/ftpusers.db
```

```
#ls -l /etc/vsftpd/ftplogin.db
```

```
#rm -rf ~/ftpusers
```

-T: 允许使用文本文件信息加载到数据库中

-t : 指定加密算法

-f : 指定文件



# GNU/Linux FTP

## 10. FTP 虚拟账户实现

### 3) 通过 PAM 验证虚拟账户

```
#cp /etc/pam.d/vsftpd  
/etc/pam.d/vsftpd.bak
```

```
#cp \
```

```
>/usr/share/doc/vsftpd*/EXAMPLE/VIRTUAL_  
USERS/vsftpd.pam \
```

```
> /etc/pam.d/vsftpd
```




# GNU/Linux FTP

## 10. FTP 虚拟账户实现

### 4) 修改 vsftpd 验证文件

```
#vim /etc/pam.d/vsftpd
```

将 so 模块路径去除，仅保留模块 

将 db=/etc/vsftpd\_login 改为数据库文件名。

如

```
db=/etc/vsftpd/ftpusers
```

/\* 特别提示，在 pam 中的验证数据库不用  
加 .db



# GNU/Linux FTP

## 10. FTP 虚拟账户实现

### 5) 创建与虚拟账户管理的实体账户

```
#useradd test
```

```
#chmod 555 -Rv /home/test
```

/\* 如出现“500 OOPS: vsftpd: refusing to run with writable root inside chroot()” 请将 /home/test 目录写权限去除

```
#chmod a-w /home/test
```



# GNU/Linux FTP

## 10. FTP 虚拟账户实现

6) 修改 vsftpd.conf 配置文件  
在其下增加

```
guest_enable=YES  
guest_username=test
```

7) 重启 vsftpd

```
#systemctl restart vsftpd
```

```
//* 客户端测试 *//
```



# GNU/Linux FTP

## 11. FTP+SSL/TLS

### 1) 生成所需密钥及证书

```
#cd /etc/pki/tls/certs  
#openssl req -x509 -nodes -newkey  
rsa:2048 -keyout  
/etc/pki/tls/certs/vsftpd.pem -out  
/etc/pki/tls/certs/vsftpd.pem
```

//\*OpenSSL 使用 PEM 文件格式存储证书和密钥。 PEM 实质上是 Base64 编码的二进制内容。本例将密钥和证书信息全部存放在 vsftpd.pem 中

# GNU/Linux FTP

## 11. FTP+SSL/TLS

### 2) 配置 vsftpd.conf

在 vsftpd.conf 最尾端加入：

```
rsa_cert_file=/etc/pki/tls/certs/vsftpd.pem
```

```
ssl_enable=YES
```

```
force_local_data_ssl=YES
```

```
force_local_logins_ssl=YES
```



# GNU/Linux FTP

## 11. FTP+SSL/TLS

3) 重新启动 vsftpd

```
#systemctl restart vsftpd
```





# GNU/Linux FTP

## 11. FTP+SSL/TLS

### 4) 文本客户端

#### (1) 配置 .ftprc 文件

```
#vi ~/.ftprc
```

```
set ftp:ssl-auth TLS
```

```
set ftp:ssl-force true
```

```
set ftp:ssl-protect-list yes
```

```
set ftp:ssl-protect-data yes
```

```
set ftp:ssl-protect-ftp yes
```

```
set ssl:verify-certificate no
```



# GNU/Linux FTP

## 11. FTP+SSL/TLS

### 4) 文本客户端

```
#lftp -u snow rh7s1.niliu.edu
```



# GNU/Linux FTP

## 11. FTP+SSL/TLS

### 5)GUI 客户端

