第 10 讲　网络资源共享

王晓庆

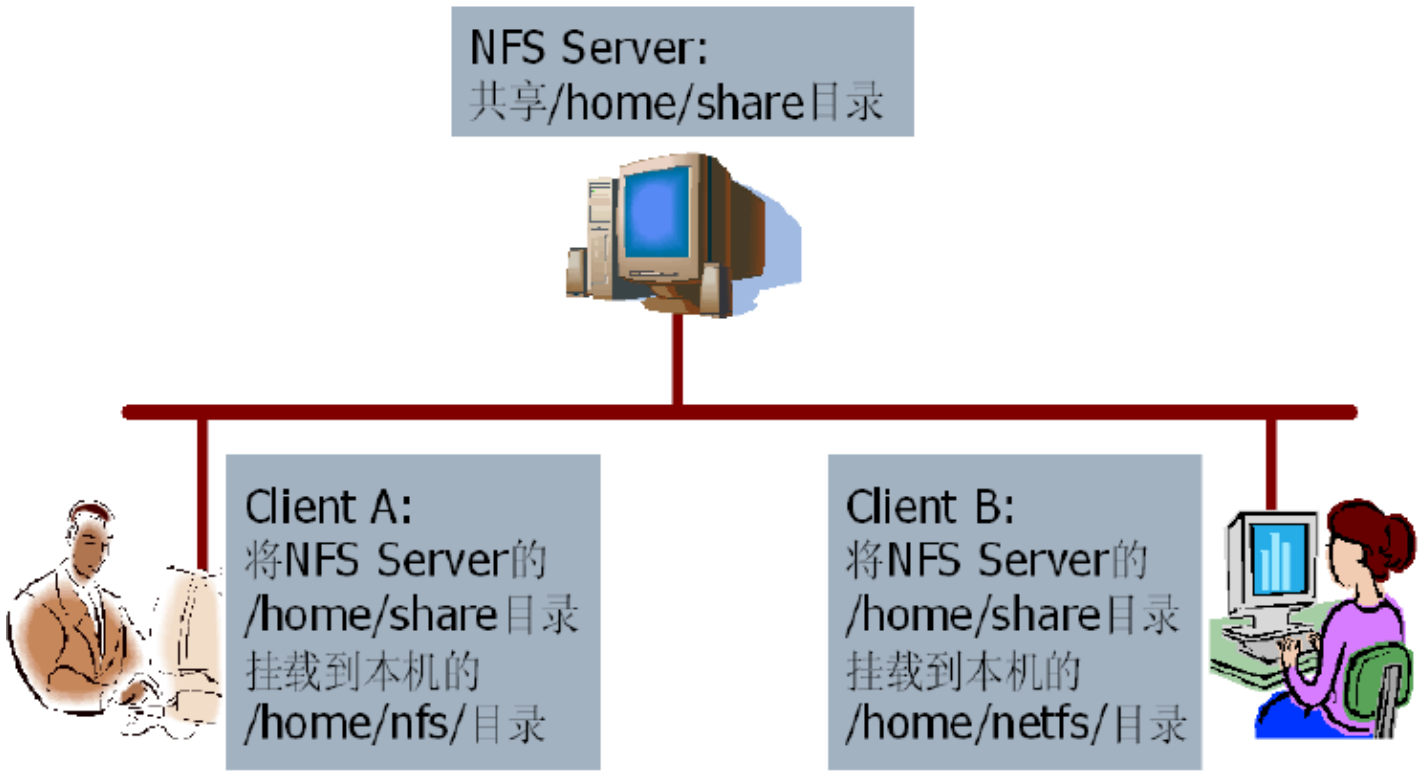
wangxiaoqing@outlook.com

June 2, 2016

Outline

NFS 工作原理 (1)

- 在 NFS 服务器上设置共享目录，客户端可以将该共享目录挂载到本地挂载点，然后就可以像本地目录一样使用。



NFS 工作原理 (2)

- NFS 可实现文件共享，但要借助于 RPC 协议实现数据传输，要使用 NFS，客户端和服务器端都需要启动 RPC。
- 对于 NFS，RPC 最主要的功能是指定每个 NFS 功能所对应的端口号，并且告知客户端，让客户端可以链接到正确的端口。
  1. 服务器启动 NFS 时随机取用多个端口，并主动向 RPC 注册各功能端口
  2. RPC 使用 111 端口监听客户端请求
  3. 客户端向服务器 RPC(111 端口) 发出 NFS 文件访问功能的查询请求
  4. RPC 将 NFS 注册的守护进程端口号告知客户端
  5. 客户端根据得到的端口号直接与 NFS 守护进程通信

NFS 必需的系统守护进程

- rpc.nfsd
  - 基本的 NFS 守护进程，主要功能是管理客户端是否能够登录服务器
- rpc.mountd
  - RPC 挂载守护进程，主要功能是管理 NFS 文件系统。当客户端通过 rpc.nfsd 成功登录 NFS 服务器后，必须通过文件使用权限的验证 (由 rpc.mountd 读取 NFS 配置文件来确认客户端权限)，才能使用 NFS 服务器提供的文件
- portmap
  - 主要用于端口映射，当客户端尝试连接并使用 RPC 所管理的服务 (如 NFS) 时，portmap 将所管理服务的对应端口号提供给客户端，从而使客户端可以通过该端口向服务器发出请求。

安装 NFS 服务器

- 正常允许 NFS 服务需要安装以下两个软件包
  - nfs-utils(NFS 主程序)，默认已安装
  - portmap(RPC 主程序)，默认已安装
- 启动 NFS(注意顺序！)

```
service portmap start # 客户端也要启动 portmap 服务！
service nfs start
rpcinfo -p            # 打印 rpc 注册端口信息
```

- 停止 NFS(注意顺序！)

```
service nfs stop
service portmap stop
```

配置 NFS 服务器 (1)

- 主配置文件/etc/exports

```
sharedir [clients][(opt,...)] [clients][(opt,...)] ...
```

示例

```
/projects       *.abc.com(rw)
/home/testnfs   192.168.0.1(rw,sync) *(ro)
```

- 说明
  - 共享目录要用绝对路径，且包含空格时要用双引号
  - 共享目录与客户端之间要用空格分隔
  - 共享目录可同时指定多个客户端，客户端之间用空格分隔
  - 客户端和该客户端的选项之间不能有空格
  - 客户端的选项放在 () 内，且选项之间要用逗号分隔

配置 NFS 服务器 (2)

- 指定客户端
  客户端指允许访问 nfs 服务的计算机，是可选的设置项（为空则代表任意客户端），支持通配符 * 或?。

示例

```
192.168.1.10                    # 指定主机 ip
192.168.1.0/24                  # 指定网段
192.168.1.*                     # 同上
192.168.1.0/255.255.255.0       # 同上
client1.abc.com                 # 指定主机域名
*.abc.com                       # 指定域
'*'                             # 任何主机
```

配置 NFS 服务器 (3)

- **常用共享选项 (未指定选项时，将使用默认选项)**

ro	只读(默认)
rw	读写
sync	同步写入(默认)
async	异步写入
root_squash	客户端root用户映射为匿名用户(默认)
no_root_squash	客户端root用户保持为root用户
all_squash	所有客户端用户映射为匿名用户
not_all_squash	所有客户端用户身份保持不变(默认)
secure	要求客户端通过1024以下端口连接NFS服务器(默认)
insecure	允许客户端通过1024以上端口连接NFS服务器
wdelay	有多个用户写入NFS共享目录时合并写入(默认)
no_wdelay	立即执行写操作(当使用async时无效)

配置 NFS 服务器 (4)

配置文件示例

```
vim /etc/exports
/home/public 192.168.1.0/24(rw) *(ro)
/ 192.168.1.10(rw,no_root_squash)
/pub (ro,insecure,all_squash)

exportfs -rv  # 重新发布/etc/exports 配置的共享目录
mkdir /home/public /pub
chmod a+w /home/public # 设置共享目录本地权限
```

注意

配置文件中给出的只是 NFS 访问权限，用户最终的权限还要看共享目录的本地权限设置！

配置 NFS 服务器 (5)

- 测试 nfs 服务

```
cat /var/lib/nfs/etab # 服务器端查看共享目录及其共享选项
showmount -e 192.168.1.200 # 查看服务器共享目录列表
# 客户端挂载共享目录
mkdir /mnt/public
mount -t nfs 192.168.1.200:/home/public /mnt/public
mount
# 分别以 root 和 mike 身份向/mnt/public 目录写入文件
echo "root test" >rootfile
echo "mike test" >mikefile
ls -l /mnt/public    # 在客户端查看文件信息
ls -l /home/public   # 在服务器端查看文件信息
```

Samba 概述

- Samba 工作原理
  - Samba 是 Linux、UNIX 与 Windows 之间进行交互操作的软件,samba 通过 SMB/CIFS 协议为不同操作系统之间提供安全、稳定、快速的文件与打印服务。
  - Samba 包括 samba(服务器端软件包)、samba-client(客户端软件包) 和 samba-common(samba 公共文件软件包)
  - Samba 由 smbd 和 nmbd 两个守护进程组成
    - smdb：为客户提供文件与打印机共享服务, 还负责用户权限验证以及锁功能, 默认监听 TCP 的 139 与 445 端口。
    - nmdb：提供 NetBIOS 名称服务, 以满足基于 Common Internet File System(CIFS) 协议的共享环境, 默认使用 UDP 的 137 端口。

Samba 服务器角色

- 域控制器
  - Samba 服务器可以充当 Windows NT4 类型的主域控制器 (PDC)、备份域控制器 (BDC)，或者活动目录安全模式的域控制器 (相当于 Windwos 2000 Server 以上的域控制器)。
- 域成员服务器
  - Samba 服务器可以充当 Windows NT4 类型的域成员服务器或者活动目录安全模式的域成员服务器，接受域控制的统一管理。域控制器可以由 Windows 服务器或 Samba 服务器来充当。
- 独立服务器
  - 工作在对等网络 (工作组)，Samba 服务器作为不加入域的独立服务器，与其他计算机是一种对等关系，各自管理自己的用户帐号。

Samba 安全模式

- share
  - 共享安全模式，用户不需要提供用户名和密码即可访问 Samba 服务器资源，适用于公共的共享资源，安全性差，需要配合其他权限设置才能保证 Samba 服务器的安全。
- user
  - 用户安全模式，用户必须提供合法的用户名和密码，通过身份验证才能访问 Samba 服务器资源，这是默认模式。
- server
  - 服务器安全模式，与用户安全模式类似，但用户名和密码需要提交到另一台 Samba 服务器进行验证，因而还要指定密码验证服务器。如果验证出现错误，客户端改用用户安全模式。
- domain
  - 域安全模式，Samba 服务器作为域成员加入到 Windows 域环境中，验证工作由 Windows 域控制器负责。
- ads
  - 活动目录安全模式，Samba 服务器具备域安全模式的所有功能，并可以作为域控制器加入到 Windows 域环境中。

Samba 的功能与应用

- 文件和打印机共享
  - Samba 的主要功能，SMB 进程实现资源共享，将文件和打印机发布到网络中供用户使用。
- 身份验证和权限设置
  - 支持用户安全模式和域安全模式等的身份验证和权限设置模式，通过加密方式可以保护共享的文件和打印机。
- 名称解析
  - 可以作为 NetBIOS 名称服务器提供计算机名称解析服务，还可作为 WINS 服务器。
- 浏览服务
  - Samba 服务器可以称为本地主浏览服务器 (LMB)，保存可用资源列表，当客户端访问网上邻居时，会提供浏览列表，显示共享目录、打印机等资源。

部署 Samba 服务器

- 1. 安装 Samba 服务器

```
yum -y install samba
service smb {start|stop|restart|status|condrestart}
```

- 2. 规划 Samba 共享资源和设置权限
- 3. 编辑主配置文件/etc/samba/smb.conf

注意

用户最终访问共享资源的权限是由配置文件中设置的权限以及 Linux 系统的本地文件权限共同决定，且以两者中最严格的为准。

- 4. 设置共享用户
- 5. 重新加载配置文件或重启 smb 服务，使配置生效
- 6. 测试 Samba 服务器及客户端访问测试

Samba 配置实例 (1)

配置要求

1. Samba 以独立服务器形式部署
2. 采用 user 安全模式
3. 作为文件服务器，为 Linux 客户端和 Windows 客户端提供文件共享服务
4. 将一个共享目录作为一个公共数据存储区，只有经过认证的用户才能读写文件，其中一个用户对该共享的所有文件具有所有权
5. 让用户通过网络访问自己的主目录

Samba 配置实例 (2)

- 1. 共享文件权限规划
  - 将目录/home/pubsmb 作为一个公共存储区，经过认证的用户才能在其中存储文件
  - 经过认证的用户都可以访问自己的主目录，但是不能访问其他用户的主目录
  - 指定用户 neo 作为公共存储区的所有者
- 2. 创建相应用户和组

```
groupadd pubsmb
useradd -g pubsmb neo
passwd neo
```

- 3. 配置共享目录

```
mkdir /home/pubsmb
chown neo:pubsmb /home/pubsmb
chmod 777 /home/pubsmb
```

Samba 配置实例 (3)

- 4. 配置/etc/samba/smb.conf 文件

```
#=====Global Settings=====
[global]
workgroup = WORKGROUP
server string = samba server
security = user
log file = /var/log/samba/%m.log
username map=/etc/samba/smbusers
#=====Share Definitions=====
[homes]
comment = Home Directories
validusers = %S
read only = no
browseable = no
writable = yes
```

Samba 配置实例 (4)

- 4. 配置/etc/samba/smb.conf 文件 (续)

```
[public]
comment = DataShare
path = /home/pubsmb
force user = neo
force group = pubsmb
read only = no
```

- 说明
  - smb.conf 文件分为若干节，每一节由一个方括号括起来的节名开始，直到下一节。
  - 每一节包含若干参数设置：参数名称 = 参数值
  - 节名和参数名称不区分大小写
  - 每行定义一个参除，可在行尾加
  - 以 # 和; 开头的行是注释行
  - 该文件包含两个部分：全局设置和共享定义

常见的 Samba 服务器全局设置参数

参数	说明	举例
workgroup	域/工作组	workgroup = WORKGROUP
server string	描述	server string = samba server
security	安全模式	security = user
netbios name	NetBIOS 名	netbios name = SMBSRV
hosts allow	允许客户端	hosts allow = 192.168.1. 192.168.2.10
guest account	匿名账户	guest account = pcguest(默认 nobody)
log file	日志文件	log file = /var/log/samba/%m.log
max log size	最大日志	max log size = 50(单位为 KB)
interfaces	侦听接口	interfaces = 192.168.1.1/24

常见共享定义参数

参数	说明	举例
comment	说明信息	comment = Home
path	共享路径	path = /home/pub
browseable	允许浏览	browseable = yes(默认)
valid users	允许用户	valid users = tom @user
invalid users	拒绝用户	invalid users = bob @sale
read only	只读	read only = yes(默认)
writable	可写	writable = no
write list	可写用户	write list = tom @user
guest ok	允许匿名	guest ok = no(默认)
force user	默认用户	force user = auser
force group	默认组	force group = agroup
create mask	权限掩码	create mask = 0744(默认)
directory mask	权限掩码	directory mask = 0755(默认)

- 注意：write list 仅当 writeable = no 时才生效。

常用 Samba 变量

参数	说明	参数	说明
%U	当前用户名	%T	当前日期时间
%G	当前用户组	%D	当前域/工作组
%h	服务器域名	%S	当前共享名
%m	客户 NetBIOS 名	%P	当前服务的根目录
%L	服务器 NetBIOS 名	%u	当前服务用户名
%M	客户端域名	%g	当前服务组名
%I	客户 IP 地址	%H	当前服务的用户主目录
%i	客户连接的 IP 地址		

Samba 配置实例 (5)

- 检测配置文件

  ```
  testparm /etc/samba/smb.conf
  ```

- 配置 Samba 用户
  - 由于 share 安全模式缺乏安全性，一般不采用 share 安全模式，这就需要添加 Samba 账户，Samba 使用 Linux 本地账户，但需要将本地账户添加到 Samba 的账户文件/etc/samba/smbpasswd 中。

    ```
    smbpasswd -a neo # 向/etc/samba/smbpasswd 中添加账户
    # -x 删除账户 -d 禁用账户 -e 启用账户
    ```

注意

smbpasswd 命令所操作的用户必须是 Linux 系统中已存在的本地账户！

Samba 配置实例 (6)

- 设置用户名映射
  - Samba 支持从客户端到服务器的用户名映射，如将 Windows 用户映射到 Linux 用户，或将多个用户映射到同一个用户，便于他们共享文件。
  - 设置用户名映射需要在 Samba 主配置文件中添加以下全局参数设置，指定一个用户映射文件：

    ```
    username map = /etc/samba/smbusers
    ```

  - 用户映射文件 (默认为/etc/samba/smbusers) 格式如下：

    ```
    root = administrator admin
    sys = @system
    nobody = guest pcguest smbguest
    neo = mike
    ```

Samba 配置实例 (7)

- 监测 Samba 服务器

```
smbclient -L localhost -U%
smbstatus
ls /var/log/samba
#Samba 自动为每个连接到 Samba 服务器的计算机分别建立日志
#nmbd.log 记录 nmdb 进程的解析信息
#smdb.log 记录用户访问 Samba 服务器的事件
```

Linux 客户端访问 Samba 服务器

- 使用 smbclient 工具访问，使用类似 ftp 的交互命令

```
smbclient -L //192.168.0.200 -U neo      # 查看共享资源
smbclient //192.168.0.200/public -U neo # 访问共享资源
Password:
smb: \> ?
smb: \> ls
smb: \> get test
smb: \> q
```

- 使用 mount 命令挂载共享目录到本机

```
mkdir /mnt/public
mount [-t cifs] -o username=neo,password=666666 \
//192.168.0.200/public /mnt/public
# 注：服务器使用 share 安全模式，则无需用户名和密码
```

Samba 客户端访问控制 (1)

- 1. share 之外的安全模式都会对用户进行身份验证
- 2. 使用 host allow 和 host deny 参数限制客户端

```
hosts allow = 192.168.2.
hosts deny = 192.168.2.200 # 无效

hosts allow = 192.168.2. EXCEPT 192.168.2.200
# 可以使用通配符 * 和?, ALL(任意客户端),LOCAL(本机) 等
```

注意

1. hosts allow 和 hosts deny 冲突时，hosts allow 优先
2. 注意 hosts allow/deny 的作用范围，即是在全局设置中还是共享定义中设置的

Samba 客户端访问控制 (2)

- 3. 使用 valid users 参数实现用户审核

```
valid users = neo, @cwb # 仅允许 neo 和财务部组成员访问
```

- 4. 使用 writable 和 write list 控制用户的写权限

```
writable = yes # 所有人都拥有写权限

writable = no  # 所有人都没有写权限
write list = neo, @cwb # 但 neo 和财务部组成员具有写权限
```

Samba 客户端访问控制 (3)

- 5. 使用 Linux 文件权限实现用户访问的最终控制

```
mkdir /cwb_data
chown -R neo:cwb /cwb_data # 设置共享目录所有者和属主
chmod -R 770 /cwb_data     # 设置权限
vim /etc/samba/smb.conf
...
[cwb]
path = /cmb_data
force user = neo           # 设置默认用户
force group = cwb          # 设置默认组名
# 所有访问该共享目录的用户都将拥有默认用户和组的权限
# 上述设置可以有效地简化权限设置
```

share 安全模式的 Samba 服务配置 (1)

- 实例 1：匿名只读文件服务器

```
vim /etc/samba/smb.conf
...
security = share
...
[doc]
comment = documents
path = /usr/share/doc
public = yes
# 注意：此处没有添加 writable = yes，默认权限为只读。

service smb restart
```

- Windows 客户端访问测试

访问UNC路径：\\192.168.0.200\doc

share 安全模式的 Samba 服务配置 (2)

- 实例 2：匿名读写文件服务器 (1)

```
#1. 添加用户
useradd fox
password fox
smbpasswd -a fox

#2. 创建共享目录
mkdir /pubdir
chown fox:fox /pubdir
chmod -R 770 /pubdir
```

share 安全模式的 Samba 服务配置 (3)

- 实例 2：匿名读写文件服务器 (2)

```
#3. 配置共享目录
vim /etc/samba/smb.conf
...
security = share
...
[pub]
path = /pubdir
force user = fox # 所有客户端映射为默认用户 fox
force group = fox # 所有客户端映射为默认组 fox
read only = no #read only=no 等价于 writable = yes
guest ok = yes #guest ok 与 public 是同义词

#4. 重启 smb 服务并在客户端测试
```

部署 vsftpd 服务器 (myserver)

- 安装 vsftpd 服务器

```
yum install vsftpd
```

- 测试 vsftpd 服务器

```
chekconfig --level 345 vsftpd on
service vsftpd start
ftp 192.168.0.200
```

- 默认的匿名 ftp 服务器
  - 匿名用户名为 anonymouse 或 ftp，密码为空
  - 匿名访问的 ftp 站点主目录默认为/var/ftp
  - 可将提供下载的文件复制到/var/ftp/pub 目录

主配置文件/etc/vsftpd/vsftpd.conf

- 指令行格式：选项 = 值 (注意：= 两边不能有空格等空白符！)
  - 每个选项都有默认值。
- 注释行：以 # 开头 (注意：# 前面也不能有空白符！)

```
rpm -qd vsftpd
man 5 vsftpd.conf
```

vsftpd 基本配置 (1)

- 设置 vsftpd 服务器运行方式

```
listen=YES # 以独立服务方式运行 (默认值 NO)
```

- 设置 vsftpd 服务监听地址与控制端口

```
listen_address # 监听 IP 地址 (默认未设置, 表示监听所有接
listen_port=21 # 监听端口号 (默认值)
```

- 设置 PORT 模式

```
port_enable=YES # 运行于 PORT(主动) 模式 (默认值)
connect_from_port_20=YES #PORT 模式数据连接使用端口 20(默
ftp_data_port=20 #PORT 模式连接发起的端口 (默认值)
```

- 设置 PASV 模式

```
pasv_enable=YES # 是否允许采用 PASV 模式 (默认值)
pasv_min_port #PASV 模式数据连接所用端口最小值 (默认值 0
pasv_max_port #PASV 模式数据连接所用端口最大值 (默认值 0
```

vsftpd 基本配置 (2)

- 设置性能选项
  - idle_session_out
    - 会话空闲断开时间，默认为 300(秒)
  - data_connection_timeout
    - 数据连接超时时间，默认为 300(秒)
  - accept_timeout
    - 客户端以 PASV 模式建立连接的超时时间，默认为 60(秒)
  - connect_timeout
    - 客户端响应 PORT 模式连接的超时时间，默认为 60(秒)
  - max_clients
    - 独立运行方式下，最大并发连接数，默认为 0(不限制)
  - max_per_ip
    - 独立运行方式下，每 IP 最大并发连接数，默认为 0(不限制)
  - anon_max_rate
    - 匿名用户的最大数据传输速率，默认为 0(byte/s)(不限制)
  - local_max_rate
    - 本地用户的最大数据传输速率，默认为 0(byte/s)(不限制)

配置匿名访问 (1)

- 用户类型
  - 匿名用户
  - 本地用户
  - 虚拟用户
- 设置匿名访问选项

选项	含义
anonymous_enable	允许匿名用户
ftp_username	匿名用户名
no_anon_password	匿名用户无需密码
deny_email_enable	与 baned_email_file 配合使用
baned_email_file	与 deny_email_enable 配合使用
anon_root	匿名用户主目录

配置匿名访问 (2)

- 设置匿名访问选项 (续)

选项	含义
anon_world_readable_only	匿名用户仅能下载全局可读文件
anon_upload_enable	允许匿名用户上传文件
anon_mkdir_write_enable	允许匿名用户新建目录
anon_other_write_enable	允许匿名用户其他写权限
chown_uploads	与 chown_username 配合使用
chown_username	与 chown_uploads 配合使用

配置面向 Internet 的匿名 FTP 站点

- vsftpd 自带了相应的示例配置文件

```
cd /usr/share/doc/vsftpd-2.0.5/EXAMPLE
vim INTERNET_SITE_NOINETD/vsftpd.conf
```

配置 FTP 站点本地用户访问

- 启用本地用户登录

```
local_enable=YES
```

- 配置用户主目录

```
local_root # 默认值 none
```

- 设置用户配置文件：基于每个用户定义配置选项

```
#1. 在主配置文件中设置用户配置文件所在目录
user_config_dir=/etc/vsftpd_user_conf
#2. 创建用户配置目录
mkdir /etc/vsftpd_user_conf
#3. 创建用户配置文件
vim /etc/vsftpd_user_conf/mike
local_max_rate=100000
```

- 设置用户磁盘配额
  - vsftpd 本身并未提供该功能，可以通过 quota 进行设置。

vsftpd 安全设置 (1)

- 用户访问控制
  - 拒绝部分用户访问

```
userlist_enable=YES
userlist_file=/etc/vsftpd/user_list
userlist_deny=YES
```

  - 允许部分用户访问

```
userlist_enable=YES
userlist_file=/etc/vsftpd/user_list
userlist_deny=NO
```

vsftpd 安全设置 (2)

- 目录访问控制

```
chroot_list_enable # 将指定用户锁定在其主目录中
chroot_list_file   # 设置要锁定的用户列表文件
# 默认文件为/etc/vsftpd/chroot_list, 格式为一行一个用户
chroot_local_user  # 所有本地用户被锁定在其主目录中
passwd_chroot_enable # 与 chroot_local_user 配合使用
```

1. chroot_list_enable=YES和chroot_local_user=YES
==> 不锁定/etc/vsftpd/chroot_list中列出的用户
2. chroot_list_enable=YES和chroot_local_user=NO
==> 仅锁定/etc/vsftpd/chroot_list中列出的用户
3. chroot_list_enable=NO和chroot_local_user=YES
==> 锁定所有用户
4. chroot_list_enable=NO和chroot_local_user=NO
==> 不锁定所有用户

vsftpd 安全设置 (3)

- **文件系统操作控制**

```
hide_ids # 将目录列表中的所有用户和组信息显示为 ftp
ls_recurse_enable # 允许使用 ls -R 指令
write_enable # 允许使用任何修改文件系统的 ftp 指令
secure_chroot_dir # 设置安全监牢目录，详见 man 手册
```

vsftpd 安全设置 (4)

- 新增文件的权限控制

```
anon_umask # 匿名用户新增文件默认权限掩码 (默认值 077)
local_umask # 本地用户新增文件默认权限掩码 (默认值 077)
file_open_mode # 上传文件的权限 (默认值 0666)
```

vsftpd 安全设置 (5)

- 客户端主机限制

```
tcp_wrappers=YES # 使用 tcp_wrappers 进行客户端访问控制

vim /etc/hosts.allow
vsftpd:192.168.0.:allow # 允许指定网段客户端访问
vsftpd:ALL:deny         # 拒绝其他客户端访问
```

配置 FTP 虚拟用户访问 (1)

- 虚拟用户
  - 直接用本地用户访问 ftp 服务器存在安全隐患，为此可使用虚拟用户 (virtual user) 来作为专门的 ftp 账户。
  - ftp 虚拟用户并不是系统本地用户，不能直接登录系统，只能访问 ftp 服务，对操作系统的影响更小。
  - 虚拟用户主要用于访问提供给非信任用户，但又不适合公开的内容。
  - 可使用 PAM 实现虚拟用户功能。

配置 FTP 虚拟用户访问 (2)

- 1. 创建虚拟用户数据库

```
#(1) 创建虚拟用户数据文件
vim /etc/vsftpd/login.txt
kehu01   # 奇数行为虚拟用户名
abc123   # 偶数行为虚拟用户密码
kehu02
123abc
#(2) 将 login.txt 转换为数据库文件 login.db
yum install db4-utils
cd /etc/vsftpd
db_load -T -t hash -f login.txt login.db
#(3) 限制数据库文件的访问权限
chmod 600 login.db
```

配置 FTP 虚拟用户访问 (3)

- 2. 修改 vsftpd 的 PAM 配置文件

```
vim /etc/pam.d/vsftpd
# 注释掉其他所有行，并添加以下两行内容
auth required /lib/security/pam_userdb.so \
db=/etc/vsftpd/login
account required /lib/security/pam_userdb.so \
db=/etc/vsftpd/login
```

注意

PAM 配置文件里面的数据库文件 login 不能写后缀.db

- 3. 为虚拟用户创建一个系统用户和主目录

```
useradd -d /home/ftpsite -s /sbin/nologin virtual
# 可将要下载的内容复制到/home/ftpsite 目录
```

配置 FTP 虚拟用户访问 (4)

- 4. 修改/etc/vsftpd/vsftpd.conf

```
anonymous_enable=NO
local_enable=YES
write_enable=NO
anon_upload_enable=NO
anon_mkdir_write_enable=NO
anon_other_write_enable=NO
chroot_local_user=YES
guest_enable=YES # 启用虚拟用户功能
guest_username=virtual # 设置虚拟用户所映射的系统用户
pam_service_name=vsftpd # 设置 PAM 配置文件名
listen=YES
listen_port=8021
pasv_min_port=30000
pasv_max_port=30999
```

配置 FTP 虚拟用户访问 (5)

- 5. 测试虚拟用户访问

```
service vsftpd restart

ftp 192.168.0.200 8021
user: kehu01
pass: abc123
```

说明

经过上述配置，匿名用户或本地用户都无法访问。通过对实际用户进行限制，可以适当修改虚拟用户的权限。