# Linux vsftpd文件共享

简介：

vsftpd是Very Secure FTP的简写形式，是一筐非常安全的FTP软件。该软件基于GPL开发的，被设置为Linux平台下稳定、快速、安全的FTP软件，它支持IPv6以及SSL加密。

vsftpd的安全性主要体现在三个方面：进程分离，处理不同任务的进程彼此是独立运行的；进程运行均以最小权限运行；多数进程都使用chroot进行了禁锢，放置客户访问非法共享目录，这里的chroot是改变根的一种技术，如果我们通过vsftpd共享了/var/ftp目录，则该目录对客户端而言就是共享的根目录。

vsftpd虽然是一款FTP软件，但FTP不同于其他的互联网协议，它使用多端口通信。当客户端连接服务器请求资源时，服务器会使用21端口与客户端进行通信，该端口专门处理客户端发送给服务器的请求命令，也就是俗称的命令端口。而最终服务器与客户端进行数据传输时，还需要使用另一个数据端口，数据端口号取决于服务器运行的模式是主动模式还是被动模式。

## FTP的工作模式

### 1，主动模式

第一步，客户端随机开启大于1024的X端口与服务端的21端口建立连接通道，通道建立后，客户端随时可以通过该通道发送上传或下载命令。

第二步，当客户端需要与服务器进行数据传输时，客户端会在开启一个大于1024的随机端口Y，并将Y端口号通过之前的命令通道(X)传送给服务器的21端口。

第三步，服务器获取到客户端的第二个端口后会主动连接客户端的该端口(Y),通过三次握手，完成服务器与客户端数据通道的建立，所有的数据均通过该数据通道进行传输。

### 2，被动模式

第一步，客户端随意开启大于1024的X端口与服务器的21端口建立连接通道。

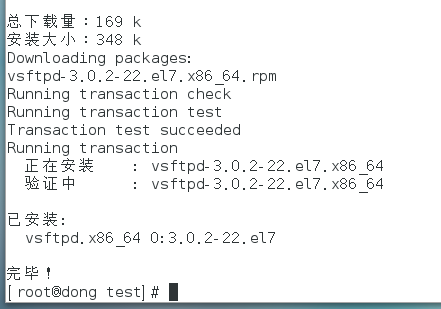
第二步，当客户端需要与服务器进行数据传输时，客户端从命令通道发送数据请求要求上传或下载数据。

第三步，服务器收到数据请求后会随机开启一个端口Y，并通过命令将该端口信息传送给客户端。

第四步，客户端在收到服务器发送过来的数据端口Y的信息后，将在客户端本地开启一个随机端口Z，此时客户端再主动通过本机的Z端口与服务器的Y端口进行连接，通过三次握手连接完成后，即可进行数据传输。

## 安装vsftpd

[root@centos7 ~]# yum -y install vsftpd



[root@centos7 ~]# systemctl start vsftpd

[root@centos7 ~]# systemctl enable vsftpd

RPM格式的vsftpd软件安装完成后，该软件的主程序是/usr/sbin/vsftpd，下面是vsftpd相关的核心文件与目录列表的说明。

/etc/logrotate.d/vsftpd # 日志轮转备份配置文件

/etc/pam.d/vsftpd # 基于PAM的csftpd验证配置文件

/etc/vsftpd # vsftpd软件目录

/etc/vsftpd/ftpusers # 默认的vsftpd黑名单

/etc/vsftpd/user\_list # 可以通过主配置文件设置该文件为黑名单或白名单

/etc/vsftpd/vsftpd.conf # vsftpd主配置文件

/usr/sbin/vsftpd # vsftpd主程序

/usr/share/doc/vsftpd-3.0.2 # vsftpd文档资料路径

/var/ftp # 默认vsftpd共享目录

## vsftp配置文件解析

vsftpd配置文件默认位于/etc/vsftpd目录下，vsftpd会自动寻找以.conf结尾的配置文件，并使用此配置文件启动FTP服务。

配置文件的格式为：选项=值(中间不可以有任何空格符)，以#开头的行会被识别为注释行。

全局设置

anonymous\_enable=NO

# 是否允许匿名登陆

local\_enable=YES

# 是否启用本地账户登陆功能。注意：主要是为虚拟宿主用户，如果该项目设定为NO那么所有虚拟用户将无法访问。

local\_umask=022

# 设定上传/新建 目录或文件 的默认权限

# 022 -rw-r--r-- 077 -rw-------

dirmessage\_enable=YES

# 是否开启目录标语功能[用户进来目录是否显示消息]

ftpd\_banner=This Vsftp server supports virtual users ^\_^

# 设定Vsftpd的登陆标语

xferlog\_enable=YES

# 是否开启日志记录功能。

xferlog\_file=/var/log/xfer.log

# 设定Vsftpd的服务日志保存路径。注意，该文件默认不存在。

xferlog\_std\_format=YES

# 设定日志使用标准的记录格式

dual\_log\_enable=YES

# 是否开启双日志功能，生成两个日志文件

vsftpd\_log\_file=/var/log/vsftpd.log

# 设定vsftpd日志存放路径。注意：日志不存在需要手动创建

nopriv\_user=vsftpd

# 设定支撑Vsftpd服务的宿主用户为手动建立的Vsftpd用户。注意，一旦做出更改宿主用户后，必须注意一起与该服务相关的读写文件的读写赋权问题。比如日志文件就必须给与该用户写入权限等：chown vsftpd:vsftpd /var/log/{vsftpd.log,xfer.log}

connect\_from\_port\_20=YES

# 使用主动模式连接，启用20端口

chown\_uploads=NO

chown\_username=whoever

# 是否允许上传文件更改宿主。更改的宿主由chown\_username指定

idle\_session\_timeout=600

# 设定空闲连接超时时间，这里使用默认。将具体数值留给每个具体用户具体指定，当然如果不指定的话，还是使用这里的默认值600，单位秒

data\_connection\_timeout=120

# 设定单次最大连续传输时间，这里使用默认。将具体数值留给每个具体用户具体指定，当然如果不指定的话，还是使用这里的默认值120，单位秒。

async\_abor\_enable=YES

# 是否支持异步传输功能。

ascii\_upload\_enable=YES

ascii\_download\_enable=YES

# 是否支持ASCII模式的上传和下载功能。

deny\_email\_enable=YES

banned\_email\_file=/etc/vsftpd/banned\_emails

# 指定不允许匿名电子邮件地址文件，写入banned\_email\_file指定的文件，一行一个。当deny\_email\_enable=YES时，banned\_email\_file有效

#chroot\_list\_enable=YES # 不建议使用，注释掉。

chroot\_list\_enable=YES

chroot\_list\_file=/etc/vsftpd/chroot\_list

# 禁止用户登出自己的FTP主目录。用户名写在chroot\_list\_file指定的文件中，一行一个。写在chroot\_list\_file文件里的用户，禁止登出自己的主目录。

ls\_recurse\_enable=NO

# 是否激活用户登陆FTP后使用"ls -R"的命令。该命令会对服务器性能造成巨大开销。如果该项被允许，那么挡多用户同时使用该命令时将会对该服务器造成威胁。

listen=YES

# 是否监听端口，独立运行守护进程

# 意思是使用standalone启动vsftpd，而不是super daemon(xinetd)控制它

listen\_port=21

# 监听入站FTP请求的端口号

download\_enable=YES

# 如何设置为NO，则拒绝所有下载请求

write\_enable=YES

# 是否允许写操作，设置为NO，则拒绝所有上传文件请求

pasv\_enable=YES

# 是否启用被动链接，默认为被动模式

pasv\_max\_port=24600

# 被动模式链接的最大端口号

pasv\_min\_port=24500

# 被动模式连接的最小端口号

userlist\_enable=YES

# 是否启用userlist用户列表文件

userlist\_deny=YES

# 是否禁止userlist文件中的账户访问FTP

max\_clients=2000

# 最大允许同时2000客户端连接，0代表无限制

max\_per\_ip=0

# 每个客户端的最大链接限制，0代表无限制

tcp\_wrappers=YES

# 是否启用tcp\_wrappers限制访问【/etc/hosts.allow , /etc/hosts.deny)

guest\_enable=YES

# 是否开启虚拟用户功能

guest\_username=ftp

# 设定虚拟用户的宿主用户

virtual\_use\_local\_privs=YES

# 虚拟用户的权限是否同步宿主用户权限

user\_config\_dir=/etc/vsftpd/conf

# 指定目录，在该目录下可以为用户设置独立的配置文件与选项。配置名必须和

[虚拟]用户名相同。

pam\_service\_name=vsftpd

# 设定PAM服务下Vsftpd的验证配置文件名。因此，PAM验证将参考/etc/pam.d/下的vsftpd文件配置。

匿名账户

anon\_root=/var/ftp # 匿名访问FTP的根路径，默认为/var/ftp

anon\_upload\_enable=YES # 是否允许匿名账户上传，默认禁止

anon\_mkdir\_write\_enable=YES # 是否允许匿名账户创建目录，默认禁止

anon\_other\_write\_enable=YES # 是否允许匿名账户进行其他所有写操作

anon\_max\_rate=0 # 匿名数据上传率(B/s)

anon\_umask=077 # 匿名上传权限掩码

本地账户

local\_enable=YES # 是否启用本机账户FTP功能

local\_max\_rate=0 # 本地账户数据上传率（B/s）

local\_umask=077 # 本地账户权限掩码

chroot\_local\_user=YES # 是否禁锢本地账户根目录，默认为NO

chroot\_list\_enable=YES # 使用户不能离开主目录

local\_root=/ftp/common # 本地账户访问FTP根路径

## 账户权限

vsftpd支持的常用登陆方式有：匿名登陆、本地账户登陆、虚拟账户登陆三种方式。

匿名账户登陆一般应用于下载服务器。vsftpd默认开启的是匿名共享，默认共享路径为/var/ftp

本地账户登陆则需要使用系统账户以及对应的系统密码才可以登录使用FTP。默认anonymous\_enable与local\_enable均被设置为YES，此时FTP为匿名访问模式，如果需要开启本地账户FTP功能，需要将anonymous\_enable设置为NO，默认共享路径为个人家目录。

需要注意的是，开启本地账户登陆后，用户可以离开家目录，从而进入系统中的其他目录，这样是非常危险的，如果在配置文件中使用chroot-local\_user，用户将被禁锢在自己的家目录下，放置用户进入系统中的其他目录。

当有大量用户需要使用FTP时，vsftpd支持虚拟账户模拟登陆FTP，从而避免了创建大量的系统账户，通过guest\_enable可以开启vsftpd的虚拟账户功能，guest\_username用阿里指定本地账户的虚拟映射名称。

vsftpd有两个文件（黑名单文件和白名单文件）可以对用户进行ACL控制，/etc/vsftpd/ftpusers默认是一个黑名单文件，存储在该文件的所有用户将无法访问FTP，格式为每行一个账户名称。/etc/vsftpd/user\_list文件会根据注配置文件中配置项设定的不通，而成为黑名单或白名单文件。此外，也可以禁用该文件。主配置文件中的userlist\_enable决定了是否启用user\_list文件。黑名单表示拒绝名单中的账户访问FTP，也就是其他所有的账户默认允许访问FTP。而白名单表示仅允许白名单中的账户访问FTP，没有在白名单中的其他所有账户则默认将拒绝访问FTP。

## 常见问题

### 1，提示错误代码：530 Login incorrect

如果登陆提示530错误，说明登陆过程中账户验证失败。则可能是因为你使用的是64位操作系统，而pam文件中库文件的调用却使用的是/lib/security/pam\_userdb.so。或者是用户名或密码输入有错误。此外，也可能是vsftpd主配置文件中pam\_service\_name设置的pam文件名称与/etc/pam.d中创建的pam文件名称不一致，导致无法验证成功。

### 2，提示错误代码：500 OOPS:cannot change directory:/home/ftp/$USER

该提示代表目录不存在或无权限导致的无法切换至目录，也有可能是SELinux导致的无法共享账户家目录，默认SELinux不允许共享家目录

### 3，使用windows系统访问主动模式的vsftpd服务器时无法访问成功

默认Windows会使用被动模式连接FTP服务器，如果需要以主动模式连接服务器，需要修改浏览器的属性，方法是查找Internet选项的“高级”选项卡，找到使用被动FTP，取消该功能即可。

### 4，账户登陆后无法上传数据

根据不用的账户类型，检查主配置文件的设置，匿名账户与虚拟账户以anon\_开头的权限设置，本地账户检查以local\_开头的权限设置，并且要确保全局write\_enable设置为YES。此外，文件系统目录的权限也需要修改，确保客户端账户有访问该目录的权限。

### 5，启动服务器报错：500 OOPS: bad bool value in config file

vsftpd配置文件设置错误，检查配置文件。配置文件要求每个设置项占用独立的一行，并且不可以有多余的空格

### 6，限定用户不能跳出主目录之后，却无法登陆？

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

解决：从2.3.5之后，vsftpd增强了安全检查，如果用户被限定在了其主目录下，则该用户的主目录不能再具有写权限了！如果检查发现还有写权限，就会报该错误。

要修复该错误有两种解决方法！

第一种：去除用户主目录的写权限

[root@centos7 ~]# chmod a-w /home/user # 注意把user目录替换为自己

第二种：编辑vsftpd.conf配置文件

加入以下配置：

allow\_writeable\_chroot=YES # 允许主目录拥有写权限

## local\_umask权限解析

umask=022中"022"是八进制的写法，如果换成二进制是000010010

在unix中文件权限是三类用户，三种权限。三类用户分别是文件所有者user(u)，文件所有者所在主群组group(g)、其它用户others(o)，三种权限分别是起读read(r)、写write(w)、执行execute(x)。

如果一个文件的权限如下：所有者有读写的权限，群组有读和执行权限、其它用户有读权限，可以写成：

rw-r-xr--

其中前三位指明了所有者的权限、中间三位指明了组权限、最后三位指明了其它用户的权限。我们用ls -l可以看到文件权限详情，列出来的是10位，最前一位如果是d表示是子目录。

说回来，如果把这上面9位字母换成二进制数则是：

110101100

如果换成八进制是多少？

因此文件权限可以用9位二进制数表示。umask在英文中是屏蔽的意思，那么

umask=022

指屏蔽文件的022权限，到底屏蔽了什么？将这个八进制数用二进制表示

000010010

换算成字母是

----w--w-

指取消组的写权限、取消其它用户的写权限。

屏蔽这些权限后，剩下什么权限呢？用字母表示是：

rwxr-xr-x

用二进制数表示是：

111101101

你可以对比一下

000010010

可以知道屏蔽前后的换算关系。

另一种算法（不建议使用）

文件夹满权限：777

文件满权限：666

777-022=755

666-022=644

## 基础案例

### 本地账户FTP

步骤：

1，修改vsftpd.conf 配置文件

anonymous\_enbale=NO # 禁止匿名用户登陆

local\_enable=YES # 开启本地账户登陆

write\_enable=YES # 开启全局写[上传]权限

local\_umask=022 # 默认上传文件权限掩码

dirmessage\_enable=YES # 开启用户主目录消息提示

xferlog\_enable=YES # 开启xferlog日志

xferlog\_std\_format=YES # 定义xferlog日志格式

chroot\_local\_user=YES # 禁锢本地账户根目录

chroot\_list\_enable=YES # 同上

chroot\_list\_file=/etc/vsftpd/chrootlist # 禁锢主目录的用户列表文件

listen=YES # 监听请求

pam\_service\_name=vsftpd # 设定PAM服务下vsftpd的验证配置文件名

userlist\_enable=YES # 设定userlist\_file里的用户，不允许登陆FTP

tcp\_warppers=YES # 开启tcp支持

allow\_writeable\_chroot=YES # 允许主目录拥有写权限

2，创建系统账户

[root@centos7 ~]# useradd -s /sbin/nologin tom

[root@centos7 ~]# useradd -s /sbin/nologin jerry

[root@centos7 ~]# useradd -s /sbin/nologin smith

3，添加用户到 chroot\_list 文件

[root@centos7 ~]# vim /etc/vsftpd/chroot\_list

tom

jerry

smith

4，服务管理

[root@centos7 ~]# setenforce 0 # 暂停SELinux

[root@centos7 ~]# systemctl stop firewalld.service # 关闭防火墙

[root@centos7 ~]# systemctl start vsftpd # 启动vsftpd服务

[root@centos7 ~]# systemctl enable vsftpd # vsftpd设为开机启动

### 虚拟账户FTP，满足大量账户的访问需求

vsftpd创建虚拟账户的数据需要博村在Berkeley DB格式的数据文件中，所以需要安装db4-utils工具来创建这样的数据文件。

1，创建虚拟用户数据库

首先需要创建明文密码文件，明文文件奇数行为用户名，偶数行为密码。然后使用db\_load工具将其转换为数据库文件，db\_load工具需要通过安装libdb-utils软件获得，最后可以通过修改文件权限已增强数据资料的安全性。

[root@centos7 ~]# yum -y install libdb-utils

[root@centos7 ~]# vim /etc/vsftpd/vlogin

tomcat

123456

jerry

654321

[root@centos7 ~]# db\_load -T -t hash -f /etc/vsftpd/vlogin /etc/vsftpd/vlogin.db

[root@centos7 ~]# chmod 600 /etc/vsftpd/{vlogin,vlogin.db}

2，创建PAM文件，设置基于虚拟账户验证

Linux 一般通过PAM文件设置账户验证机制，然后通过创建新的PAM文件，使用新的数据文件进行登录验证，PAM文件中的db选项于指定并验证账户和密码的数据库问及那，数据库文件无须.db的名称后缀。注意：因为本环境使用的是CentOS 7 64 位操作系统，所以验证模块调用的是lib64目录下的文件。

[root@centos7 ~]# vim /etc/pam.d/vsftpd

auth required /lib64/securitypam\_userdb.so db=/etc/vsftpd/vlogin.db

account required /lib64/security/pam\_userdb.so db=/etc/vsftpd/vlogin.db

3，设置虚拟账户共享目录

因为所有的虚拟最终都需要映射到一个真实的系统账户，所以这里需要添加一个系统账户并设置家目录。

[root@centos7 ~]# useradd -s /sbin/nologin -d /home/ftp virtual

4，修改主配置文件

与本地账户FTP一样，我们需要修改vsftpd的主配置文件，使用guest\_enable选项开启虚拟账户功能，所有的虚拟账户都被映射为guest\_username指定的一个系统真实账户。如果需要对虚拟账户进行权限设置，使用与匿名账户一样的设置项即可。如anon\_mkdir\_write\_enable=NO就可以控制虚拟账户无法创建目录。

anonymous\_enable=NO # 禁止匿名用户登陆

local\_enable=YES # 开启本地账号登陆功能

write\_enable=YES # 允许写操作

anon\_upload\_enable=YES # 允许匿名用户上传文件

anon\_mkdir\_write\_enbale=NO # 不允许匿名用户创建文件

anon\_other\_write\_enable=NO # 不允许匿名用户其他写操作

chroot\_local\_user=YES # 禁锢用户只能在家目录操作

guest\_enable=YES # 开启虚拟账户功能

guest\_username=virtual # 虚拟用户的映射名

listen=YES # 独立运行vsftpd

listen\_port=21 # 监听的端口

pasv\_enable=YES # 开启被动模式连接

pasv\_min\_port=30000 # 被动模式连接最小端口号

pasv\_max\_port=30999 # 被动模式连接最答端口号

pam\_service\_name=vsftpd.pam # vsftpd的PAM认证文件

user\_config\_dir=/etc/vsftpd\_user\_conf # 在指定目录为用户设置独立的配置

user\_sub\_token=$USER # 设置用户名变量令牌

5，为每个用户设置独立的共享路径

通过在主配置文件中使用user\_config\_dir选项，设置一个基于账户的配置目录，在该目录下可以创建若干个账户名称同名的文件，并在文件中为此账户设置独立的配置选项，包括权限与共享路径等设置，这样就可以为每个账户做单独的权限设置等操作。

[root@centos7 ~]# mkdir /etc/vsftpd\_user\_conf

[root@centos7 ~]# mkdir -p /home/ftp/tomcat

[root@centos7 ~]# vim /etc/vsftpd\_user\_conf/tomcat

local\_root=/home/ftp/$USER

6，重启服务

[root@centos7 ~]# systemctl restart vsftpd

细节注意：

1，要使用$USER变量，必须先配置user\_sub\_token=$USER

2，PAM文件不存在可以自己创建。

3，若想使虚拟用户和系统用户或者匿名用户同时均可同时登陆FTP，则需要修改默认的 vsftpd PAM文件。

默认PAM文件：/etc/pam.d/vsftpd。[系统版本的不同，位置或者名称可能不一样]

[root@centos7 ~]# cat /etc/pam.d/vsftpd

**session optional pam\_keyinit.so force revoke**

**auth required pam\_listfile.so item=user sense=deny file=/etc/vsftpd/ftpusers onerr=succeed**

**auth required pam\_shells.so**

**auth include password-auth**

**account include password-auth**

**session required pam\_loginuid.so**

**session include password-auth**

若需要时虚拟用户可以登陆，需要加入自己定义的PAM文件。如：

auth sufficient /lib64/security/pam\_userdb.so db=/etc/vsftpd/vlogin

account sufficient /lib64/security/pam\_userdb.so db=/etc/vsftpd/vlogin

解释：

sufficient # 这个标志的含义是如果这个模块验证通过，就不必使用后面的层叠模块进行验证了；但如果失败了，就继续后面的认证。

注意：位置需要放好，不然配置家目录的时候可能会冲突。

假设：配置在最前面，可能会导致所有的用户登陆，都当成虚拟用户登陆。

## PAM文件解析

### 一、pam简介

Linux-PAM(linux可插入认证模块)是一套共享库,使本地系统管理员可以随意选择程序的认证方式. 换句话说,不用(重新编写)重新编译一个包含PAM功能的应用程序,就可以改变它使用的认证机制. 这种方式下,就算升级本地认证机制,也不用修改程序.

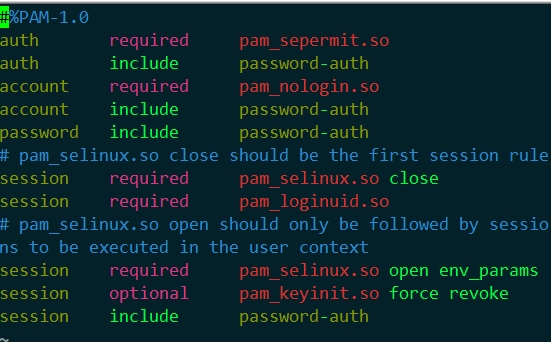
PAM使用配置/etc/pam.d/下的文件,来管理对程序的认证方式.应用程序 调用相应的配置文件,从而调用本地的认证模块.模块放置在/lib/security下,以加载动态库的形式进，像我们使用su命令时,系统会提示你输入root用户的密码.这就是su命令通过调用PAM模块实现的.

### 二、PAM的配置文件介绍

PAM配置文件有两种写法:

一种是写在/etc/pam.conf文件中，但centos6之后的系统中，这个文件就没有了。

另一种写法是,将PAM配置文件放到/etc/pam.d/目录下,其规则内容都是不包含 service 部分的，即不包含服务名称，而/etc/pam.d 目录下文件的名字就是服务名称。如: vsftpd,login等.,只是少了最左边的服务名列.如:/etc/pam.d/sshd



由上图可以将配置文件分为四列，

第一列代表模块类型

第二列代表控制标记

第三列代表模块路径

第四列代表模块参数

1. PAM的模块类型

Linux-PAM有四种模块类型,分别代表四种不同的任务

它们是:认证管理(auth),账号管理(account),会话管理(session)和密码(password)管理,一个类型可能有多行,它们按顺序依次由PAM模块调用.

|  |  |
| --- | --- |
| 管理方式 | 说明 |
| auth | 用来对用户的身份进行识别.如:提示用户输入密码,或判断用户是否为root等. |
| account | 对帐号的各项属性进行检查.如:是否允许登录,是否达到最大用户数,或是root用户是否允许在这个终端登录等. |
| session | 这个模块用来定义用户登录前的,及用户退出后所要进行的操作.如:登录连接信息,用户数据的打开与关闭,挂载文件系统等. |
| password | 使用用户信息来更新.如:修改用户密码. |

2.PAM的控制标记

PAM使用控制标记来处理和判断各个模块的返回值.（在此只说明简单的认证标记）

|  |  |
| --- | --- |
| 控制标记 | 说明 |
| required | 表示即使某个模块对用户的验证失败，也要等所有的模块都执行完毕后,PAM 才返回错误信息。这样做是为了不让用户知道被哪个模块拒绝。如果对用户验证成功，所有的模块都会返回成功信息。 |
| requisite | 与required相似,但是如果这个模块返回失败,则立刻向应用程序返回失败,表示此类型失败.不再进行同类型后面的操作. |
| sufficient | 表示如果一个用户通过这个模块的验证，PAM结构就立刻返回验证成功信息（即使前面有模块fail了，也会把 fail结果忽略掉），把控制权交回应用程序。后面的层叠模块即使使用requisite或者required 控制标志，也不再执行。如果验证失败，sufficient 的作用和 optional 相同 |
| optional | 表示即使本行指定的模块验证失败，也允许用户接受应用程序提供的服务，一般返回PAM\_IGNORE(忽略). |

3.模块路径

模块路径.即要调用模块的位置. 如果是64位系统，一般保存在/lib64/security,如: pam\_unix.so

同一个模块,可以出现在不同的类型中.它在不同的类型中所执行的操作都不相同.这是由于每个模块针对不同的模块类型,编制了不同的执行函数.

4.模块参数

模块参数,即传递给模块的参数.参数可以有多个,之间用空格分隔开,如:

password required pam\_unix.so nullok obscure min=4 max=8 md5

### 常用的PAM模块介绍

|  |  |  |
| --- | --- | --- |
| PAM模块 | 结合管理类型 | 说明 |
| pam\_unix.so | auth | 提示用户输入密码,并与/etc/shadow文件相比对.匹配返回0 |
| account | 检查用户的账号信息(包括是否过期等).帐号可用时,返回0. |
| password | 修改用户的密码. 将用户输入的密码,作为用户的新密码更新shadow文件 |
| pam\_shells.so | auth  account | 如果用户想登录系统，那么它的shell必须是在/etc/shells文件中之一的shell |
| pam\_deny.so | account  auth  password  session | 该模块可用于拒绝访问 |
| pam\_permit.so | auth  account  password  session | 模块任何时候都返回成功. |
| pam\_securetty.so | auth | 如果用户要以root登录时,则登录的tty必须在/etc/securetty之中. |
| pam\_listfile.so | auth  account  password session | 访问应用程的控制开关 |
| pam\_cracklib.so | password | 这个模块可以插入到一个程序的密码栈中,用于检查密码的强度. |
| pam\_limits.so | session | 定义使用系统资源的上限，root用户也会受此限制，可以通过/etc/security/limits.conf或/etc/security/limits.d/\*.conf来设定 |

更详细信息请参考：<https://www.cnblogs.com/ilinuxer/p/5087447.html>

