1、理论

1.1 FTP 概述

```
# FTP 服务器 < File Transfer Protocol Server 文件传输服务器 >
    互联网上提供文件存储和访问服务的计算机

# FTP < File Transfer Protocol 文件传输协议 >
    互联网上用来传输文件的协议
```

```
# 常见 FTP 服务器:

Windows:
    Serv-U
    FTP Server
    filezilla_server
Linux:
    ProFTPD
```

1.2 VSFTP 概述

FTP是File Transfer Protocol < 文件传输协议 > 的英文简称，用户Internet上文件的上传/下载。使用FTP来传输时，具有一定的危险性，因为数据在Internet上是采用明文传输方式。

VSFTP是一种基于 GPL 发布的类Unix系统上使用的FTP服务器软件，它的全称是Very Secure FTP，解决了FTP传输安全性的问题。

1.3 安全特性

1. vsftp程序的运行着一般是普通用户，降低了相对应进程的权限，提高了安全性。
2. 任何需要执行较高权限的指令都需要上层程序许可。
3. ftp所使用的的命令都是ftp的独立命令，基本不需要系统额外提供命令。
4. 拥有chroot功能，可以改变用户的根目录，限制用户只能在自己的家目录。

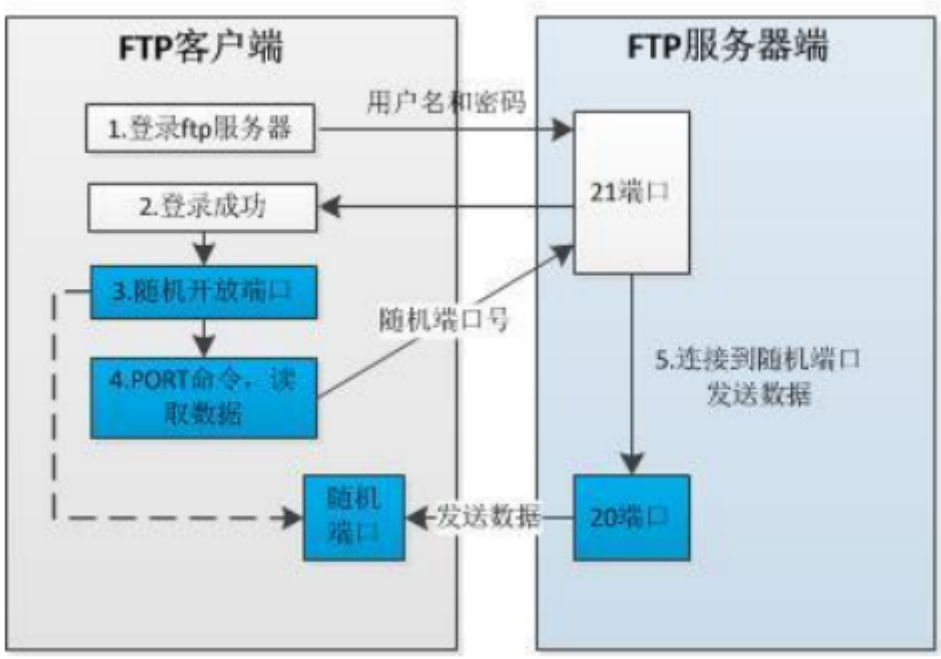
1.4 VSFTP 连接类型

控制连接 < 持续连接 > ▶ TCP 21 < 命令信道 > ▶ 用户收发FTP命令

数据连接 < 按需连接 > ▶ TCP 20 < 数据信道 > ▶ 用户上传下载数据

1.5 VSFTP 工作模式

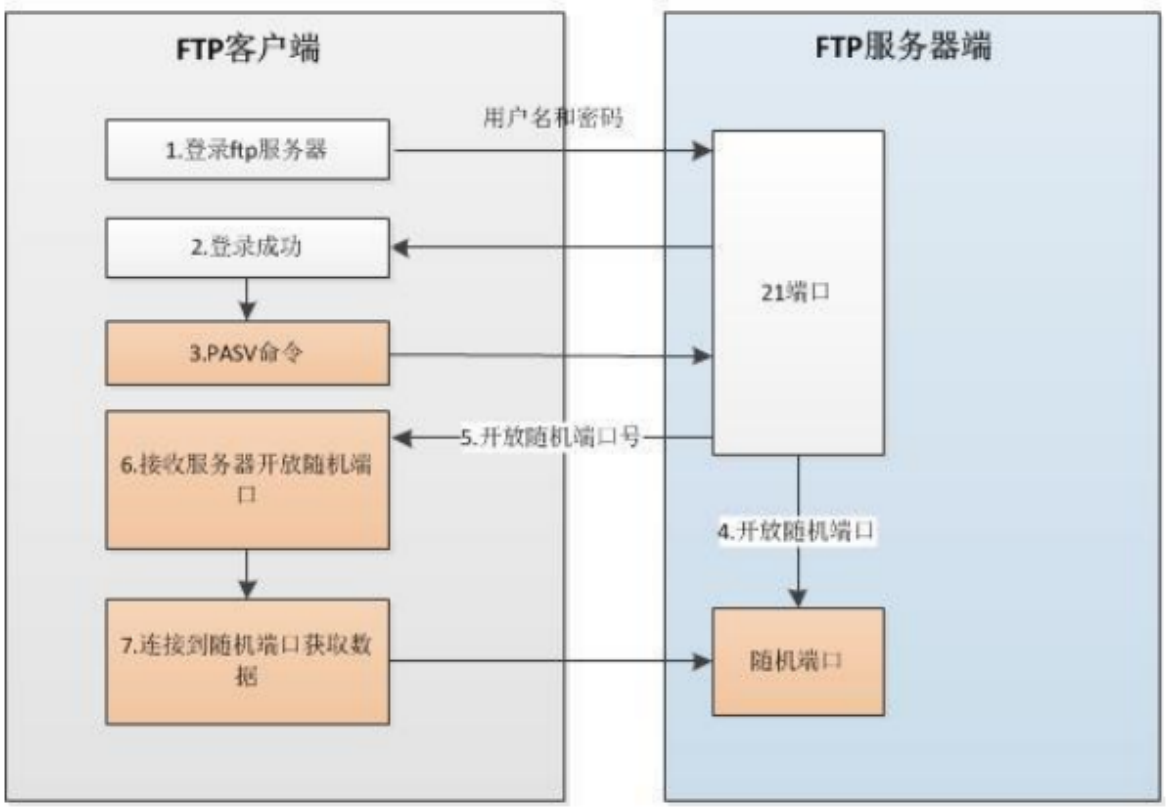
1.5.1 主动模式



Port模式：FTP客户端首先和服务器的21端口建立连接，用来建立控制通道，**客户端开放随机端口**，并通过PORT命令

告诉服务器自己的端口号。在传输数据时，服务器通过自己的20端口连接客户端的随机端口发送数据。

1.5.2 被动模式



Passive模式：FTP客户端首先和服务器的21端口建立连接，用来建立控制通道，建立连接后，客户端发送PASV命令

服务器收到PASV命令后，**开放随机端口** < 1023-65535 > 并且告诉客户端开放的端口号。客户端连接服务器开放的随机

端口，最后服务器通过这个端口传输数据。

1.6 VSFTP 传输模式

1.6.1 Binary 模式

不对数据进行任何处理，适合进行可执行文件、压缩文件、图片等。

1.6.2 ASCII 模式

进行文本传输时，自动适应目标操作系统的结束符，如回车符等。

1.6.3 切换方式

```
在 ftp> 提示符下输入 ascii，转换到 ACSII 方式
在 ftp> 提示符下输入 bin，转换到 Binary 方式
```

1.7 VSFTP 基本信息

1.7.1 服务端软件名

```
vsftpd
```

1.7.2 客户端软件名

```
lftp
```

1.7.3 服务名

```
vsftpd
```

1.7.4 端口号

```
数据端口 20/tcp
控制端口 21/tcp
指定范围内的随机端口
```

1.7.5 配置文件

```
# vsftpd 核心配置文件
/etc/vsftpd/vsftpd.conf

# 指定哪些用户不能访问 FTP 服务器
/etc/vsftpd/ftpusers

# 指定允许使用 vsftp 的用户列表
/etc/vsftpd/user_list

# vsftpd 操作的一些变量和设置脚本
/etc/vsftpd/vsftpd_conf_migrate.sh

# 默认情况下匿名用户的根目录
/var/ftp/
```

1.8 登录验证方式

1.8.1 匿名用户验证

用户账号名称：ftp/anonymous

用户账号密码：无密码

工作目录：/var/ftp

默认权限：默认可下载不可上传，上传权限有两部分组成 < 主配置文件和文件系统 >

```
# 匿名权限控制：
    anonymous_enable=YES              # 启用匿名访问
    anon_umask=022                    # 匿名用户所上传文件的权限掩码
    anon_root=/var/ftp                # 匿名用户的 FTP 根目录
    anon_upload_enable=YES            # 允许上传文件
    anon_mkdir_write_enable=YES       # 允许创建目录
    anon_other_write_enable_YES       # 开放其它权限 < 删除、覆盖、重命名 >
    anon_max_rate=0                   # 限制最大传输速率 < 0为不限速，单位：bytes/秒
>
```

1.8.2 本地用户验证

用户账号名称：本地用户

用户账号密码：本地用户密码

工作目录：登录用户的宿主目录

权限：最大权限 < drwx------ >

```
# 创建用户/密码

useradd -s /sbin/nologin god
passwd god
```

```
# 本地用户权限控制

local_enable=YES              # 是否启用本地系统用户
local_umask=022               # 本地用户所上传文件的权限掩码
local_root=/var/ftp           # 设置本地用户的 FTP 根目录
chroot_local_user=YES         # 是否将用户禁锢在主目录
local_max_rate=0              # 限制最大传输速率
ftpd_banner=Hello             # 用户登录时显示的欢迎信息
userlist_enable=YES&userlist_deny=YES       # 禁止 /etc/vsftpd/user_list 文件中出现的用户名登录
FTPuserlist_enable=YES&userlist_deny=NO     # 仅允许 /etc/vsftpd/user_list 文件中出现的用户名登录 FTP

# 配置文件：ftpusers
# 禁止 /etc/vsftpd/ftpusers 文件中出现的用户名登录 FTP，权限比 user_list 更高，即时生效
```

1.8.3 虚拟用户验证

创建虚拟用户用来代替本地用户，减少本地用户曝光率

使用本地用户作为虚拟用户的映射用户，为虚拟用户提供工作目录和权限控制

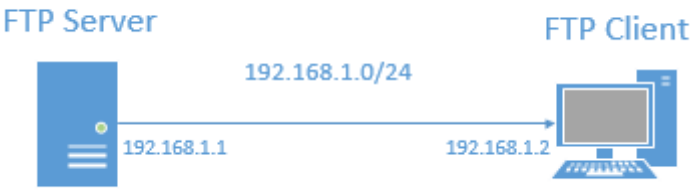
能够设置严格的权限 < 为每一个用户生成单独的配置文件 >

2、实验

2.1 匿名用户验证

2.1.1 实验需求及拓扑

```
1 公司技术部准备搭建一台功能简单的 FTP 服务器，允许所有员工上传和下载文件，并允许创建用户自己的目录。
```



-

2.1.2 关闭防护

```
1 systemctl stop firewalld
2 getenforce 0
```

1.1.3 安装软件包

```
1 yum install -y vsftpd
```

2.1.4 修改配置文件

```
1  cd /etc/vsftpd/
2
3  # 备份配置文件
4  cp vsftpd.conf{,.bak}
5
6  # 修改配置文件
7  vim /etc/vsftp/vsftp.conf
8  12 anonymous_enable=YES                # 启用匿名用户
9  29 anon_upload_enable=YES              # 允许匿名用户上传文件
10 33 anon_mkdir_write_enable=YES         # 允许匿名用户创建目录
11 34 anon_other_write_enable=YES         # 允许匿名用户删除、重命名目录
12
13 # 修改 FTP 共享目录权限
14 chown ftp:ftp /var/ftp/pub/
15
16 # 重启服务
```

```
systemctl restart vsftpd
```

注意，默认匿名用户家目录的权限是 755，这个权限是不能改变的！

2.1.5 测试

```
# 客户端登录测试
```

2.2 本地用户验证

2.2.1 实验背景及拓扑

```
公司内部现在有一台 FTP 和 WEB 服务器，FTP 的功能主要用于维护公司的网站内容，包括上传文件、创建目录、更新网页等等。公司现有两个部门负责维护任务，他们分别使用 team1 和 team2 帐号进行管理。先要求仅允许 team1 和 team2 帐号登录 FTP 服务器，但不能登录本地系统，并将两个帐号的根目录限制为 /var/www/html，不能进入该目录以外的任何目录。

# 只允许：team1 和 team2 用户可以上传，禁止匿名用户登录。

分析：
将 FTP 和 WEB 服务器做在一起是企业经常采用的方法，这样方便实现对网站的维护，为了增强安全性，首先需要使用仅允许本地用户访问，并禁止匿名用户登录。其次使用 chroot 功能将 team1 和 team2 锁定在 /var/www/html 目录下。如果需要删除文件则还需要注意本地权限。
```

2.2.2 关闭防护

```
systemctl stop firewalld
getenforce 0
```

2.2.3 安装软件包

```
yum install -y vsftpd
```

2.2.4 创建用户并禁止本地登录

```
useradd -s /sbin/nologin team1
useradd -s /sbin/nologin team2
echo 123 | passwd --stdin team1
echo 123 | passwd --stdin team2
echo "/sbin/nologin" >> /etc/shells
```

```
[root@hl11 ~]# useradd -s /sbin/nologin team1
[root@hl11 ~]# useradd -s /sbin/nologin team2
[root@hl11 ~]# echo 123 | passwd --stdin team1
更改用户 team1 的密码 。
passwd: 所有的身份验证令牌已经成功更新。
[root@hl11 ~]# echo 123 | passwd --stdin team2
更改用户 team2 的密码 。
passwd: 所有的身份验证令牌已经成功更新。
[root@hl11 ~]# echo "/sbin/nologin" >> /etc/shells
[root@hl11 ~]# cat /etc/shells
/bin/sh
/bin/bash
/usr/bin/sh
/usr/bin/bash
/bin/tcsh
/bin/csh
/sbin/nologin
[root@hl11 ~]#
```

2.2.5 修改配置文件

```
cd /etc/vsftpd/
vim /etc/vsftpd/vsftpd.conf
12 anonymous_enable=NO                # 禁止匿名用户登录
16 local_enable=YES                   # 允许本地用户登录
102 local_root=/var/www/html          # 设置本地用户的根目录。默认没有，添加即可
103 chroot_list_enable=YES            # 开启 chroot 功能
105 chroot_list_file=/etc/vsftpd/chroot_list    # 指定存放锁定用户的列表文件。此文件存放要锁定的用户名
106 allow_writeable_chroot=YES        # 允许锁定的用于有写的权限，默认没有添加即可
```

2.2.6 创建用户列表文件

```
vim /etc/vsftpd/chroot_list
team1
team2
```

2.2.7 创建测试文件

```
mkdir -p /var/www/html
chmod -R o+w /var/www/html/
```

```
[root@hl11 vsftpd]# mkdir -p /var/www/html
[root@hl11 vsftpd]# chmod -R o+w /var/www/html/
[root@hl11 vsftpd]# ll -d /var/www/html/
drwxr-xrwx 2 root root 6 1月    7 18:57 /var/www/html/
[root@hl11 vsftpd]#
```

2.2.8 重启服务

```
systemctl restart vsftpd
```

2.2.9 测试

```

```

2.3 虚拟用户验证

```
# 未测试

https://www.modb.pro/db/111811
```

2.4 openssl + vsftpd 证书加密验证

```
FTP 与 HTTP 一样缺省状态都是基于明文传输，希望 FTP 服务器端与客户端传输保证安全，可以为 FTP 配置 SSL
```

2.4.1 安装软件

```
yum -y install vsftpd
```

2.4.2 使用 OpenSSL 生成自签证书

```
# 创建证书文件存放目录
cd /etc/vsftpd
mkdir .sslkey

# 生成自签证书
openssl req -new -x509 -nodes -out vsftpd.pem -keyout vsftpd.pem -days 3650
OpenSSL 参数：
    req          # 是 X.509 < Certificete Signing Request CSR 证书签名请求 > 管理的一个命令
    x509         # X.509 证书数据管理
    days         # 定义证书的有效日期
    newkey       # 指定证书密钥处理器
    keyout       # 设置密钥存储文件。
    out          # 设置证书存储文件，注意证书和密钥都保存在一个相同的文件

mv vsftpd.pem .sslkey/
chmod 400 .sslkey/vsftpd.pem
```

```
[root@hl11 vsftpd]# openssl req -new -x509 -nodes -out vsftpd.pem -keyout vsftpd.pem -days 36
50
Generating a 2048 bit RSA private key
.....................................+++
.........................................+++
writing new private key to 'vsftpd.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:ZH
State or Province Name (full name) []:JS
Locality Name (eg, city) [Default City]:NJ
Organization Name (eg, company) [Default Company Ltd]:XS
Organizational Unit Name (eg, section) []:XS
Common Name (eg, your name or your server's hostname) []:XS.COM
Email Address []:XS@163.COM
[root@hl11 vsftpd]#
```

```
[root@hl11 vsftpd]# ll .sslkey/vsftpd.pem
-r-------- 1 root root 3038 1月   7 19:27 .sslkey/vsftpd.pem
[root@hl11 vsftpd]#
```

2.4.3 修改配置文件

```
vim /etc/vsftpd.conf
ssl_enable=YES      # 启用 SSL 支持
allow_anon_ssl=NO
# 以下四行表示强制匿名用户使用加密登录和数据传输
force_local_data_ssl=YES
force_local_logins_ssl=YES
force_anon_logins_ssl=YES
force_anon_data_ssl=YES
ssl_tlsv1=YES               # 指定 vsftpd 支持 TLS v1
ssl_sslv2=YES               # 指定 vsftpd 支持 TLS v2
ssl_sslv3=YES               # 指定 vsftpd 支持 TLS v3
require_ssl_reuse=NO        # 不重用 SSL 会话,安全配置项
ssl_ciphers=HIGH            # 允许用于加密 SSL 连接的 SSL 算法。可以极大地限制那些尝试发现使用存在缺陷的特定算法的攻击者。
rsa_cert_file=/etc/vsftpd/.sslkey/vsftpd.pem    # 指定 SSL 证书位置
rsa_private_key_file=/etc/vsftpd/.sslkey/vsftpd.pem     # 指定密钥文件位置

# 注意：上面的配置项不要添加到 vsftpd.conf 文件最后也不要加注释，且不能有多余的空格。
```

```
# with the listen_ipv6 directive.
listen=NO

#config
ssl_enable=YES
allow_anon_ssl=NO
force_local_data_ssl=YES
force_local_logins_ssl=YES
force_anon_logins_ssl=YES
force_anon_data_ssl=YES
ssl_tlsv1=YES
ssl_sslv2=YES
ssl_sslv3=YES
require_ssl_reuse=NO
ssl_ciphers=HIGH
rsa_cert_file=/etc/vsftpd/.sslkey/vsftpd.pem
rsa_private_key_file=/etc/vsftpd/.sslkey/vsftpd.pem
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
```

2.4.4 重启服务

```
systemctl restart vsftpd
```

2.4.5 测试

```
通过 FileZilla 测试
```

2.5 配置文件详解

```
/etc/vsftpd/vsftpd.conf
anonymous_enable=YES              # 是否允许匿名 ftp
local_enable=YES                  # 是否允许本地用户登录
write_enable=YES                  # 是否开放本地用户的写权限
local_umask=022                   # 设置本地用户文件权限掩码
anon_upload_enable=YES            # 是否允许匿名用户上传文件
anon_mkdir_write_enable=YES       # 是否允许匿名用户创建文件夹
dirmessage_enable=YES             # 当切换到某个目录时，显示该目录下的隐含文件
.message 的内容。
xferlog_enable=YES                # 激活上传下载日志
connect_from_port_20=YES          # 启用 FTP 数据端口的连接请求
chown_uploads=YES                 # 是否改变上传文件的属主
chown_username=username           # 指定上传文件的属主
xferlog_file=/var/log/vsftpd.log     # 日志存放位置
```

```
xferlog_std_format=YES          # 日志保存格式
idle_session_timeout=600        # 设置会话连接超时时间
data_connection_timeout=120     # 设置数据传输超时时间
nopriv_user=ftpsecure           # 运行 vsftpd 需要的非特权系统用户,默认是 nobody
ascii_upload_enable=YES         # 是否使用 ascii 码方式上传文件
ascii_download_enable=YES       # 是否使用 ascii 码方式下载文件
ftpd_banner=Welcome to blah FTP service!    # 定制欢迎信息
banner_file=/etc/vsftpd/welcome.txt         # 定制欢迎信息
chroot_local_user=YES                       # 是否将系统用户限制在自己家目录下
chroot_list_file=/etc/vsftpd.chroot_list    # 限制家目录用户列表
pam_service_name=vsftpd         # 设置 PAM 认证服务，该文件存放在 /etc/pam.d/ 目录下
userlist_enable=YES             # 启用用户访问控制。
userlist_file=/etc/vsftpd/user_list         # 用户访问控制列表
listen=YES                      # 使 vsftpd 处于独立启动模式
listen_address=192.168.9.201    # 监听 IP 地址
listen_port=21                  # 监听 21 号端口
tcp_wrappers=YES                # 使用 tcp_wrappers 作为主机的访问控制方式
max_clients=100                 # 最大客户端连接数
max_per_ip=5                    # 每客户端最大并发数


```