## 1、修改密码

apache ftpserver具有两种管理用户的方式，一种是文件管理，一种是数据库管理。

用户管理的需要再ftpd-typical.xml中配置

<file-user-manager file="./res/conf/users.properties" />

文件路径可以修改，文件内的具体配置不作说明。

数据库管理的同样需要在ftpd-typical.xml中配置

<db-user-manager>

<data-source>

<beans:bean class="org.apache.commons.dbcp.BasicDataSource">

<beans:property name="driverClassName" value="com.mysql.jdbc.Driver"/>

<beans:property name="url" value="jdbc:mysql://127.0.0.1/ftpserver"/>

<beans:property name="username" value="root"/>

<beans:property name="password" value=""/>

</beans:bean>

</data-source>

<insert-user>INSERT INTO users (userid, userpassword,

homedirectory, enableflag, writepermission, idletime, uploadrate,

downloadrate) VALUES ('{userid}', '{userpassword}',

'{homedirectory}',

{enableflag}, {writepermission}, {idletime},

{uploadrate},

{downloadrate})

</insert-user>

<update-user>UPDATE users SET

userpassword='{userpassword}',homedirectory='{homedirectory}',enableflag={enableflag},writepermission={writepermission},idletime={idletime},uploadrate={uploadrate},downloadrate={downloadrate}

WHERE userid='{userid}'

</update-user>

<delete-user>DELETE FROM users WHERE userid = '{userid}'

</delete-user>

<select-user>SELECT userid, userpassword, homedirectory,

enableflag, writepermission, idletime, uploadrate, downloadrate,

maxloginnumber, maxloginperip FROM

users WHERE userid = '{userid}'

</select-user>

<select-all-users>SELECT userid FROM users ORDER BY userid

</select-all-users>

<is-admin>SELECT userid FROM users WHERE userid='{userid}'

AND

userid='admin'

</is-admin>

<authenticate>SELECT userpassword from users WHERE

userid='{userid}'</authenticate>

</db-user-manager>

可以修改里面的表名，或者增加查询字段。

注意：使用数据库管理用户时，必须要开启mysql服务，否则无法启动ftpserver，会报Failed to open connection to user database错误。

apache ftpserver可以自定义命令，在ftpd-typical.xml中配置

<commands>

<!--修改密码，必须使用数据库管理用户-->

<command name="SETPASS">

<beans:bean class="com.ftplet.command.SETPASS"></beans:bean>

</command>

<!--ad域验证-->

<command name="PASS">

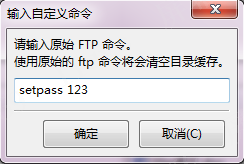
<beans:bean class="com.ftplet.command.ADPASS"></beans:bean>

</command>

</commands>

name：命令名，如果与ftp命令相同会覆盖原始ftp命令。

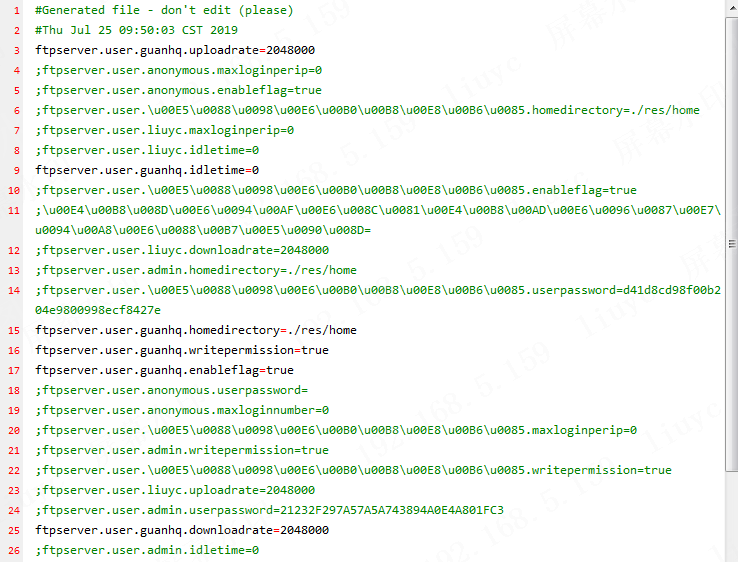
通过这种自定义命令的方法，就能实现修改密码的功能，我们自定义一个setpass的命令的逻辑，就能在客户端发送自定义命令来修改密码。



对于这个修改密码的逻辑，使用系统原有的函数即可，它会判断你是文件管理还是数据库管理而对应的修改该用户的信息。

注意：两种管理方式在修改密码后都不用重启服务器，之前我说当文件管理的时候需要重启才生效，那个时候我不是调用系统的函数，而是自己实现的修改文件的方法，所以过程是错误的。

（1）文件管理的时候，会修改文件的内容，大概会变为这个样子



文件的内容都会乱了，而且原本用来测试中文用户名的也看到已经被转码（文档编码ANSI），再次测试value值有中文的情况也会被转码，但仍能继续访问

ftpserver.user.guanhq.homedirectory=./res/home/中文

ftpserver.user.guanhq.homedirectory=./res/home/\u00D6\u00D0\u00CE\u00C4

而用户名为中文的情况下，就算被转码了，也是无法登录。

（2）数据库管理的时候就是根据配置的sql语句（来自ftpd-typical.xml）来修改数据库的内容。

## 2、域验证

由于要使用不同的登录逻辑，而客户端在登录的时候是固定发送user和pass命令的，所以我们只能修改pass来达到改变登录的逻辑。

在自定义命令中配置PASS命令来覆盖原有的PASS命令。

大概的逻辑是，获取user命令中的用户名和pass命令中的密码，然后使用ldap进行ad域验证，如果验证通过，就使用系统原有的函数getUserByName来获取用户的基本信息，即仍然是要使用文件或数据库管理用户，并且ftp的用户上要有通过ad域验证的用户，这样才能使用原有的函数方便的获取该用户的信息。即：ad域中有liuyc，那么用户文件或数据库中也要有liuyc，才能获取liuyc的主目录等所有信息。

当然，我们可以不使用系统原有的函数来获取用户信息，可以通过其他的途径来获取例如网络获取、工具获取等。当使用这种方法时，ftpd-typical.xml中可以完全不用配置<file-user-manager>或者<db-user-manager>，影响后果未发现，因为在登录的时候就已经为用户设置相应的用户配置，所以后续的使用都没有读取file或者db了。如果在登录之外的其他功能上仍然有获取file或db的信息的代码，那就还需要在ftpd-typical.xml中配置file或db管理。

## 3、加密通讯

### 3.1、简要

我们先看一下官方的文档[**TLS-SSL Support**](http://mina.apache.org/ftpserver-project/configuration_ssltls_support.html)，主要是说了如何使用TLS为通讯加密。默认是显式ssl，在此模式下，服务器同时支持安全连接（ftps）和非安全连接（ftp）。根据来自客户机(AUTH SSL)的请求，服务器切换到SSL/TLS模式。如果服务端设为隐式ssl，那客户端以显式ssl就无法访问会超时。

### 3.2、相关概念

**explicit-ssl：（显式ssl）**， 在与ftp服务器建立连接后，ftp客户端要以命令（"AUTH SSL" 或者 "AUTH TLS"）显式地告诉服务器端来初始化相应的安全连接。此时使用的是默认的ftp端口21。

**implicit-ssl：（隐式ssl）**，当ftp客户端连接到服务器端时，服务器端自动建立安全连接。此时，客户端默认以990端口来安全连接服务器端，而服务器端端口可设置。

参考[**SSL与TLS的区别以及介绍**](https://www.cnblogs.com/susanhonly/p/7489532.html)

**SSL：（Secure Socket Layer，安全套接字层）**，位于可靠的面向连接的网络层协议和应用层协议之间的一种协议层。SSL通过互相认证、使用数字签名确保完整性、使用加密确保私密性，以实现客户端和服务器之间的安全通讯。

**TLS：（Transport Layer Security，传输层安全协议）**，用于两个应用程序之间提供保密性和数据完整性。

**防混淆：**

（1）FTPS (FTP encrypted with TLS)，ftp-over-ssl或者ftp-over-tls，这个apache ftpserver可以在配置文件ftpd-typical.xml的ssl节点上配置protocol="SSL"或者protocol="TLS"，即apache ftpserver支持ftps。

（2）SFTP (SSH)，ftp-over-ssh，利用ftp服务器和ssh协议加密传输数据。apache ftpserver好像没有提到过ssh，官网也只出现TLS-SSL，可能是不支持sftp。

原文[Ftp、Ftps与Sftp之间的区别](https://www.cnblogs.com/Javi/p/6904587.html) ，提到**SFTP 和FTPS的区别：**

SFTP 和FTPS都是为ftp连接加密，协议非常相似。一个是借助ssl协议加密，一个时借助ssh加密。

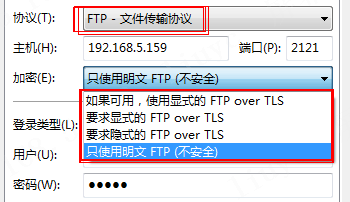
ssl是为http/smtp等加密设计的，ssh是为telnet/ftp等加密、建立传输通道而设计的。

其实ssh建立传输通道就是为了加密和传输，而且这个通道是可以用来远程登录。如果只说它们的功能，通俗的讲，ssh就像铺管子，ssl就像打包裹，铺管子和打包裹都会使数据安全，都是一个制作密钥的过程，而因为ssh是一个管子所以它很适合ftp的安全传输。

### 3.3、配置测试

3.3.1、FileZilla

客户端使用FileZilla。修改图示的地方进行测试：



（1）

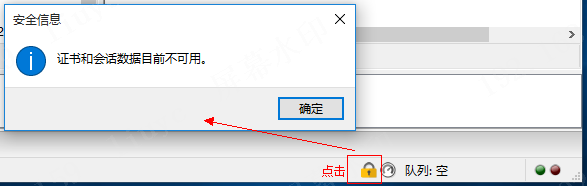
apache ftpserver不做任何配置，即按官网文档说的默认显式ssl；

客户端设置显式ssl连接方式（显式的FTP over TLS）。

结果：



然后在FileZilla客户端下方状态栏看到：



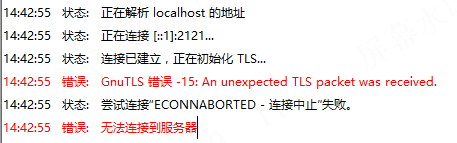
所以很有可能是客户端和服务端仍然配置有问题，或者不支持过期证书等可能。

（2）

apache ftpserver不做任何配置；

客户端设置隐式ssl连接方式（隐式的FTP over TLS）。

结果：

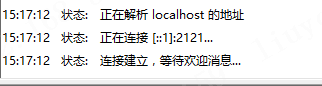


（3）

apache ftpserver为nio-listener节点设置implicit-ssl="true"，即要求隐式ssl；

客户端设置显式ssl连接方式。

结果：

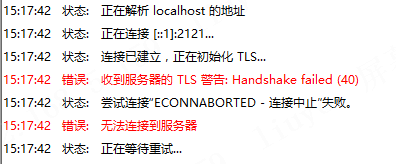
直至超时。

（4）

apache ftpserver为nio-listener节点设置implicit-ssl="true"，要求隐式ssl；

客户端设置隐式ssl连接方式。

结果：



（5）

apache ftpserver不作任何配置；

客户端设置sftp连接方式。

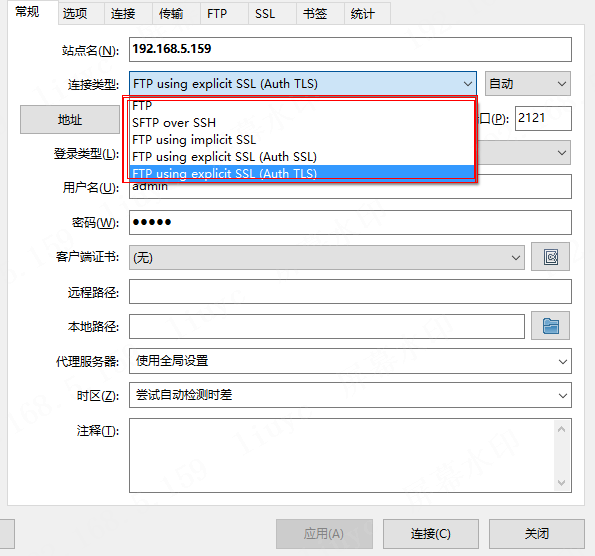


结果：

没有响应直到超时。

3.3.2、FlashFXP

客户端使用FlashFXP，这里说明FlashFXP的原因是FlashFXP能通过TLS连接上apache ftpserver，而FileZilla一直未连上，未清楚解决方法（[第3.4](#_3.4、修改证书)更换证书已经能解决）。

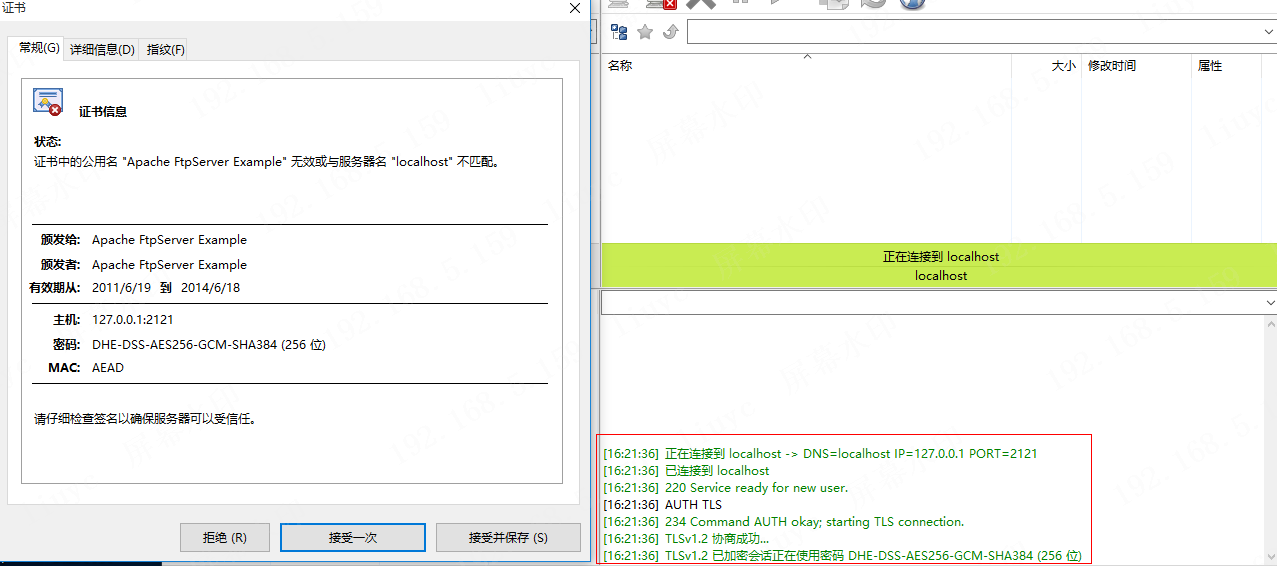


（1）

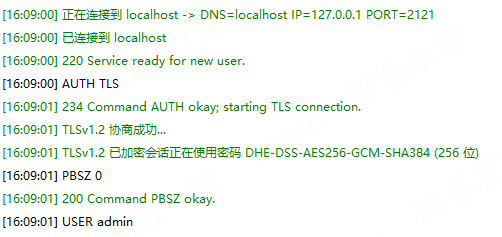
apache ftpserver不作任何配置；

FlashFXP设置连接方式为显式ssl（FTP using explicit SSL（Auth TLS））。

收到提示



点击接受，成功连接上服务器：



并且它是使用TLSv1.2协议。

如果这个时候，把客户端的协议改为自动、TLSv1、TLSv1.2，效果一样。

FTP using explicit SSL（Auth SSL）效果一样，正常访问。

（2）

apache ftpserver不作任何配置；

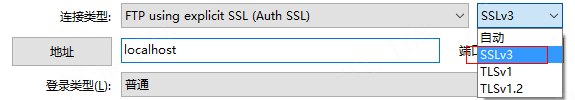
FTP using implicit SSL无法连上，符合预期。



（3）

apache ftpserver不作任何配置，即保持默认的protocol="TLS"；

把客户端连接方式改为显式ssl（Auth SSL），和使用SSLv3版本；



无法连接，提示：



就算服务端设置protocol="SSL"，同样无法连接。根据报错，可能FileZilla也是使用了ssl，所以同样报handshake failure.（[第3.4点](#_3.4、修改证书)已经验证，不是协议版本的问题，FileZilla也是使用TLSv1.2，应该是会自动切换的）

（4）

apache ftpserver不作任何配置；

客户端改为隐式连接（FTP using explicit SSL（Auth TLS））和自动选择协议版本。

结果：能正常访问。

（5）

服务端设置隐式ssl（implicit-ssl="true"）；

客户端设置显式连接（（Auth SSL）或者（Auth TLS）），同样的结果：

，无法访问，和FileZilla client一样的然后会超时，超时是符合预期的。

（6）

服务端设置隐式ssl（implicit-ssl="true"）；

客户端设置隐式连接（FTP using implicit SSL）

结果：能正常访问。

（7）设置连接类型为SFTP



结果：



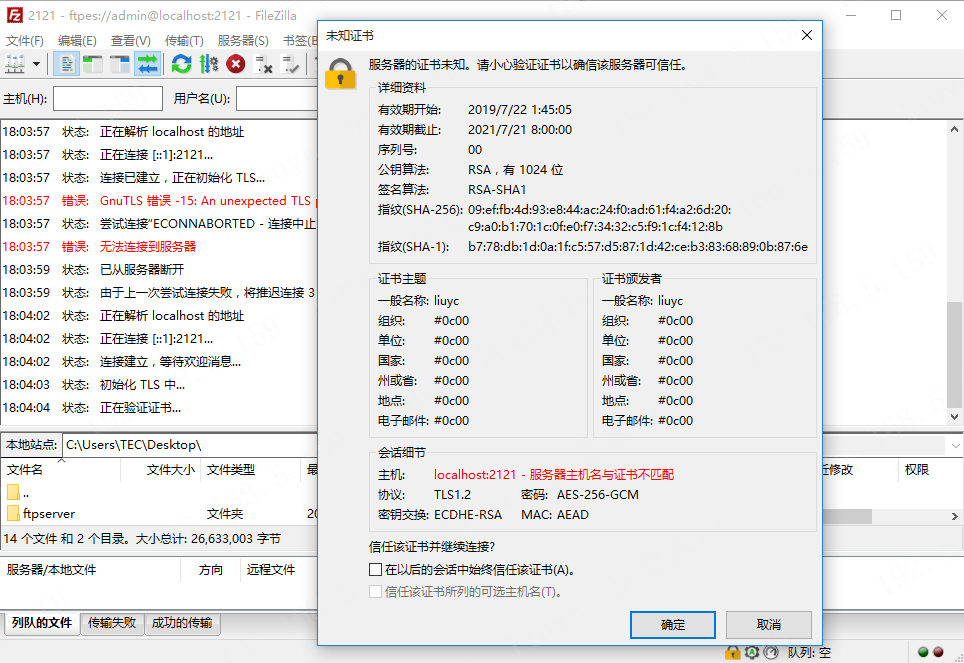
### 3.4、修改证书

创建一个新的jks文件。这里创建证书的步骤是使用FlashFXP的密钥管理器来生成

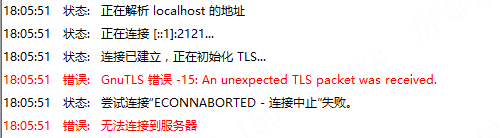


然后导出为pem格式，然后在<https://myssl.com/cert_convert.html>中进行pem转jks，转的时候还需要私钥，使用openssl提取私钥在网上搜索方法，最后记得在ftpd-typical.xml中修改jsk路径。

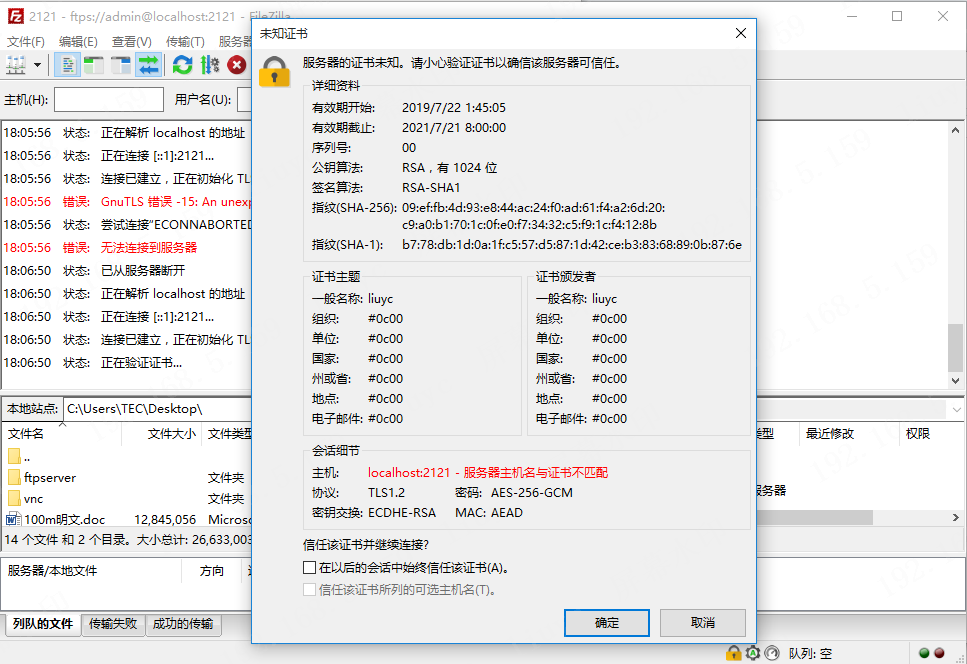
（1）服务端不作任何设置，然后FileZilla使用显式TLS能成功连接上，提示：



但隐式TLS无法连接，符合预期：



（2）然后服务器改为隐式ssl，客户端可以通过隐式TLS连接了



这时客户端改为显式，符合预期会超时无法访问。

### 3.5、总结

通过3.4，FileZilla应该是不接受过期的证书，所以才一直都连不上，改为了自己创建的证书就成功了，而FlashFXP一直都能成功连接，这里说的连接是指ftps的连接。而对于sftp，个人认为apache ftpserver不支持，文档中没有提到，在配置方面也不太清楚，所以一直都未成功使用sftp连接。有人说道ftpserver不支持sftp：<http://mail-archives.apache.org/mod_mbox/mina-ftpserver-users/201207.mbox/%3C1342452432.49817.YahooMailNeo@web45114.mail.sp1.yahoo.com%3E>，而另一个项目apache sshd才是提供sftp支持。