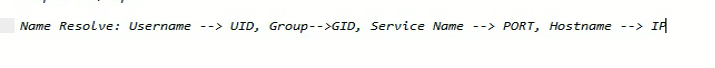
**名称解析 Name Resolve**

将一个服务名转换为服务的端口号 也是名称解析

将一个主机名转换成主机IP也是名称解析



名称解析的面非常多 ----- 每一种名称解析的解析库不太一样

对于用户名和ID对应 应该存储到一个数据库中 linux中 是 UserName 🡪 UID: /etc/passwd



昨天每一个邮件用户就是系统用户 ----- 那么/ect/passwd也可以实现名称解析

同理 这个对应关系也可以存储在MySQL中

**==== 主要有一种存储 或者数据存储格式 帮我们保存这种对应关系 都可以称之为解析库 ---- 不同的库中 需要依赖的程序不一样**

文件中的话 按行找

但是 这种解析库存放在MySQL中 需要SQL

**===== 不同的解析库 需要不同的查询方式**



Smtps并不好用 ---- 后来有一个从用户到用户的邮件安全的协议 S/MIME 对应的工具是OpenSSL和gpg(PGP的一种实现)

============================== 以上回顾的是邮件服务 ========

FTP：



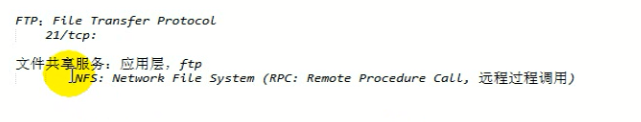
FTP是文件传输协议 File Transmission Protocol 监听在TCP的21端口 ----- 是古老的协议 和SMTP诞生的时间差不多 比HTTP早很多 ----- 是不同主机之间的文件共享服务 ---- 是工作在应用层 TCP上层的一种特殊的协议完成的 ----- **这个和基于RPC的文件共享服务有着巨大的不同 ---RPC能够让两台主机上的进程直接依赖于二进制协议完成通信**

RPC：Remote Procedure Call ---- 远程过程调用 ----很多服务都用到了

----**让两台主机上的进程直接依赖于二进制协议进行通信**

基于RPC的共享服务也有很多 比如 Hadoop OpenStack都依赖于RPC

文件共享服务中 FTP工作在应用层 后面的NFS（网络文件系统） 实际上依赖于RPC



还有第三种一种文件共享服务 Samba ---- 是CIFS/SMB协议的实现 ----- 跨平台文件共享 共享的机制是比较底层 CIFS ---Common Interface File System

Windows的网上邻居就是基于CIFS Linux的Samba就是基于这种协议 所以可以让异构系统 --- 也就是让linux出现在windows的网上邻居中

企业中FTP也会很多 但是配置和应用很简单 互联网上直接提供FTP服务器少 因为我们使用HTTP就能实现文件共享服务

FTP是文件传输协议 工作于应用层 只需要依赖于互联网Internet就可以完成通信 --- 因为是古老的协议 考虑了文件传输的可靠性 所以 基于了传输层的TCP协议 ----

还有一点就是FTP的诡异的地方 就是FTP有两个连接

说到HTTP的时候 需要一个连接 客户端发起请求 就和服务端建立起来一个连接 客户端和服务器直接通过这个链接完成数据传输

但是 对于FTP进行数据传输需要两个链接 一个是命令链接(也叫控制链接) 另一个是数据连接【非常类似于计算机组成原理中的 命令总线和数据总线进行数据传输 ---- 只是 计算机组成中 总线是物理存在的 但是 这里面的链接 都是无线的 基于TCP的连接】

控制连接是一直在线的 但是 数据连接只有在客户端发起请求的时候 才会按照需要打开数据连接 文件传输完成 数据连接就关闭了 ---- 所以 数据连接是按需打开 按需关闭

FTP类似HTTP一样 支持文本协议 --- 支持文本协议 都可以telnet

有专用的客户端 类似于CS模式

当客户端发起

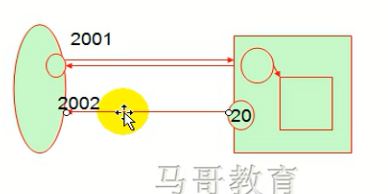
基于TCPIP协议 要套接字完成

数据连接监听的端口是什么？

控制连接是监听在TCP的21号端口

但是数据连接未必 要看数据连接是谁主动建立的

数据连接可以工作在20号端口 可以主动连接客户端的端口 客户端的端口是随机的

 如果客户端的2002被占用 就会依次向后找 找到之后 建立数据连接

---- 这时候服务器是主动连接（站在服务器的角度来看 服务器主动）

FTP主动用自己的20号端口连接客户端【HTTP协议都是被动的 不会服务器主动】

---- 主动模式有不好的地方 要考虑防火墙的 一般考虑安全问题 客户端都会有防火墙

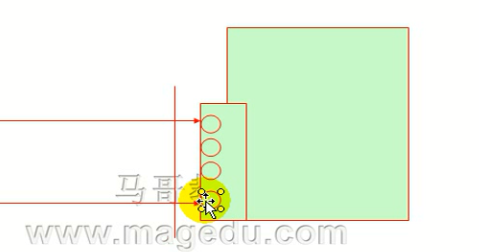
**------ 防火墙和杀毒软件有什么区别？**

杀毒软件是检测能访问的每一个文件或者报文是否有病毒存在 --- 通过检测文件的内容判断是否有危险 并且防止我们访问危险的文件或者软件的软件

TCP和UDP上都提供了65536个端口

每个端口就相当于外部访问你的主机的大门 接入互联网之后 相当于12万多个大门可以被访问 随意进出 ---防火墙就是把没用的门关掉 ---- 避免非请自来的连接挡在门外 ---- 防火墙就是关闭用不到的门

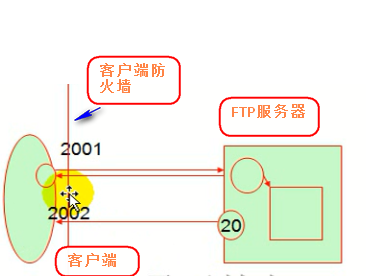
有些门是有用的 Web服务器需要客户端80端口

 可以屏蔽非请自来的外部请求 但是 可以自己随意进出 ---- 这个门如何开放关闭 都是自己定义 什么时候出去 什么时候进来 都是可以定义的 ---- 这就是防火墙规则 ---- 拉一个人过来的时候 是可以进来的

这种会话连接中 分谁请求进来的

----- 主要我们自己服务器 只有我们访问别人 才响应我 但是 如果不提供任何服务 别人是不需要访问的 这样客户端才是足够安全的

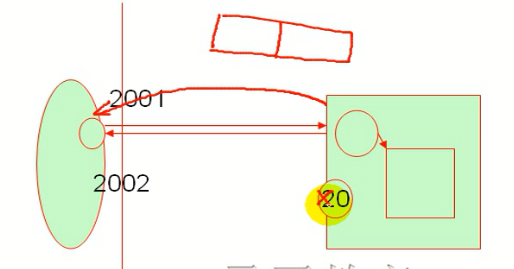
----- FTP服务器 客户端为了安全做了防火墙

 这样如果服务器主动发起连接到客户端的2002端口 但是 客户端做了防火墙 这个时候 服务器的主动请求就会被拒之门外 ---- 这样FTP主动对客户端发起的连接被防火墙戛然而止

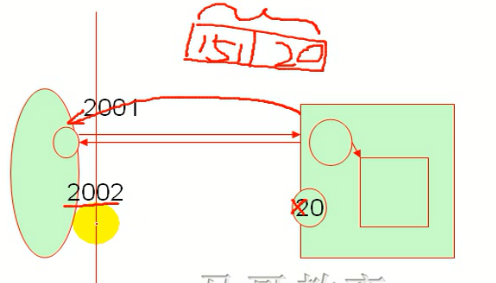
----- **这种情况并不理想 所以 就有了后面的被动模式**

还有被动模式

此时FTP服务器仍然通过命令连接 发一个报文给客户端 告诉客户端我监听在一个端口上 你来连接我吧



客户端使用自己的2002端口来连接刚才服务器端通过命令连接发送过来的新端口 ---- 这个新端口的表现方式如下：

 这个报文有两个数字 第一个数字是151 另一个是20 ---- 这样服务器可以被客户端连接的端口 就是 151\*256+20 这个服务器端口可以被连接

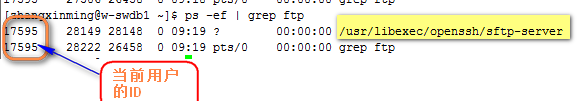
【比如 一开始的时候 我的linux虚拟机是没有ftp服务的



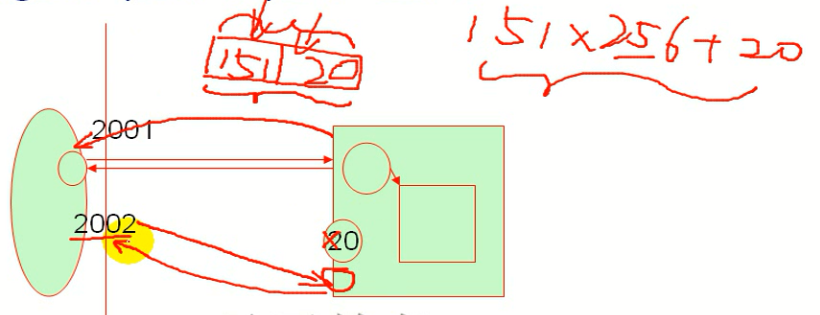
之后呢 我点击了下面的按钮

这样我就可以建立文件共享 说明 linux虚拟机此时有Ftp服务器的功能

此时重新检查一下进程

 就是我这个客户端的用户建立了一个ftp服务器】

这时候 ftp服务器是被动的 所以数据连接在服务器不再在20端口





这时候又会出现问题 因为服务器端的端口是随机的

如果服务器端有防火墙 ---- 每次客户端要连接FTP服务器的端口都是不确定的

如果服务器上的每一个端口都开放 这样安全就不存在 --- 这样做不理想

现在防火墙都有一种连接追踪的功能

防火墙能够识别请求响应彼此之间的关系

TCPIP第一次握手是new状态

服务器建立连接的响应 是established

客户端请求之后 服务端才打开 --- 每一个数据连接一定和命令连接相关联 但是 这两个连接还是两个连接 ---- 但是 控制连接和数据连接这两个连接是有关联的 是related的

----- 所以 防火墙无论这两个连接的端口是什么 会追踪出来相关联的connection

所以 这就是FTP的主动模式 被动模式 和 防火墙的关系【后面会在防火墙详细讲】

【这样 因为21号端口是开放的 所以 服务器上的防火墙就会查询到和这个21号端口相关联的连接的端口是哪个 也会开放】

**FTP的数据传输模式**

HTTP协议把数据包从客户端传向服务端 发送的是文本文档 二进制文档

如果把一个文件在FTP客户端传到FTP服务端 是哪种二进制 还是文本？

HTTP可以把二进制通过MIME这种 以文本的形式传送二进制文件 --- 有了MIME 在客户端会把这个文本重新还原

但是FTP没有这种能力 --- 所以FTP实现数据传输 到底是文本还是二进制 要遵循文件本身的特性 --- 也就是文件本身是文本的 那就按照文本传 否则就按照二进制传 不能交叉编码

所以 不能统一传输模式 所以 FTP传输模式有两种模式 二进制 和 文本 ---遵循文件本身的格式 ---- 所以这时候 需要服务端和客户端自行选择数据传输的模式 --- 否则会带来致命的错误

数据在互联网上传输 两个计算机交互 A计算机的内容 B未必能理解

 这个A发出来 给B 是怎么理解？因为过来的就是01 所以 是从左向右 还是从右向左 是文本还是二进制

两台计算机 数据共享或者传输是非常麻烦 更何况还有编码的情况

所以 数据在互联网有三中类型：结构化数据 半结构化数据 和 非结构化数据 具有这三种类型

FTP是一种协议 要实现这种协议 需要客户端和服务器端的程序

上面是Linux的

服务器端：

\*\*\*\*\*wuftpd ---- Washington university ftp Daemon ----最早的 ftp是鼻祖级的 没有采用模块化的设计思想 导致一个地方出现问题

\*\*\*\*\*红帽默认使用 vsftpd ---- very secure ftp Daemon ---短小精悍 着眼于安全精悍 功能未必丰富

\*\*\*\*\*Proftpd --- 功能强大 专门做ftp的 professional ftp

\*\*\*\*\*pureftpd ---

Windows上面也有ftp

\*\*\*\*\*Serv-U

\*\*\*\*\*Linux和Windows两端都有的开源FTP服务器FileZilla

客户端和服务器端都有

============

客户端程序

CLI格式：

ftp iftp

GUI格式

Linux gftpd

Windows flashfxp cuteftp filezilla

红帽默认提供的是vsftpd这个

Vsftpd是一个服务端的程序 支持 主动传输 被动传输

Rpm安装 配置文件在/etc/vsftpd下面

服务器脚本在/etc/init.d/vsftpd 主程序

Vsftpd支持虚拟用户 不是系统用户 放在另外一个存储空间的 这种都成为虚拟用户 不能登陆

Vsftpd基于PAM进行用户认证

所以 要让vsftpd支持虚拟用户 需要修改PAN

PAM是框架 可以按需定制一种认证模式



PAM 可插入式认证模块

确保关闭selinux 才能让ftp工作

Vsftp对文件权限检查是非常严格的

对/var/ftp root用户以外的用户不能有写权限的 这一点非常重要

一般ftp应该支持上传和下载

上传 ---- 文件从客户端和服务器 这种服务器的风险大得多

下载 --- 反过来 风险小

运行vsftp这个进程的用户和组通常是ftp用户 ftp组 那么 这个用户 无法把文件放置到/var/ftp这个目录下面 ==== 这个进程 vsftp进程一定是不能让运行这个进程的用户在/var/ftp目录具有写权限的

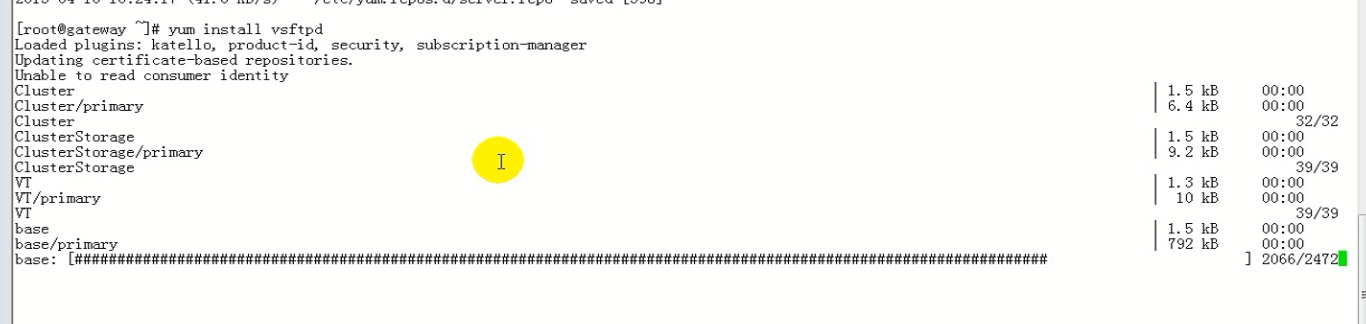
如果需要上传 可以在/var/ftp下面建立一个子目录 根目录不能有写权限（root用户除外）

---- 这是安全性的表现

========= 安装ftp服务器软件 ============



程序名称就叫vsftpd



 大小只有143K

下面看一下安装完之后 生成了哪些文件



/etc/logrotate.d/vsftpd.log这个是日志文件

/etc/pam.d/vsftpd因为vsftpd是基于PAM进行用户认证 座椅 在pam.d下面构建了一个vsftpd文件

/etc/rc.d/init.d/vsftpd是vsftpd这个服务的服务控制脚本

/etc/vsftpd这个是vsftpd本身的配置文件的目录

这个下面生成了四个配置文件

 这个是vsftpd的主配置文件

/ect/vsftpd/user\_list和/ect/vsftpd/ftpusers这是控制用户登陆的

/usr/sbin/vsftpd是vsftpd的主程序

剩下大部分就是vsftpd的文档

 /var/ftp是ftp服务的根目录

ftp --- 一个人可以到我们的主机下载文件 那么 应该有一个账号 ---- 这样ftp有用户的说法

ftp有三类用户：

匿名用户 系统用户 虚拟用户 ---- 这三类ftp用户 无论是哪一种 都要映射为linux的系统用户

这样匿名用户 和 虚拟用户都有一个与之对应的系统用户

所以 一个用户到ftp服务器上面访问对应的文件 一定是访问那个用户的家目录

比如/var/ftp就是某一个用户的家目录 ----- 应该是ftp用户的家目录

【使用finger命令查看一个用户的家目录



Finger表示指纹的意思

直接yum install finger

】



看一下系统是否有一个叫vsftp的用户？



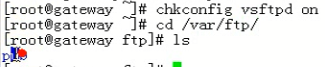
没有 说明 运行的时候 用户的名称是ftp

**ftp这个用户 就是*匿名用户*访问的目录 --- 被映射为ftp用户**

下面直接启动服务

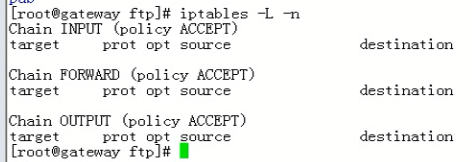


检查一下vsftpd服务启动的情况

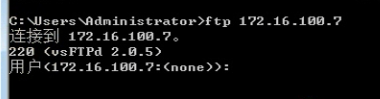


有一个pub目录 共享文件都放到pub下面 默认用Pub存储共享的文件

确保防火墙关闭



使用windows的ftp连接linux的ftp

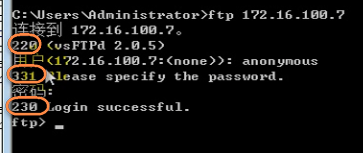
 --- -告诉对方的ftp服务器是vsftpd服务器

这个windows连接上去 就返回一个响应码220

刚才说了 匿名用户访问的目录 都是/var/ftp 同时 匿名用户被映射为系统的ftp用户

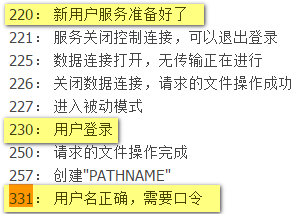
【】

所以 这里面登录的时候 用户名虽然是匿名的 但是 也要写anonymous 密码是空的

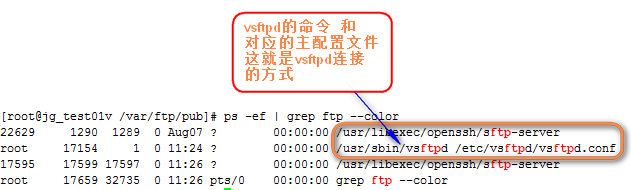


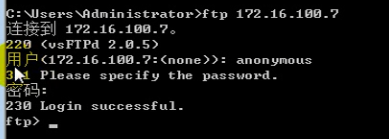
331是3开头的 说明所填写的信息不全 需要不全 这里面仅仅填写了用户名 没有填写密码

最后 就是返回230这个状态码

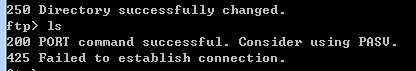
 口令指的就是密码 这里面列举出刚才看到的链接

【如果自身安装完ftp 又启动了那个按钮

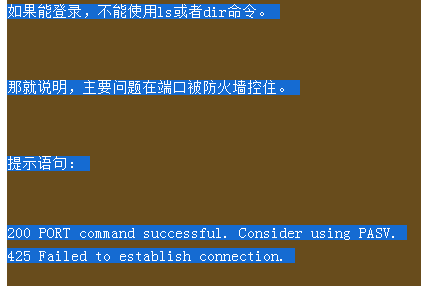
】

登陆成功

使用help命令 ftp客户端可以使用的众多命令

【

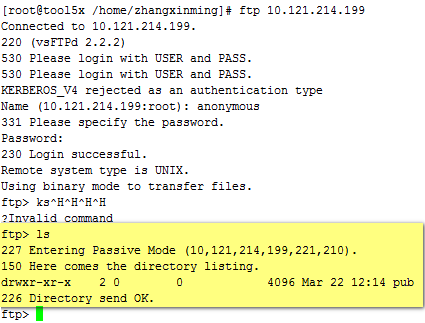
出现了425的响应码

查询了一看  
 

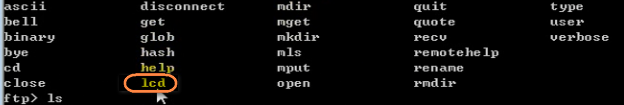
修改了服务器的防火墙 都没有问题 但是windows客户端 还是出现这个问题

最后 使用linux服务器 来访问这台ftp server就可以了

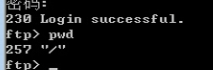
只需要在这台ftp server上面使用yum install ftp 安装ftp客户端

 ---- windows端 关闭防火墙之类的 都无法解决 --- 后面自己解决】

使用get下载 下载到哪里？ 就在登录之前的目录中

如果想下载到别的目录 该怎么办 使用这里面提示的lcd命令

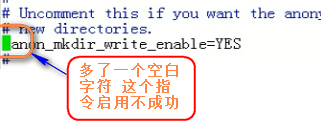
Mget可以一次下载多个文件 put是上传 mput批量上传

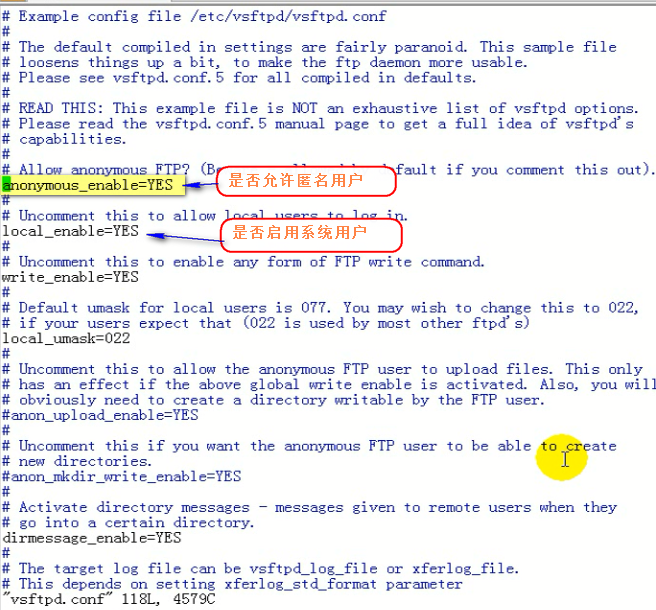
Pwd显示当前路径  ftp的/根目录 实际上是对应的linux上面的/var/ftp/目录

这样 锁定用户在自己的家目录下

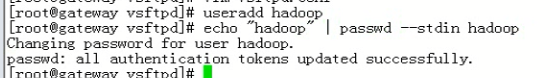
---- 看看vsftpd的配置文件

在配置文件中 任何一个指令之前 如果去掉了注释 并不代表这个指令启用了 必须去掉空白字符

 有空白字符是语法错误

 这个配置文件中 既开启了匿名用户 又开启了系统用户

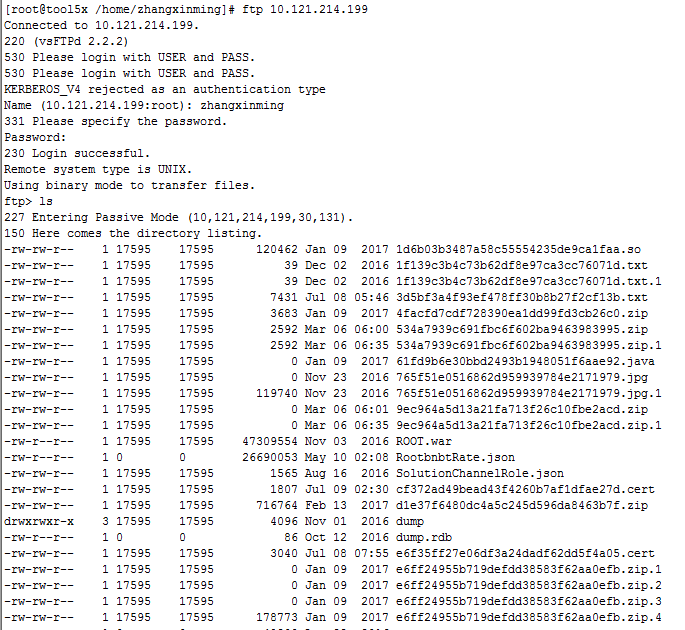
现在测试下系统用户



创建了一个Hadoop用户 并且 密码是hadoop



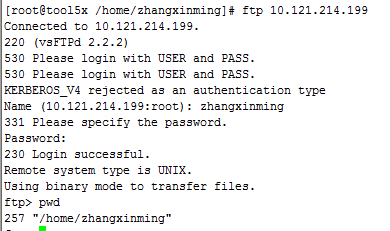
【我使用自己的用户 登录199的ftp服务器

】

说了 每个用户通过ftp访问的是对应服务器的用户的家目录

每个用户在自己的家目录下面

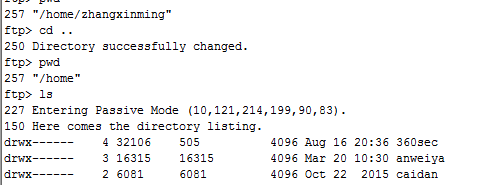
 对于系统用户而言 使用的是linux用户真正的家目录 而不是锁定在改变之后的根目录

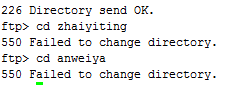
【】

现在能否到对应的父目录

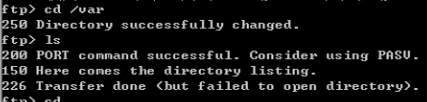
 这个到别人的home目录是不可以的

看看能否到etc下面

【 但是 进入别人的目录是不可以的

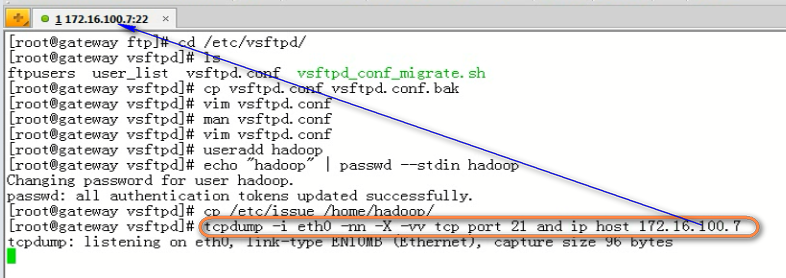
】

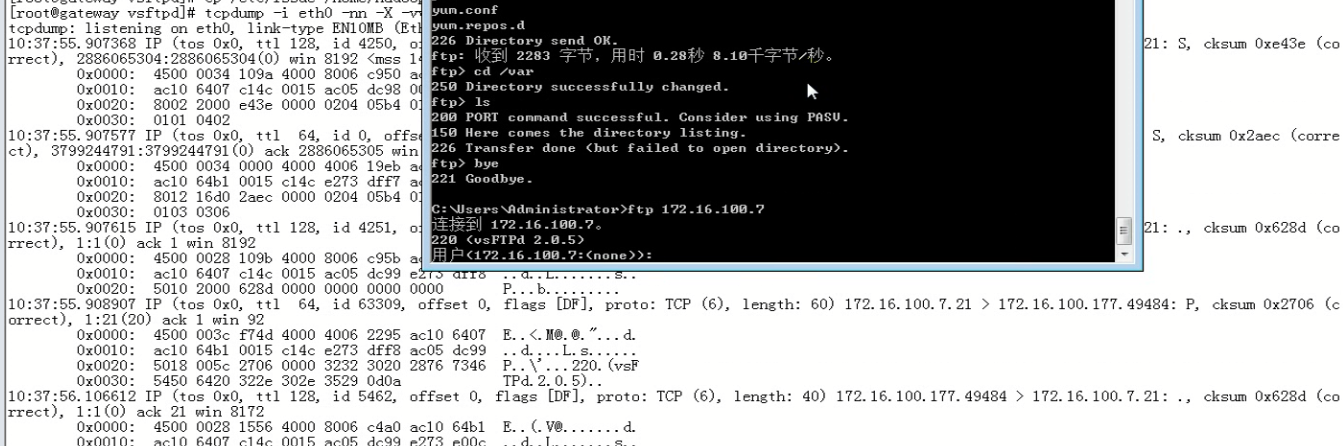




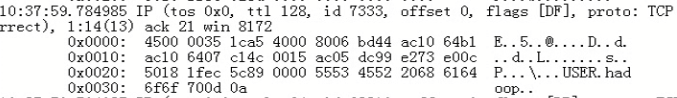
不禁锢在家目录是非常危险的 可以非常游荡

现在抓包试试



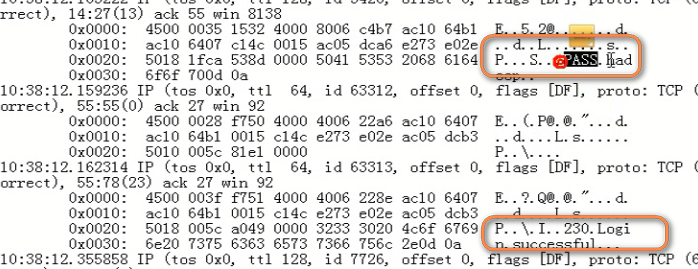


登录hadoop



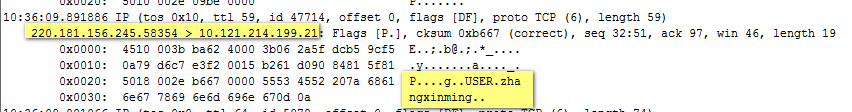
抓到这个hadoop的包

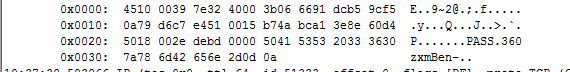
登录看看密码



这样 用户名 密码的认证都是明文的 所以 人家很容易拿到你的保密信息

所以 最好只能锁定用户在自己的家目录中 

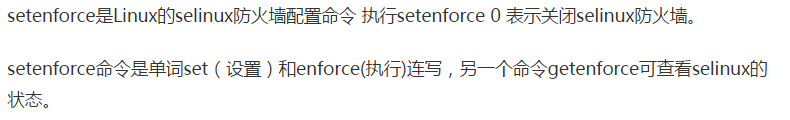
【

】

这样是两类用户 以及访问时候的特性 看下面的配置



使用这个命令之前 确保setenforce这个命令是关着的

【】



Write\_enable是用来定义系统用户是否可以上传文件

刚才的hadoop用户是可以上传文件的

我们使用浏览器来登录

这个是匿名用户

【可以看到 浏览器是可以解析ftp请求的 ftp://172.16.100.7】

如果要使用linux的系统用户登录 ftp://hadoop@172.16.100.7



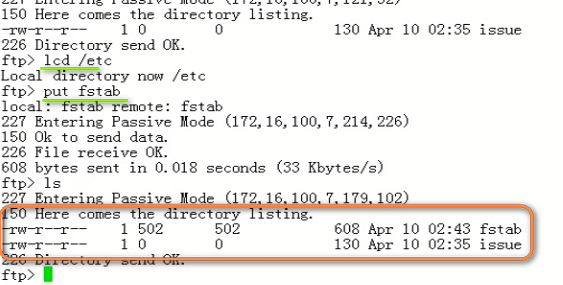
【windows上面 我可以使用我的浏览器进行登录 没有问题





  这个用户被禁锢在自己的家目录了】

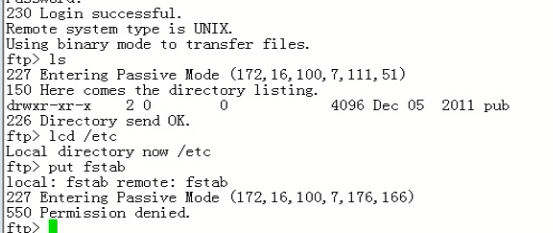
现在使用shell客户端直接连接上去

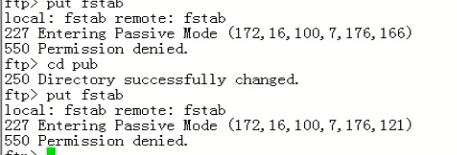


【lcd ---- 是ftp客户端所在的操作系统的 也有自己的目录结构 所以这个时候lcd是切换到本地文件目录的某一个位置 但是 cd是切换目标远程ftp服务器的某一个目录位置】

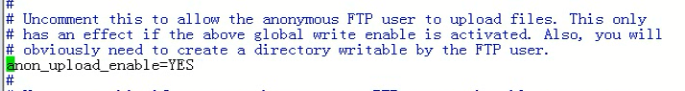
【上传的文件】

匿名用户也可以上传 --- 匿名用户映射到ftp用户



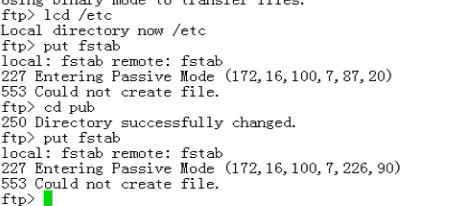


匿名用户对任何目录都没有上传文件 编辑配置文件



这个anon表示anonymous 是表示是否允许匿名用户上传文件

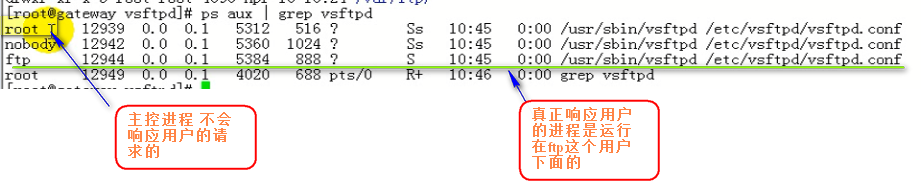
重启vsftpd服务器



还是不可以 但是Could not create file ---- 权限不是打开了么



但是ftp这个进程是谁运行？

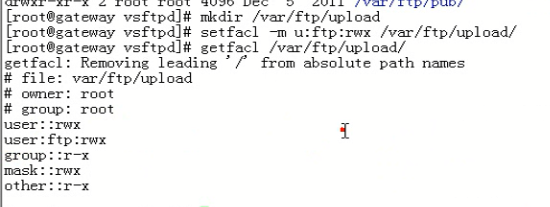


很显然 我们的进程对这个目录没有写权限 ----- 不能完成文件上传

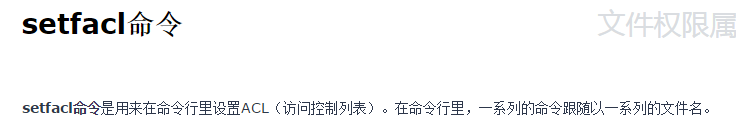


这个pub也是不让ftp用户又写权限 可以修改这个权限

还可以构建一个新的目录



【



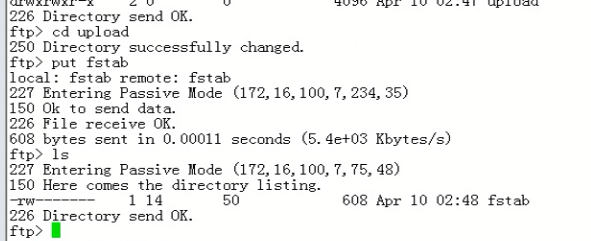
Set file acl



那么这里面 setfacl –m u:ftp:rwx /var/ftp/upload 就是对/var/ftp/upload对用户ftp开放rwx这三个权限

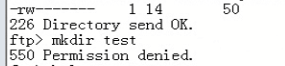
设置完之后 再看这个目录的acl权限】

此时 客户端切换到



共享权限取决于两个功能：

**文件系统权限(linux本身) 和 文件共享权限(vsftpd本身) 这两个必须同时具有某种权限**

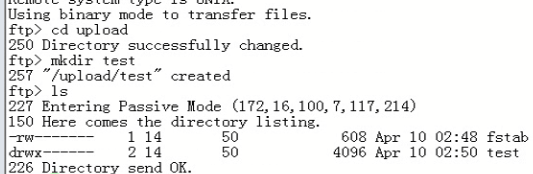
创建文件夹不可以

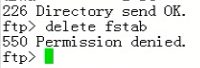
 删除文件也不行

因为这些属于不同的权限

 是否可以创建目录

重启服务

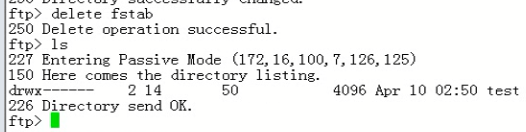




每个权限都是单独控制的

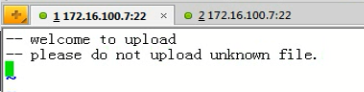


重新删除



匿名用户上传 下载 删除 都有 很危险 ---- 一般这些权限都是不开启的







这个是欢迎信息的配置 就是在对应的下载目录下面 创建一个.message的隐藏文件就可以了

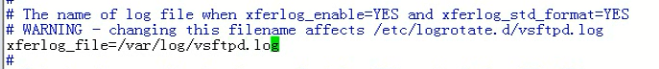
这样 切换到这个目录下

 直接给出状态码 并且展示刚才的欢迎信息

是否打开传输日志

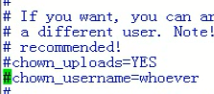
日志文件在哪？

因此还有一项 要启用传输日志的位置



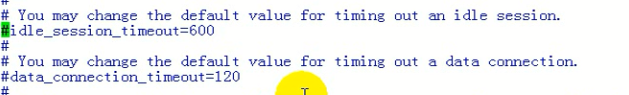
重启服务 访问试试

日志文件的格式

 就是用户上传文件之后 是否要修改一下这个文件的属主信息

这个就是是否修改 要是允许修改 这个修改的用户应该是谁

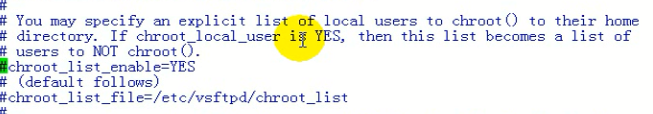
======= Ftp02 =====



定义绘话超时时间 ---- idle\_session\_timeout是控制连接相关的 data\_connection\_timeout是数据连接相关的

 启用文本模式进行上传下载 --- 有时候是二进制的 所以 一般不打开

对于系统用户账号 应该把用户锁定在家目录下面 不要让用户随便走



Chroot\_list\_enable --- 是否锁在家目录下

Chroot\_list\_file --- 哪些用户被禁锢

这个/etc/vsftpd/chroot\_list是实现不存在的



加入我们的用户

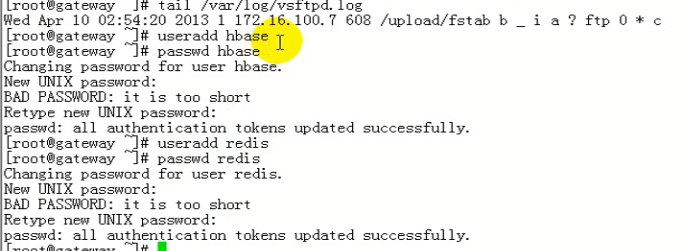
加入

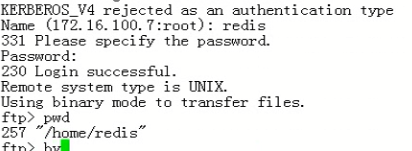
hadoop

---- 一行一个用户 然后保存退出

重启服务





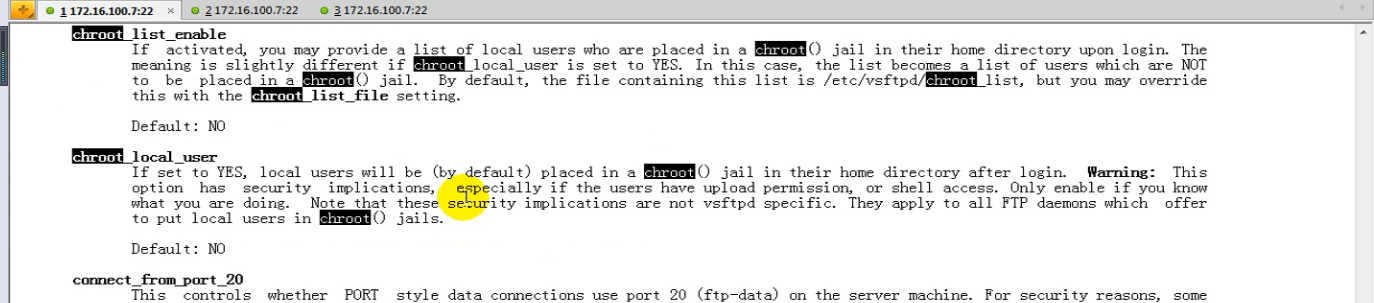


这表示只有写在这个列表中的用户才会被禁锢在家目录下面

下面的Hbase没有被禁锢在家目录

如何批量添加用户

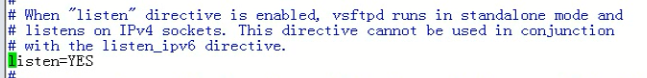
man vsftpd.conf



可以看到 这个选项叫chroot\_local\_user



这时候所有的linux系统用户都被禁锢家目录



表示vsftpd是否工作在一个独立的守护进程

守护进程有两类：独立守护进程 瞬时守护进程

瞬时守护进程都是xinetd代为管理 --- xinted本身就是一个独立守护进程

独立守护进程是用户用户访问量大 或者connection连接时间很长的情况 ---standalone

瞬时守护进程 就是用户访问量小 或者 connection的在线时间不长的情况 ----

**listen=YES表示把这个vsftpd进程设置为独立守护进程**

---- 否则配置为瞬时守护进程

此时 在/etc/xinetd.d下面配置一个脚本可以设置为瞬时守护进程 后面说



之前说过 vsftpd是接受pam管理的 那么pam下面这个vsftpd的配置文件叫什么名字是通过pam\_service\_name进行配置的



---------------------------



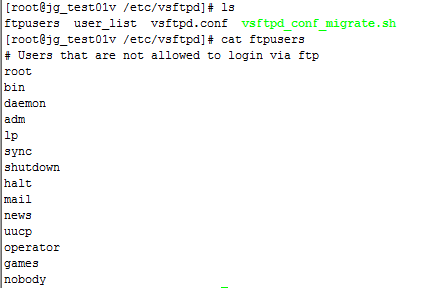
使用root登录ftp服务器 看下效果

 root登录直接就是fail掉了

直接告诉你Permission Denied

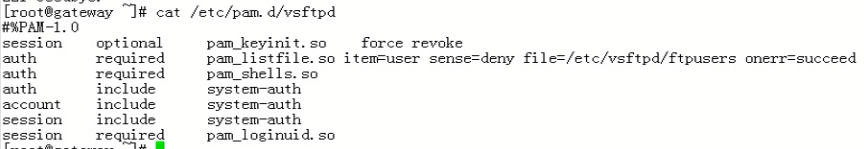
默认写在 /etc/vsftpd下面的ftpusers文件中的用户 都是禁止访问ftp服务器的



 这些用户不允许访问Ftp服务器

为什么这样呢？刚说过 默认vsftp认证依靠的是pam

看下这个认证文件的配置





这有一行写着 item=user sense=deny file=/etc/vsftpd/ftpusers onerr=succeed

这个写的是item=user表示 这个后面file中指出来的文件中写的都是用户名 sense=deny 表示敏感度是拒绝 --- 也就是列在后面file中的用户名都是被拒绝的

------ 所以 这里面就说明了file=/etc/vsftpd/ftpusers中的用户都是不可以登录ftp服务器的

Linux有些用户只是运行系统 管理员root的账号都是非常重要的 --- 这样别人抓包分析就知道你的用户名 和 密码 非常危险

除了ftpusers以外 还有一个user\_list

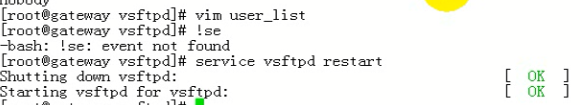




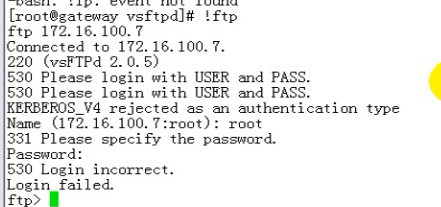
这些用户是否可以访问么？

先把这个文件清空

重启服务



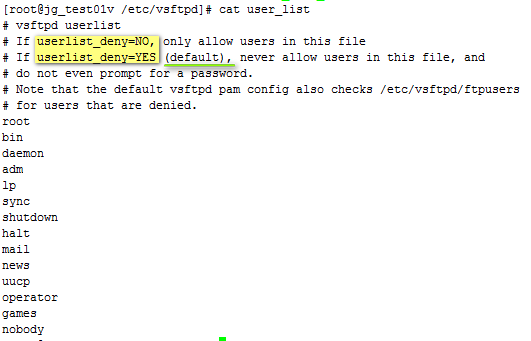
删除这个文件之后 可以输入密码 但是 还是不能登陆



输入密码也不行

**User\_list清空之后 才真的编程由vsftpusers这个文件控制的**

User\_list也控制了哪些登陆 哪些不能登陆 ----这个受userlist\_deny指令控制的

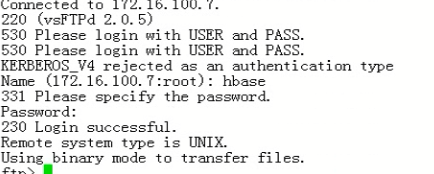


默认情况 userlist\_deny=YES 也就是userlist表示user\_list这个文件的deny开启 ---- 那么写在user\_list文件中的用户不能访问

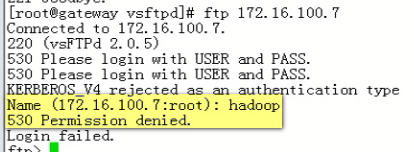
如果userlist\_deny=NO 表示user\_list文件的deny关闭 ---- 这个里面的用户就可以访问ftp服务器

 这个列表中的用户允许登录

现在 这个user\_list加入用户hbase



但是 此时 hadoop在user\_list中不存在 但是hadoop用户无法登录



这样 就说明 userlist\_deny=NO的时候 user\_list变成了可以登录ftp服务器中的白名单 --- 仅允许 ---- 设置为YES 这个user\_list就是黑名单

所以 userlist\_enable和userlist\_deny要配合使用

这就是ftp服务的基本配置

---------------

ftp配置可以控制用户如何使用资源 ---- 如果一次涌向ftp服务器的用户特别多 会导致这个ftp服务器无法正常对用户进行服务

所以 有下面的配置

 一共最多多少请求

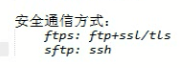
 一个IP最多多少个请求

这两个指令可以定义vsftpd的资源的访问

----------------------------- 传输保密性设置 -----------------------------

FTP本身是明文的 账号密码发送都是明文的 ----- 连用户名密码都是明文的 这个时候 为了安全 要加密

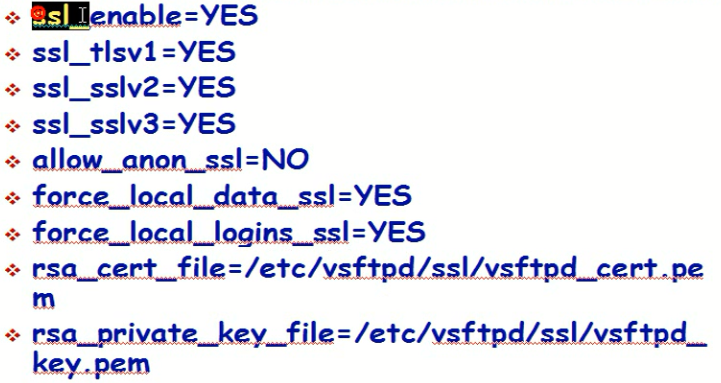
安全通信：ftps和sftp



Openssh提供了一个子系统 叫subsystem ---sftp ---- 就是基于ssh实现的安全传输功能

要配置vsftpd支持**ftps** --- 也就是基于ftp+ssl/tls 这种功能是非常简单的 ---- 首先创建一个证书 CA创建一个私有证书 发给vsftpd

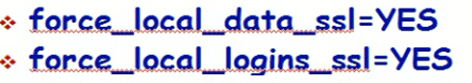
然后vsftpd中启用如下的指令



ssl\_enable --- 是否支持ssl协议

那么下面支持了ssl 那么 具体支持ssl哪几种协议？tlsv1 sslv2 sslv3 ---通常sslv2不建议启用

 匿名 文件就是给所有人的 所以 匿名不用ssl



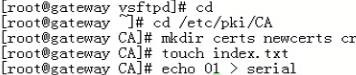
强制本地用户传输数据和登录必须使用ssl

那么 证书的位置 --- RSA的秘钥证书

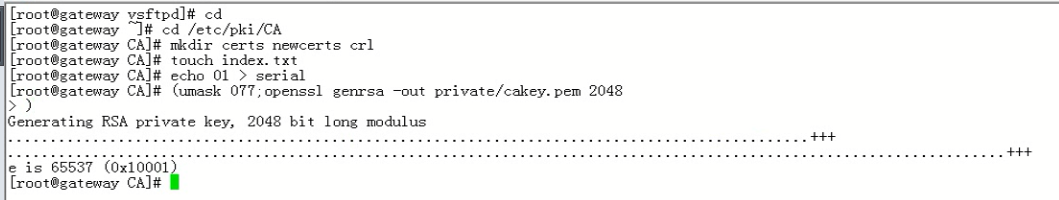
----- vsftpd明确要求使用rsa格式的秘钥

---- 创建CA非常恶心



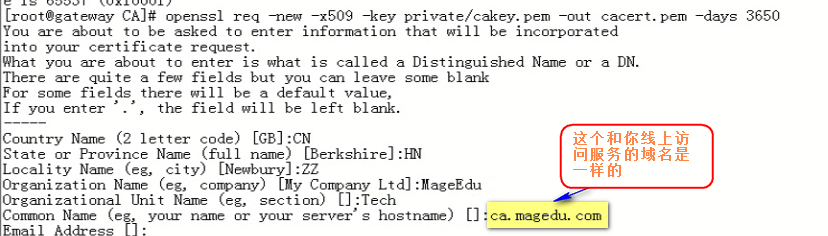


下面是创建私钥

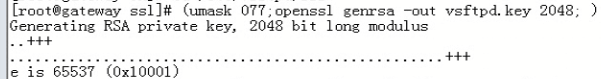


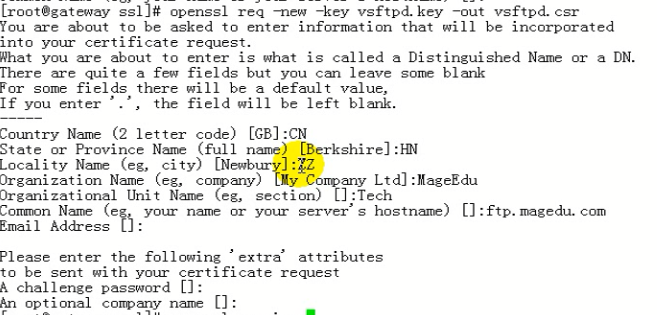
这个2048是指定的秘钥的长度

生成自签证书

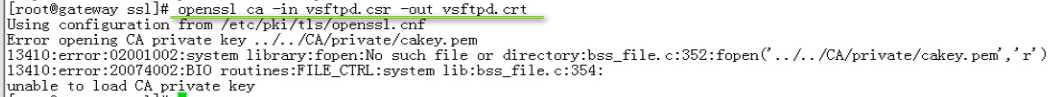


？？？哪台机器上？再生成一个私钥

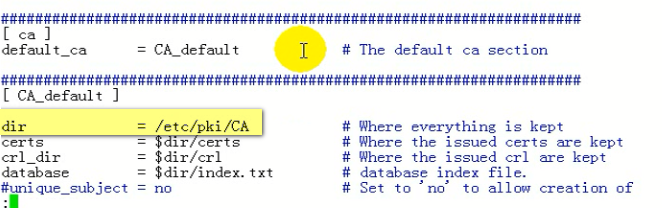


 ---- 这个是[ftp.magedu.com](ftp://ftp.magedu.com)

现在进行签署





 仅仅修改这个dir即可

再来一次 就签署完成

24:00

【补一下证书整块的知识】

17:28

HTTP是无状态的 每次