FTP

一 实验目的

通过远程查看 FTP 服务器的配置文件，了解 FTP 服务器的配置内容；然后通过抓包软件分析 FTP 传输交互的整个流程。

二 预备知识

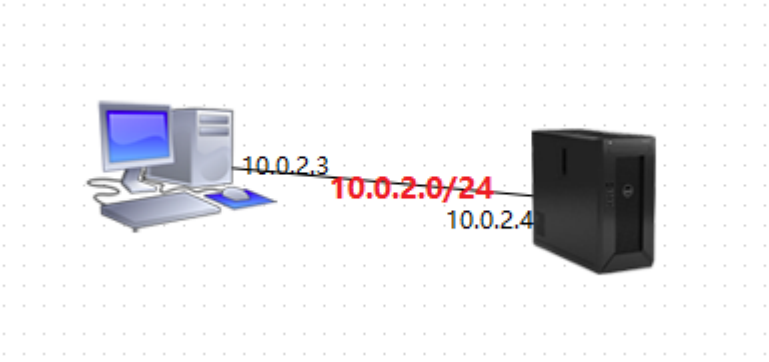
文件传输协议 FTP(File Transfer Protocol)是因特网中使用最广泛的文件传输协议。FTP 使用交互式的访问，允许客户指定文件的类型和格式(如指明是否使用 ASCII 码)，并允许文件具有存取权限(如访问文件的用户必须经过授权，并输入有效的口令)。

FTP 屏蔽了各计算机系统的细节，因而适合在异构网络中任意计算机之间传送文件。FTP 只提供文件传送的一些基本服务，它使用 TCP 可靠地运输服务，FTP 主要功能是减小或消除在不同系统下处理文件的不兼容性。

FTP 控制连接在整个会话期间都保持打开，只用来发送连接/传送请求。当客户进程向服务器发送连接请求时，寻找连接服务器进程的熟知端口 21，同时还要告诉服务器进程自己的另一个端口号码，用于建立数据传送连接。接着，服务器进程用自己传送数据的熟知端口 20 与客户进程所提供的端口号码建立数据传送连接，FTP 使用了 2 个不同的端口号，所以数据连接和控制连接不会混乱。

三 实验环境

在右上方的实验拓扑图中选择”**FTP**”，点击连线配置子网网段（10.0.2.0/24），实验拓扑如下图所示：

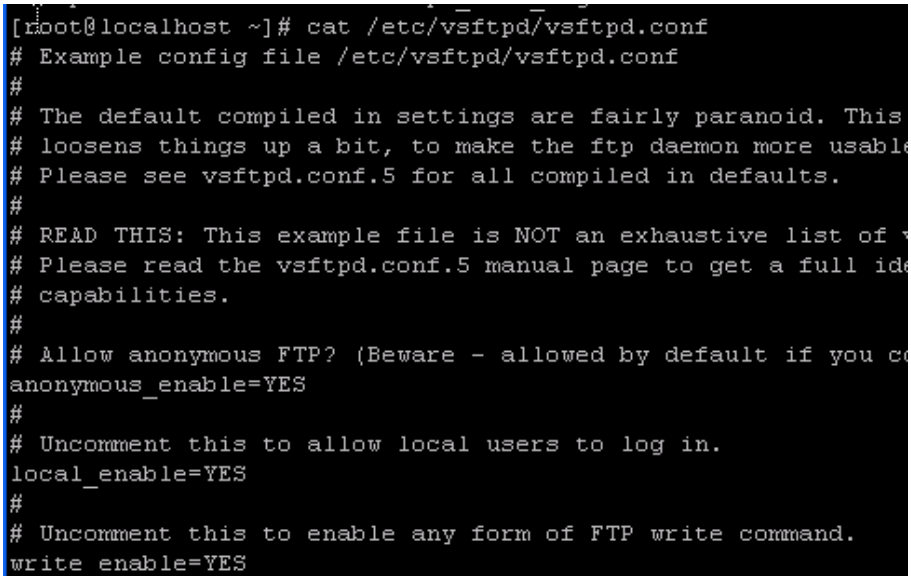


然后点击提交实验，等待资源分配成功后，点击图标按全屏访问即可进入设备。

四 实验步骤

1. 配置 FTP

我们首先进入 PC，打开桌面的 Putty，SSH 登录到 10.0.2.4（账号：centos 密码：centos）。输入命令 cat /etc/vsftpd/vsftpd.conf 来查看 ftp 配置文件的相关信息：



其中相关配置的意义为：

Allow anonymous FTP? (Beware - allowed by default if you comment this out).

anonymous_enable=YES 允许匿名用户登录

#

Uncomment this to allow local users to log in.

local_enable=YES 允许系统用户名登录

#

Uncomment this to enable any form of FTP write command.

write_enable=YES 允许使用任何可以修改文件系统的 FTP 的指令

#

Default umask for local users is 077. You may wish to change this to 022,

if your users expect that (022 is used by most other ftpd's)

local_umask=022 本地用户新增档案的权限

#

Uncomment this to allow the anonymous FTP user to upload files. This only

has an effect if the above global write enable is activated. Also, you will

obviously need to create a directory writable by the FTP user.

#anon_upload_enable=YES 允许匿名用户上传文件

Uncomment this if you want the anonymous FTP user to be able to create

new directories.

#anon_mkdir_write_enable=YES 允许匿名用户创建新目录

#

Activate directory messages - messages given to remote users when they

go into a certain directory.

dirmessage_enable=YES 允许为目录配置显示信息,显示每个目录下面的 message_file 文件的内容

#

Activate logging of uploads/downloads.

xferlog_enable=YES 开启日记功能

#

Make sure PORT transfer connections originate from port 20 (ftp-data).

connect_from_port_20=YES 使用标准的 20 端口来连接 ftp

#

If you want, you can arrange for uploaded anonymous files to be owned by

a different user. Note! Using "root" for uploaded files is not

recommended!

#chown_uploads=YES 所有匿名上传的文件的所属用户将会被更改成 chown_username

#chown_username=whoever 匿名上传文件所属用户名

#

You may override where the log file goes if you like. The default is shown

below.

#xferlog_file=/var/log/vsftpd.log 日志文件位置

#

If you want, you can have your log file in standard ftpd xferlog format

xferlog_std_format=YES 使用标准格式

#

You may change the default value for timing out an idle session.

#idle_session_timeout=600 空闲连接超时

#

You may change the default value for timing out a data connection.

#data_connection_timeout=120 数据传输超时

#

It is recommended that you define on your system a unique user which the

ftp server can use as a totally isolated and unprivileged user.

#nopriv_user=ftpsecure 当服务器运行于最底层时使用的用户名

#

Enable this and the server will recognise asynchronous ABOR requests. Not

recommended for security (the code is non-trivial). Not enabling it,

however, may confuse older FTP clients.

#async_abor_enable=YES 允许使用\"async ABOR\"命令,一般不用,容易出问题

```
#
# By default the server will pretend to allow ASCII mode but in fact ignore
# the request. Turn on the below options to have the server actually do ASCII
# mangling on files when in ASCII mode.
# Beware that on some FTP servers, ASCII support allows a denial of service
# attack (DoS) via the command "SIZE /big/file" in ASCII mode. vsftpd
# predicted this attack and has always been safe, reporting the size of the
# raw file.
# ASCII mangling is a horrible feature of the protocol.
#ascii_upload_enable=YES  管控是否可用 ASCII 模式上传。默认值为 NO
#ascii_download_enable=YES 管控是否可用 ASCII 模式下载。默认值为 NO
# You may fully customise the login banner string:
#ftpd_banner=Welcome to blah FTP service.  login 时显示欢迎信息.如果设置了 banner_file 则此设置无效
#
# You may specify a file of disallowed anonymous e-mail addresses. Apparently
# useful for combatting certain DoS attacks.
#deny_email_enable=YES  如果匿名用户需要密码,那么使用 banned_email_file 里面的电子邮件地址的用户不能登录
# (default follows)
#banned_email_file=/etc/vsftpd/banned_emails  禁止使用匿名用户登陆时作为密码的电子邮件地址
#
# You may specify an explicit list of local users to chroot() to their home
# directory. If chroot_local_user is YES, then this list becomes a list of
# users to NOT chroot().
#chroot_list_enable=YES 如果启动这项功能，则所有列在 chroot_list_file 中的使用者不能更改根目录
# (default follows)
#chroot_list_file=/etc/vsftpd/chroot_list  定义不能更改用户主目录的文件
#
# You may activate the "-R" option to the builtin ls. This is disabled by
# default to avoid remote users being able to cause excessive I/O on large
# sites. However, some broken FTP clients such as "ncftp" and "mirror" assume
# the presence of the "-R" option, so there is a strong case for enabling it.
#ls_recurse_enable=YES  是否能使用 ls -R 命令以防止浪费大量的服务器资源
```

```
# When "listen" directive is enabled, vsftpd runs in standalone mode and
# listens on IPv4 sockets. This directive cannot be used in conjunction
# with the listen_ipv6 directive.listen=YES 绑定到 listen_port 指定的端口,既然都绑定了也就是每时都开着的,就是那个什么 standalone 模式
# This directive enables listening on IPv6 sockets. To listen on IPv4 and IPv6
# sockets, you must run two copies of vsftpd whith two configuration files.
# Make sure, that one of the listen options is commented !!
#listen_ipv6=YES
pam_service_name=vsftpd 定义 PAM 所使用的名称，预设为 vsftpd
userlist_enable=YES 若启用此选项,userlist_deny 选项才被启动
tcp_wrappers=YES 开启 tcp_wrappers 支持
pasv_enable =YES 支持被动模式
port_enable = YES 支持主动模式
```

了解完服务器端的配置后，下面我们开始分析下载文件的数据流量；首先我们打开 PC 的 cmd，和抓包工具 Wireshark，在抓包->选项中选择本地连接接口，然后点击开始；

接下来我们在 cmd 中输入 ftp 10.0.1.1（账号密码都是 anonymous, anonymous 是匿名用户的意思）：

然后输入 literal 表示需要向远程 FTP 服务器发送协商参数，在 command line to send 后输入 pasv 后 windows 进入被动模式：

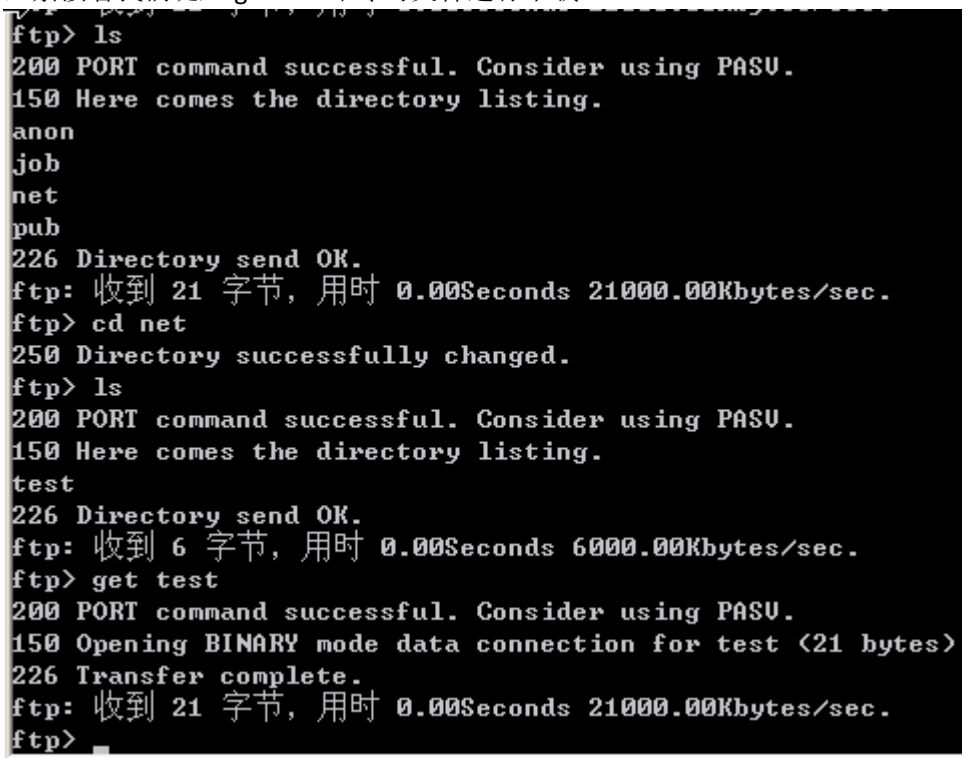
2. FTP 相关命令

首先我们来了解一下 ftp 的一些基本命令：

1. open {[ftp_server_Name_or_IP]} [port]--建立会话，连接到 ftp 服务器，提示输入用户名和密码 登录
2. user {[username]} {[password]}--以 username 登录，系统会提示输入密 7801
3. type {ascii|binary}--更改 ftp 传输模式

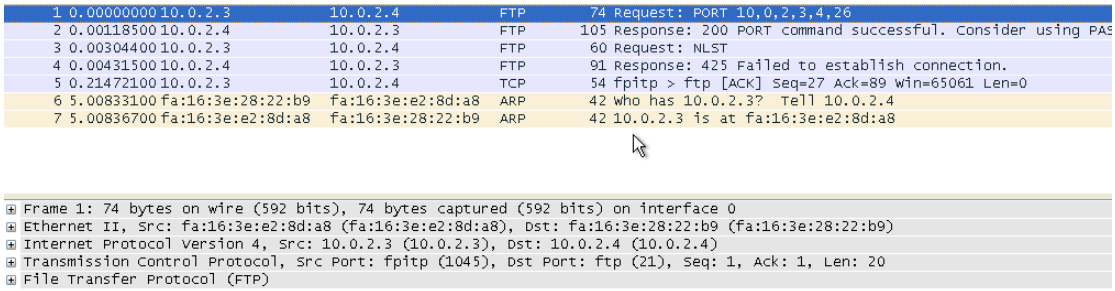
```
4. verbose、bell、hash、debug、trace、glob、staus--设置 ftp 的使用习惯、信息输出等级等
5. lcd {localpath|空}--设置本地工作路径，或恢复默认的工作路径
6. pwd--远程 ftp 服务器的当前目录
7. cd {fulldirname}--进入 ftp 的其他目录
8. dir {dirpath} [locallistfile]、ls {dirpath} [locallistfile] --列出远程目录的信息
9. get {fulpathfile} {[localfile]}、recv {file} {[localfile]}--下载文件
10. put {locafile} {[fullpahtfile]}、send {locafile} {[fullpahtfile]}--上传文件
11. delete {fullpahfile}--删除 ftp 文件
12. mkdir {fulldirname}--创建 ftp 目录
13. rmdir {fulldirname}--删除 ftp 目录
14. close、disconnect--结束回话
15. bype、quit—结束会话、退出 ftp 模式，回到系统调用
16. bype、quit—结束会话、退出 ftp 模式，回到系统调用
```

回到我们刚才的实验，进入 ftp 后我们首先键入 ls 查看目录下有哪些文件夹，然后我们进入 net 文件夹（命令：cd net），然后继续 ls 查看文件夹下有哪些文件，发现下面有个 test 文件，紧接着我们键入 get test 命令对文件进行下载：



3. 思考

最后我们需要对捕获的流量包进行分析，并思考以下问题：



1	0.00000000	10.0.2.3	10.0.2.4	FTP	74 Request: PORT 10,0,2,3,4,26
2	0.00118500	10.0.2.4	10.0.2.3	FTP	105 Response: 200 PORT command successful. Consider using PAS
3	0.00304400	10.0.2.3	10.0.2.4	FTP	60 Request: NLST
4	0.00431500	10.0.2.4	10.0.2.3	FTP	91 Response: 425 Failed to establish connection.
5	0.21472100	10.0.2.3	10.0.2.4	TCP	54 fpitp > ftp [ACK] Seq=27 Ack=89 Win=65061 Len=0
6	5.00833100	fa:16:3e:28:22:b9	fa:16:3e:e2:8d:a8	ARP	42 Who has 10.0.2.3? Tell 10.0.2.4
7	5.00836700	fa:16:3e:e2:8d:a8	fa:16:3e:28:22:b9	ARP	42 10.0.2.3 is at fa:16:3e:e2:8d:a8


```
⊞ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
⊞ Ethernet II, Src: fa:16:3e:e2:8d:a8 (fa:16:3e:e2:8d:a8), Dst: fa:16:3e:28:22:b9 (fa:16:3e:28:22:b9)
⊞ Internet Protocol Version 4, Src: 10.0.2.3 (10.0.2.3), Dst: 10.0.2.4 (10.0.2.4)
⊞ Transmission Control Protocol, Src Port: fpitp (1045), Dst Port: ftp (21), Seq: 1, Ack: 1, Len: 20
⊞ File Transfer Protocol (FTP)
```

1.FTP 应用是基于 TCP 还是 UDP 的？

2.你的主机发起到目的主机的 FTP 连接的端口号和目的主机的端口号分别是多少？

3.观测用户和 FTP 服务器连接时都交互了哪些信息？

4.在传送数据前是否新开了一个 TCP 连接？

5.传送数据时的源端口号和目的端口号分别是多少？还是开始建立连接时的端口号吗？由此你可以得出什么结论？