NAME

ftpd, in.ftpd, setup.anonftp − DARPA Internet File Transfer Protocol server

SYNOPSIS

ftp stream tcp nowait root /usr/sbin/in.ftpd in.ftpd
tcpd ftp /usr/sbin/in.ftpd

DESCRIPTION

Ftpd is the DARPA Internet File Transfer Protototocol server process. The server uses the TCP protocol and listens at the port specified in the ‘‘ftp’’ service specification; see **services**(5).

The ftp server currently supports the following ftp requests; case is not distinguished.

Request	**Description**
ABOR	abort previous command
ACCT	specify account (ignored)
ALLO	allocate storage (vacuously)
APPE	append to a file
CDUP	change to parent of current working directory
CWD	change working directory
DELE	delete a file
HELP	give help information
LIST	give list files in a directory (‘‘ls -lA’’)
MKD	make a directory
MODE	specify data transfer *mode*
NLST	give name list of files in directory (‘‘ls’’)
NOOP	do nothing
PASS	specify password
PASV	prepare for server-to-server transfer
PORT	specify data connection port
PWD	print the current working directory
QUIT	terminate session
RETR	retrieve a file
RMD	remove a directory
RNFR	specify rename-from file name
RNTO	specify rename-to file name
STOR	store a file
STOU	store a file with a unique name
STRU	specify data transfer *structure*
TYPE	specify data transfer *type*
USER	specify user name
XCUP	change to parent of current working directory
XCWD	change working directory
XMKD	make a directory
XPWD	print the current working directory
XRMD	remove a directory

The remaining ftp requests specified in Internet RFC 959 are recognized, but not implemented.

The ftp server will abort an active file transfer only when the ABOR command is preceded by a Telnet "Interrupt Process" (IP) signal and a Telnet "Synch" signal in the command Telnet stream, as described in Internet RFC 959.

Ftpd interprets file names according to the ‘‘globbing’’ conventions used by **csh**(1). This allows users to utilize the metacharacters ‘‘*?[]{}~’’.

Ftpd authenticates users according to three rules.

1) The user name must be in the password data base, **/etc/passwd**, and not have a null password. In this case a password must be provided by the client before any file operations may be performed.

2) The user name must not appear in the file **/etc/ftpusers**.

3) If the user name is ‘‘anonymous’’ or ‘‘ftp’’, an anonymous ftp account must be present in the password file (user ‘‘ftp’’). In this case the user is allowed to log in by specifying any password (by convention this is given as the client host’s name).

In the last case, **ftpd** takes special measures to restrict the client’s access privileges. The server performs a **chroot**(2) command to the home directory of the ‘‘ftp’’ user. In order that system security is not breached, it is recommended that the ‘‘ftp’’ subtree be constructed with care; the following rules are recommended.

~ftp) Make the home directory owned by ‘‘ftp’’ and unwritable by anyone.

~ftp/bin)
Make this directory owned by the super-user and unwritable by anyone. The program **ls**(1) must be present to support the list commands. This program should have mode 111.

~ftp/etc) Make this directory owned by the super-user and unwritable by anyone. The files **passwd**(5) and **group**(5) must be present for the **ls** command to work properly. These files should be mode 444.

~ftp/pub)
Make this directory mode 755 and owned by the super-user. Create directories in it owned by users if those users want to manage an anonymous ftp directory.

~ftp/pub/incoming)
Optionally create this directory for anonymous uploads. Make it mode 777. The FTP daemon will create files with mode 266, so remote users can write a file, but only local users can do something with it.

The script **setup.anonftp** can be used to create or check an anonymous FTP tree.

SEE ALSO

ftp(1).

BUGS

The anonymous account is inherently dangerous and should avoided when possible.