NAME

in.ftpd, ftpd – file transfer protocol server

SYNOPSIS

in.ftpd [**–dl**] [**–t***timeout*]

AVAILABILITY

SUNWcsu

DESCRIPTION

in.ftpd is the Internet File Transfer Protocol (FTP) server process. The server is invoked by the Internet daemon **inetd**(1M) each time a connection to the FTP service (see **services**(4)) is made.

OPTIONS

–d	Debugging information is logged to the system log daemon **syslogd**(1M).
–l	Each FTP session is logged to the system log daemon **syslogd**(1M).
–t*timeout*	Set the inactivity timeout period to *timeout* seconds. The FTP server will timeout an inactive session after 15 minutes.

Requests

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

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 –lg**)
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 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 RFC 959.

in.ftpd interprets file names according to the "globbing" conventions used by **sh**(1). This allows users to utilize the metacharacters: ∗ **? [ ] { } ˜**

in.ftpd authenticates users according to four rules.

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

2) If the user name appears in the file **/etc/ftpusers**, **ftp** access is denied.

3) **ftp** access is denied if the user's shell (from **/etc/passwd**) is not listed in the file **/etc/shells.** If the file **/etc/shells** does not exist, then the user's shell must be one of the following:

/usr/bin/sh	**/usr/bin/csh**	**/usr/bin/ksh**
/usr/bin/jsh	**/bin/sh**	**/bin/csh**
/bin/ksh	**/bin/jsh**	**/sbin/sh**
/sbin/jsh		

4) If the user name is "anonymous" or "ftp", an entry for the user name *ftp* must be present in the password and shadow files. The user is then allowed to log in by specifying any password — by convention this is given as the user's e-mail address (such as **user@host.Sun.COM**). Do not specify a valid shell in the password entry of the *ftp* user, and do not give it a valid password (use NP in the encrypted password field of the shadow file).

For anonymous ftp users, **in.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 suggested.

˜ftp Make the home directory owned by **root** and unwritable by anyone. This directory should not be on a file system mounted with the **nosuid** option.

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

˜ftp/usr/lib Make this directory owned by the super-user and unwritable by anyone. Copy the following shared libraries from **/usr/lib** into this directory.:

ld.so∗
libc.so∗
libdl.so∗
libintl.so∗
libw.so∗
libnsl.so∗
libsocket.so∗
nss_nis.so∗
nss_nisplus.so∗
nss_dns.so∗
nss_files.so∗
straddr.so∗

˜ftp/etc Make this directory owned by the super-user and unwritable by anyone. Copies of the files **passwd**(4), **group**(4), and **netconfig**(4) must be present for the **ls**(1) command to work properly. These files should be mode 444.

˜ftp/pub Make this directory mode 777 and owned by **ftp**. Users should then place files which are to be accessible via the anonymous account in this directory.

˜ftp/dev Make this directory owned by the super-user and unwritable by anyone. First perform **ls** –**lL** on the device files listed below to determine their major and minor numbers, then use **mknod** to create them in this directory.

/dev/zero
/dev/tcp
/dev/udp
/dev/ticotsord

˜ftp/usr/share/lib/zoneinfo
Make this directory mode 555 and owned by the super-user. Copy its contents from **/usr/share/lib/zoneinfo**. This enables **ls** –**l** to display time and date stamps correctly.

EXAMPLES

To set up anonymous ftp, add the following entry to the **/etc/passwd** file. In this case, **/export/ftp** was chosen to be the anonymous ftp area, and the shell is the non-existent file **/nosuchshell**. This prevents users from logging in as the ftp user.

ftp:x:30000:30000:Anonymous FTP:/export/ftp:/nosuchshell

Add the following entry to /etc/shadow:

ftp:NP:6445::::::

The following is a shell script that will set up the anonymous ftp area. It presumes that names are resolved using NIS.

```
#!/bin/sh
# script to setup anonymous ftp area
#
# handle the optional command line argument
case $# in

  # the default location for the anon ftp comes from the passwd file
  0) ftphome="`grep '^ftp:' /etc/passwd | cut -d: -f6`"
    ;;

  1) if [ "$1" = "start" ]; then
      ftphome="`grep '^ftp:' /etc/passwd | cut -d: -f6`"
    else
      ftphome=$1
    fi
    ;;

  *) echo "Usage: $0 [anon-ftp-root]"
    exit 1
    ;;
esac

if [ -z "${ftphome}" ]; then
  echo "$0: ftphome must be non-null"
  exit 2
fi

# This script assumes that ftphome is neither / nor /usr so ...
if [ "${ftphome}" = "/" -o "${ftphome}" = "/usr" ]; then
  echo "$0: ftphome must not be / or /usr"
  exit 2
fi

# If ftphome does not exist but parent does, create ftphome
if [ ! -d ${ftphome} ]; then
   # lack of -p below is intentional
```

```
   mkdir ${ftphome}


fi
echo Setting up anonymous ftp area ${ftphome}

# Ensure that the /usr/bin directory exists
if [ ! -d ${ftphome}/usr/bin ]; then
   mkdir -p ${ftphome}/usr/bin
fi

cp /usr/bin/ls ${ftphome}/usr/bin
chmod 111 ${ftphome}/usr/bin/ls

# Now set the ownership and modes to match the man page
chown root ${ftphome}/usr/bin
chmod 555 ${ftphome}/usr/bin

# this may not be the right thing to do
# but we need the bin -> usr/bin link
if [ -r ${ftphome}/bin ]; then
   mv -f ${ftphome}/bin ${ftphome}/Obin
fi
ln -s usr/bin ${ftphome}

# Ensure that the /usr/lib and /etc directories exist
if [ ! -d ${ftphome}/usr/lib ]; then
   mkdir -p ${ftphome}/usr/lib
fi
if [ ! -d ${ftphome}/etc ]; then
   mkdir -p ${ftphome}/etc
fi

#Most of the following are needed for basic operation, except
#for libnsl.so, nss_nis.so, libsocket.so, and straddr.so which are
#needed to resolve NIS names.

cp /usr/lib/ld.so /usr/lib/ld.so.1 ${ftphome}/usr/lib

for lib in libc libdl libintl libw libnsl libsocket \
  nss_nis nss_nisplus nss_dns nss_files
do
  cp /usr/lib/${lib}.so.1 ${ftphome}/usr/lib
  rm -f ${ftphome}/usr/lib/${lib}.so
  ln -s ./${lib}.so.1 ${ftphome}/usr/lib/${lib}.so
```

```
done

cp /usr/lib/straddr.so.2 ${ftphome}/usr/lib
rm -f ${ftphome}/usr/lib/straddr.so
ln -s ./straddr.so.2 ${ftphome}/usr/lib/straddr.so

cp /etc/passwd /etc/group /etc/netconfig ${ftphome}/etc

# Copy timezone database
mkdir -p ${ftphome}/usr/share/lib/zoneinfo
(cd ${ftphome}/usr/share/lib/zoneinfo
  (cd /usr/share/lib/zoneinfo; find . -print | cpio -o) | cpio -imdu
  find . -print | xargs chmod 555
  find . -print | xargs chown root
)

chmod 555 ${ftphome}/usr/lib/*
chmod 444 ${ftphome}/etc/*

# Now set the ownership and modes
chown root ${ftphome}/usr/lib ${ftphome}/etc
chmod 555 ${ftphome}/usr/lib ${ftphome}/etc

# Ensure that the /dev directory exists
if [ ! -d ${ftphome}/dev ]; then
    mkdir -p ${ftphome}/dev
fi

# make device nodes. ticotsord and udp are necessary for
# 'ls' to resolve NIS names.

for device in zero tcp udp ticotsord
do
   line=`ls -lL /dev/${device} | sed -e 's/,//'`
   major=`echo $line | awk '{print $5}'`
   minor=`echo $line | awk '{print $6}'`
   rm -f ${ftphome}/dev/${device}
   mknod ${ftphome}/dev/${device} c ${major} ${minor}
done

chmod 666 ${ftphome}/dev/*

## Now set the ownership and modes
chown root ${ftphome}/dev
chmod 555 ${ftphome}/dev
```

```
if [ ! -d ${ftphome}/pub ]; then
  mkdir -p ${ftphome}/pub
fi
chown ftp ${ftphome}/pub
chmod 777 ${ftphome}/pub
```

SEE ALSO

ftp(1), **ls**(1), **aset**(1M), **inetd**(1M), **mknod**(1M), **syslogd**(1M), **chroot**(2), **getsockopt**(3N), **group**(4), **inetd.conf**(4), **netconfig**(4), **netrc**(4), **passwd**(4), **services**(4)

Postel, Jon, and Joyce Reynolds, *File Transfer Protocol (* FTP), RFC 959, Network Information Center, SRI International, Menlo Park, Calif., October 1985.

DIAGNOSTICS

in.ftpd logs various errors to **syslogd**, with a facility code of **daemon**.

Info Severity

These messages are logged only if the –**l** flag is specified.

FTPD: connection from *host* **at** *time*
: A connection was made to **ftpd** from the host *host* at the date and time *time*.

FTPD: User *user* **timed out after** *timeout* **seconds at** *time*
: The user *user* was logged out because they had not entered any commands after *timeout* seconds; the logout occurred at the date and time *time*.

Debug Severity

These messages are logged only if the –**d** flag is specified.

FTPD: command: *command*
: A command line containing *command* was read from the FTP client.

lost connection
: The FTP client dropped the connection.

<—— *replycode*
<—— *replycode*–
: A reply was sent to the FTP client with the reply code *replycode*. The next message logged will include the message associated with the reply. If a – follows the reply code, the reply is continued on later lines.

NOTES

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

The server must run as the super-user to create sockets with privileged port numbers. It maintains an effective user id of the logged in user, reverting to the super-user only when binding addresses to sockets. The possible security holes have been extensively scrutinized, but are possibly incomplete.

/etc/ftpusers contains a list of users who cannot access the system; the format of the file is one user name per line.