ProFTP Server

Scenario: combine several webcams at different locations into one screen, have the cameras upload images automatically to the webserver.

Installation

```
apt-get install proftpd
```

Configuration: proftpd.conf

```
UseIPv6                         off
ServerName                      "<machine name>"
DefaultRoot                     ~/path/to/images <user>
Port                            21
MasqueradeAddress               yourdomain.com
DynMasqRefresh                  28800
Umask                           027 027
PassivePorts                    65000 65534
SetEnv TZ :/etc/localtime
```

To restrict ProFTP to serve only one directory, and limit access to write only commands, set the DefaultRoot and configure the directory. Or, configure the server to jail all users within their home directory (1st entry) or a sub directory in home (2nd entry):

```
DefaultRoot ~
DefaultRoot /home/%u/ftp
```

Bug fix

There is a bug in the current init script to start / restart the server, fix it as follows:

```
# vim /etc/init.d/proftpd
@@ -105,3 +105,3 @@
if [ -f "$PIDFILE" ]; then
- start-stop-daemon --stop --signal $SIGNAL --quiet --pidfile "$PIDFILE"
+ start-stop-daemon --stop --signal $SIGNAL --retry 1 --quiet --pidfile
"$PIDFILE"
if [ $? = 0 ]; then
```

Configuration: conf.d/myconfig.conf

Restrict access to write only commands and login from specific domains or ip address ranges, and jail loged in user to specific directory path. In Debian, the TCP wrapper module mod_wrap2 is loaded by default.

```
# use TCP wrapper....
WrapEngine on
WrapTables file:/etc/hosts.allow file:/etc/hosts.deny
# ....or Limit LOGIN
<Limit LOGIN>
      Order allow, deny
      Allow from .domain.tld
      Allow from 999.999.999.0/24
      DenyAll
</Limit>
<Directory /path/to/directory>
      <Limit ALL>
      DenyAll
      </Limit>

      <Limit CDUP CWD PWD XCWD XCUP>
      AllowAll
      </Limit>

      <Limit MKD STOR STOU>
      AllowAll
      </Limit>
</Directory>
```

Remark: MKD is necessary, if the client needs to create sub-directories when storing files.

If you use the TCP wrapper as explained above, then you need to add the following lines to /etc/hosts.deny:

```
# hosts.deny
ALL: ALL
```

and /etc/hosts.allow:

```
# hosts.allow
.domain.tld
ddns.domain.tld
```

Refer to the ProFTPD user documentation for details:

- ProFTPD module mod_wrap2
- ProFTPD module mod_wrap2_file

Configuration: client

Enable passive mode and set root directory for the upload to /. For some clients, this means leaving the root directory entry empty. Some cameras need a crontab entry to rename and / or move the uploaded image to the camera root. Remove the crontab entry if you take the camera offline. The command “rename” is explained here.

Panasonic BL-C30

For some clients, like my Panasonic BL-C30 camera, we need to deal with the uploaded file server side. The BL-C30 uploads the current camera image only if option “Save as New File with Time Stamp” is selected in the FTP Timer Settings. For example, the camera would upload the file “image20140618120030500.jpg”, which would indicate the time as 18 June 2014, 12:00, 30 seconds and 500 milli-seconds. To convert this file to image.jpg, add the following command to your crontab file, so the command gets executed automatically every minute:

```
$ crontab -e
* * * * * rename -f s/[0-9]//g image*
```

iPux ICS-2330

FTP Settings under Event Server Settings:

```
Host Address: yourdomain.com
Port Number: 22
User Name: user
Password: ***
Directory Path: /cam-name/
Passive Mode: yes
FTP Upload With: One snapshot
```

```
$ crontab -e
* * * * * ~/batch/ipux cam-name 2
```

Prepare a batch file like so:

```
#!/bin/sh
# camera name for subdirectory
if [ "$1" = "" ]; then
      echo "$1: camera name, $2: time offset (default 0)"
      exit 1
else
      CAM=$1
fi
# hour offset camera - server
if [ "$2" = "" ]; then
      RELTZ=0
```

```
else
      RELTZ=$2
fi

# settings
DEBUG=0
HOME='/path/to/camera/files'
SNAPSHOT='snapshot' # do not change
TODAY=$(date -d "$RELTZ hours ago" +"%Y%m%d")
YESTERDAY=$(date -d "$((24+$RELTZ)) hours ago" +"%Y%m%d")
NOW=$(date -d "$RELTZ hours ago" +"%H")
PASTHOUR=$(date -d "$((1+$RELTZ)) hours ago" +"%H")

cd $HOME
if [ "$DEBUG" = "1" ]; then
      echo Today ... = $TODAY
      echo Yesterday = $YESTERDAY
      echo Now ..... = $NOW
      echo Past hour = $PASTHOUR
fi
#
# rename uploaded file and move it to camera root
rename -f 's/(\d{6}\.\d{1})/snapshot/' $CAM/$TODAY/$NOW/*
if [ -f "$CAM/$TODAY/$NOW/$SNAPSHOT.jpg" ]; then
      mv $CAM/$TODAY/$NOW/$SNAPSHOT.jpg ./$CAM.jpg
fi
#
# remove yesterdays entries
rm -rf $CAM/$YESTERDAY/*
if [ -d "$CAM/$YESTERDAY" ]; then
      rmdir $CAM/$YESTERDAY
fi
#
# remove past hour entries
rm -rf $CAM/$TODAY/$PASTHOUR/*
if [ -d "$CAM/$TODAY/$PASTHOUR" ]; then
      rmdir $CAM/$TODAY/$PASTHOUR
fi
#
# write error log entry and remove invalid time stamp entries, e.g. after
power failure
if [ $ISVALID -eq 1 ]; then
      DIRLS=$(ls -ld $CAM/$INVALID)
      if [ $? -eq 0 ]; then
              DIRDATE=${DIRLS:26:12}
              echo $(date +"%Y")", $DIRDATE - camera '$CAM' reset" >> camlog
              rm -R $CAM/$INVALID
      fi
fi

exit 0
```

Internal non-camera client

I need a file from another linux client on the LAN to upload regularly a file. To handle lan external and internal transfers, you need to have separate <VirtualHost> sections for the daemon's internal and external IP addresses, and to have the correct MasqueradeAddress in the respective vhost configurations.

Delete! Create the ~/.netrc file in the home dir of the user that will run the (upload) ftp command on the linux client, give it appropriate perms (chmod 0600 ~/.netrc), and add the following:

```
# ~/.netrc
machine <machine> # ftp server ip
login <user> # user name to login to server
password <secret> # pwd to login to server
```

Delete! end

The upload is initiated with the following command, login credentials are taken from ~/.netrc

```
$ echo put my-local-file.txt /remote-path/my-remote-file.txt | ftp <machine>
<port> # server in active mode
$ echo put my-local-file.txt /remote-path/my-remote-file.txt | pftp
<machine> <port> # server in passive mode
```

To autmatically create a file every Monday at 07.00 and move it to the cloud, do the following:

1. on the client, run a crontab -e as user with the following entry:

   ```
   0 7 * * 1 echo put <my-file> | pftp <machine> <port>
   ```

2. on the server, run a crontab -e as root with the following entry:

   ```
   5 7 * * 1 mv <source-dir/*> <dest-dir>
   ```

Alternatively, create the following file and invoke ftp like so from within the directory the file to upload resides in:

```
ftp -n < ~/.ftp/upload
```

Content of upload:

```
open sub.domain.tld port
user <user> <pwd>
put file /remote/path/file
bye
```

Trouble shooting

Check whether ProFTP is running:

```
ps -ef | grep proftpd
```

Restart the server:

```
/etc/init.d/proftp start
```

Make sure ProFTP is automatically run after boot:

```
# cd /etc/rc5.d
# ln -s ../init.d/proftpd S25proftpd
```

FTP server connectivity test

Todo Links

- Firewalls, Routers, NAT
- ProFTPD mini-HOWTO - ServerType
- nmap - How to find live hosts on my network
- Server does not support non-ASCII characters
- ProFTPD: Configuring <Limits>