

```
#####
# BigBrother CCTV Recording & Live Viewing (mirroring) software #
# Copyright 2016-2019 Andrew Wood #
# #
# www.bigbrothercctv.org #
# #
# Licensed under the GNU Public License v 3 #
# The full license can be read at www.gnu.org/licenses/gpl-3.0.txt #
# and is included in the License.txt file included with this #
# software. #
# #
# BigBrother is free open source software but if you find it #
# useful please consider making a donation to the Communications #
# Museum Trust at www.comms.org.uk/donate #
#####
```

BigBrother is a fairly simple wrapper around the open source ffmpeg program to harness it for CCTV recording and live viewing (referred to as mirroring) without having to fiddle around with complicated settings for ffmpeg or writing your own scripts to handle segmenting the recording files.

It offers the following features:

- Recording of a live stream from an IP camera in any format supported by ffmpeg

- Recording files are automatically segmented into one file per hour with the start time of each file automatically aligned to zero minutes past.

- Ability to organise files by day (with files kept until the same day next week) or by groups of cameras in their own folder each of which could if needed be mounted on a separate disk.

- Access to recorded files can be provided using a third party file server (Samba is recommended) and instructions are provided on how to do this. This will allow the recorded files to be viewed using the standard file browser on Windows, Mac OS X or most Linux desktop GUIs such as Gnome.

- If a camera stream becomes unavaialbe due to a network issue or the camera being rebooted, BigBrother will keep trying it until it comes back up.

- Live viewing of cameras using streaming over HTTP and a web browser interface
- The live viewing is referred to as mirroring because it can mirror a camera stream from one network to another if desired, for example from a private LAN to a WAN for remote viewing.

- HTTP Live Streaming (HLS) format output which is currently compatible with the following clients:

- Microsoft Edge (Windows)

- Apple Safari (Mac & iOS)

- Google Chrome (Android)

- VLC Media Player (Windows, Mac, Linux) can view a specific camera stream

- Requires third party web server software with PHP support (such as Apache HTTPD, Nginx or Lighttpd) to be running on the same host

- Support for the number of cameras, simultaneous recordings and mirroring is limited only by hardware and network capacity.

- Configured via two simple text config files, which which defines program settings and the other which defines the camera parameters (name, url etc) and the actions you want to perform on it (recording mode, mirroring mode)

- Runs on Unix-like operating systems with a Bourne Shell, Python run time and (optionally) a webserver supporting PHP if you want to do mirroring. Also requires an intallation of the ffmpeg command line program.

- A rcNG init script is provided for FreeBSD

```
=====
INSTALLATION & CONFIGURATION
=====
```

## Installation

A FreeBSD pkgng and Debian deb package is provided for installation which sets up the necessary users and associated ownership & permissions. However you are responsible for creating the directory where you want the recorded files to be placed. This has to contain certain subdirectories and have appropriate permissions set so this will be described below.

This directory can be placed anywhere on the filesystem and is specified on a per camera basis which means you can share one directory amongst all or several cameras or have different directories for some. By combining this with different mount points you can spread recordings over different disks.

In this example we will use a directory called /cctvrecordings  
This MUST contain the following subdirectories:

```

/cctvrecordings
|
|-byday
|   |
|   |--Monday
|   |--Tuesday
|   |--Wednesday
|   |--Thursday
|   |--Friday
|   |--Saturday
|   |--Sunday
|
|-bycamera
|   |
|   |--GroupName

```

where GroupName is a directory for EACH group you define in the camera config file  
Likewise you can specify different mount points for the subdirectories to split file amongst disks.  
See the documentation for your operating system for details of how to specify a mount point.

The username that BigBrother runs as is defined in the config file.

A default installation using the package will create a user called bigbrother and two groups, cctvviewers and cctvwriters. The ownership of all the recording directory tree shown above should be bigbrother:cctvviewers with mode 750 (rwxr-x--).

You should also set the set group id (chmod g+s) on all of these directories

The software itself is installed under /usr/local/bigbrother  
You should not rename, or remove anything under this main directory.

The permissions on it should be as owner bigbrother:cctvwriters mode 755  
its contents should have the same ownership mode 740 WITH THE EXCEPTION  
OF THE mirrorwebroot subdirectory. The permissions on this should be  
owner bigbrother:cctvviewers mode 750

The permissions on all files under this should be owner bigbrother:cctvviewers  
mode 640

The main directory also contains two sample config files which you can use as the basis of your own. Note the permissions set on these sample files and ensure that you match it on any custom files you create.

For FreeBSD there is an init script also provided in this directory which is symlinked to from the main OS init directory (/usr/local/etc/rc.d)

This can be used to start, stop or restart BigBrother.

To set it to start automatically at boot time you need to edit /etc/rc.conf and insert the following two lines:

```
bigbrotherd_enable="YES"
bigbrotherd_conf="/usr/local/bigbrother/bigbrother.conf"
```

Obviously you can change the path to a custom config file if you wish. The file referred to here is the 'global config file' which contains server wide settings. The other config file is the camera config file, the path to which you specify in the global conf file. The syntax of both files will be discussed later in this document.

The init script accepts the standard start|stop|restart arguments

For Debian a SystemD unit file is provided and installed to /etc/systemd/system. The software can then be controlled using systemctl command bigbrotherd where command is start|stop|restart|status|enable|disable

## Configuration

BigBrother has two plain text config files. The global config file contains server wide settings. It's format is one entry per line in the format key value

Blank lines are permitted and comments can be entered by starting the line with a #

The following keys are mandatory and are shown with sample values:

```
cameraconf /usr/local/etc/bigbrother_camera.conf
logfile /var/log/bigbrother.log
ffmpegcommand /usr/local/bin/ffmpeg
user bigbrother
```

Most of them should be self explanatory, user is the username you want BigBrother to run as, ffmpeg command is the full path to the copy of ffmpeg you want it to use. This must be the full path as init will start the software without the context of a PATH environment variable.

Ensure that the user BigBrother is running as, has write permission to the specified log file, and of course execute permission for ffmpeg.

The camera configuration file is where you define each camera you want BigBrother to work with and the actions you want it to take. Again this is a plain text file with one entry (camera) per line, blank lines are permitted and comments can be entered by starting the line with a #

The format of each camera definition is with values in columns separated by tabs. The column order is fixed. If a particular column doesn't apply you insert a \*

The column order is as follows:

```
CameraName  URL  GroupName  RecordMode  MirrorMode  Folder  Container
```

CameraName is a string containing only letters or numbers which must be unique for every camera. You can use this to describe the camera location e.g FrontEntrance

URL is the URL of the camera live stream. You need to check the documentation of your camera to see what the direct stream URL is.

e.g rtsp://cameraip:554

e.g rtsp://cameraip/StreamID

GroupName is an optional string containing only letters or numbers. It is mandatory if using the RecordMode C (see below). It allows you to group recording files together in a folder and must therefore exist as a directory under your main recording directory's/bycamera subfolder (see above)

If you dont want to specify a GroupName use \*

GroupNames can also be used to group related cameras together for display (see the URL Format section below)

e.g Group1

e.g CarPark

e.g \*

RecordMode is an optional string defining if you want BigBrother to record this camera (if not use \*) and if so, the type of recording. Valid values are C or D which indicated 'bycamera' and 'byday' mode and will result in the recording files being put in the corresponding directory (see above).

MirrorMode is an optional string defining if you want BigBrother to do live viewing mirroring for this camera. This allows you to view the live output from the camera in a webpage and can be routed from one network to another if you want to monitor cameras from a remote location or if you have a separate VLAN for cameras and want to monitor them from a different VLAN. If you don't wish to do this use \* otherwise this sets the stream format that BigBrother/ffmpeg will produce. Currently the only valid value is HLS which provides HTTP Live Streaming.

e.g HLS

e.g \*

Folder is the main parent directory you want the recordings to be placed in. This directory must contain mandatory sub directories as described above.

ContainerType is the type of codec container delivered by the camera. Currently the only valid value is MP4 for an MPEG4 container.

Here are some full examples:

```
#Camera1 is recorded ('byday') and is mirrored
```

```
Camera1 rtsp://192.168.111.1:554 Group1 D HLS /cctvrecordings MP4
```

```
#Camera2 is not recorded only mirrored
```

```
Camera2 rtsp://192.168.111.2:554 * * HLS * *
```

```
#Outside cameras are recorded ('byday') and mirrored. A group is used
```

```
#for mirroring output grouping only
```

```
FrontEntrance rtsp://192.168.111.10:554 Outside D HLS /cctvrecordings MP4
```

```
CarParkCam1 rtsp://192.168.111.11:554 Outside D HLS /cctvrecordings MP4
```

```
CarParkCam2 rtsp://192.168.111.12:554 Outside D HLS /cctvrecordings MP4
```

```
BuildingSideEast rtsp://192.168.111.13:554 Outside D HLS /cctvrecordings MP4
```

```
RearYard rtsp://192.168.111.14:554 Outside D HLS /cctvrecordings MP4
```

```
#Camera covering cash office is recorded 'bycamera' so it will be kept indefinitely
```

```
#it is also mirrored
```

```
CashOffice rtsp://192.168.111.20:554 HighImportance C HLS /cctvrecordings MP4
```

Remember the difference between 'byday' and 'bycamera' is that 'bycamera' will put the files under mainfolder/bycamera/GroupName and they will be kept forever. If you want to auto delete them you will need to set a cron job to do this at whatever interval you desire. On the other hand 'byday' will put the files under mainfolder/byday/DayOfWeek and they will be kept only until the same day next week when they will be automatically deleted.

The config files should have ownership bigbrother:cctvviewers and be readable by the group.

```
=====
MIRRORING
=====
```

To provide mirroring BigBrother writes the stream files to its mirrorwebroot subdirectory (you should not rename, move or delete anything in this directory).

The contents of this directory can then be served up by any HTTP server such as Apache, Nginx or Lighttpd. This README does not cover configuring this third party software as it is well documented elsewhere, but you need to be aware of some points before the two will work together.

Firstly the HTTP server you setup must have PHP support and be configured to pass files with names ending .php to the PHP interpreter.

Secondly the username the webserver is running as must have permission to read BigBrothers mirrorwebroot directory and be configured to serve it.

You should add the webserver username to the cctvviewers group to allow this.

You will also need to add the webserver username to the cctvwriters group to allow it to read the global and camera config files.

If there is already an HTTP server on the machine you can configure a virtual site with its own hostname e.g bigbrother.mydomain.local

Thirdly BigBrother does not provide any authentication / access control to the contents of mirrorwebroot coming via HTTP. You will need to configure the HTTP server for HTTP Basic or Digest Authentication if you want to restrict access.

The url will be `http://server/index.php`

#### URL Format

-----

The URL used by the mirroring system accepts the following optional parameters:

`perRow=integer`

e.g `perRow=4`

Specifies the number of cameras to show per row on the page

`groupName[]=string`

e.g `groupName[]=Group1`

Displays only the cameras in that group. Multiple groups can be specified

e.g `groupName[]=Group1&groupName[]=Group2`

Note that the use of square brackets [] is required if you specify more than one value, otherwise only the last one will be acted on.

`cameraName[]=string`

e.g `cameraName[]=Camera1`

Displays only the specified camera. Multiple cameras can be specified

e.g `cameraName[]=Camera1&cameraName[]=Camera2`

Note that the use of square brackets [] is required if you specify more than one value, otherwise only the last one will be acted on.

If you specify both a groupName and a cameraName and the camera is in that group, it will only be shown once (as part of its groups display).

An example of the URL in full:

`http://server/index.php?`

`groupName[]=Group1&groupName[]=Group2&cameraName[]=CameraX&cameraName[]=CameraY&perRow=5`

If no parameters are specified all cameras setup in the config file for mirroring will be shown

An optional parameter `allownewfilefromweb` can be specified in the global config file:

`allownewfilefromweb=True`

If it is set to true then a button will be displayed on the mirroring webpage which will cause all recording files

to be closed early and new ones started immediately, rather than having to wait until the next hour.

This can be useful if you have spotted something suspicious on the live mirroring display and want to review the

recording straight away.

By default `allownewfilefromweb` will be assumed to be false if not present.

In order for this feature to work, the webserver process must have permission to send a hangup signal (SIGHUP)

to `bigbrotherd`. Unix security requires that a signal can only be sent from one process to another if they are

both running under the same username/id, you therefore need to take an additional step to make it work:

1. Copy the systems kill command from its location (run 'which kill' to locate it) to a file called `sendSIG` in the bigbrother directory:

```
cp /bin/kill /usr/local/bigbrother/sendSIG
```

2. Change the ownership and permissions on it as follows:

```
chown bigbrother:cctvwriters sendSIG
```

```
chmod 750 sendSIG
```

```
chmod u+s sendSIG
```

### 3. Put the webservers username into the cctvwriters group

```
=====
APPENDIX A - Remote access to recordings using Samba
=====
```

It is quite likely that BigBrother will be running on a headless server or a server with commandline only access locked away in a secure location. To allow easy viewing of the recorded files you can setup a fileserver on the same machine to provide access from a different machine.

Samba is a free open source SMB file and print server which is secure, reliable and widely used. SMB is the protocol used by Microsoft Windows for accessing file and print servers so by setting up Samba you can access recorded files using the Windows Explorer file browser on a Windows PC on your desk or in the security office.

Mac OS X also has built in support for SMB (Apple uses Samba in OS X to provide this) and it can be accessed by a Linux PC too.

If you are interested in all the things Samba can do and its full configuration I recommend you read

'Using Samba' published by O'Reilly. At the present time the latest edition of this book covers Samba 3

but not Samba 4. Samba 4 was a major upgrade (released in 2012) so you might want to email [bookquestions@oreilly.com](mailto:bookquestions@oreilly.com) and suggest they produce an updated edition!

This appendix will just give a simple standalone file server configuration to allow people not familiar with Samba to get started quickly.

Samba is configured using a text config file. For Samba 4 this is usually `smb4.conf`. On FreeBSD the Samba 4 package is called `samba4x` where `x` is the minor version, currently `samba43`.

Below is a sample `smb4.conf` file suitable for a standalone file server for accessing BigBrother recordings. Comments are lines starting with a `#`

```
#global section defines server wide settings
```

```
[global]
```

```
    passdb backend = tdbsam
    bind interfaces only = yes
    interfaces = 192.168.253.202 127.0.0.1
    server string = Samba %v on %h
    workgroup = CCTV
    netbios name = bigbrother
    #use this servers local passwd db for user auth
    security = user
    #use smbpasswd file for user DB not /etc/passwd (needed for WinNT and securing passwords
over network )
    encrypt passwords = true
    #this machine is the WINS server
    wins support = yes
    #participate in subnet lmb elections
    local master =yes
    #force an lmb election on startup
    preferred master = yes
    #and always win
    os level = 100
```

```
#this is a file share called cctvrecordings
#accessible by anyone in the cctvviewers group
#it is read only for security against CCTV deletion
```

```
[cctvrecordings]
```

```
    comment = CCTV recordings file share
    #the directory we want to serve up
    path = /cctvrecordings
    read only = yes
    guest ok = no
    browseable = yes
    writeable = no
```

```
create mode = 0660 #default for new files = rw-rw----
directory mode = 0770 #default for new sub dirs = rwxrwx---
validusers = @cctvviewers
adminusers = root
```

#End of smb4.conf

To use this config you will also need to add the usernames of people who can access the share to the cctvviewers group and then run the smbpasswd command. This is because Samba does not use the Unix password stored in the Unix passwd file but instead uses its own password database as the encryption used by SMB is different.

```
smbpasswd -a username
```

Remember that the Samba password and the Unix password are then separate.

Also note if your OS is using SELinux (for example RHEL or CentOS) then you may need to run the following command before any users can access the Samba share:

```
chcon -R -t samba_share_t /cctvrecordings/
```

TIP: You can scan through a recording on 'fast forward' using VLC. You can start VLC from the command line with the --rate option. Using --rate 20 allows you to view an hours footage in 3 minutes. To do this you either need a wired Ethernet connection to the SMB server or you need to copy the file locally first. Trying to do this over Wifi or remotely over the internet wont work well as it can't usually read the data fast enough consistently.

TIP: There is an extract.sh shell script provided which uses ffmpeg to extract a segment from a recording file into a new file:

```
extract.sh -in /path/to/inputfile -out /path/to/outputfile -start HH:MM:SS -duration HH:MM:SS"
```

For example to extract 30 seconds of video from the input file, starting 45 minutes into it:

```
extract.sh -in /path/to/inputfile -out /path/to/outputfile -start 00:45:00 -duration 00:00:30"
```

```
=====
APPENDIX B - Building from source
=====
```

A FreeBSD pkgng package can be build by cd'ing to the source directory and running

```
pkg create -M ./+MANIFEST -r .
```

A Debian .deb package can be built by cd'ing to the source directory and running  
make deb

To build a Debian package you will need checkinstall which can be installed using apt-get and GNU Make

To build an RPM cd to the source directory and run make rpm