

Rtorrent_rutorrent_lighttpd

What this article covers:

Installing a uTorrent like web interface on dd-wrt which is rutorrent running on lighttpd connecting to rtorrent. This article is geared towards advanced linux users but if you aren't, this is the time to learn!

Intro

Rutorrent's interface and functionality is closest to uTorrent compared to other bittorrent implementations that can run on dd-wrt. Transmission is very basic and does not offer too much control. rtorrent by itself is not much usable unless you are at ease with the shell. Rutorrent comes in as an interface for rtorrent.

Try rutorrent/rtorrent [here](#) before diving into this installation.

1. user:demo
2. pass:demo

click on Rutorrent. You can play with it :)

Contents

- [1 rtorrent](#)
 - ◆ [1.1](#)
[/opt/etc/rtorrent.conf](#)
- [2 lighttpd](#)
 - ◆ [2.1](#)
[/opt/etc/lighttpd/lighttpd.conf](#)
 - ◆ [2.2](#)
[Troubleshooting](#)
- [3 ruTorrent](#)
- [4 swap](#)
 - ◆ [4.1](#)
[/opt/etc/init.d/S10swap](#)
- [5 SSL & Password Protection](#)
 - ◆ [5.1 File Transfer from ubuntu](#)
 - ◆ [5.2 htpasswd](#)
 - ◆ [5.3 SSL certificate](#)

rtorrent

```
ipkg install rtorrent
ipkg-opt install screen dtach adduser
```

screen for daemon

This will install libtorrent, rtorrent and xmlrpc-c

Rtorrent_rutorrent_lighttpd

After all the installations are done,

```
mkdir /mnt/rtorrent
mkdir /mnt/rtorrent/work
mkdir /mnt/rtorrent/session
```

Edit **rtorrent.conf** as shown below

/opt/etc/rtorrent.conf

```
# This is an example resource file for rTorrent. Copy to
# ~/.rtorrent.rc and enable/modify the options as needed. Remember to
# uncomment the options you wish to enable.

# Minumum amount of peers to connect per torrent, if available.
min_peers = 20

# Minumum amount of peers to connect per torrent.
max_peers = 40

# Same as above but for seeding completed torrents (-1 = same as downloading)
#min_peers_seed = 10
#max_peers_seed = 20

# Maximum number of simultaneous uploads per torrent.
max_uploads = 5

# Global download rate in KiB. "0" for unlimited.
download_rate = 0

# Global upload rate in KiB. "0" for unlimited.
upload_rate = 22

# Default directory to save downloaded files. Note it doesn't support
# space yet.
directory = /mnt/rtorrent/work/

# Watch a directory for new torrents, and stop those that have been deleted.
#schedule = watch_directory,5,5,load_start=/opt/share/torrent/dl/*.torrent
schedule = watch_directory,5,5,load_start=/mnt/rtorrent/dl/*.torrent
schedule = untied_directory,5,5,stop_untied=

# Close torrents when disk space is low.
#schedule = low_diskspace,5,60,close_low_diskspace=100M

# Default session directory. Make sure you don't run multiple instance
# of rtorrent using the same session directory. Perhaps using a
# relative path?
#session = /opt/share/torrent/session
session = /mnt/rtorrent/session

# The ip address reported to the tracker.
#ip = rakshasa

# The ip address the listening socket and outgoing connections is
# bound to.
#bind = rakshasa

# Port range to use for listening.
```

Rtorrent_rutorrent_lighttpd

```
port_range = 51777-51780

# Start opening ports at a random position within the port range.
#port_random = no

# Check hash for finished torrents. Might be usefull until the bug is
# fixed that causes lack of diskpace not to be properly reported.
#check_hash = no

# Set whatever the client should try to connect to UDP trackers.
#use_udp_trackers = yes

# Alternative calls to bind and ip that should handle dynamic ip's.
#schedule = ip_tick,0,1800,ip=rakshasa
#schedule = bind_tick,0,1800,bind=rakshasa

#
# Do not modify the following parameters unless you know what you're doing.
#

# Hash read-ahead controls how many MB to request the kernel to read
# ahead. If the value is too low the disk may not be fully utilized,
# while if too high the kernel might not be able to keep the read
# pages in memory thus end up trashing.
#hash_read_ahead = 10

# Interval between attempts to check the hash, in milliseconds.
#hash_interval = 100

# Number of attempts to check the hash while using the mincore status,
# before forcing. Overworked systems might need lower values to get a
# decent hash checking rate.
#hash_max_tries = 10

# Max number of files to keep open simultaneously.
#max_open_files = 128

# Number of sockets to simultaneously keep open.
#max_open_sockets = <no default>

# Example of scheduling commands: Switch between two ip's every 5
# seconds.
#schedule = "ip_tick1,5,10,ip=torretta"
#schedule = "ip_tick2,10,10,ip=lampedusa"

# Remove a scheduled event.
#schedule_remove = "ip_tick1"

# Stop torrents when reaching upload ratio in percent,
# when also reaching total upload in bytes, or when
# reaching final upload ratio in percent.
# example: stop at ratio 2.0 with at least 200 MB uploaded, or else ratio 20.0
#schedule = ratio,60,60,"stop_on_ratio=200,200M,2000"
schedule = ratio,30,60,stop_on_ratio=200

# Encryption options, set to none (default) or any combination of the following:
# allow_incoming, try_outgoing, require, require_RC4, enable_retry, prefer_plaintext
#
# The example value allows incoming encrypted connections, starts unencrypted
# outgoing connections but retries with encryption if they fail, preferring
```

Rtorrent_rutorrent_lighttpd

```
# plaintext to RC4 encryption after the encrypted handshake
#
# encryption = allow_incoming,enable_retry,prefer_plaintext

# Enable DHT support for trackerless torrents or when all trackers are down.
# May be set to "disable" (completely disable DHT), "off" (do not start DHT),
# "auto" (start and stop DHT as needed), or "on" (start DHT immediately).
# The default is "off". For DHT to work, a session directory must be defined.
#
# dht = auto

# UDP port to use for DHT.
#
# dht_port = 6881

# Enable peer exchange (for torrents not marked private)
#
# peer_exchange = yes

#
# Do not modify the following parameters unless you know what you're doing.
#
scgi_port = 127.0.0.1:5000
```

lighttpd

```
ipkg install lighttpd
```

/opt/etc/lighttpd/lighttpd.conf

```
# lighttpd configuration file
#
# use it as a base for lighttpd 1.0.0 and above
#
# $Id: lighttpd.conf 8860 2008-08-05 10:56:16Z engy $

##### Options you really have to take care of #####

## modules to load
# at least mod_access and mod_accesslog should be loaded
# all other module should only be loaded if really necessary
# - saves some time
# - saves memory
server.modules = (
#         "mod_rewrite",
#         "mod_redirect",
#         "mod_alias",
#         "mod_access",
#         "mod_cml",
#         "mod_trigger_b4_dl",
#         "mod_auth",
#         "mod_status",
#         "mod_setenv",
#         "mod_fastcgi",
#         "mod_proxy",
#         "mod_simple_vhost",
#         "mod_evhost",
```

Rtorrent_rutorrent_lighttpd

```
#
#               "mod_userdir",
#               "mod_cgi",
#               "mod_compress",
#               "mod_ssi",
#               "mod_usertrack",
#               "mod_expire",
#               "mod_secdownload",
#               "mod_rrdtool",
#               "mod_accesslog" )

server.modules += ( "mod_scgi" )

## a static document-root, for virtual-hosting take look at the
## server.virtual-* options
server.document-root      = "/opt/share/www/lighttpd"

## where to send error-messages to
server.errorlog           = "/opt/var/log/lighttpd/error.log"

# files to check for if .../ is requested
index-file.names         = ( "index.php", "index.html",
                             "index.htm", "default.htm",
                             "lighttpd/index.html" )

## set the event-handler (read the performance section in the manual)
# server.event-handler = "frebsd-kqueue" # needed on OS X
server.event-handler = "poll" # needed on OS X

# mimetype mapping
mimetype.assign          = (
  ".pdf"                => "application/pdf",
  ".sig"                => "application/pgp-signature",
  ".spl"                => "application/futuresplash",
  ".class"              => "application/octet-stream",
  ".ps"                 => "application/postscript",
  ".torrent"            => "application/x-bittorrent",
  ".dvi"                => "application/x-dvi",
  ".gz"                 => "application/x-gzip",
  ".pac"                => "application/x-ns-proxy-autoconfig",
  ".swf"                => "application/x-shockwave-flash",
  ".tar.gz"             => "application/x-tgz",
  ".tgz"                => "application/x-tgz",
  ".tar"                => "application/x-tar",
  ".zip"                => "application/zip",
  ".mp3"                => "audio/mpeg",
  ".m3u"                => "audio/x-mpegurl",
  ".wma"                => "audio/x-ms-wma",
  ".wax"                => "audio/x-ms-wax",
  ".ogg"                => "application/ogg",
  ".wav"                => "audio/x-wav",
  ".gif"                => "image/gif",
  ".jpg"                => "image/jpeg",
  ".jpeg"               => "image/jpeg",
  ".png"                => "image/png",
  ".xbm"                => "image/x-xbitmap",
  ".xpm"                => "image/x-xpixmap",
  ".xwd"                => "image/x-xwd",
  ".css"                => "text/css",
  ".html"               => "text/html",
  ".htm"                => "text/html",
  ".js"                 => "text/javascript",
  ".asc"                => "text/plain",
```

Rtorrent_rutorrent_lighttpd

```
".c"          => "text/plain",
".cpp"       => "text/plain",
".log"       => "text/plain",
".conf"      => "text/plain",
".text"      => "text/plain",
".txt"       => "text/plain",
".dtd"       => "text/xml",
".xml"       => "text/xml",
".mpeg"     => "video/mpeg",
".mpg"       => "video/mpeg",
".mov"       => "video/quicktime",
".qt"        => "video/quicktime",
".avi"       => "video/x-msvideo",
".asf"       => "video/x-ms-asf",
".asx"       => "video/x-ms-asf",
".wmv"       => "video/x-ms-wmv",
".bz2"       => "application/x-bzip",
".tbz"       => "application/x-bzip-compressed-tar",
".tar.bz2"   => "application/x-bzip-compressed-tar"
)

# Use the "Content-Type" extended attribute to obtain mime type if possible
#mimetype.use-xattr      = "enable"

## send a different Server: header
## be nice and keep it at lighttpd
# server.tag             = "lighttpd"

#### accesslog module
accesslog.filename       = "/opt/var/log/lighttpd/access.log"

## deny access the file-extensions
#
# ~      is for backupfiles from vi, emacs, joe, ...
# .inc   is often used for code includes which should in general not be part
#       of the document-root
url.access-deny          = ( "~", ".inc" )

$HTTP["url"] =~ "\.pdf$" {
    server.range-requests = "disable"
}

##
# which extensions should not be handle via static-file transfer
#
# .php, .pl, .fcgi are most often handled by mod_fastcgi or mod_cgi
static-file.exclude-extensions = ( ".fcgi", ".php", ".pl", ".py", ".rb" )

##### Options that are good to be but not necessary to be changed #####

## bind to port (default: 80)
server.port              = 8010

## bind to localhost (default: all interfaces)
#server.bind             = "grisu.home.kneschke.de"

## error-handler for status 404
#server.error-handler-404 = "/error-handler.html"
#server.error-handler-404 = "/error-handler.php"

## to help the rc.scripts
```

Rtorrent_rutorrent_lighttpd

```
server.pid-file          = "/var/run/lighttpd.pid"

##### virtual hosts
##
## If you want name-based virtual hosting add the next three settings and load
## mod_simple_vhost
##
## document-root =
##   virtual-server-root + virtual-server-default-host + virtual-server-docroot
## or
##   virtual-server-root + http-host + virtual-server-docroot
##
#simple-vhost.server-root   = "/home/weigon/wwwroot/servers/"
#simple-vhost.default-host = "grisu.home.kneschke.de"
#simple-vhost.document-root = "/pages/"

##
## Format: <errorfile-prefix><status-code>.html
## -> ../status-404.html for 'File not found'
#server.errorfile-prefix  = "/home/weigon/projects/lighttpd/doc/status-"

## virtual directory listings
#dir-listing.activate     = "enable"

## enable debugging
#debug.log-request-header = "enable"
#debug.log-response-header = "enable"
#debug.log-request-handling = "enable"
#debug.log-file-not-found = "enable"

### only root can use these options
#
# chroot() to directory (default: no chroot() )
#server.chroot            = "/"

## change uid to <uid> (default: don't care)
#server.username         = "nobody"

## change gid to <gid> (default: don't care)
#server.groupname       = "nobody"

##### compress module
#compress.cache-dir     = "/tmp/lighttpd/cache/compress/"
#compress.filetype      = ("text/plain", "text/html")

##### proxy module
## read proxy.txt for more info
#proxy.server           = ( ".php" =>
#                          ( "localhost" =>
#                            (
#                              "host" => "192.168.0.101",
#                              "port" => 80
#                            )
#                          )
#                        )
#

scgi.server = (
    "/RPC2" =>
        ( "127.0.0.1" =>
            (
```

Rtorrent_rutorrent_lighttpd

```
"host" => "127.0.0.1",
"port" => 5000,
"check-local" => "disable"
)
)

#### fastcgi module
## read fastcgi.txt for more info
#fastcgi.server = ( ".php" =>
#                 ( "localhost" =>
#                 (
#                   "socket" => "/tmp/php-fcgi.sock",
#                   "bin-path" => "/opt/bin/php-fcgi",
#                   "bin-environment" => (
#                     "PHP_FCGI_MAX_REQUESTS" => "4000"
#                   )
#                 )
#                 )
#                 )
#                 )

#### CGI module
#cgi.assign = ( ".pl" => "/opt/bin/perl",
#              ".cgi" => "/opt/bin/perl" )

#### SSL engine
#ssl.engine = "enable"
#ssl.pemfile = "server.pem"
#ssl.ca-file = "ca.crt"
#ssl.verifyclient.activate = "enable"
#ssl.verifyclient.enforce = "enable"
#ssl.verifyclient.depth = 1

#### status module
#status.status-url = "/server-status"
#status.config-url = "/server-config"

#### auth module
## read authentication.txt for more info
#auth.backend = "plain"
#auth.backend.plain.userfile = "lighttpd.user"
#auth.backend.plain.groupfile = "lighttpd.group"

#auth.backend.ldap.hostname = "localhost"
#auth.backend.ldap.base-dn = "dc=my-domain,dc=com"
#auth.backend.ldap.filter = "(uid=$)"

#auth.require = ( "/server-status" =>
#                 (
#                   "method" => "digest",
#                   "realm" => "download archiv",
#                   "require" => "user=jan"
#                 ),
#                 "/server-config" =>
#                 (
#                   "method" => "digest",
#                   "realm" => "download archiv",
#                   "require" => "valid-user"
#                 )
#                 )
```


Rtorrent_rutorrent_lighttpd

```
##### url handling modules (rewrite, redirect, access)
#url.rewrite          = ( "^/$"          => "/server-status" )
#url.redirect        = ( "^/wishlist/(.+)" => "http://www.123.org/$1" )
##### both rewrite/redirect support back reference to regex conditional using %n
#$HTTP["host"] =~ "^www\.(.*)" {
# url.redirect        = ( "^/(.*)" => "http://%1/$1" )
#}

#
# define a pattern for the host url finding
# %% => % sign
# %0 => domain name + tld
# %1 => tld
# %2 => domain name without tld
# %3 => subdomain 1 name
# %4 => subdomain 2 name
#
#evhost.path-pattern = "/home/storage/dev/www/%3/htdocs/"

##### expire module
#expire.url          = ( "/buggy/" => "access 2 hours", "/asdhas/" => "access plus 1 second" )

##### ssi
#ssi.extension      = ( ".shtml" )

##### rrdtool
#rrdtool.binary     = "/opt/bin/rrdtool"
#rrdtool.db-name    = "/var/www/lighttpd.rrd"

##### setenv
#setenv.add-request-header = ( "TRAV_ENV" => "mysql://user@host/db" )
#setenv.add-response-header = ( "X-Secret-Message" => "42" )

## for mod_trigger_b4_dl
# trigger-before-download.gdbm-filename = "/home/weigon/testbase/trigger.db"
# trigger-before-download.memcache-hosts = ( "127.0.0.1:11211" )
# trigger-before-download.trigger-url = "^/trigger/"
# trigger-before-download.download-url = "^/download/"
# trigger-before-download.deny-url = "http://127.0.0.1/index.html"
# trigger-before-download.trigger-timeout = 10

## for mod_cml
## don't forget to add index.cml to server.indexfiles
# cml.extension      = ".cml"
# cml.memcache-hosts = ( "127.0.0.1:11211" )

##### variable usage:
## variable name without "." is auto prefixed by "var." and becomes "var.bar"
#bar = 1
#var.mystring = "foo"

## integer add
#bar += 1
## string concat, with integer cast as string, result: "www.fool.com"
#server.name = "www." + mystring + var.bar + ".com"
## array merge
#index-file.names = (foo + ".php") + index-file.names
#index-file.names += (foo + ".php")

##### include
#include /etc/lighttpd/lighttpd-inc.conf
## same as above if you run: "lighttpd -f /etc/lighttpd/lighttpd.conf"
```

Rtorrent_rutorrent_lighttpd

```
#include "lighttpd-inc.conf"

#### include_shell
#include_shell "echo var.a=1"
## the above is same as:
#var.a=1

# Configuration from other optware packages
include_shell "cat /opt/etc/lighttpd/conf.d/*.conf"
```

After completing the configurations, restart lighttpd and rtorrent

```
service lighttpd restart
service rtorrent start
```

Check if lighttpd has successfully started by browsing to the router's ip address and port of the configured lighttpd.conf file which is 8010 here. <http://192.168.1.2:8010>

The browser should display **lighttpd server is running**.

htop command should show rTorrent running like this:

```
screen -dm -S rtorrent
rtorrent -n -o import=/opt/etc/rtorrent.conf
```

Troubleshooting

If [rtorrent -n -o import=/opt/etc/rtorrent.conf] cannot be found in the htop list, then rtorrent has not started correctly. Navigate to /mnt/rtorrent/session and remove rtorrent.lock file. Issue an rtorrent restart

```
service rtorrent restart
```

Reference:

1. <http://code.google.com/p/rutorrent>http://www.dd-wrt.com/wiki/WebSERVER#Lighttpd_Specific_SCGI_Conf
2. http://www.mattgibson.ca/2009/11/05/fix-dd-wrt-lighttpd-error-server-c-1105-fdevent_init-failed-on-asus-wl-5

ruTorrent

1. Goto <http://code.google.com/p/rutorrent/downloads/list>
2. Click on ruTorrent core files. Right click on the download link and copy link address
3. In the router console,

```
cd /opt/share/www/lighttpd/
```

Type wget and press shift key + insert key so that something like this appears

```
wget http://rutorrent.googlecode.com/files/rutorrent-3.2.tar.gz
```

Now the rutorrent archive has been downloaded to the /opt/share/www/lighttpd directory

```
tar -xvf rutorrent-3.2.tar.gz
```

Rtorrent_rutorrent_lighttpd

or

```
tar -xvf rutorrent*.gz
```

if you have download another version
You should now see the rutorrent folder

In web browser, ip address of router:port of lighttpd/rutorrent

<http://192.168.1.2:8010/rutorrent/>

You should now see the web interface of rutorrent loading.

As usual, setup port forwarding on ports 51777-51780 to your dd-wrt router

Congratulations! You have successfully installed and configured rtorrent + rutorrent on a dd-wrt device.

Enjoy seeding. :) --[Leonidy](#) 18:55, 11 May 2011 (CEST)

swap

Using rtorrent requires a lot of ram. Even on an Asus RT-N16, free memory drops to nearly 2MB. So, why this swap section here? Normally, a typical optware installation should auto mount a swap partition, but on my setup with an RT-N16, it does not seem to find the swap partition. Investigation leads to dd-wrt not able to label a swap partition as "swap". The S10swap script found in /opt/etc/init.d/ tries to look for the swap partition, but it fails due to the partition not being labeled. (If I'm wrong, please correct me). If you are in the same situation, then modify the swap script accordingly.

Before modifying the file, make a backup

Note: The script assumes that your swap partition is /dev/sda2

Change as required.

```
cp /opt/etc/init.d/S10swap /opt/etc/init.d/S10swap.bak
```

/opt/etc/init.d/S10swap

```
#!/bin/sh
export PATH=/opt/bin:/opt/sbin:/opt/usr/sbin:$PATH
#scriptname=${0##*/}
#scriptname="swap"

DESC="swap"
NAME=swap
DAEMON=$NAME
SCRIPTNAME=/etc/init.d/$NAME

optlog "$scriptname" "Managing swap partitions"

if [ -z "$1" ] ; then
    case `echo "$0" | sed 's:^.*\/(.*)$:1:g'` in
        S??*) rc="start" ;;
        K??*) rc="stop" ;;
        *) rc="usage" ;;
    esac
else
    rc="$1"
fi
```

Rtorrent_rutorrent_lighttpd

```
fi

case "$rc" in
start)
size=""
#/opt/sbin/fdisk -l | grep "^/dev/" | grep " swap" | awk '{print $1}' >/tmp/swaps
/dev/sda2 >/tmp/swaps
#while read swapdev ; do
swapdev="/dev/sda2"
/opt/sbin/swapon ${swapdev} 2>/dev/null
if [ ${?} -eq 0 ] ; then
size=`/opt/sbin/swapon -s | grep ${swapdev} | awk '{print $3}'`
if [ -z "${size}" ] ; then
optlog "$scriptname" "Failed to mount swap ${swapdev}"
else
optlog "$scriptname" "${size} kB allocated on ${swapdev}"
break
fi
else
size=`/opt/sbin/swapon -s | grep ${swapdev} | awk '{print $3}'`
if [ -z "${size}" ] ; then
optlog "$scriptname" "Failed to mount swap ${swapdev}"
else
optlog "$scriptname" "swap device ${swapdev} already allocated (${size} kB)"
break
fi
fi
fi
;;
*)
#/opt/sbin/swapon /dev/sda2 2>/dev/null

#done</tmp/swaps

if [ -z "${size}" ] ; then
if [ -f /opt/swapfile ] ; then
/opt/sbin/swapon /opt/swapfile 2>/dev/null
size=`/opt/sbin/swapon -s | grep /opt/swapfile | awk '{print $3}'`
if [ -z "${size}" ] ; then
optlog "$scriptname" "Failed to mount swap ${swapdev}"
else
optlog "$scriptname" "swap device ${swapdev} already allocated (${size} kB)"
fi
fi
fi
;;
stop)
/opt/sbin/swapon -s | grep "^/dev/" | awk '{print $1}' >/tmp/swaps
while read swapdev ; do
/opt/sbin/swaponoff ${swapdev}
if [ ${?} -eq 0 ] ; then
optlog "$scriptname" "Unmounted swap partition ${swapdev}"
else
optlog "$scriptname" "Failed to unmount swap partition ${swapdev}"
fi
done</tmp/swaps
;;
*)
echo "Usage: $0 (start|stop)"
;;
esac
exit 0
```

Rtorrent_rutorrent_lighttpd

When finished editing, reboot your router.

Issue this command to check if swap has been mounted or not.

```
swapon -s
```

You should see /dev/sda2 under filename.

--[Leonidv](#) 17:43, 17 May 2011 (CEST)

SSL & Password Protection

SSL and a password protected page will be necessary for safe remote login. What this section will explain:

1. Setup an encrypted web access for rutorrent (<https://routerip/rutorrent>)
2. Password protect rutorrent

I strongly suggest updating the firmware to a latest one before proceeding. This implementation was crashing an RT-N16 router having build `dd-wrt.v24-14929_NEWD-2_K2.6_big.bin`

I've been testing the SSL feature with a `dd-wrt.v24-16994_NEWD-2_K2.6_mega.bin` on an ASUS RT-N16 which is stable. So, consult the forum on which firmware is best for your router.

There will be a series of commands to be executed that is best done in Linux, outside the router. Fire up an ubuntu environment or whichever you are comfortable with.

- To transfer files easily from Ubuntu to the router, we will be using a webserver on ubuntu. If you have your own way of doing this, skip this step.

File Transfer from ubuntu

```
sudo apt-get update
sudo apt-get install lighttpd
sudo service lighttpd start
```

This will install and start the lighttpd webserver. From Ubuntu, browse to <http://localhost>
The browser should display **lighttpd server is running**. or **Index of** displaying the lighttpd version on the bottom of the page.

```
ip a
```

Under eth0, find inet. The ip address next to inet is the ip address of Ubuntu. Note this address.

htpasswd

```
sudo apt-get install apache2-utils
```

for htpasswd command

```
cd /var/www
```

Rtorrent_rutorrent_lighttpd

```
sudo htpasswd -c /var/www/.auth yourUserName
```

Type and confirm your password. To see the file,

```
ls -a /var/www/
```

The `-a` switch shows hidden files. Files starting with a period (.) are hidden in Linux environments. This **.auth** file will be used to authenticate to the rutorrent server. You will need to supply the same username and password used for the `htpasswd` command to login to rutorrent.

SSL certificate

```
cd /var/www
sudo openssl req -new -newkey rsa:1024 -days 365 -nodes -x509 -keyout lighttpd.pem -out lighttpd.
```

You don't need to provide your true identity. Now, `ls -a` to check if the `lighttpd.pem` file is there. Ok, the **.auth** and **lighttpd.pem** files are ready to be used.

Now, log into your router

```
cd /opt/share/www/lighttpd
wget http://[ip of ubuntu]/.auth
wget http://[ip of ubuntu]/lighttpd.pem
```

These two files have now been saved in your router.

As usual, winSCP and open `/opt/etc/lighttpd/lighttd.conf`
Before making any changes to this file, backup with

```
cp lighttpd.conf lighttpd.conf.bak
```

We'll need to do some editing. Don't forget to check with winSCP asking for permission to save the file after modifying it. To be able to use winSCP with dd-wrt, open the dd-wrt web interface, services -> services -> secure shell. Enable SSHd, SSH TCP Forwarding and Password Login. In winSCP, select SCP for the file protocol.

Find these lines in the config file

```
server.modules      = (
    mod_***,
    mod_xxx,
)
```

Add the following block of codes on a newline after the closing bracket

```
#ssl configuration
$SERVER["socket"] == ":443" {
    ssl.engine = "enable"
    ssl.pemfile = "/opt/etc/lighttpd/lighttpd.pem"
}

server.modules += ( "mod_auth" ) #for authentication
```

Rtorrent_rutorrent_lighttpd

```
auth.backend = "htpasswd"
auth.backend.htpasswd.userfile = "/opt/etc/lighttpd/.auth"
auth.debug = 2
auth.require = ( "" =>
    (
        "method" => "basic",
        "realm" => "Authorized users only",
        "require" => "Basic"
    )
)

service lighttpd restart
```

Browse to <https://routerip/rutorrent>

Proceed anyway

--Leonidv 18:43, 8 June 2011 (CEST)