

Web Server Configuration (uHTTPd)

The `/etc/config/uhttpd` configuration is provided by the `uhttpd` web server package used since OpenWrt 10.03 (Backfire). This file defines the behaviour of the server and default values for certificates generated for SSL operation. `uhttpd` supports multiple instances (i.e. multiple listen ports, each with its own document root and other features) as well as `cgi`, `php5`, `perl` and `lua`.

Sections

There are two sections defined, the section of type `uhttpd` contains general server settings while the `cert` one defines the default values for SSL certificates.

Server Settings

A minimal `uhttpd` config section must consist of at least the document root and HTTP listen options:

```
config 'uhttpd' 'main' option 'listen_http' '80' option 'home' '/www'
```

The options defined for this section are outlined below.

| Name | Type | Required | Default | Description |
|------------------------------|--|--|------------------------------|---|
| <code>listen_http</code> | list of port numbers or address:port pairs | yes, if 'listen_https' is not given | <i>(none)</i> | Specifies the ports and addresses to listen on for plain HTTP access. If only a port number is given, the server will attempt to serve both IPv4 and IPv6 requests. Use <code>0.0.0.0:80</code> to bind at port 80 only on IPv4 interfaces or <code>:::80</code> to serve only IPv6. To run on multiple interfaces, specifying each, you can list one interface (or interface:port) per line. |
| <code>listen_https</code> | list of port numbers or address:port pairs | yes, if 'listen_http' is not given | <i>(none)</i> | Specifies the ports and addresses to listen on for encrypted HTTPS access. The format is the same as for <code>listen_http</code> . Read below for extra details |
| <code>home</code> | directory path | yes | <code>/www</code> | Defines the server document root |
| <code>cert</code> | file path | yes if <code>listen_https</code> is given, else no | <code>/etc/uhttpd.crt</code> | ASN.1/DER certificate used to serve HTTPS connections |
| <code>key</code> | file path | yes if <code>listen_https</code> is given, else no | <code>/etc/uhttpd.key</code> | ASN.1/DER private key used to serve HTTPS connections |
| <code>cgi_prefix</code> | string | no | <code>/cgi-bin</code> | Defines the prefix for CGI scripts, relative to the document root. CGI support is disabled if this option is missing |
| <code>lua_prefix</code> | string | no | <i>(none)</i> | Defines the prefix for dispatching requests to the embedded Lua interpreter, relative to the document root. Lua support is disabled if this option is missing |
| <code>lua_handler</code> | file path | yes if <code>lua_prefix</code> is given, else no | <i>(none)</i> | Lua handler script used to initialize the Lua runtime on server start |
| <code>script_timeout</code> | integer | no | 60 | Maximum wait time for CGI or Lua requests in seconds. Requested executables are terminated if no output was generated until the timeout expired |
| <code>network_timeout</code> | integer | no | 30 | Maximum wait time for network activity. Requested executables are terminated and connection is shut down if no network activity occurred for the specified number of seconds |
| <code>realm</code> | string | no | <i>local hostname</i> | Basic authentication realm when prompting the client for credentials (HTTP 400) |

| Name | Type | Required | Default | Description |
|-----------------|-----------|----------|---|--|
| config | file path | no | /etc/httpd.conf | Config file in Busybox httpd format for additional settings (currently only used to specify Basic Auth areas) |
| index_file | file name | no | index.html, index.htm, default.html, default.htm | Index file to use for directories, e.g. add index.php when using php |
| index_page | file name | no | index.html | Index file to use for directories, e.g. add index.php when using php (last, 20131015, replace index_file ?) should be noted: list index_page "index.html index.htm default.html default.htm index.php" |
| error_page | string | no | (none) | Virtual URL of file or CGI script to handle 404 request. Must begin with '/' |
| no_symlinks | boolean | no | 0 | Do not follow symbolic links if enabled |
| no_dirlists | boolean | no | 0 | Do not generate directory listings if enabled |
| http_keepalive | integer | no | 20 | connection reuse. Some bugs have been seen, you <i>may</i> wish to disable this by setting to 0 (BB or later only) |
| max_requests | integer | no | 3 | Maximum number of concurrent requests. If this number is exceeded, further requests are queued until the number of running requests drops below the limit again. |
| max_connections | integer | no | 100 | Maximum number of concurrent connections. If this number is exceeded, further TCP connection attempts are queued until the number of active connections drops below the limit again. |

Multiple sections if the type *uhttpd* may exist - the init script will launch one webserver instance per section.

HTTPS Enable and Certificate Settings and Creation

First of all, you need to install the `uhttpd-mod-tls` package in order to pull into the system the *'TLS plugin which adds HTTPS support to uHTTPd'*.

Then if `listen_https` is defined in the server configuration, the certificate and private key is missing. In this case (as of 10.03.1) you'll need to install the `luci-ssl` meta-package which in turn will pull also the `px5g` script. With this utility the init script will generate the appropriate certificate and key files when the server is started for the first time, either by reboot or by manual restart.

The `/etc/config/uhttpd` file contains in the end a section detailing the certificate and key files creation parameters:

| Name | Type | Required | Default | Description |
|------------|---------|----------|---------|---|
| days | integer | no | 730 | Validity time of the generated certificates in days |
| bits | integer | no | 1024 | Size of the generated RSA key in bits |
| country | string | no | DE | ISO country code of the certificate issuer |
| state | string | no | Berlin | State of the certificate issuer |
| location | string | no | Berlin | Location/city of the certificate issuer |
| commonname | string | no | OpenWrt | Common name covered by the certificate |

Those will be needed only once, at the next restart.

Basic Authentication (httpd.conf)

For backward compatibility reasons, *uhttpd* uses the old *Busybox httpd* config file `/etc/httpd.conf` to define authentication areas and the associated usernames and passwords. This configuration file is **not** in UCI format and usually shipped or generated by external packages like `webif` (X-Wrt).

Authentication realms are defined in the format `prefix:username:password` with one entry per line followed by a newline.

- `prefix` is the URL part covered by the realm, e.g. `/cgi-bin` to request basic auth for any CGI program
- `username` specifies the username a client has to login with
- `password` defines the secret password required to authenticate

The password can be either in plain text format, MD5 encoded or in the form `puser` where `user` refers to an account in `/etc/shadow` or `/etc/passwd`.

A plain text password can be converted to MD5 encoding by using the `-m` switch of the *uhttpd* executable:

```
root@OpenWrt:~# uhttpd -m secret
$1$ysVNzQc4CTMkp5da0dZ.3/
```

If the `p...` format is used, *uhttpd* will compare the client provided password against the one stored in the `shadow` or `passwd` database.

URL decoding

Like *Busybox HTTPd*, the `URL` decoding of strings on the command line is supported through the `-d` switch:

```
root@OpenWrt:/# uhttpd -d "An%20URL%20encoded%20String%21%0a"
An URL encoded String!
```

Using PHP5

A minimal `php5` installation includes

- `php5`
- `php5-cgi`

In `/etc/php.ini` ensure that the `doc_root` is empty if you are using multiple `uhttpd` instances (each on its own port). This enables the `uhttpd` 'home' variable to work for you.

Ensure that you uncomment the extension interpreter line for PHP in the main section of the `uHTTPd` config file:

```
list interpreter ".php=/usr/bin/php-cgi"
```

Securing uHTTPd

By default, `uHTTPd` is bind to `0.0.0.0` which also includes the WAN port of your router. To bind `uHTTPd` to the `LAN` port only you have to change the `listen_http` and `listen_https` options to your `LAN` IP address.

To get your current `LAN` IP address run this command:

```
uci get network.lan.ipaddr
```

```
config uhttpd main # HTTP listen addresses, multiple allowed list listen_http 192.168.1.1:80 # list listen_http [::]:80 # HTTPS listen
addresses, multiple allowed list listen_https 192.168.1.1:443 # list listen_https [::]:443
```

Embedded Lua

`uHTTPd` supports running Lua in-process, which can speed up Lua CGI scripts. ~~It is unclear whether LuCI supports running in this embedded interpreter.~~ LuCI seems to work fine (if not better) with the embedded Lua interpreter. See the next subsection for instructions on how to set it up.

Here is an example using a test file `test.lua` to show it works:

```
root@OpenWrt:~# opkg install uhttpd-mod-lua
Installing uhttpd-mod-lua (18) to root...
Downloading http://downloads.openwrt.org/snapshots/trunk/ar71xx/packages/uhttpd-mod-lua_18_ar71xx.ipk.
Configuring uhttpd-mod-lua.
root@OpenWrt:~# uci set uhttpd.main.lua_prefix=/lua
root@OpenWrt:~# uci set uhttpd.main.lua_handler=/root/test.lua
root@OpenWrt:~# cat /root/test.lua
function handle_request(env)
    uhttpd.send("Status: 200 OK\r\n")
    uhttpd.send("Content-Type: text/plain\r\n\r\n")
    uhttpd.send("Hello world.\n")
end
root@OpenWrt:~# /etc/init.d/uhttpd restart
root@OpenWrt:~# wget -qO- http://127.0.0.1/lua/
Hello world.
root@OpenWrt:~#
```

~~Taken from <http://pastebin.com/8H9eqiJ3> [<http://pastebin.com/8H9eqiJ3>]. Tested on Backfire 10.03.1 with uHTTPd 28.~~ Taken from <http://pastebin.com/8H9eqiJ3> [<http://pastebin.com/8H9eqiJ3>] with minor fix to make it pass on Barrier Breaker. Tested on Barrier Breaker 14.07 with uHTTPd 2014-08-25.

LuCI with embedded Lua interpreter

You need to install `uhttpd-mod-lua` and `luci-sgi-uhttpd` to get it to work:

```
root@OpenWrt:~# opkg install uhttpd-mod-lua luci-sgi-uhttpd
```

In Chaos Calmer 15.05, the `luci-sgi-uhttpd` package is not needed. The appropriate files are included in `luci-base`.

Then uncomment the following lines in `/etc/config/uhttpd` (or add them if you don't have them):

```
option lua_prefix      /luci
option lua_handler     /usr/lib/lua/luci/cgi/uhttpd.lua
```

Then restart the server:

```
root@OpenWrt:~# /etc/init.d/uhttpd restart
```

One thing remains to be done. By default `/www/index.html` redirects you to `/cgi-bin/luci` which is the default CGI gateway for LuCI. The config above puts LuCI through embedded interpreter under `/luci` (`lua_prefix` is what causes it) so you have to change that in `/www/index.html`. The path appears there twice (one for the meta tag which does the redirect and one for the anchor). You can also copy/paste the code below if you don't want to meddle with it on your own.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="refresh" content="0; URL=/luci" />
</head>
<body style="background-color: black">
<a style="color: white; text-decoration: none" href="/luci">LuCI - Lua Configuration Interface</a>
</body>
</html>
```

Also remember to flush the browser's cache if you relied on the redirection because otherwise it will probably keep redirecting you to `/cgi-bin/luci` until the cache expires by itself.