

NAME

sws — a simple web server

SYNOPSIS

```
sws [-dh] [-c dir] [-i address] [-l file] [-p port] [-s dir -k key] dir
```

DESCRIPTION

sws is a very simple web server. It behaves quite like you would expect from any regular web server, in that it binds to a given port on the given address and waits for incoming HTTP/1.0 requests. It serves content from the given directory. That is, any requests for documents is resolved relative to this directory (the document root).

OPTIONS

The following options are supported by **sws**:

- c** *dir* Allow execution of CGIs from the given directory (relative to the document root). See CGIs for details.
- d** Enter debugging mode. That is, do not daemonize, only accept one connection at a time and enable logging to stdout.
- h** Print a short usage summary and exit.
- i** *address* Bind to the given IP address. If not provided, **sws** will listen on all addresses on this host.
- l** *file* Log all requests to the given file. See LOGGING for details.
- p** *port* Listen on the given port. If not provided, **sws** will listen on port 8080.
- s** *dir* -**k** *key* Enable "secure" mode for the given directory. That is, encrypt all content from this directory with the given key. See ENCRYPTION for details.

DETAILS

sws speaks a crippled dialect of HTTP/1.0: it responds to GET and HEAD requests according to RFC1945, but may not support POST requests and only supports the If-Modified-Since Request-Header.

When a connection is made, **sws** will respond with the appropriate HTTP/1.0 status code and the following headers:

Date	The current timestamp in GMT.
Server	A string identifying this server and version.
Last-Modified	The timestamp in GMT of the file's last modification date.
Content-Type	<i>text/html</i> or <i>x-blowfish</i>
Content-Length	The size in bytes of the data returned.

If the request type was a GET, then it will subsequently return the data of the requested file. After serving the request, the connection is terminated.

FEATURES

sws supports a number of interesting features:

CGIs	Execution of CGIs as described in <code>CGIs</code> .
Directory Indexing	If the request was for a directory and the directory does not contain a file named "index.html", then <code>sws</code> will generate a directory index, listing the contents of the directory in alphanumeric order. Files starting with a "." are ignored.
User Directories	If the request begins with a "~", then the following string up to the first slash is translated into that user's <code>sws</code> directory (ie <code>/home/<user>/sws/</code>).
Encrypted Content	If the <code>-s</code> and <code>-k</code> flags were specified, then <code>sws</code> will encrypt the contents of the given directory before returning them to the client.

CGIs

Instead of returning the contents of each file under the directory specified via the `-c` flag, they are executed and any output generated is returned instead. Executing of CGIs follows the specification at <http://hoohoo.ncsa.uiuc.edu/cgi/>.

LOGGING

Per default, `sws` does not do any logging. If explicitly enabled via the `-l` flag, `sws` will log every request in a slight variation of Apache's so-called "common" format: `'%a %t "%r" %>s %b'`. That is, it will log:

- `%a` The remote IP address.
- `%t` The time the request was received (in GMT).
- `%r` The (quoted) first line of the request.
- `%>s` The status of the request.
- `%b` Size of the response in bytes. Ie, "Content-Length".

All lines will be appended to the given file unless `-d` was given, in which case all lines will be printed to std-out.

ENCRYPTION

`sws` allows for a given directory to be only served encrypted if the `-s dir` flag was specified. (This flag turns off directory indexing for the given directory as well.)

The contents of any file under this directory are encrypted using OpenSSL's blowfish algorithm (see `BF_cbc_encrypt(3)`). The key to encrypt the data is provided via the `-k key` flag. `sws` prevents leaking of the key into the process table via `setproctitle(3)`.

For files served from this directory, `sws` will set the `Content-Type` header to "x-blowfish".

EXIT STATUS

The `sws` utility exits 0 on success, and >0 if an error occurs.

SEE ALSO

`RFC1945`, `BF_cbc_encrypt(3)`, `setproctitle(3)`

HISTORY

A simple http server has been a frequent final project assignment for the class *Advanced Programming in the UNIX Environment* at Stevens Institute of Technology. This variation was first assigned in the Fall 2008 by Jan Schaumann.