htsh – Installation and Administration Guide

http://www.exolution.de/wapsh
mailto:wapsh@exolution.de

Version 1.0, last revised 21 Nov 2000

1. Introduction

For an overview of the operation of *htsh*, please read the *htsh User Guide* before reading this document.

2. Installation

2.1. Software requirements

The following software is required to run *htsh*:

- *Apache* web server (installed on *htsh HTTP Server Host*)
- SSL certificate for *htsh* (optional)
- PHP 4.0.2 or later (installed on *htsh HTTP Server Host*)
- *htsh* web-server application (installed on *htsh HTTP Server Host*)
- *htshd* (*htsh* server daemon) (installed on *htsh Login Host*)

2.2. HTTP Server Host Setup

2.2.1. Create a `loginhosts` file (optional)

By default, the *HTTP Server* and the *htsh* server daemon (*htshd*) must reside on the same machine (i.e. *localhost*). This means that users can only login on the machine running the *HTTP Server*. If you wish to place the *HTTP Server* and *htshd* on separate machines, or to allow the *HTTP Server* to establish login connections to other *Login Hosts* in addition to *localhost*, then two steps must be performed:

1. When starting the *htshd* program the set of *HTTP Server* addresses from which login connections will be accepted must be listed either in the *htshd* configuration file or on the *htshd* command line. This step is described in detail later in this guide.

2. In the directory on the *HTTP Server* which contains the *htsh* web application, create a file called `loginhosts` in the *htsh* web directory. This file lists the *Login Hosts* on which logins are permitted. This step is described in detail in the remainder of this section.

The `loginhosts` file lists the *Login Hosts* on which the *htsh* web application will allow users to login. (These hosts are displayed as a selectable list in the *Client Browser* on the *htsh* login page, unless the list consists of just one host, in which case a selectable list would be redundant, and is therefore not displayed.) The format of each line in this file is as follows:

[*alias*] *address*

The *alias* specifies the name to be displayed to the user on the *htsh* login page. The *address* is the DNS name or numeric dot address to which this name corresponds. If *alias* is omitted then the *address* value is displayed in the *htsh* login dropdown list. Empty lines and lines beginning with a hash (#) character in the *loginhosts* file are ignored. Here is a sample `loginhosts` file:

```
freyr
localhost 127.0.0.1
mimir 192.168.1.61
jord jord.exolution.lan
```

2.2.2. Create an *htsh* service in /etc/services

A service named **htsh** must be created in the file **`/etc/services`** on the *HTTP Server* host. The port number specified for this service must match the port number specified for the *htsh* service on the *Login Host*.

2.3. htshd *(*htsh *server daemon) Setup and execution*

The following steps must be performed in order to set up the *htshd* server.

2.3.1. Create an *htsh* service in /etc/services

A service named **htsh** must be created in the file **`/etc/services`** on each *Login Host*. (If the *Login Host* and the *HTTP Server Host* are the same machine, then only a single `/etc/services` file needs to be updated).

2.3.2. Building the *htshd* application daemon

A short shell script (`Build_htshd`) is provided which checks the flavour of Unix in use, as well as a few other system options, and then compiles and links the *htshd* program.

2.3.3. *htshd* installation

Just copy the *htshd* executable produced by the previous step into any desired directory, for example `/usr/sbin`.

2.3.4. *htshd* startup

There are two ways to start the *htsh* server daemon.

Startup via inetd(8)

This is the preferred method for starting *htshd*. Creating an *inetd* entry for the *htsh* service, ensures that *htshd* is automatically started whenever a user tries to connect to the service. To configure *htshd* to be run by *inetd*, add a line of the following form to `/etc/inetd.conf`:

```
htsh  stream  tcp   nowait  root  /some-path/htshd  htshd  cmd-line-args
```

Where *some-path* should be the pathname of the directory where *htshd* resides, and *cmd-line-args* are any desired command line arguments (as defined below). (Note that some versions of *inetd* place quite a short limit on the number of command line arguments that may be specified in `inetd.conf`. If you need to specify more command line arguments than are permitted, you should instead use the *htshd* configuration file (described below) to specify startup options for *htshd*.

Manual startup

While logged in as *root*, it is possible to simply run *htshd* from a shell prompt. Simply enter the command:

```
# htshd
```

In this case *htshd* automatically becomes a daemon (unless the *–D n* command line option or the *runasdaemon n* configuration option is specified). Starting *htshd* manually may be useful for running *htshd* in debug mode (command line option *–d 7777*) or for one-off testing of the operation of *htshd*.

2.4. htshd *initialisation*

Upon startup, *htshd* first processes its configuration file (if present) and then processes the *htshd* command line options. Running the daemon without specifying a configuration file or specifying any command line options results in the following defaults:

- Login requests are accepted only from an *HTTP Server* residing on the same host (i.e. *localhost*) as *htshd*.
- Any user with a valid username/password may login on the *Login Host*.

Most initialisation options can appear in either the configuration file, or on the *htshd* command line. Note that command line options may override configuration file options, since the command line is processed after the configuration file.

Note that certain *htshd* configuration file options and command line options have corresponding commands which may be used in the (global or user specific) *htsh* initialisation files to set the same values on a per-user basis. Since the initialisation files are processed at the time a user logs in, settings in these files will override corresponding *htshd* configuration file and command line options. The *htshd* options which may be overridden in the *htsh* initialisation files are noted below.

2.4.1. *htshd* configuration file

The *htsh* configuration file can be used to specify configuration options for *htshd*. By default *htshd* expects to find the configuration file at `/etc/htshd.conf`. If desired, an alternate location can be specified using the *–f* command line option to *htshd*, or processing of the configuration file can be skipped entirely using the *–F* option. If the default configuration file is missing, and no alternate file is specified, *htshd* silently skips configuration file processing.

Configuration file format

- The configuration file is line oriented. Commands consist of series of an initial keyword followed by one or more space-delimited arguments.
- Blank lines, and comment lines (those whose *initial* character is '#") are ignored.
- Lines (other than comment lines) can be continued onto the following line by preceding the newline with a backslash. (The backslash and newline are removed.)

Configuration File commands

The following commands may be placed in the configuration file. Each command may be repeated multiple times. In the case of commands setting a single value, only the last instance of the command will have effect. For those commands (*httpserver* and *user*) which can take a list of arguments, all arguments across all instances of the command are accumulated into a single set.

`allowedprotocols` *`proto-name...`*	Specify the set of protocols which may be used to login to *htsh*. Permitted protocol names are (lowercase) "wap" and "http". The default is to allow logins using any protocol. This option corresponds to the *set csmaxtransfersize* command within either of the *htsh* initialisation files. Note that the initialisation file command does *not* override this configuration file option – instead the initialisation file command can only be used to reduce the available protocols from the set specified by this configuration file option. Example: `allowedprotocols http`

`csmaxtransfersize` ***`nbytes`***	Specify the largest number of bytes that will be transferred in a single block by the *Client Shell* to the *HTTP Server*. This is useful to prevent large outputs from choking the browser client. Otherwise, the *HTTP Server*/*Client Browser* could be inundated with large amounts of output and the user would be prevented from sending further input (for example a *Control-C* to abort the command generating the output) until all of the output has been completed. This option corresponds to (and is overridden by) the *set csmaxtransfersize* command within either of the *htsh* initialisation files. Note however, that this configuration file option specifies an upper limit on the value which can be set via the corresponding initialisation file command. Default = 10000 bytes. Attempts to set this value less than 1000 bytes result in a setting of 1000 bytes. Example: `csmaxtransfersize 5000`
`csoutputtimeout` ***`nsecs`***	Specify the time for which the *Client Shell* will wait for any further shell output (after receiving shell input) before informing the *HTTP Server* that output is complete in response to the most recent shell input. To avoid slow response times, this should be set to some small value (usually less than one second). This option corresponds to (and is overridden by) the *set csoutputtimeout* command within either of the *htsh* initialisation files. Default is 0.5 seconds Example: `csoutputtimeout 0.2`
`globalprofile` ***`pathname`***	Identifies name of global initialisation file to be executed by all *htsh* logins. This can be used to set options and shortcuts defined for all users. Default is to use (if present) `/etc/htsh_profile`
`httpserver` ***`server-addr...`***	Specify the (*HTTP Server*) hosts from which the *htshd* daemon will accept login requests and shell input messages. This option may be specified repeatedly: the set of permitted servers is the union of all servers listed in all *httpserver* configuration file options and using the *–H* command line option to *htshd*. The *HTTP Server* addresses may specified as symbolic host names or in Internet dot-address notation. Default is to only allow connections from *localhost (127.0.0.1)*, so that the *HTTP Server* and the *htshd* server daemon must reside on the same host. Example: `httpserver 192.168.1.110 localhost`

`logfiledirectory` ***`path`***	Specify a directory into which *script(1)* type logs of each login session will be recorded. Default is to create no log files. Example: `logfiledirectory /root/htsh`
`outputbufferlimit` ***`nbytes`***	Set the upper limit (in bytes) for the size of the buffer used to record shell output. *htsh* maintains a separate output buffer for each login session. Default is 100000 bytes. This option corresponds to (and is overridden by) the *set outputbufferlimit* command within either of the *htsh* initialisation files. Note however, that this configuration file option specifies an *upper limit* on the value which can be set via the corresponding initialisation file command. Example: `outputbufferlimit 200000`
`port` ***`port-number`***	Specifies the TCP port number (or TCP service name in `/etc/services`) to be used by *htshd* to listen for login requests. The default is to use the port specified in `/etc/services` under the TCP service name *htsh* Example: `port 60001`
`runasdaemon` ***`{y|n}`***	Specifies whether to run *htshd* as a background daemon. By default, *htshd* is run as a daemon. The only reason not to run *htshd* as a daemon is to for debugging purposes: if *htshd* is run as a foreground process, debugging output will be sent to standard error instead of the system error logger. If started via *inetd(8)*, then *htshd* is automatically made a daemon, and this option has no effect.
`shelltimeout` ***`num-secs`***	Specify (in seconds) the timeout to be used whenever the *Client Shell* waits for further shell input from the *Client Browser*. If this timeout is exceeded then the *Client Shell* automatically terminates. This facility is required to deal with the possibility that the user shuts down their browser without doing a logout. This option corresponds to (and is overridden by) the *set shelltimeout* command within either of the *htsh* initialisation files. Default = 1800 seconds *(30 minutes)* Example: `shelltimout 600` *(10 minutes)*
`user` ***`username...`***	Specify users who is allowed to login via *htsh*. This option may be specified repeatedly: the set of permitted users is the union of all users listed in all *user* configuration file options and using the *–u* command line option to *hthsd*. If this option is not given, then any user with a valid username/password on the *Login Host* is permitted to login. Example: `user mtk thk hat`

2.4.2. *htshd* command line options

The following command line options are permitted for the `htshd` command. Most of these options can also be set using corresponding configuration file commands.

`-b` *nbytes*	Set the upper limit (in bytes) for the size of the buffer used to record shell output. This corresponds to (and overrides) the *outputbufferlimit* configuration file option (and further information can be found under the description of that option). Example: `-b 100000`
`-c` *nbytes*	Specify the largest number of bytes that will be transferred in a single block by the *Client Shell* to the *HTTP Server*. This corresponds to (and overrides) the *csmaxtransfersize* configuration file option (and further information can be found under the description of that option). Example: `-c 10000`
`-d` *octal-mask*	Run in debugging mode. Voluminous debugging output (according to the given *octal-mask* value) is sent either to the system error logger (if running as a daemon) or standard error (if running as foreground process).
`-D` *[y|n]*	Specifies whether to run *htshd* as a background daemon. This corresponds to (and overrides) the *runasdaemon* configuration file option (and further information can be found under the description of that option). Example: `-D y`
`-f` *config-file*	Specifies an alternate name for the file which contains configuration options for *htshd*. (This file represents an alternative/and or complementary way of setting options that are set on the *htshd* command) Default is to use (if present) `/etc/htshd.conf`
`-F`	Do not execute the default configuration file. *htshd* will give an error, and fail to start, if you specify both the `–f` and `–F` options.
`-g` *global-rc-file*	Identifies name of global initialisation file to be executed by all *htsh* logins. This can be used to set options and shortcuts defined for all users. This corresponds to (and overrides) the *globalprofile* configuration file option (and further information can be found under the description of that option). Default is to use (if present) `/etc/htsh_profile`
`-h`	Show usage message, including default values used for configuration settings

`-H` *`http-server-addr`*	Specify the (*HTTP Server*) hosts from which the daemon will accept login requests and shell input messages. This option may be specified repeatedly, permitting connections to be made from multiple *HTTP Server*s. This corresponds to the *httpserver* configuration file option (and further information can be found under the description of that option). Example: `-H 192.168.1.10 -H localhost`
`-L` *`path`*	Specify a directory in which script(8) type logs of each login session will be recorded. This corresponds to the *logfiledirectory* configuration file option (and further information can be found under the description of that option). Example: `-L /root/htsh`
`-o output-resp-delay`	Specify the time for which the *Client Shell* will wait for any further shell output (after receiving shell input) before informing the *HTTP Server* that output is complete in response to the most recent shell input. This corresponds to (and overrides) the *csoutputtimeout* configuration file option (and further information can be found under the description of that option). Example: `-o 0.2`
`-p` *`port-num`*	Specifies the TCP port number (or TCP service name in `/etc/services`) to be used by *htshd* to listen for login requests. This corresponds to (and overrides) the *port* configuration file option (and further information can be found under the description of that option). Example: `-p 60000`
`-P '`*`protocol-name...`*`'`	Specify the list of protocols which may be used to login to *htsh*. If more than one protocol is specified, then the protocol names should be separated by spaces, and enclosed in quotes (to prevent word splitting by the shell). Permitted protocol names are (lowercase) “wap” and “http”. This corresponds to (and overrides) the *allowedprotocols* configuration file option (and further information can be found under the description of that option). Example: `-P 'wap http'`
`-t` *`numsecs`*	Specify (in seconds) the timeout to be used whenever the *Client Shell* waits for further shell input from the *Client Browser*. This corresponds to (and overrides) the *shelltimeout* configuration file option (and further information can be found under the description of that option). Example: `-t 600` *(10 minutes)*

`-u username`	Specify a user who is allowed to login via *htsh*. This option may be specified repeatedly, thus allowing a set of users to be specified. If this option is not given, then any user with a valid username/password on the *Login Host* is permitted to login. This corresponds to the *user* configuration file option (and further information can be found under the description of that option). Example: `-u mtk –u thk –u ht`

2.5. Global user initialisation file

When each user logins, *htsh* processes two files which specify how the *htsh* session should be specified. These are:

1. The global initialisation file executed by all users
2. The user-specific initialisation file (`~/.htshrc`).

The first of these files (by default the file `/etc/htsh_profile`) can be used by the system administrator to create shortcuts and set options which apply to all *htsh* logins. For details on the format of this file, see the *htsh User Guide*.

System administrators should take special note of the *set +o allowuserinit* initialisation file command which disables the processing of user-specific login initialisation files. This command can be used to create "captive" *htsh* logins whose shortcuts and other settings are defined by the administrator (in the *htsh* global initialisation file) and cannot be changed by users.

2.6. htshd exit status

htshd may exit with the following status values

0	Success (only occurs after displaying usage message in response to the *–h* option to *htshd*)
1	Error during startup
2	Error in configuration file

3. Security Features of *htsh*

The following points regarding security are noteworthy:

- To allow secure transmission of data across the Internet, an SSL certificate should be installed on the *HTTP Server*.
- It is possible for the *htsh* server daemon (*htshd*) on the *Login Host* to handle login requests from multiple *HTTP Server*s which may be on separate hosts from the *Login Host*. By default, connections are only accepted from an *HTTP Server* on the same computer as the *htsh* daemon. When starting the *htsh* daemon it is possible to specify a series of *HTTP Servers* from which the daemon will accept login requests.
- To prevent "spoofing" of shell input from malicious processes on an *HTTP Server*, at login, *htsh* generates a random *authorisation key* which is passed back to the *HTTP Server*. This authorisation must be transmitted by the *HTTP Server* with every subsequent shell input request. The *Client Shell* will reject any input which does not provide the correct authorisation key.
- For high security environments making use of WAP access, it is possible to install a WAP gateway on the same host as the *HTTP Server*, eliminating the need to transmit data across the Internet.