NAME

htpasswd - Manage user files for basic authentication

SYNOPSIS

htpasswd [**-c**] [**-i**] [**-m** | **-B** | **-d** | **-s** | **-p**] [**-C** *cost*] [**-D**] [**-v**] *passwdfile username*

htpasswd **-b** [**-c**] [**-m** | **-B** | **-d** | **-s** | **-p**] [**-C** *cost*] [**-D**] [**-v**] *passwdfile username password*

htpasswd **-n** [**-i**] [**-m** | **-B** | **-d** | **-s** | **-p**] [**-C** *cost*] *username*

htpasswd **-nb** [**-m** | **-B** | **-d** | **-s** | **-p**] [**-C** *cost*] *username password*

SUMMARY

htpasswd is used to create and update the flat-files used to store usernames and password for basic authentication of HTTP users. If **htpasswd** cannot access a file, such as not being able to write to the output file or not being able to read the file in order to update it, it returns an error status and makes no changes.

Resources available from the Apache HTTP server can be restricted to just the users listed in the files created by **htpasswd**. This program can only manage usernames and passwords stored in a flat-file. It can encrypt and display password information for use in other types of data stores, though. To use a DBM database see dbmmanage or htdbm.

htpasswd encrypts passwords using either bcrypt, a version of MD5 modified for Apache, SHA1, or the system's **crypt()** routine. Files managed by **htpasswd** may contain a mixture of different encoding types of passwords; some user records may have bcrypt or MD5-encrypted passwords while others in the same file may have passwords encrypted with **crypt()**.

This manual page only lists the command line arguments. For details of the directives necessary to configure user authentication in httpd see the Apache manual, which is part of the Apache distribution or can be found at http://httpd.apache.org/.

OPTIONS

-b Use batch mode; *i.e.*, get the password from the command line rather than prompting for it. This option should be used with extreme care, since **the password is clearly visible** on the command line. For script use see the **-i** option. Available in 2.4.4 and later.

-i Read the password from stdin without verification (for script usage).

-c Create the *passwdfile*. If *passwdfile* already exists, it is rewritten and truncated. This option cannot be combined with the **-n** option.

-n Display the results on standard output rather than updating a file. This is useful for generating password records acceptable to Apache for inclusion in non-text data stores. This option changes the syntax of the command line, since the *passwdfile* argument (usually the first one) is omitted. It cannot be combined with the **-c** option.

-m Use MD5 encryption for passwords. This is the default (since version 2.2.18).

-B Use bcrypt encryption for passwords. This is currently considered to be very secure.

-C This flag is only allowed in combination with **-B** (bcrypt encryption). It sets the computing time used for the bcrypt algorithm (higher is more secure but slower, default: 5, valid: 4 to 17).

-d Use **crypt()** encryption for passwords. This is not supported by the httpd server on Windows and Netware. This algorithm limits the password length to 8 characters. This algorithm is **insecure** by today's standards. It used to be the default algorithm until version 2.2.17.

-s Use SHA encryption for passwords. Facilitates migration from/to Netscape servers using the LDAP Directory Interchange Format (ldif). This algorithm is **insecure** by today's standards.

-p Use plaintext passwords. Though **htpasswd** will support creation on all platforms, the httpd daemon will only accept plain text passwords on Windows and Netware.

-D Delete user. If the username exists in the specified htpasswd file, it will be deleted.

-v Verify password. Verify that the given password matches the password of the user stored in the specified htpasswd file. Available in 2.4.5 and later.

passwdfile

Name of the file to contain the user name and password. If **-c** is given, this file is created if it does not already exist, or rewritten and truncated if it does exist.

username

The username to create or update in *passwdfile*. If *username* does not exist in this file, an entry is added. If it does exist, the password is changed.

password

The plaintext password to be encrypted and stored in the file. Only used with the **-b** flag.

EXIT STATUS

htpasswd returns a zero status ("true") if the username and password have been successfully added or updated in the *passwdfile*. **htpasswd** returns **1** if it encounters some problem accessing files, **2** if there was a syntax problem with the command line, **3** if the password was entered interactively and the verification entry didn't match, **4** if its operation was interrupted, **5** if a value is too long (username, filename, password, or final computed record), **6** if the username contains illegal characters (see the Restrictions section), and **7** if the file is not a valid password file.

EXAMPLES

```
htpasswd /usr/local/etc/apache/.htpasswd-users jsmith
```

Adds or modifies the password for user **jsmith**. The user is prompted for the password. The password will be encrypted using the modified Apache MD5 algorithm. If the file does not exist, **htpasswd** will do nothing except return an error.

```
htpasswd -c /home/doe/public_html/.htpasswd jane
```

Creates a new file and stores a record in it for user **jane**. The user is prompted for the password. If the file exists and cannot be read, or cannot be written, it is not altered and **htpasswd** will display a message and return an error status.

```
htpasswd -db /usr/web/.htpasswd-all jones Pwd4Steve
```

Encrypts the password from the command line (**Pwd4Steve**) using the **crypt()** algorithm, and stores it in the specified file.

SECURITY CONSIDERATIONS

Web password files such as those managed by **htpasswd** should *not* be within the Web server's URI space -- that is, they should not be fetchable with a browser.

This program is not safe as a setuid executable. Do *not* make it setuid.

The use of the **-b** option is discouraged, since when it is used the unencrypted password appears on the command line.

When using the **crypt()** algorithm, note that only the first 8 characters of the password are used to form the password. If the supplied password is longer, the extra characters will be silently discarded.

The SHA encryption format does not use salting: for a given password, there is only one encrypted representation. The **crypt()** and MD5 formats permute the representation by prepending a random salt string, to make dictionary attacks against the passwords more difficult.

The SHA and **crypt()** formats are insecure by today's standards.

RESTRICTIONS

On the Windows platform, passwords encrypted with **htpasswd** are limited to no more than **255** characters in length. Longer passwords will be truncated to 255 characters.

The MD5 algorithm used by **htpasswd** is specific to the Apache software; passwords encrypted using it will not be usable with other Web servers.

Usernames are limited to **255** bytes and may not include the character **:**.

The cost of computing a bcrypt password hash value increases with the number of rounds specified by the **-C** option. The **apr-util** library enforces a maximum number of rounds of 17 in version **1.6.0** and later.