NAME

pass – a simple password manager

SYNOPSIS

`pass` [`-f` *file*] [`-de`] [*keywords...*]

DESCRIPTION

pass manages a plaintext database of passwords, letting you encrypt/decrypt or search the encrypted database using *auth/aescbc(1)*. The plaintext database is `$home/lib/passwords` by default, but can be changed with the *-f file* flag. The encrypted database has the same name, but adds an *aes* suffix, eg. `$home/lib/passwords.aes`.

With no arguments *pass* will simply print the database. If you supply keyword arguments, it will search for lines in the database containing those keywords. (in both cases this will only work if you actually have an encrypted database)

Aside from *-f*, there are two other flags: *-d* will decrypt the password database, and *-e* will copy the plaintext file into an encrypted database, then overwrite the plaintext file with garbage before deleting it.

EXAMPLE

```
pass -d
echo Bank Pincode 12345 >> $home/lib/passwords
pass -e
pass bank
```
Bank Pincode 12345

BUGS

Setting up a *secstore(1)* on a cpu server makes this script redundant.

pass tries to purge the plaintext passwords from disk when encrypting the database, but it is technically impossible to guarantee that this is done. Filesystems and harddisks often do their own internal caching, which the script has no control over. Additionally no attempt is made to purge the plaintext passwords from system memory. *pass* makes it more difficult to recover your plaintext passwords, but not impossible.

Make sure that no filesystem dumps are taken when the database is decrypted, otherwise your plaintext passwords will be forever preserved in the filesystem backups!

There is no way to restore a corrupted database, so its wise to backup the encrypted database before making adjustments to it.

SEE ALSO

secstore(1)