NAME

EVP_KDF−PBKDF2 − The PBKDF2 EVP_KDF implementation

DESCRIPTION

Support for computing the **PBKDF2** password-based KDF through the **EVP_KDF** API.

The EVP_KDF−PBKDF2 algorithm implements the PBKDF2 password-based key derivation function, as described in SP800−132; it derives a key from a password using a salt and iteration count.

Identity

‘‘PBKDF2’’ is the name for this implementation; it can be used with the **EVP_KDF_fetch()** function.

Supported parameters

The supported parameters are:

‘‘pass’’ (**OSSL_KDF_PARAM_PASSWORD**) <octet string>
‘‘salt’’ (**OSSL_KDF_PARAM_SALT**) <octet string>
‘‘iter’’ (**OSSL_KDF_PARAM_ITER**) <unsigned integer>

This parameter has a default value of 2048.

‘‘properties’’ (**OSSL_KDF_PARAM_PROPERTIES**) <UTF8 string>
‘‘digest’’ (**OSSL_KDF_PARAM_DIGEST**) <UTF8 string>

These parameters work as described in ‘‘PARAMETERS’’ in **EVP_KDF** (3).

‘‘pkcs5’’ (**OSSL_KDF_PARAM_PKCS5**) <integer>

This parameter can be used to enable or disable SP800−132 compliance checks. Setting the mode to 0 enables the compliance checks.

The checks performed are:

− the iteration count is at least 1000.
− the salt length is at least 128 bits.
− the derived key length is at least 112 bits.

The default provider uses a default mode of 1 for backwards compatibility, and the fips provider uses a default mode of 0.

The value string is expected to be a decimal number 0 or 1.

NOTES

A typical application of this algorithm is to derive keying material for an encryption algorithm from a password in the ‘‘pass’’, a salt in ‘‘salt’’, and an iteration count.

Increasing the ‘‘iter’’ parameter slows down the algorithm which makes it harder for an attacker to perform a brute force attack using a large number of candidate passwords.

No assumption is made regarding the given password; it is simply treated as a byte sequence.

CONFORMING TO

SP800−132

SEE ALSO

EVP_KDF (3), **EVP_KDF_CTX_new** (3), **EVP_KDF_CTX_free** (3), **EVP_KDF_CTX_set_params** (3), **EVP_KDF_derive** (3), ‘‘PARAMETERS’’ in **EVP_KDF** (3)

HISTORY

This functionality was added to OpenSSL 3.0.

COPYRIGHT