NAME

SSL_CTX_use_psk_identity_hint, SSL_use_psk_identity_hint, SSL_CTX_set_psk_server_callback, SSL_set_psk_server_callback - set PSK identity hint to use

SYNOPSIS

```
#include <openssl/ssl.h>

int SSL_CTX_use_psk_identity_hint(SSL_CTX *ctx, const char *hint);
int SSL_use_psk_identity_hint(SSL *ssl, const char *hint);

void SSL_CTX_set_psk_server_callback(SSL_CTX *ctx,
unsigned int (*callback)(SSL *ssl, const char *identity,
unsigned char *psk, int max_psk_len));
void SSL_set_psk_server_callback(SSL *ssl,
unsigned int (*callback)(SSL *ssl, const char *identity,
unsigned char *psk, int max_psk_len));
```

DESCRIPTION

SSL_CTX_use_psk_identity_hint() sets the given **NULL**-terminated PSK identity hint **hint** to SSL context object **ctx**. *SSL_use_psk_identity_hint()* sets the given **NULL**-terminated PSK identity hint **hint** to SSL connection object **ssl**. If **hint** is **NULL** the current hint from **ctx** or **ssl** is deleted.

In the case where PSK identity hint is **NULL**, the server does not send the ServerKeyExchange message to the client.

A server application must provide a callback function which is called when the server receives the ClientKeyExchange message from the client. The purpose of the callback function is to validate the received PSK identity and to fetch the pre-shared key used during the connection setup phase. The callback is set using functions *SSL_CTX_set_psk_server_callback()* or *SSL_set_psk_server_callback()*. The callback function is given the connection in parameter **ssl**, **NULL**-terminated PSK identity sent by the client in parameter **identity**, and a buffer **psk** of length **max_psk_len** bytes where the pre-shared key is to be stored.

RETURN VALUES

SSL_CTX_use_psk_identity_hint() and *SSL_use_psk_identity_hint()* return 1 on success, 0 otherwise.

Return values from the server callback are interpreted as follows:

0 PSK identity was not found. An ‘‘unknown_psk_identity’’ alert message will be sent and the connection setup fails.

>0 PSK identity was found and the server callback has provided the PSK successfully in parameter **psk**. Return value is the length of **psk** in bytes. It is an error to return a value greater than **max_psk_len**.

If the PSK identity was not found but the callback instructs the protocol to continue anyway, the callback must provide some random data to **psk** and return the length of the random data, so the connection will fail with decryption_error before it will be finished completely.

COPYRIGHT