NAME

libssh2_userauth_publickey_sk - authenticate a session with a FIDO2 authenticator

SYNOPSIS

```
#include <libssh2.h>

int
libssh2_userauth_publickey_sk(LIBSSH2_SESSION *session,
                              const char *username,
                              size_t username_len,
                              const unsigned char *publickeydata,
                              size_t publickeydata_len,
                              const char *privatekeydata,
                              size_t privatekeydata_len,
                              const char *passphrase,
                              LIBSSH2_USERAUTH_SK_SIGN_FUNC((*sign_callback)),
                              void **abstract);
```

CALLBACK

```
#define LIBSSH2_SK_PRESENCE_REQUIRED     0x01
#define LIBSSH2_SK_VERIFICATION_REQUIRED 0x04

typedef struct _LIBSSH2_SK_SIG_INFO {
   uint8_t flags;
   uint32_t counter;
   unsigned char *sig_r;
   size_t sig_r_len;
   unsigned char *sig_s;
   size_t sig_s_len;
} LIBSSH2_SK_SIG_INFO;

int name(LIBSSH2_SESSION *session, LIBSSH2_SK_SIG_INFO *sig_info,
      const unsigned char *data, size_t data_len, int algorithm,
      uint8_t flags, const char *application,
      const unsigned char *key_handle, size_t handle_len,
      void **abstract);
```

DESCRIPTION

session - Session instance as returned by **libssh2_session_init_ex(3)**

username - Name of user to attempt authentication for.

username_len - Length of username parameter.

publickeydata - Buffer containing the contents of a public key file. If NULL, the public key will be extracted from the privatekeydata. When using certificate authentication, this buffer should contain the public certificate data.

publickeydata_len - Length of public key data.

privatekeydata - Buffer containing the contents of a private key file.

privatekeydata_len - Length of private key data.

passphrase - Passphrase to use when decoding private key file.

sign_callback - Callback to communicate with FIDO2 authenticator.

abstract - User-provided data to pass to callback.

Attempt FIDO2 authentication. using either the sk-ssh-ed25519@openssh.com or sk-ecdsa-sha2-nistp256@openssh.com key exchange algorithms.

This function is only supported when libssh2 is backed by OpenSSL.

CALLBACK DESCRIPTION

session - Session instance as returned by **libssh2_session_init_ex(3)**

sig_info - Filled in by the callback with the signature and accompanying information from the authenticator.

data - The data to sign.

data_len - The length of the data parameter.

algorithm - The signing algorithm to use. Possible values are LIBSSH2_HOSTKEY_TYPE_ED25519 and LIBSSH2_HOSTKEY_TYPE_ECDSA_256.

flags - A bitmask specifying options for the authenticator. When

LIBSSH2_SK_PRESENCE_REQUIRED is set, the authenticator requires a touch. When LIBSSH2_SK_VERIFICATION_REQUIRED is set, the authenticator requires a PIN. Many servers and authenticators do not work properly when LIBSSH2_SK_PRESENCE_REQUIRED is not set.

application - A user-defined string to use as the RP name for the authenticator. Usually "ssh:".

key_handle - The key handle to use for the authenticator's allow list.

handle_len - The length of the key_handle parameter.

abstract - User-defined data. When a PIN is required, use this to pass in the PIN, or a function pointer to retrieve the PIN.

The *sign_callback* is responsible for communicating with the hardware authenticator to generate a signature. On success, the signature information must be placed in the '*sig_info* sig_info parameter and the callback must return 0. On failure, it should return a negative number.

The fields of the LIBSSH2_SK_SIG_INFO are as follows.

flags - A bitmask specifying options for the authenticator. This should be read from the authenticator and not merely copied from the flags parameter to the callback.

counter - A value returned from the authenticator.

sig_r - For Ed25519 signatures, this contains the entire signature, as returned directly from the authenticator. For ECDSA signatures, this contains the r component of the signature in a big-endian binary representation. For both algorithms, use LIBSSH2_ALLOC to allocate memory. It will be freed by the caller.

sig_r_len - The length of the sig_r parameter.

sig_s - For ECDSA signatures, this contains the s component of the signature in a big-endian binary representation. Use LIBSSH2_ALLOC to allocate memory. It will be freed by the caller. For Ed25519 signatures, set this to NULL.

sig_s_len - The length of the sig_s parameter.

RETURN VALUE

Return 0 on success or negative on failure. It returns LIBSSH2_ERROR_EAGAIN when it would otherwise block. While LIBSSH2_ERROR_EAGAIN is a negative number, it is not really a failure per

se.

ERRORS

Some of the errors this function may return include:

LIBSSH2_ERROR_ALLOC - An internal memory allocation call failed.

LIBSSH2_ERROR_SOCKET_SEND - Unable to send data on socket.

LIBSSH2_ERROR_AUTHENTICATION_FAILED - failed, invalid username/key.

AVAILABILITY

Added in libssh2 1.10.0

SEE ALSO

libssh2_session_init_ex(3)