

Files

c483a5c

Go to file

> .github

> contrib

> m4

> openbsd-compat

> regress

.depend

.gitignore

.skipped-commit-ids

CREDITS

INSTALL

LICENCE

Makefile.in

OVERVIEW

PROTOCOL

PROTOCOL.agent

PROTOCOL.certkeys

PROTOCOL.chacha20poly1305

PROTOCOL.key

PROTOCOL.krl

PROTOCOL.mux

PROTOCOL.sshsig

PROTOCOL.u2f

README

README.dns

README.md

README.platform

README.privsep

README.tun

SECURITY.md

TODO

addr.c

addr.h

addrmatch.c

atomicio.c

atomicio.h

audit-bsm.c

openssh-portable / ssherr.c

 djmdjm upstream: improve the error message for u2f enrollment ... 59d01f1 · 5 years ago

History

Code

Blame

151 lines (149 loc) · 5.17 KB







Raw







```
1  /*      $OpenBSD: ssherr.c,v 1.10 2020/01/25 23:13:09 djm Exp $ */
2  /*
3   * Copyright (c) 2011 Damien Miller
4   *
5   * Permission to use, copy, modify, and distribute this software for any
6   * purpose with or without fee is hereby granted, provided that the above
7   * copyright notice and this permission notice appear in all copies.
8   *
9   * THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
10  * WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
11  * MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
12  * ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
13  * WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
14  * ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
15  * OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
16  */
17
18  #include <errno.h>
19  #include <string.h>
20  #include "ssherr.h"
21
22  const char *
23  ssh_err(int n)
24  {
25      switch (n) {
26      case SSH_ERR_SUCCESS:
27          return "success";
28      case SSH_ERR_INTERNAL_ERROR:
29          return "unexpected internal error";
30      case SSH_ERR_ALLOC_FAIL:
31          return "memory allocation failed";
32      case SSH_ERR_MESSAGE_INCOMPLETE:
33          return "incomplete message";
34      case SSH_ERR_INVALID_FORMAT:
35          return "invalid format";
36      case SSH_ERR_BIGNUM_IS_NEGATIVE:
37          return "bignum is negative";
38      case SSH_ERR_STRING_TOO_LARGE:
39          return "string is too large";
40      case SSH_ERR_BIGNUM_TOO_LARGE:
41          return "bignum is too large";
42      case SSH_ERR_ECPOINT_TOO_LARGE:
43          return "elliptic curve point is too large";
44      case SSH_ERR_NO_BUFFER_SPACE:
45          return "insufficient buffer space";
46      case SSH_ERR_INVALID_ARGUMENT:
47          return "invalid argument";
48      case SSH_ERR_KEY_BITS_MISMATCH:
49          return "key bits do not match";
50      case SSH_ERR_EC_CURVE_INVALID:
51          return "invalid elliptic curve";
52      case SSH_ERR_KEY_TYPE_MISMATCH:
53          return "key type does not match";
54      case SSH_ERR_KEY_TYPE_UNKNOWN:
55          return "unknown or unsupported key type";
56      case SSH_ERR_EC_CURVE_MISMATCH:
57          return "elliptic curve does not match";
```

-  audit-linux.c
-  audit.c
-  audit.h
-  auth-bsdauth.c
-  auth-krb5.c
-  auth-options.c

```
37         return "elliptic curve does not match";
38     case SSH_ERR_EXPECTED_CERT:
39         return "plain key provided where certificate required";
40     case SSH_ERR_KEY_LACKS_CERTBLOB:
41         return "key lacks certificate data";
42     case SSH_ERR_KEY_CERT_UNKNOWN_TYPE:
43         return "unknown/unsupported certificate type";
44     case SSH_ERR_KEY_CERT_INVALID_SIGN_KEY:
45         return "invalid certificate signing key";
46     case SSH_ERR_KEY_INVALID_EC_VALUE:
47         return "invalid elliptic curve value";
48     case SSH_ERR_SIGNATURE_INVALID:
49         return "incorrect signature";
50     case SSH_ERR_LIBCRYPTO_ERROR:
51         return "error in libcrypto"; /* XXX fetch and return */
52     case SSH_ERR_UNEXPECTED_TRAILING_DATA:
53         return "unexpected bytes remain after decoding";
54     case SSH_ERR_SYSTEM_ERROR:
55         return strerror(errno);
56     case SSH_ERR_KEY_CERT_INVALID:
57         return "invalid certificate";
58     case SSH_ERR_AGENT_COMMUNICATION:
59         return "communication with agent failed";
60     case SSH_ERR_AGENT_FAILURE:
61         return "agent refused operation";
62     case SSH_ERR_DH_GEX_OUT_OF_RANGE:
63         return "DH GEX group out of range";
64     case SSH_ERR_DISCONNECTED:
65         return "disconnected";
66     case SSH_ERR_MAC_INVALID:
67         return "message authentication code incorrect";
68     case SSH_ERR_NO_CIPHER_ALG_MATCH:
69         return "no matching cipher found";
70     case SSH_ERR_NO_MAC_ALG_MATCH:
71         return "no matching MAC found";
72     case SSH_ERR_NO_COMPRESS_ALG_MATCH:
73         return "no matching compression method found";
74     case SSH_ERR_NO_KEX_ALG_MATCH:
75         return "no matching key exchange method found";
76     case SSH_ERR_NO_HOSTKEY_ALG_MATCH:
77         return "no matching host key type found";
78     case SSH_ERR_PROTOCOL_MISMATCH:
79         return "protocol version mismatch";
80     case SSH_ERR_NO_PROTOCOL_VERSION:
81         return "could not read protocol version";
82     case SSH_ERR_NO_HOSTKEY_LOADED:
83         return "could not load host key";
84     case SSH_ERR_NEED_REKEY:
85         return "rekeying not supported by peer";
86     case SSH_ERR_PASSPHRASE_TOO_SHORT:
87         return "passphrase is too short (minimum five characters)";
88     case SSH_ERR_FILE_CHANGED:
89         return "file changed while reading";
90     case SSH_ERR_KEY_UNKNOWN_CIPHER:
91         return "key encrypted using unsupported cipher";
92     case SSH_ERR_KEY_WRONG_PASSPHRASE:
93         return "incorrect passphrase supplied to decrypt private key";
94     case SSH_ERR_KEY_BAD_PERMISSIONS:
95         return "bad permissions";
96     case SSH_ERR_KEY_CERT_MISMATCH:
97         return "certificate does not match key";
98     case SSH_ERR_KEY_NOT_FOUND:
99         return "key not found";
100    case SSH_ERR_AGENT_NOT_PRESENT:
101        return "agent not present";
102    case SSH_ERR_AGENT_NO_IDENTITYES:
103        return "agent contains no identities";
104    case SSH_ERR_BUFFER_READ_ONLY:
105        return "internal error: buffer is read-only";
106    case SSH_ERR_KRL_BAD_MAGIC:
107        return "KRL file has invalid magic number";
108    case SSH_ERR_KEY_REVOKED:
109        return "Key is revoked";
110    case SSH_ERR_CONN_CLOSED:
111        return "Connection closed";
```

```
132         case SSH_ERR_CONN_TIMEOUT:
133             return "Connection timed out";
134         case SSH_ERR_CONN_CORRUPT:
135             return "Connection corrupted";
136         case SSH_ERR_PROTOCOL_ERROR:
137             return "Protocol error";
138         case SSH_ERR_KEY_LENGTH:
139             return "Invalid key length";
140         case SSH_ERR_NUMBER_TOO_LARGE:
141             return "number is too large";
142         case SSH_ERR_SIGN_ALG_UNSUPPORTED:
143             return "signature algorithm not supported";
144         case SSH_ERR_FEATURE_UNSUPPORTED:
145             return "requested feature not supported";
146         case SSH_ERR_DEVICE_NOT_FOUND:
147             return "device not found";
148         default:
149             return "unknown error";
150     }
151 }
```