NAME

SSL_check_chain - check certificate chain suitability

SYNOPSIS

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

int SSL_check_chain(SSL *s, X509 *x, EVP_PKEY *pk, STACK_OF(X509) *chain);
```

DESCRIPTION

SSL_check_chain() checks whether certificate **x**, private key **pk** and certificate chain **chain** is suitable for use with the current session **s**.

RETURN VALUES

SSL_check_chain() returns a bitmap of flags indicating the validity of the chain.

CERT_PKEY_VALID: the chain can be used with the current session. If this flag is **not** set then the certificate will never be used even if the application tries to set it because it is inconsistent with the peer preferences.

CERT_PKEY_SIGN: the EE key can be used for signing.

CERT_PKEY_EE_SIGNATURE: the signature algorithm of the EE certificate is acceptable.

CERT_PKEY_CA_SIGNATURE: the signature algorithms of all CA certificates are acceptable.

CERT_PKEY_EE_PARAM: the parameters of the end entity certificate are acceptable (e.g. it is a supported curve).

CERT_PKEY_CA_PARAM: the parameters of all CA certificates are acceptable.

CERT_PKEY_EXPLICIT_SIGN: the end entity certificate algorithm can be used explicitly for signing (i.e. it is mentioned in the signature algorithms extension).

CERT_PKEY_ISSUER_NAME: the issuer name is acceptable. This is only meaningful for client authentication.

CERT_PKEY_CERT_TYPE: the certificate type is acceptable. Only meaningful for client authentication.

CERT_PKEY_SUITEB: chain is suitable for Suite B use.

NOTES

SSL_check_chain() must be called in servers after a client hello message or in clients after a certificate request message. It will typically be called in the certificate callback.

An application wishing to support multiple certificate chains may call this function on each chain in turn: starting with the one it considers the most secure. It could then use the chain of the first set which returns suitable flags.

As a minimum the flag **CERT_PKEY_VALID** must be set for a chain to be usable. An application supporting multiple chains with different CA signature algorithms may also wish to check **CERT_PKEY_CA_SIGNATURE** too. If no chain is suitable a server should fall back to the most secure chain which sets **CERT_PKEY_VALID**.

The validity of a chain is determined by checking if it matches a supported signature algorithm, supported curves and in the case of client authentication certificate types and issuer names.

Since the supported signature algorithms extension is only used in TLS 1.2, TLS 1.3 and DTLS 1.2 the results for earlier versions of TLS and DTLS may not be very useful. Applications may wish to specify a different "legacy" chain for earlier versions of TLS or DTLS.

SEE ALSO

SSL_CTX_set_cert_cb(3), **ssl**(7)

COPYRIGHT