NAME

X509_check_host, X509_check_email, X509_check_ip, X509_check_ip_asc - X.509 certificate matching

SYNOPSIS

```
#include <openssl/x509.h>

int X509_check_host(X509 *, const char *name, size_t namelen,
unsigned int flags, char **peername);
int X509_check_email(X509 *, const char *address, size_t addresslen,
unsigned int flags);
int X509_check_ip(X509 *, const unsigned char *address, size_t addresslen,
unsigned int flags);
int X509_check_ip_asc(X509 *, const char *address, unsigned int flags);
```

DESCRIPTION

The certificate matching functions are used to check whether a certificate matches a given host name, email address, or IP address. The validity of the certificate and its trust level has to be checked by other means.

X509_check_host() checks if the certificate Subject Alternative Name (SAN) or Subject CommonName (CN) matches the specified host name, which must be encoded in the preferred name syntax described in section 3.5 of RFC 1034. By default, wildcards are supported and they match only in the left-most label; but they may match part of that label with an explicit prefix or suffix. For example, by default, the host **name** ‘‘www.example.com’’ would match a certificate with a SAN or CN value of ‘‘*.example.com’’, ‘‘w*.example.com’’ or ‘‘*w.example.com’’.

Per section 6.4.2 of RFC 6125, **name** values representing international domain names must be given in A-label form. The **namelen** argument must be the number of characters in the name string or zero in which case the length is calculated with strlen(**name**). When **name** starts with a dot (e.g ‘‘.example.com’’), it will be matched by a certificate valid for any sub-domain of **name**, (see also **X509_CHECK_FLAG_SINGLE_LABEL_SUBDOMAINS** below).

When the certificate is matched, and **peername** is not NULL, a pointer to a copy of the matching SAN or CN from the peer certificate is stored at the address passed in **peername**. The application is responsible for freeing the peername via *OPENSSL_free()* when it is no longer needed.

X509_check_email() checks if the certificate matches the specified email **address**. Only the mailbox syntax of RFC 822 is supported, comments are not allowed, and no attempt is made to normalize quoted characters. The **addresslen** argument must be the number of characters in the address string or zero in which case the length is calculated with strlen(**address**).

X509_check_ip() checks if the certificate matches a specified IPv4 or IPv6 address. The **address** array is in binary format, in network byte order. The length is either 4 (IPv4) or 16 (IPv6). Only explicitly marked addresses in the certificates are considered; IP addresses stored in DNS names and Common Names are ignored.

X509_check_ip_asc() is similar, except that the NUL-terminated string **address** is first converted to the internal representation.

The **flags** argument is usually 0. It can be the bitwise OR of the flags:

X509_CHECK_FLAG_ALWAYS_CHECK_SUBJECT,
X509_CHECK_FLAG_NEVER_CHECK_SUBJECT,
X509_CHECK_FLAG_NO_WILDCARDS,
X509_CHECK_FLAG_NO_PARTIAL_WILDCARDS,
X509_CHECK_FLAG_MULTI_LABEL_WILDCARDS.
X509_CHECK_FLAG_SINGLE_LABEL_SUBDOMAINS.

The **X509_CHECK_FLAG_ALWAYS_CHECK_SUBJECT** flag causes the function to consider the subject DN even if the certificate contains at least one subject alternative name of the right type (DNS name or email address as appropriate); the default is to ignore the subject DN when at least one corresponding subject alternative names is present.

The **X509_CHECK_FLAG_NEVER_CHECK_SUBJECT** flag causes the function to never consider the subject DN even if the certificate contains no subject alternative names of the right type (DNS name or email address as appropriate); the default is to use the subject DN when no corresponding subject alternative names are present. If both **X509_CHECK_FLAG_ALWAYS_CHECK_SUBJECT** and **X509_CHECK_FLAG_NEVER_CHECK_SUBJECT** are specified, the latter takes precedence and the subject DN is not checked for matching names.

If set, **X509_CHECK_FLAG_NO_WILDCARDS** disables wildcard expansion; this only applies to **X509_check_host**.

If set, **X509_CHECK_FLAG_NO_PARTIAL_WILDCARDS** suppresses support for ''*'' as wildcard pattern in labels that have a prefix or suffix, such as: ''www*'' or ''*www''; this only applies to **X509_check_host**.

If set, **X509_CHECK_FLAG_MULTI_LABEL_WILDCARDS** allows a ''*'' that constitutes the complete label of a DNS name (e.g. ''*.example.com'') to match more than one label in **name**; this flag only applies to **X509_check_host**.

If set, **X509_CHECK_FLAG_SINGLE_LABEL_SUBDOMAINS** restricts **name** values which start with ''.'', that would otherwise match any sub-domain in the peer certificate, to only match direct child sub-domains. Thus, for instance, with this flag set a**name** of ' '.example.com'' would match a peer certificate with a DNS name of ''www.example.com'', but would not match a peer certificate with a DNS name of ''www.sub.example.com''; this flag only applies to **X509_check_host**.

RETURN VALUES

The functions return 1 for a successful match, 0 for a failed match and -1 for an internal error: typically a memory allocation failure or an ASN.1 decoding error.

All functions can also return -2 if the input is malformed. For example, *X509_check_host()* returns -2 if the provided **name** contains embedded NULs.

NOTES

Applications are encouraged to use *X509_VERIFY_PARAM_set1_host()* rather than explicitly calling *X509_check_host(3)*. Host name checks may be out of scope with the *DANE-EE(3)* certificate usage, and the internal checks will be suppressed as appropriate when DANE support is enabled.

SEE ALSO

SSL_get_verify_result(3), *X509_VERIFY_PARAM_set1_host(3)*, *X509_VERIFY_PARAM_add1_host(3)*, *X509_VERIFY_PARAM_set1_email(3)*, *X509_VERIFY_PARAM_set1_ip(3)*, *X509_VERIFY_PARAM_set1_ipasc(3)*

HISTORY

These functions were added in OpenSSL 1.0.2.

COPYRIGHT