NAME

CURLOPT_SSL_VERIFYHOST − verify the certificate's name against host

SYNOPSIS

```
#include <curl/curl.h>

CURLcode curl_easy_setopt(CURL *handle, CURLOPT_SSL_VERIFYHOST, long verify);
```

DESCRIPTION

Pass a long set to 2L to make libcurl verify the host in the server's TLS certificate.

When negotiating a TLS connection, the server sends a certificate indicating its identity.

When *CURLOPT_SSL_VERIFYHOST(3)* is set to 1 or 2, the server certificate must indicate that it was made for the hostname or address curl connects to, or the connection fails. The certificate has to have the same name as is used in the URL you operate against.

curl considers the server the intended one when the Common Name field or a Subject Alternate Name field in the certificate matches the hostname in the URL to which you told curl to connect.

When the *verify* value is 0, the connection succeeds regardless of the names in the certificate. Use that ability with caution.

This option controls checking the server's certificate's claimed identity. The separate *CURLOPT_SSL_VERIFYPEER(3)* options enables/disables verification that the certificate is signed by a trusted Certificate Authority.

WARNING: disabling verification of the certificate allows bad guys to man−in−the−middle the communication without you knowing it. Disabling verification makes the communication insecure. Having encryption on a transfer is not enough as you cannot be sure that you are communicating with the correct end−point.

When libcurl uses secure protocols it trusts responses and allows for example HSTS and Alt−Svc information to be stored and used subsequently. Disabling certificate verification can make libcurl trust and use such information from malicious servers.

MATCHING

A certificate can have the name as a wildcard. The only asterisk (*) must then be the left−most character and it must be followed by a period. The wildcard must further contain more than one period as it cannot be set for a top−level domain.

A certificate can be set for a numerical IP address (IPv4 or IPv6), but then it should be a Subject Alternate Name kind and its type should correctly identify the field as an IP address.

DEFAULT

2

PROTOCOLS

This functionality affects all TLS based protocols: HTTPS, FTPS, IMAPS, POP3S, SMTPS etc.

All TLS backends support this option.

EXAMPLE

```
int main(void)
{
 CURL *curl = curl_easy_init();
 if(curl) {
  CURLcode result;
```

```
  curl_easy_setopt(curl, CURLOPT_URL, "https://example.com");

  /* Set the default value: strict name check please */
  curl_easy_setopt(curl, CURLOPT_SSL_VERIFYHOST, 2L);

  result = curl_easy_perform(curl);
  curl_easy_cleanup(curl);
 }
}
```

AVAILABILITY

Added in curl 7.8.1

HISTORY

In 7.28.0 and earlier: the value 1 was treated as a debug option of some sorts, not supported anymore due to frequently leading to programmer mistakes.

From 7.28.1 to 7.65.3: setting it to 1 made *curl_easy_setopt(3)* return an error and leaving the flag untouched.

From 7.66.0: libcurl treats 1 and 2 to this option the same.

RETURN VALUE

curl_easy_setopt(3) returns a CURLcode indicating success or error.

CURLE_OK (0) means everything was OK, non−zero means an error occurred, see *libcurl−errors(3)*.

SEE ALSO

CURLOPT_CAINFO(3), **CURLOPT_DOH_SSL_VERIFYHOST**(3), **CURLOPT_PINNEDPUBLICKEY**(3), **CURLOPT_PROXY_SSL_VERIFYHOST**(3), **CURLOPT_SSL_VERIFYPEER**(3)