

NAME

CURLOPT_SSH_KEYFUNCTION – callback for known host matching logic

SYNOPSIS

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

```
enum curl_khstat {
    CURLKHSTAT_FINE_ADD_TO_FILE,
    CURLKHSTAT_FINE,
    CURLKHSTAT_REJECT, /* reject the connection, return an error */
    CURLKHSTAT_DEFER, /* do not accept it, but we cannot answer right
                        now. Causes a CURLE_PEER_FAILED_VERIFICATION error but
                        the connection is left intact */
    CURLKHSTAT_FINE_REPLACE
};

enum curl_khmatch {
    CURLKHMATCH_OK, /* match */
    CURLKHMATCH_MISMATCH, /* host found, key mismatch */
    CURLKHMATCH_MISSING, /* no matching host/key found */
};

struct curl_khkey {
    const char *key; /* points to a null-terminated string encoded with
                     base64 if len is zero, otherwise to the "raw"
                     data */
    size_t len;
    enum curl_khkeytype keytype;
};

int ssh_keycallback(CURL *easy,
    const struct curl_khkey *knownkey,
    const struct curl_khkey *foundkey,
    enum curl_khmatch match,
    void *clientp);

CURLcode curl_easy_setopt(CURL *handle, CURLOPT_SSH_KEYFUNCTION,
    ssh_keycallback);
```

DESCRIPTION

Pass a pointer to your callback function, which should match the prototype shown above.

It gets called when the known_host matching has been done, to allow the application to act and decide for libcurl how to proceed. The callback is only called if *CURLOPT_SSH_KNOWNHOSTS(3)* is also set.

This callback function gets passed the curl handle, the key from the known_hosts file *knownkey*, the key from the remote site *foundkey*, info from libcurl on the matching status and a custom pointer (set with *CURLOPT_SSH_KEYDATA(3)*). It must return one of the following return codes to tell libcurl how to act:

CURLKHSTAT_FINE_REPLACE

The new host+key is accepted and libcurl replaces the old host+key into the known_hosts file before continuing with the connection. This also adds the new host+key combo to the known_host pool kept in memory if it was not already present there. The adding of data to the file is done by completely replacing the file with a new copy, so the permissions of the file must allow this. (Added in 7.73.0)

CURLKHSTAT_FINE_ADD_TO_FILE

The host+key is accepted and libcurl appends it to the known_hosts file before continuing with the connection. This also adds the host+key combo to the known_host pool kept in memory if it was not already present there. The adding of data to the file is done by completely replacing the file with a new copy, so the permissions of the file must allow this.

CURLKHSTAT_FINE

The host+key is accepted libcurl continues with the connection. This also adds the host+key combo to the known_host pool kept in memory if it was not already present there.

CURLKHSTAT_REJECT

The host+key is rejected. libcurl denies the connection to continue and it is closed.

CURLKHSTAT_DEFER

The host+key is rejected, but the SSH connection is asked to be kept alive. This feature could be used when the app wants to return and act on the host+key situation and then retry without needing the overhead of setting it up from scratch again.

DEFAULT

NULL

PROTOCOLS

This functionality affects scp and sftp

EXAMPLE

```
struct mine {
    void *custom;
};

static int keycb(CURL *easy,
                const struct curl_khkey *knownkey,
                const struct curl_khkey *foundkey,
                enum curl_khmatch match,
                void *clientp)
{
    /* 'clientp' points to the callback_data struct */
    /* investigate the situation and return the correct value */
    return CURLKHSTAT_FINE_ADD_TO_FILE;
}

int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        struct mine callback_data;
        curl_easy_setopt(curl, CURLOPT_URL, "sftp://example.com/thisfile.txt");
        curl_easy_setopt(curl, CURLOPT_SSH_KEYFUNCTION, keycb);
        curl_easy_setopt(curl, CURLOPT_SSH_KEYDATA, &callback_data);
        curl_easy_setopt(curl, CURLOPT_SSH_KNOWNHOSTS, "/home/user/known_hosts");

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

AVAILABILITY

Added in curl 7.19.6

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_SSH_KEYDATA(3), CURLOPT_SSH_KNOWNHOSTS(3)