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 can't answer right
                        now so this causes a CURLE_DEFER error but
                        otherwise the connection will be left intact
                        etc */
};

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 zero-terminated string encoded with
                      base64 if len is zero, otherwise to the "raw"
                      data */
  size_t len;
  enum curl_khtype keytype;
};

int ssh_keycallback(CURL *easy,
                    const struct curl_khkey *knownkey,
                    const struct curl_khkey *foundkey,
                    enum curl_khmatch,
                    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 will only be 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_ADD_TO_FILE
: The host+key is accepted and libcurl will append it to the known_hosts file before continuing with the connection. This will also add the host+key combo to the known_host pool kept in memory if it wasn't 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 will continue with the connection. This will also add the host+key combo to the known_host pool kept in memory if it wasn't already present there.

CURLKHSTAT_REJECT
: The host+key is rejected. libcurl will deny the connection to continue and it will be 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 somehow return back and act on the host+key situation and then retry without needing the overhead of setting it up from scratch again.

DEFAULT

NULL

PROTOCOLS

SFTP and SCP

EXAMPLE

TODO

AVAILABILITY

Added in 7.19.6

RETURN VALUE

Returns CURLE_OK if the option is supported, and CURLE_UNKNOWN_OPTION if not.

SEE ALSO

CURLOPT_SSH_KEYDATA(3), **CURLOPT_SSH_KNOWNHOSTS**(3),