LIBSSH2

Your humble presenter

- Daniel Stenberg
- Consultant by trade at Haxx
- Open Source since 15 years
- Contributed to 20 projects
- 15K public commits (counted by ohloh)
- cURL, Rockbox, c-ares, Subversion, ...

Agenda

- Background
- Why a lib is needed
- The project
- Basic design
- License
- App examples
- Competitors
- Future
- Contributors

LIBSSH2

Questions

Please do interrupt and ask if you have questions!

A little story

Once upon a time...

1997

Wouldn't it be nice with a tool that grabs HTTP?

1998

Wouldn't it be nice if it did FTP too?

1999

And Gopher!

1999

Wouldn't it be nice if it did HTTPS too?

2000

You get the picture. Time moved on.
It was now also a lib

2006

Supporting HTTP, HTTPS, FTP, FTPS, TELNET, LDAP, LDAPS, DICT, TFTP and FILE. Using several underlying libraries.

Wouldn't it be nice if it did SCP and SFTP too?

How to select a SSH lib

- OpenSSH – no lib
- Putty – no lib
- Libssh (license, blocking, thread-safety...)
- Libssh2 – there you go

Why libssh2

- Because neither OpenSSH nor Putty “did it right”
- Embeddability, command line tools aren't enough
- Portability – not limited to POSIX (C89)
- License – as free as possible
- Only 20K LOC

Libssh2 Background

- Created by Sara Golemon, Dec 2004
- Client and server
- SSH2 only (no v1)
- For PHP
- Sara set the API, made it run
- Daniel entered Nov 2006
- For libcurl (non-blocking!)
- Client-side only (for simplicity)

The project

- Small
- Low key
- Mailing list driven
- Meritocracy
- Start-up feeling
- Responsive to new ideas and features
- Not many users (apps)
- No umbrella org
- git
- Individual copyrights

Basic design

- Sessions
- Channels
- SCP
- X11
- SFTP
- Publickey
- Knownhost

- (non-)blocking
- pass in socket
- external poll()
- Crypto-layer abstracted

Blocking API SCP client

```
/* create a socket */
sockfd = socket();

/* connect to server */
connect(sockfd, address);

/* get a libssh2 session handle (blocking behavior is default) */
session = libssh2_session_init();

/* pass in the socket to the SSH2 session */
libssh2_session_startup(session, sockfd);

/* password authentication (known host handling skipped) */
libssh2_userauth_password(session, username, password));

/* request a file with SCP */
channel = libssh2_scp_recv(session, “/home/daniel/textfile”, &fileinfo);

while (!done)
        libssh2_channel_read(channel, mem, amount);

/* free the SCP channel */
libssh2_channel_free(channel);

/* disconnect session */
libssh2_session_disconnect(session, "shutdown”);

/* free the session */
libssh2_session_free(session);

/* close the socket */
close(socket);
```

Blocking API SFTP client

```
/* create a socket */
sockfd = socket();

/* connect to server */
connect(sockfd, address);

/* get a libssh2 session handle (blocking behavior is default) */
session = libssh2_session_init();

/* pass in the socket to the SSH2 session */
libssh2_session_startup(session, sockfd);

/* password authentication (known host handling skipped) */
libssh2_userauth_password(session, username, password))

/* request a file with SCP */
channel = libssh2_scp_recv(session, “/home/daniel/textfile”, &

while (!done)
          libssh2_channel_read(channel, mem, amount);

/* free the SCP channel */
libssh2_channel_free(channel);
```

```
/* create the SFTP session */
sftp_session = libssh2_sftp_init(session);

sftp_handle = libssh2_sftp_open(sftp_session, “/home/daniel/file”,
                                              LIBSSH2_FXF_READ, 0);

while (!done)
         libssh2_sftp_read(sftp_handle, mem, amount);

/* close the SFTP handle */
libssh2_sftp_close(sftp_handle);

/* close the SFTP session */
libssh2_sftp_shutdown(sftp_session);
```

```
/* disconnect session */
libssh2_session_disconnect(session, "shutdown”);

/* free the session */
libssh2_session_free(session);

/* close the socket */
close(socket);
```

Or use libcurl!

Libcurl supports SCP and SFTP URLs

http://curl.haxx.se/

License

- Modified BSD
- Free for virtually every use
- Sara G picked it
- Suitable for re-use by other libs/platforms

Competitor(s)

- Libssh
  - Slower
  - License
  - non-blocking
  - Name-space polluting
  - Unstable API
- Others? Mostly non-C (Java, .NET ...)
- Commercial?

Contributors

- Sara G
- Me
- Simon J
- A few other happy campers
- No particular company backing
- Company-funded features

Future

- Remain client-side
- Make more stable
- Improve speed
- Use less mallocs
- Use less memory (adjust windows etc)
- Missing features?
- Scratch itches

Summary

- Open source library for SSH2
- Written in C
- The best one available
- Small team

Join in!

libssh2 needs you!