



FOSDEM '09

FREE AND OPEN SOURCE SOFTWARE DEVELOPERS' EUROPEAN MEETING

БЕЗПЛАТНО ОТКРИТО ПОСРЕДСТВО РАЗРАБОТЧИКОВ, ЕВРОПЕЙСКО СЪБИТИЕ

Introduction to GnuTLS



<http://www.gnutls.org/>

Simon Josefsson
simon@josefsson.org
<http://josefsson.org/>

What is GnuTLS?

- Implementation of Transport Layer Security
 - Provides strong encryption and authentication
 - The S in HTTPS
 - Application protocol standardized by the IETF
- Similar to OpenSSL but does not implement non-TLS stuff like S/MIME, low-level crypto, etc

What is GnuTLS?

- Part of the GNU project
- Copyrights assigned to the FSF
- Core library is LGPLv2.1+, tools under GPLv3+

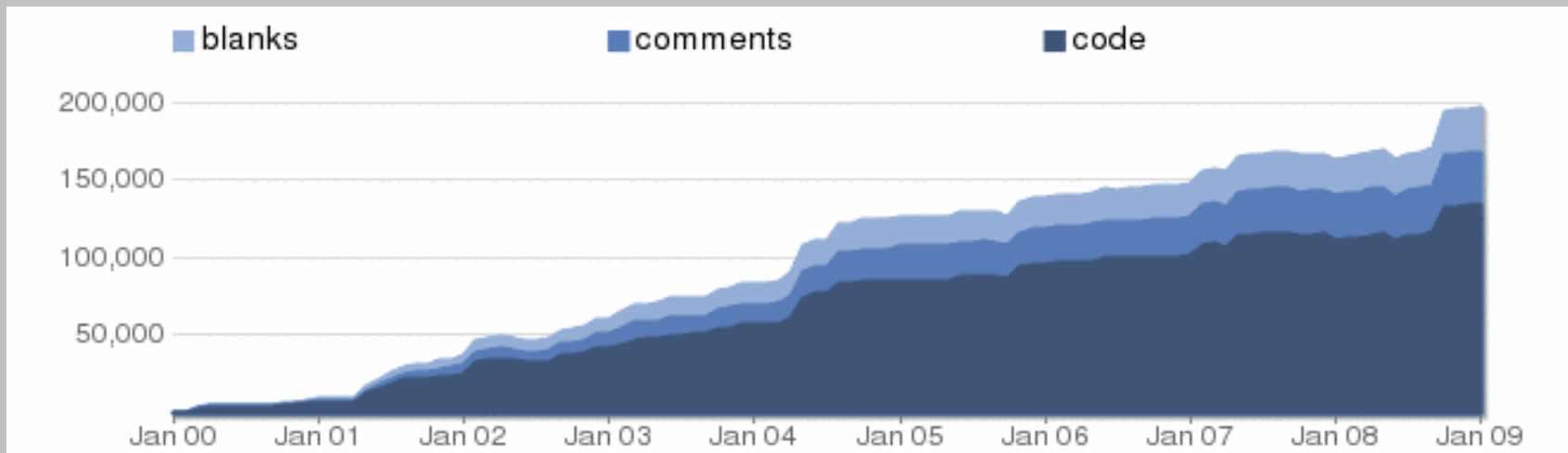


What is GnuTLS?

- Required dependencies:
 - Libgcrypt (replaceable), Libtasn1
- Optional dependencies
 - Libz, Liblzo (compression)
- Official API bindings for Guile and C++
- Windows installer (DLL, tools, etc) available from <http://josefsson.org/gnutls4win/>

GnuTLS History

- ~8 year old project – started late 2000
- Originally written by Nikos Mavrogiannopoulos
- Currently maintained by Simon Josefsson
- Small dev team: ~15 significant contributors
- Version 1.0 in 2003 - version 2.0 in 2007



Well documented

The image displays two overlapping web browser windows. The left window, titled "gnutls", shows the reference page for `gnutls-gnutls.html`. It contains documentation for the `gnutls_handshake_get_last_in()` and `gnutls_record_send()` functions. The right window, titled "GNU TLS 2.7.4", shows the manual index page with a "Table of Contents" section listing various topics such as "1 Preface", "2 The Library", "3 Introduction to TLS", and "4 Authentication Methods".

gnutls

File Edit View Go Bookmarks Tools Tabs Help

Back Forward Up Stop Reload Home

http://www.gnu.org/software/gnutls/reference/gnutls-gnutls.html

GNU T

Top | Description

This function is only useful to check where the last performed handshake failed. If the previous ha
Check `gnutls_handshake_description_t` in `gnutls.h` for the available handshake descriptions.

`session` : is a `gnutls_session_t` structure.
`Returns` : the last handshake message type sent, a `gnutls_handshake_description_t`

gnutls_handshake_get_last_in ()

```
gnutls_handshake_description_t gnutls_handshake_get_last_in  
(gnutls_session_t ses
```

This function is only useful to check where the last performed handshake failed. If the previous ha
Check `gnutls_handshake_description_t` in `gnutls.h` for the available handshake descriptions.

`session` : is a `gnutls_session_t` structure.
`Returns` : the last handshake message type received, a `gnutls_handshake_descripti`

gnutls_record_send ()

```
ssize_t gnutls_record_send (gnutls_session_t ses  
const void *data,  
size_t sizeofdata);
```

This function has the similar semantics with `send()`. The only difference is that it accepts a GNUT
Note that if the send buffer is full, `send()` will block this function. See the `send()` documentation
with a call to `send()` with a `MSG_DONTWAIT` flag if blocking is a problem.

If the `EINTR` is returned by the internal push function (the default is `send()`) then `GNUTLS_E_INT`
function again, with the same parameters; alternatively you could provide a NULL pointer for data

`session` : is a `gnutls_session_t` structure.
`data` : contains the data to send
`sizeofdata` : is the length of the data
`Returns` : the number of bytes sent, or a negative error code. The number of bytes s
call depends on the negotiated maximum record size.

gnutls_record_rcv ()

```
ssize_t gnutls_record_rcv (gnutls_session_t ses
```

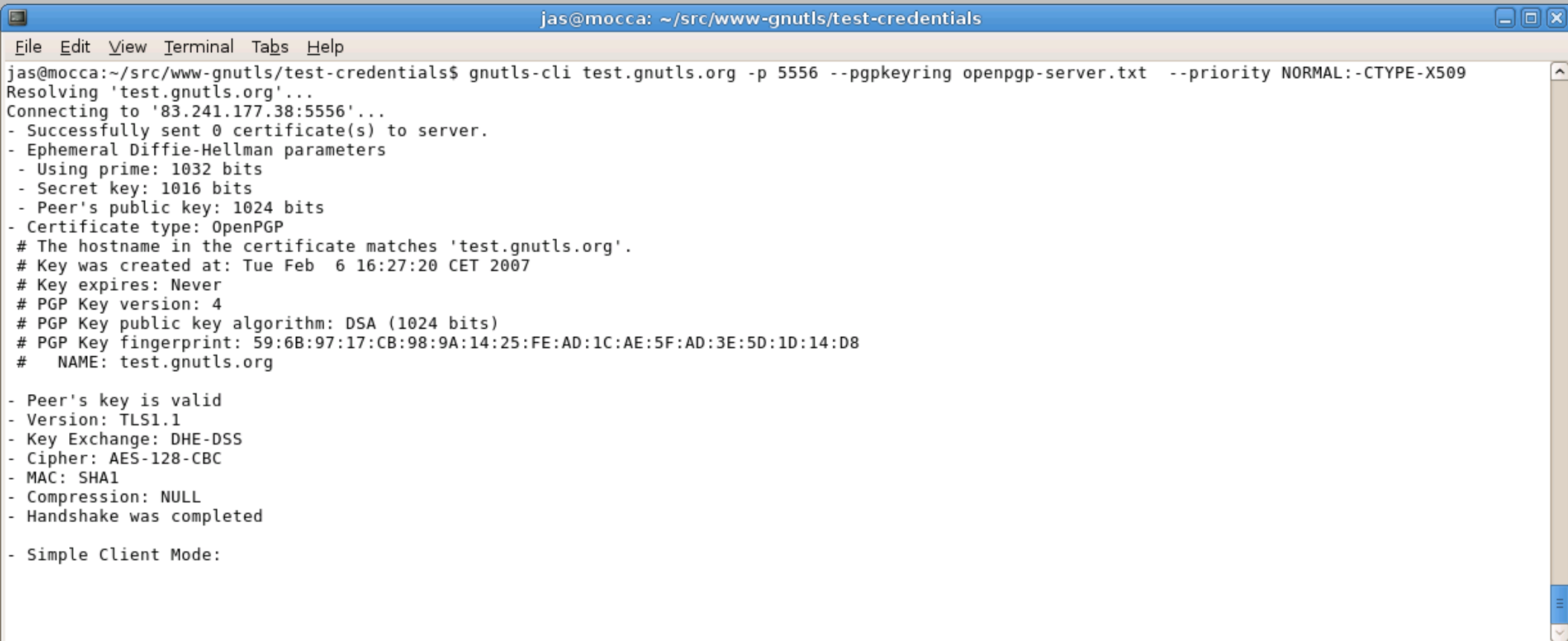
GNU TLS 2.7.4

Table of Contents

- [GNU TLS](#)
- [1 Preface](#)
 - [1.1 Getting Help](#)
 - [1.2 Commercial Support](#)
 - [1.3 Downloading and Installing](#)
 - [1.4 Bug Reports](#)
 - [1.5 Contributing](#)
- [2 The Library](#)
 - [2.1 General Idea](#)
 - [2.2 Error Handling](#)
 - [2.3 Memory Handling](#)
 - [2.4 Callback Functions](#)
- [3 Introduction to TLS](#)
 - [3.1 TLS Layers](#)
 - [3.2 The Transport Layer](#)
 - [3.3 The TLS Record Protocol](#)
 - [3.3.1 Encryption Algorithms Used in the Record Layer](#)
 - [3.3.2 Compression Algorithms Used in the Record Layer](#)
 - [3.3.3 Weaknesses and Countermeasures](#)
 - [3.4 The TLS Alert Protocol](#)
 - [3.5 The TLS Handshake Protocol](#)
 - [3.5.1 TLS Cipher Suites](#)
 - [3.5.2 Client Authentication](#)
 - [3.5.3 Resuming Sessions](#)
 - [3.5.4 Resuming Internals](#)
 - [3.6 TLS Extensions](#)
 - [3.6.1 Maximum Fragment Length Negotiation](#)
 - [3.6.2 Server Name Indication](#)
 - [3.7 Selecting Cryptographic Key Sizes](#)
 - [3.8 On SSL 2 and Older Protocols](#)
 - [3.9 On Record Padding](#)
- [4 Authentication Methods](#)
 - [4.1 Certificate Authentication](#)
 - [4.1.1 Authentication Using X.509 Certificates](#)
 - [4.1.2 Authentication Using OpenPGP Keys](#)

Features

- Supports authentication of server and user using OpenPGP



```
jas@mocca: ~/src/www-gnutls/test-credentials
File Edit View Terminal Tabs Help
jas@mocca:~/src/www-gnutls/test-credentials$ gnutls-cli test.gnutls.org -p 5556 --pgpkeyring openpgp-server.txt --priority NORMAL:-CTYPE-X509
Resolving 'test.gnutls.org'...
Connecting to '83.241.177.38:5556'...
- Successfully sent 0 certificate(s) to server.
- Ephemeral Diffie-Hellman parameters
  - Using prime: 1032 bits
  - Secret key: 1016 bits
  - Peer's public key: 1024 bits
- Certificate type: OpenPGP
# The hostname in the certificate matches 'test.gnutls.org'.
# Key was created at: Tue Feb  6 16:27:20 CET 2007
# Key expires: Never
# PGP Key version: 4
# PGP Key public key algorithm: DSA (1024 bits)
# PGP Key fingerprint: 59:6B:97:17:CB:98:9A:14:25:FE:AD:1C:AE:5F:AD:3E:5D:1D:14:D8
#   NAME: test.gnutls.org

- Peer's key is valid
- Version: TLS1.1
- Key Exchange: DHE-DSS
- Cipher: AES-128-CBC
- MAC: SHA1
- Compression: NULL
- Handshake was completed

- Simple Client Mode:
```

Features

- Strong password authentication via Secure Remote Password (SRP)
- Shared symmetric key authentication using Pre-Shared Key (PSK)
- Server Name extension
 - Used by Apache mod_gnutls
- X.509 tools to create private keys, self-signed certificates, certificate requests, etc

Software Patent Blues

- The network security area has many overlapping patents for some techniques
- RSA the historical example, but the patent expired (and a free implementation existed)
- Secure Remote Password (SRP) is patented but freely implementable
- TLS-AUTHZ extension support in GnuTLS removed!

The End

Thanks for listening!