Introduction to GnuTLS

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

http://www.gnutls.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

GNU TLS 2.7.4

Table of Contents

- GNU TLS
  - 1 Interface
    - 1.1 Setting help
    - 1.2 Commercial Support
    - 1.3 Development and Installing
    - 1.4 Bug Reports
    - 1.5 Contributions
  - 2 The Library
    - 2.1 General Ideas
    - 2.2 Error Handling
    - 2.3 Memory Handling
    - 2.4 Call-Back Functions
  - 3 Introduction to TLS
    - 3.3.15 Layer
    - 3.2 The Transport Layer
    - 3.3 The TLS Record Protocol
      - 3.3.2 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 Client Protocol
      - 3.3.1 TLS Client Sockets
      - 3.3.2 Client Authentication
      - 3.3.3 Resuming Sessions
      - 3.3.4 Resuming Sessions
    - 3.5 TLS Extensions
      - 3.5.1 Maximum Fragment Length Negotiation
      - 3.5.2 Server Name Indication
      - 3.5.3 Selecting Cryptographic Key Sizes
      - 3.5.4 Use of SSL 2 and Older Protocols
      - 3.5.5 Use of Record Padding
  - 4 Authentication Methods
    - 4.1 Certificate Authentication
      - 4.1.1 Authentication Using X.509 Certificates
      - 4.1.2 Authentication Using Issued X.PGP Keys
Features

- Supports authentication of server and user using OpenPGP

```
jas@mocca:~$ gnutls-cli test.gnutls.org -p 5556 --pgpkeyring openpgp-server.txt --priority NORMAL:-CTYPE-X509
Resolving 'test.gnutls.org'...
Connecting to '83.241.177.39:5556'...
- Successfully sent 0 certificate(s) to server.
- Ephemeral Diffie-Hellman parameters
  - Using prime: 1024 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)
# 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!