SSL History

- Developed by Netscape in 1994
- Now de facto Internet community standard
- The IETF Transport Layer Security (TLS),
  - TLS 1.0 = SSL 3.1
- X.509v3 cert.format
- The OpenSSL Project
- JSSE – Java Secure Sockets Extensions reference implementation provided by Sun.

What kind of security?

- The easiest problem: Secure channel
- End-to-end
- Uses digital certificates
- Multiple cryptographic functions
- Authentication
  - server
  - client
  - data
- Confidentiality
  - data
- Separation of services

https://www.openssl.org/doc.html

Security-awareness: Hide most of the details from the user
PLEASE REMEMBER THAT EXPORT/IMPORT AND/OR USE OF STRONG CRYPTOGRAPHY SOFTWARE, PROVIDING CRYPTOGRAPHY HOOKS OR EVEN JUST COMMUNICATING TECHNICAL DETAILS ABOUT CRYPTOGRAPHY SOFTWARE IS ILLEGAL IN SOME PARTS OF THE WORLD. SO, WHEN YOU IMPORT THIS PACKAGE TO YOUR COUNTRY, RE-DISTRIBUTE IT FROM THERE OR EVEN JUST EMAIL TECHNICAL SUGGESTIONS OR EVEN SOURCE PATCHES TO THE AUTHOR OR OTHER PEOPLE YOU ARE STRONGLY ADVISED TO PAY CLOSE ATTENTION TO ANY EXPORT/IMPORT AND/OR USE LAWS WHICH APPLY TO YOU. THE AUTHORS OF OPENSSL ARE NOT LIABLE FOR ANY VIOLATIONS YOU MAKE HERE. SO BE CAREFUL, IT IS YOUR RESPONSIBILITY.

Internet Protocol Stack With SSL

<table>
<thead>
<tr>
<th>Layer</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Layer</td>
<td>HTTP, NNTP, Telnet, LDAP</td>
</tr>
<tr>
<td>Secure Sockets Layer</td>
<td>SSL</td>
</tr>
<tr>
<td>Transport Layer</td>
<td>TCP, X.25, ... (not UDP)</td>
</tr>
<tr>
<td>Internet Layer</td>
<td>IP</td>
</tr>
</tbody>
</table>

Unix ports used

<table>
<thead>
<tr>
<th>Appl. protocol</th>
<th>Port</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>https</td>
<td>443/tcp</td>
<td>Web service</td>
</tr>
<tr>
<td>ssntp</td>
<td>465/tcp</td>
<td>Mailsending</td>
</tr>
<tr>
<td>news</td>
<td>563/tcp</td>
<td>Unetnews</td>
</tr>
<tr>
<td>ssl-ldap</td>
<td>636/tcp</td>
<td>Directory</td>
</tr>
<tr>
<td>sftp</td>
<td>995/tcp</td>
<td>Mailretrieving</td>
</tr>
</tbody>
</table>

SSL Protocol Stack
Cryptographic functions

- **DES.** Data Encryption Standard, an encryption algorithm used by the U.S. Government.
- **Triple-DES.** DES applied three times.
- **DSA.** Digital Signature Algorithm, part of the digital authentication standard used by the U.S. Government.
- **KEA.** Key Exchange Algorithm, an algorithm used for key exchange by the U.S. Government.
- **MD5.** Message Digest algorithm developed by Rivest.
- **RC2 and RC4.** Rivest encryption ciphers developed for RSA Data Security.
- **RSA.** A public-key algorithm for both encryption and authentication. Developed by Rivest, Shamir, and Adleman.
- **RSA key exchange.** A key-exchange algorithm for SSL based on the RSA algorithm.
- **SHA-1.** Secure Hash Algorithm, a hash function used by the U.S. Government.

Two important SSL concepts

- **Connection**
  - Transient peer-to-peer transport service
  - Multiple conn. associated with one session
- **Session**
  - Cryptographic association client-server
  - Created by Handshake protocol

SSL Handshake

```
1. Client Hello
2. Server Hello
3. Certificate (optional)
4. Server Certificate (optional)
5. Change Cipher Spec
6. Finished
7. Certificates (optional)
8. Certificates (optional)
9. Change Cipher Spec
10. Finished
11. Encrypted data
12. Change Cipher Spec
13. Encrypted data
```

Client

- Establish security capabilities including compression methods and shared secret
- Send certificate
- Send key exchange
- Notify application

Server

- Send certificate
- Send key exchange
- Notify application
- Validate certificate
- Send Change Cipher Spec
- Notify application
- Send encrypted data

Client

- Notify application
- Send Change Cipher Spec
- Notify application
- Send encrypted data

Server

- Notify application
- Send Change Cipher Spec
- Notify application
- Send encrypted data
```
Example Certificate Details

Certificate: Data: Version: 3 (0x2) Serial Number: 204 (0xcc) Signature Algorithm: sha1WithRSAEncryption Issuer: C=NO, ST=Sor-Trondelag, L=Trondheim, O=NTNU, OU=Institutt for Telematikk, CN=SIE5040 - Root CA

Not Before: Feb 6 12:03:49 2003 GMT Not After: Oct 14 12:03:49 2003 GMT Subject: C=NO, ST=Sor-Trondelag, L=Trondheim, O=NTNU, OU=Institutt for Telematikk, CN=Stig Frode Mjølsnes/Email=sfm@item.ntnu.no


X.509 Certificate Structure

How a client authenticates a server certificate

How a server authenticates a client certificate
SSL Record Protocol

• Confidentiality
  – By a shared secret key for symmetric cipher

• Message Integrity
  – By a shared secret key for MAC function

SSL Record Protocol (cont)

Two more subprotocols

• Change Cipher Spec
  – Simple message
  – One byte of ones
  – Pending state ➔ Current state

• Alert
  – Simple message
  – Two bytes
    • Level of severity
    • Alert code
  – Examples
    • Unexpected_message
    • Bad_record_MAC
    • Bad_certificate
    • ……
Java Secure Sockets Extensions Architecture

JSSE packages

- `com.sun.net.ssl` -- Provides classes related to creating and configuring secure socket factories.
- `javax.net` -- Provides optional classes for networking applications.
- `javax.net.ssl` -- Provides the classes for the secure socket optional package.
- `javax.security.cert` -- Provides optional classes for public key certificates.

Java Software Resources

- JSSE Reference Guide
- Intro. to JSSE at Java One
- JSSE download for JDK 1.3 and prior (Export restrictions may still apply for the US version)
- Openssl (Use for generating server certificates)

Package `javax.net.ssl`

- provides SSL Socket level API
- Classes
  - `SSLServerSocket`
  - `SSLServerSocketFactory`
  - `SSLSocket`
  - `SSLSocketFactory`