Implementing new TLS versions.

A summary of protocol update problems.

Yngve N. Pettersen
Opera Software ASA

Implementing new TLS versions.

A summary of protocol update problems.

The theory:

- SSL/TLS is a forward compatible protocol.
- SSL/TLS implementations (in particular servers) will be able to work with implementations of newer protocol versions.

The reality

Many SSL/TLS server implementations are broken with respect to forward compatibility:

- SSL v3 servers that will not talk to TLS 1.0 clients.
- TLS 1.0 servers that will not accept TLS Extensions.
- TLS 1.0 servers that will not talk to TLS 1.1 clients.
- TLS 1.0 servers using the wrong protocol field to negotiate version.

SSL v3 -> TLS 1.0

Problems with the SSL v3 to TLS 1.0 transition

- Servers that would not accept the TLS 1.0 version number.
- Servers that used the wrong version number during key exchange.

Consequence:

- Clients had to consider any SSL v3 negotiation failure as an indication that the server could not talk to TLS 1.0 clients and disable TLS 1.0 for that server.

TLS 1.0 and TLS Extensions

Despite the fact that support for an extension of the TLS Client Hello is required by RFC 2246 many servers does not accept TLS Extensions.

The reaction can be error codes or shutting down the connection without an error.

In some cases Extensions were accepted with a TLS 1.1 hello, but not a TLS 1.0 hello, and sometimes only with TLS 1.0 and not TLS 1.1.

It is unknown whether or not this is caused by server implementation errors, or firewall rules. However, at least one banking site that originally accepted TLS Extensions did not accept them 4 weeks later.

TLS 1.0 -> TLS 1.1

As with the transition to TLS 1.0, many servers does not accept Client Hello's from TLS 1.1 clients.

Various reactions:

- No response from server.
- Connection closed immediately, with or without error.
- Some refused to accept TLS 1.1 with TLS Extensions.
- Failure due to incorrect handling of the RSA Premaster secret (wrong version used).
- Falls back to SSL v3, even if TLS 1.0 is supported.

As with extensions it is unknown whether or not this is caused by server implementation errors, or firewall rules.

Incorrect version negotiation

Both RFC 2246 and TLS 1.1 may be unclear about which version number to use in the first record protocol record to a server with unknown version support.

The correct interpretation may be that the client should use the lowest version the server is known to support, not the same as in the Client Hello (highest client supported version).

This have revealed a new class of problems: Some servers use the Protocol Record version number of the Client Hello record, not the Client Hello record's version field to negotiate the supported version.

Results

- Clients have to perform a version rollback in order to connect to a server. This causes the SSL v3+ version rollback protection in the RSA suites to be worthless.
- Clients have to perform functionality testing of a server before establishing a secure connection.

The alternative would be to fail to connect, which is not acceptable for a general client.

What does this mean for TLS 1.2?

Based on the experience so far, non-compliant servers are likely to cause as many problems for the transition to TLS 1.2 as they have for previous versions.

Given that TLS 1.2 is intended to solve expected problems with digest algorithms, version rollbacks to solve the negotiation problem may introduce security problems.

Recommendation: The TLS WG should consider how to handle non-compliant implementations.