Analysis of the SSL 3.0 protocol

David Wagner
University of California, Berkeley
daw@cs.berkeley.edu

Bruce Schneier
Counterpane Systems
schneier@counterpane.com

Note: A corrected version of the paper may be found at

`http://www.cs.berkeley.edu/~daw/ssl3.0.ps`

Thanks to Martín Abadi, Tim Dierks, and others for pointing out flaws they discovered in the published paper.

Introduction

- SSL: practical application-layer communications security
  - Widely used to protect web traffic
- Record layer
  - Symmetric crypto for confidentiality, message authentication, and replay protection
- Handshake layer
  - Public-key key-exchange protocol
  - Layered on top of the record layer

Limitations of SSL 2.0

- 40-bit MAC keys in export-weakened modes
- Weak MAC construction
- Unauthenticated security-critical context
  - MAC applied to data and padding bytes but not to the cleartext padding length field
- Ciphersuite rollback attacks [Simon]

- SSL 3.0: fix these flaws, support more algorithms

SSL 3.0 Record Layer

- A significant improvement over SSL 2.0 record layer
  - 128-bit MAC keys, even in export-weakened modes
  - Independent keying for encryption and authentication
  - HMAC
- Defends against many known attacks
  - Replay attacks
  - Connection truncation
  - Cut-and-paste, short block attacks [Bellovin]

Record layer (cont.)

Traffic-analysis attack on SSL-protected web traffic [Yee]

- Observation: Ciphertext length reveals plaintext length

- Client requests URL with `http` GET
  - URL is SSL-encrypted for confidentiality

- Eavesdropper can typically recover interesting information:
  - Length of URL
  - IP address of web server
  - Length of web page returned by server

- Attacker catalogues web pages available on server

- Look for matches; answer is probably unique

  - If not, other tricks are possible

- Attacker recovers URL and requested web page

- Fix: random padding to obscure plaintext lengths

Handshake Layer

`KeyExchangeAlgorithm`-spoofing

- In some modes, server provides a (signed) ephemeral public key, e.g.

$$\begin{array}{ll} \{n, e\}_{K_s} & \text{(RSA)} \\ \{p, g, g^y\}_{K_s} & \text{(DH)} \end{array}$$

- Note: field that selects between RSA & DH is *not* signed!

- The attack:

  – Ciphersuite "rollback": server uses DH, client RSA

  – Server sends $\{p, g, g^y\}_{K_s}$

  – Client interprets p as RSA modulus, g as exponent, sends $k^g \bmod p$

- Attacker recovers k by taking g-th roots mod p (easy), spoofs rest of key exchange

- Fix: sign field that selects RSA or DH

Handshake Layer

Limitations of the analysis

- Not thoroughly analyzed:
  - Nonces properly signed? [Abadi]
  - Session resumption + version rollback
- No formal analysis performed
- Room for future work

Conclusions

- SSL 3.0 a significant improvement over SSL 2.0
- Some minor weaknesses found:
  - Traffic analysis of message lengths
  - `KeyExchangeAlgorithm`-spoofing
  - Dropping the **change cipher spec**
- On the whole, SSL 3.0 is a valuable tool for communications security