Podstawy Secure Sockets Layer

Michał Grzejszczak

20 stycznia 2003

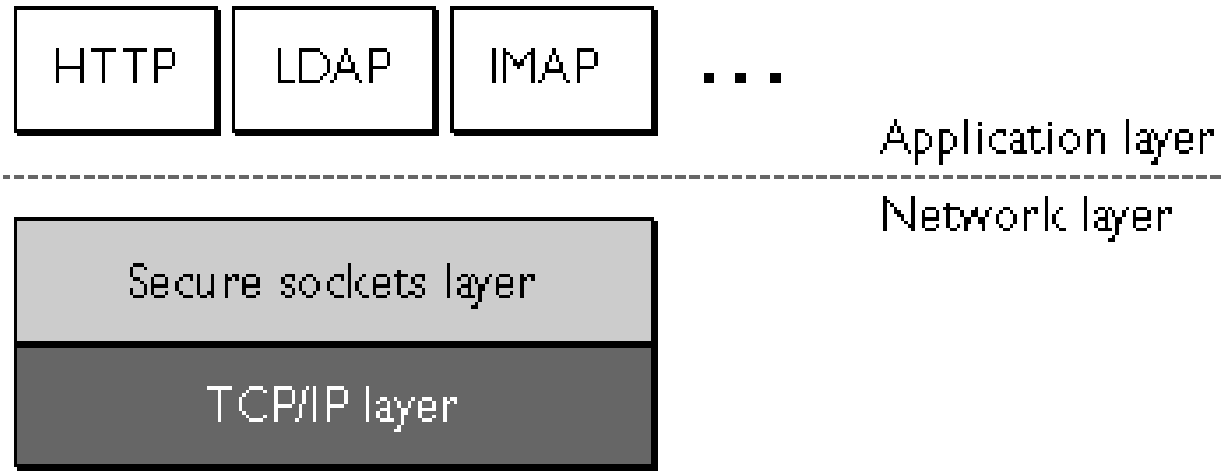
Spis treści

1 Wstęp

Protokół SSL (Secure Sockets Layer) został opracowany przez firmę Netscape. Przyjęto go jako uniwersalny standard bezpiecznych połączeń szyfrowanych klient - serwer.

2 Protokół SSL

Protokół TCP/IP (Transmission Control Protocol/Internet Protocol) zarządza przesyłaniem danych przez internet. Inne protokoły np. HTTP (HyperText Transfer Protocol), LDAP (Lightwieght Directory Access Protocol) lub IMAP (Internet Messaging Access Protocol) działają "ponad"TCP/IP, tzn. wykorzystują go do wyświetlania stron WWW lub przesyłania poczty.



SSL działa pomiędzy TCP/IP a innymi protokołami. Używa protokołu TCP/IP w imieniu innych protokołów otwierając możliwość autentykacji klient - serwer i ustanowienia szyfrowanego połączenia.
Te możliwości rozwiązują podstawowe problemy bezpiecznej komunikacji za pomocą TCP/IP:

- Autentykacja serwera
  Pozwala klientowi potwierdzić tożsamość serwera. Klient może przy wykorzystaniu standardowych technik klucza publicznego sprawdzić prawdziwość i aktualność certyfikatu i identyfikatora serwera jak również to czy certyfikat został wydany przez CA (Certificate Authority) występujące na liście zaufanych CA klienta.

- Autentykacja klienta
  To samo co wyżej tylko w drugą stronę - serwer sprawdza tożsamość klienta.

- Szyfrowane połączenie SSL
  Wymagane jest szyfrowanie danych przesyłanych pomiędzy klientem a serwerem. Dodatkowo SSL zapewnia ochronę przed „tamperingiem”, czyli zmianą danych w trakcie ich

transportu ze źródła do celu (czy to przez błąd lub umyślne działanie) [zapewnia integralność].

SSL zawiera dwa podprotokoły:

- SSL record protocol
- SSL handshake protocol.

Record protocol definiuje format używany do transmisji danych. Handshake protocol używa poprzedniego w celu wymiany serii komunikatów między klientem a serwerem zaraz po nawiązaniu połączenia. Ma to na celu:

- Uwierzytelnić serwer klientowi.
- Wybrać szyfr, który obaj obsługują.
- Ewentualnie uwierzytelnić klienta serwerowi.
- Wygenerować klucze dla sesji.
- Ustanowić szyfrowane połączenie.

3 Szyfry używane przez SSL

Każdy serwer czy klient SSL może mieć różny zestaw szyfrów lub algorytmów kryptograficznych w zależności od wersji SSL, polisy bezpieczeństwa systemu lub ograniczeń na eksport technologii SSL nałożonych w danym kraju. Z tego powodu protokół handshake odpowiedzialny jest również za wybór techniki szyfrującej do autentykacji, transmisji certyfikatów i generowania kluczy.

Algorytmy odpowiedzialne za tworzenie symetrycznych kluczy dla klienta i serwera to algorytmy typu key-exchange np. KEA (Key Exchange Algorithm) lub RSA key exchange. Najczęściej używanym jest RSA key exchange.

3.1 Lista szyfrów

Siła szyfrowania	Zalecane użycie	Szyfry
Najsilniejsza	Dla banków i instytucji przesyłających ściśle tajne dane (tylko US)	**Triple DES, autentykacja SHA-1**. Potrójne zastosowanie algorytmu DES daje 168-bitową siłę szyfrowania. Wolniejszy niż RC4. Obsługiwane przez SSL 3.0 i SSL 2.0
Silna	Dla większości instytucji rządowych i firm	**RC4 z szyfrowaniem 128-bitowym i autentykacją MD5.** Obsługiwany przez SSL 3.0 i SSL 2.0
		RC2 z szyfrowaniem 128-bitowym i autentykacją MD5. Obsługiwany przez SSL 2.0
		DES z szyfrowaniem 56-bitowym i autentykacją SHA-1. Obsługiwany przez SSL 2.0 i SSL 3.0 (SSL 2.0 używa MD5 zamiast SHA-1)
Średnia	US pozwala eksportować te i słabsze	**RC4 z szyfrowaniem 40-bitowym i autentykacją MD5.** Obsługiwany przez SSL 3.0 i SSL 2.0
		RC2 z szyfrowaniem 40-bitowym i autentykacją MD5. Obsługiwany przez SSL 2.0 i SSL 3.0
Słaba		**Bez szyfrowania, tylko autentykacja MD5.** Zapewnia jedynie autentykację i ochronę przed tamperingiem. Jest używana gdy klient i serwer nie mają żadnego wspólnego szyfru.

4 Jak działa handshake?

Po ustanowieniu połączenia dokonuje się autentykacja serwera przy użyciu techniki klucza publicznego. Wtedy tworzone są klucze symetryczne do (szybszego) kryptażu dekryptażu i ochrony przed tamperingiem.

Przebieg handshake'a:

1. Klient wysyła do serwera numer jego wersji SSL, ustawienia szyfrów, losowo wygenerowane dane itp.

2. Serwer odsyła klientowi numer swojej wersji SSL, ustawienia szyfrów, losowo wygenerowane dane oraz swój certyfikat (ewentualnie dane potrzebne klientowi do obustronnej autentykacji).

3. Klient przeprowadza autentykację serwera. Jeśli się to nie powiedzie, wysyłamy jest komunikat do użytkownika, że autentykacja się nie udała.

4. Klient generuje (z pomocą serwera, to zależy od szyfru) **„premaster secret”**. Szyfruje go kluczem publicznym serwera i wysyła go do serwera. (jeśli klient ma się autentykować to wysyła też swój certyfikat)

5. Serwer autentykuje klienta o ile tego wymaga. Jeśli się uda odszyfrowuje **„premaster secret”**, po czym wykonuje na nim kilka operacji (klient robi po swojej stronie to samo) w rezultacie czego powstaje **„master secret”**.

6. Klient i serwer generują przy użyciu **„master secret”** klucze symetryczne dla sesji.

7. Klient wysyła do serwera komunikat informujący o tym, że kolejne komunikaty będą już zaszyfrowane, po czym wysyła komunikat zaszyfrowany oznaczający zakończenie przez niego procesu handshake. To samo robi serwer.

8. Połączenie SSL jest gotowe do użycia.

5 Literatura

Informacje o SSL można znaleźć na stronie Netscape:
http://developer.netscape.com/docs/manuals/security/sslin/contents.htm.