UnB - UNIVERSIDADE DE BRASÍLIA
INSTITUTO DE CIÊNCIAS EXATAS

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Monografia sobre SSL para o Curso de Extensão
Segurança em Redes de Computadores
(5ª Turma)

por
Luiz Aristides Rios Largura

Brasília/DF, 13 de Agosto de 2000

O *Secure Socket Layer* (SSL, atualmente na versão 3) é um protocolo de comunicação que implementa um duto seguro para comunicação de aplicações na Internet, de forma transparente e independente da plataforma.

Foi desenvolvido pela Netscape Communications em sua versão inicial em julho de 1994. Em abril de 1995 foi lançada a referência para implementação da versão 2 (sendo distribuído junto os Browsers Netscape e Internet Explorer e os servidores web mais comuns - Apache, NCSA httpd, IIS, Netscape Server etc), apoiado pela Verisign e Sun, transformando-se em um padrão em *e-commerce*, tendo a sua especificação submetida ao grupo de trabalho W3C (especificação disponível em um "draft" do IETF com o nome de TLS - Transaction Layer Security – rfc2246, disponível em http://www.cis.ohio-state.edu/htbin/rfc/rfc2246.html ou http://www.ietf.org/html.charters/tls-charter.html).

Em novembro de 1995 foi lançada a versão 3 do SSL, tendo como melhorias a diminuição no número de redadas de negociação, a escolha das cifras e compressão por parte do servidor, um suporte mais completo para a troca de chaves de algoritmos de cifragem, a possibiliadde de renegociação das cifres em uso e a separação das chaves de autenticação e encriptação. Embora as diferenças entre o TLSv1 e o SSL v3 não sejam grandes, são suficientes para que eles não possam interoperar diretamente. Caso seja necessário, o TLS v1 pode emular o SSL v3.

Sua proposta é permitir a autenticação de servidores, encriptação de dados, integridade de mensagens e, como opção, a autenticação do cliente, operando nas comunicações entre aplicativos de forma interoperável.

Visa garantir as seguintes objetivos: Segurança criptográfica para o estabelecimento de uma ligação segura entre duas máquinas/aplicativos, assegurando a privacidade na conexão, com a utilização de algoritmos simétricos (como o DES ou RC4) que negociam uma chave secreta na primeira fase do handshaking (usando chaves públicas – assimétricas); Autenticação do Servidor (e, opcionalmente do Cliente) por meio de algoritmos assimétricos como o RSA ou o DSS; Confiabilidade na conexão, conseguida com o uso de Códigos de Autenticação de Mensagens (MAC).

O SSL também permite a montagem de uma *framework* onde outras chaves públicas e métodos de encriptação podem ser utilizados, evitando a necessidade de implementação de toda uma pilha de protocolos (com os riscos da introdução de fraquezas).

Como uma vantagem adicional, a questão do desempenho foi levada em consideração no projeto, para reduzir o número de conexões e minimizar o tráfego na rede pode ser usado opcionalmente um esquema de cache em memória durante o estabelecimento da sessão, com a finalidade de reduzir o número de conexões e reduzir a atividade no acesso à rede.

O SSL atua entre as camadas transporte (TCP) e aplicação, sendo independente do protocolo de alto nível podendo rodar sob HTTP, Telnet, FTP, SMTP e outras, de forma transparente (para saber mais sobre http sobre TLS, ver

http://www.ietf.org/rfc/rfc2818.txt).

Ele implementa duas novas camadas, sobre o TCP/IP, conforme o esquema e as descrições abaixo:

***Pilha de Camadas (Layers)* do TCP/IP com SSL**	
	Camada das Aplicações (http, ftp...)
SSL	Change Cipher, Alert Protocol, Handshake Protocol
SSL	Camada SSL Record
	Camada TCP
	Camada IP

- SSL Handshake Protocol: Faz a autenticação entre cliente e servidor, cuidando da inicialização da comunicação, permitindo a negociação do algoritmo de encriptação e as chaves criptográficas iniciais. Utiliza as chaves assimétricas para fazer a negociação inicial, abrindo um canal seguro para o envio da chave simétrica de sessão, criada de forma aleatória. Opera sobre o Record Layer. Todas as mensagens da negociação utilizam o MAC e funções de hash (como SHA, MD5 e outras) para aumentar a segurança do processo inicial. A ordem das mensagens é absoluta e seus conteúdos são manuseados pela "Record Layer". Segue abaixo uma descrição simplificada desde protocolo:

1. O Cliente envia uma Client Hello Message ao servidor;
2. O servidor responde com uma Server Hello Message;
3. Servidor envia seu Certificate;
4. Servidor envia Server Key Exchange Message;
5. Servidor solicita Certificate do cliente;
6. Servidor envia Server Hello Done Message;
7. Cliente manda Certificate Message ou No Certificate Alert;
8. Cliente manda Client Key Exchange Message;
9. Cliente manda Change Cipher Spec Message;
10. Cliente manda Finished Message;
11. Servidor manda Change Cipher Spec Message;
12. Servidor manda Finished Message;
13. Fim de Handshake. Inicio do protocolo de aplicação.

As mensagens Client Hello e Server Hello, estabelecem os seguintes atributos: Versão de Protocolo, Identificação de Sessão. Cipher Suite e Método de Compressão. Adicionalmente, dois valores randômicos são gerados e intercambiados entre cliente e servidor.

Ao estabelecer a conexão, o Handshake Protocol estabeleceu o identificador de sessão, o conjunto criptográfico (*cypher suite*) a ser adotado e o método de compressão a ser utilizado. O conjunto criptográfico negociado define três algoritmos:

1. Um algoritmo para troca de chaves;
2. Um algoritmo para cifragem de dados, e
3. Um algoritmo para inserção de redundância nas mensagens.

- SSL Change Cipher SPEC Protocol: Sinaliza as transições nas estratégias de cifragem. Constitui-se de uma única mensagem que pode ser transmitida tanto pelo cliente como pelo servidor para notificar que os próximos blocos utilizarão chaves de encriptação recém negociadas;

- SSL Alert Protocol: Acompanha os erros na Record Layer, fazendo troca de mensagens para sinalizar problemas com a seqüência de mensagens, erros de certificação ou encriptação.

- "SSL Record Layer Protocol": Encapsula as camadas de nível mais alto (quando conjugado com o HTTP, implementa o HTTPS), provendo os serviços de fragmentação (transforma blocos de dados em registros SSLPlaintext de, pelo menos, 2^{24} bytes), compressão, (transforma os registros SSLPlaintext em registros SSLCompressed, utilizando os algoritmos negociados no handshake), autenticação de mensagem, acréscimo do MAC e número sequenceial (antes da encriptação) e encriptação (as funções definidas no Handshake são definidas na mensagem SSLCipherSpec e são utilizadas para transformar o SSLCompressed em SSLCiphertext).

A comunicação é iniciada pelo estabelecimento de uma sessão, caracterizada pelo Estado da Sessão e o Estado da Conexão.

O Estado de Sessão é constituído pelos seguintes elementos:

- Session Identifier: Um seqüência arbitrária de bytes escolhida pelo servidor para identificar a sessão.
- Peer certificate: Certificado do peer (opcionalmente pode ser nulo).
- Compressiom Metod: Algoritmo utilizado na compressão.
- Cipher Spec: Specifica o algorimo usado na encriptação (null, DES, etc) e um algoritmo MAC (MD5 ou SHA).
- Master Secret: Uma chave secreta de 48 bytes trocada entre cliente e servidor.
- Is Resumable: Flag que indica se a sessão pode ser utilizada em outras conexões.

O Estado de Conexão é constituído pelos seguintes elementos:

- Server and Client Random: Seqüência de bytes aleatórios escolhidos pelo servidor e cliente a cada conexão.
- MAC Secret: Segredo usado nas operações MAC na escrita de dados
- Write Key: Chave de cifragem usada para encriptação e decriptação pelo cliente e servidor.
- Inicialization Vetors: Utilizados no algoritmo de encriptação
- Sequence Numbers: Utilizados no algoritmo de encriptação

No SSL versão 3, estão disponíveis os seguintes algoritmos criptográficos:

- Algoritmos para troca de chaves de sessão, durante o handshake:
NULL,RSA,Diffie-Hellman RSA,Diffie-Hellman DSS,DHE_DSS,
DHE_RSA,DH_anonymous, Fortezza/DMS

- Algoritmos para definição de chave de encriptação:
NULL, RC2, RC4, IDEA, DES, 3DES, Fortezza

- Algoritmos que implementam a função de Hash para definição do MAC:
NULL , SHA , MD5

- Tipos de Certificados:
X.509 v1,X.509 v2,X.509 v3

Algumas bibliotecas para implementação de SSL:

· SSLref : Implementação da Netscape. Distribuído em código fonte ANSIis C para compilação nas plataformas Windows 95/98/NT e Solaris. Ë gratuita para uso não comercial, entretanto apresenta sérias restrições do governo dos EUA para distribuição em outros países.

· SSLplus: Implementação comercial do SSL v3 da Consensus Development Corporation, exige a compra da licença do RSA BSAFE cryptographic toolkit para seu uso e também apresenta restrições para exportação fora dos EUA.

· SSLava: Um pacote para implementação de SSL na linguagem Java.

· SSLeay: Freeware, amplamente disponível e utilizado.

· SSL e-Sec: Implementação comercial do SSL2 desenvolvida pela empresa brasiliense e-Sec (antiga 4Safe).

Vantagens do uso do SSL:

- Um dos protocolos mais convenientes e utilizados para implementação de transações seguras.
- A implementação é relativamente simples, colocando-se o SSL no top[o da

pilha TCP/Ip e substituindo as chamadas TCP pelas chamadas SSL;

- Trabalha independente das aplicações utilizadas e, após o handshake inicial, comporta-se como um canal seguro que permite que se execute todas as funções que normalmente estão disponíveis no TCP/IP.
- Existem várias implementações gratuitas e comerciais, disponíveis para UNIX, Linux, Win 95/98/NT/2000 e outros;
- A maioria dos servidores e clientes (browsers) WEB já têm suporte nativo para ele, fazendo do SSL um padrão de fato;
- O IETF (Internet Engineering Task Force) está trabalhando na sua padronização formal, denominada TLS;
- Disponibiliza todas as primitivas necessárias para conexões seguras, a saber: Autenticação, troca de chaves de sessão com o uso de criptografia assimétrica prévia, encriptação com métodos simétricos, MAC e certificação.

Desvantagens do uso do SSL:

- Por ser implementado no topo do TCP/IP, o programador deve conhecer bem as características do sistema operacional e as especificações do TCP para manipular as chamadas do sistema.

Análise de segurança do SSL:

- Dependendo dos protocolos de segurança e algoritmos escolhidos durante o handshake, o SSL pode ser alvo de tentativas de: “Quebradores” (cracking) de cifras, ataque de texto vazio, tentativa de Replay e Espelhamento. Estes ataques são descritos em muitos livros (exemplo: http://www3.tsl.uu.se/~micke/ssl_links.html), bem como o SSL pode resistir a eles ou torná-los infrutíferos. Como o TLS v1 (ou SSL v3) possuem várias me;horias em relação ao SSL v2, os atacantes podem também tentar forçar uma conexão para “baixar” para o SSL v2. Este ataque aconteses se (e só se) ambos os lados estejam aptos a fazer o handshake na versão 2. O TLS aceita conexões em três modos: Servidor e Cliente autenticados; Só o Servidor autenticado e Nenhum dos dois autenticados. Quando apenas o Servidor é autenticado, o ataque por espelhamento é evitado, embora cliente completamente anônimos possam ser potencialmente perigosos. Quanto ao uso de funções de Hash, é recomendável o uso conjunto do MD5 e do SHA, para evitar que falhas em um dos dois algoritmos venha a comprometer todo o protocolo. Uma descrição mais detalhada das questões sobre a implementação e segurança do TLS v1 pode ser obtida em http://www.ietf.org/rfc/rfc2246.txt e o SSL v3 está em http://home.netscape.com/eng/ssl3/draft302.txt .

Referências:

APACHE – Manual, in
http://info.iqm.unicamp.br/manuais/apache/mod/mod_ssl/ssl_reference.html

Araújo, Gorgonio – Transações Seguras Via WEB, in
http://www.rnp.br/newsgen/9803/https.shtml

IETF - Network Working Group - RFC2246, The TSL Protocol Version 1.0, in
http://www.ietf.org/rfc/rfc2246.txt

IETF - Transport Layer Security (tls), in http://www.ietf.org/html.charters/tls-charter.html

Netscape - COMMERCE AND SECURITY, in
http://developer.netscape.com/misc/developer/conference/proceedings/cs2/index.html

Netscape - Transport Layer Security Working Group - INTERNET-DRAFT, The SSL Protocol Version 3.0, in http://home.netscape.com/eng/ssl3/draft302.txt

Pettersson, Micke & Sporrong, Ulf - SSL - Secure Sockets Layer ,TLS - Transport Layer Security, in http://www3.tsl.uu.se/~micke/ssl_links.html

Phaos Technology - SSL RESOURCE CENTER, in
http://www.phaos.com/sslresource.html

PUC-Rio – Curso de Redes, in

Rezende, Pedro A. D. – Material de Estudo do Curso de Criptografia e Segurança Computacional e Página de Tópicos em Segurança Computacional / Publicações / Links, in http://www.cic.unb.br/docentes/pedro/segdadtop.htm

RNP – Introdução ao SSL, in http://www.rnp.br/cgi-bin/newsgen-cgi/webglimpse/usr/local/etc/httpd/htdocs/newsgen/?query=SSL

Schiffman, Allan - The Secure Sockets Layer Protocol and Applications (Terisa Systems, Inc.), in http://www.terisa.com/presentations/ams/ssl/index.htm

Silva Filho, Joel G. (UnB – ENE) - ENIGMA - Cryptography Related Links, in
http://www.enigma.ene.unb.br/pub/crypto/crypto.htm

UFRGS – Curso de Redes, in http://penta.ufrgs.br/redes296/https/doc2.htm , doc3.htm , doc4.htm , doc5.htm

UNICAMP - Equipe de Segurança em Sistemas e Redes , in
http://www.security.unicamp.br/links.html