SSL

INE 5429
Segurança em Computação

Junho de 2013

SSL

SSL, do inglês "Secure Socket Layer" é o protocolo de segurança mais utilizado na internet para evitar intrusão e adulteração.

SSL

Seu objetivo é a realização de conexões seguras entre clientes e servidores.

SSL

O SSL utiliza criptografia assimétrica para gerar uma chave de sessão com a qual cliente e servidor cifram as mensagens posteriores trocadas entre eles utilizando criptografia simétrica.

Ou utilizam Diffie-Hellman para troca de chaves.

SSL

SSL

O cliente pode realizar a comunicação de forma segura graças ao certificado de segurança SSL

O ciframento via protocolo SSL impede a intrusão de um terceiro usuário na comunicação.

SSL

O protocolo SSL é composto por duas camadas e tem o seguinte funcionamento:

A primeira camada, o SSL Record Protocol, encapsula os protocolos de nível mais alto;

A segunda camada, o SSL Handshake Protocol, gerencia a negociação dos algoritmos de criptografia e a autenticação entre cliente e servidor.

SSL

O cliente ao fazer a conexão informa os pacotes de criptografia que dispõe.

O servidor responde com um identificador da sessão, seu certificado digital e informações sobre o pacote de criptografia escolhido.

SSL

O cliente deverá verificar o certificado digital do servidor. Então gerar uma pré-chave cifrada com a chave pública do servidor caso estejam usando o RSA para geração da chave mestre.

SSL

Esta é uma das chaves do protocolo SSL, ela será utilizada para geração da chave mestre que será utilizada em todas as mensagens após o handshake, com o cifrador simétrico escolhido.

SSL

Quando este processo estiver finalizado, os protocolos tomam controle no nível de aplicação.

SSL garante que as mensagens não sejam modificadas e nem lidas por "pessoas" não autorizadas, além de garantir que quem recebe a mensagem é quem deveria receber.

SSL

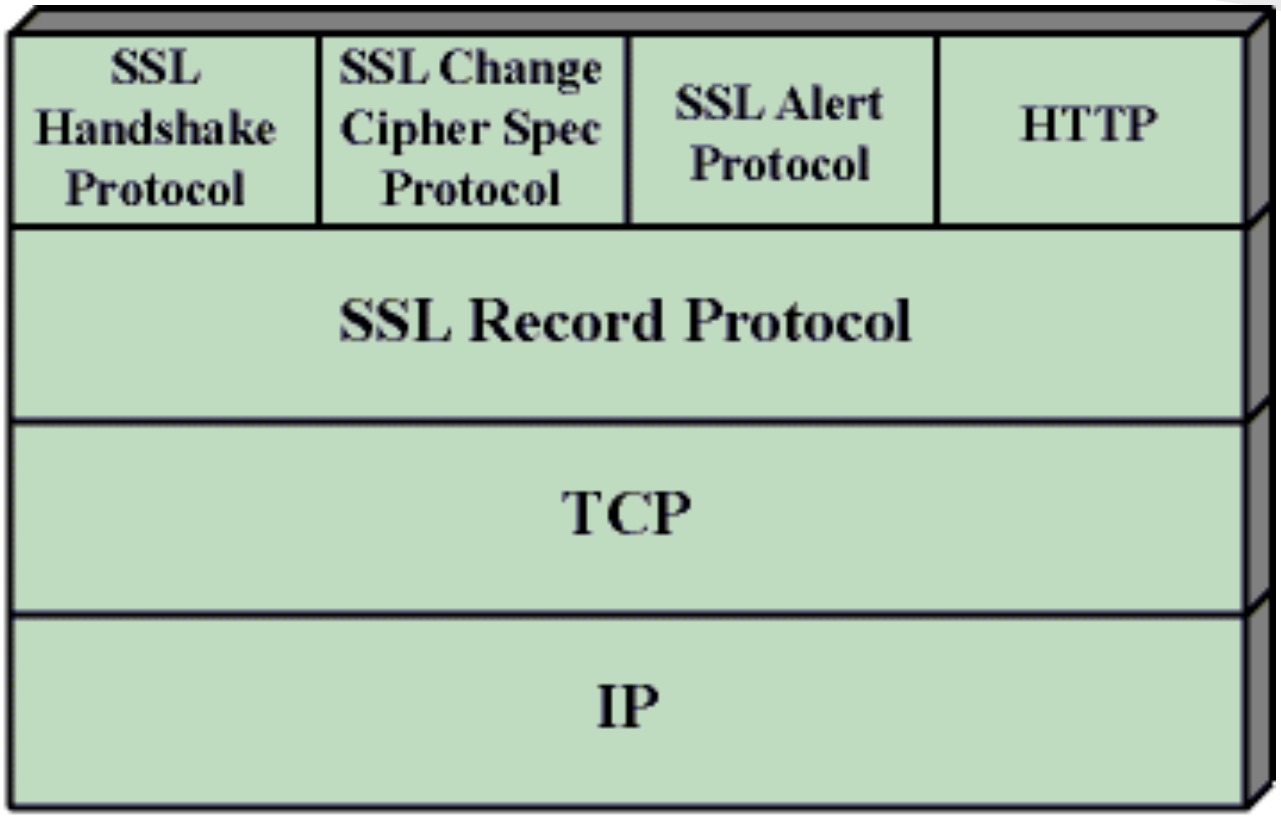


Figure 14.2 SSL Protocol Stack

Histórico do SSL

Secure Network Programming API

Primeiros esforços de pesquisa em razão de uma camada de transporte segura incluíram a Secure Network Programming API (SNP), que em 1993 explorou a abordagem de ter uma API para camada de transporte segura que muito se assemelhava ao Berkeley sockets, para facilitar a adaptação de aplicações de rede pré-existentes com medidas seguras.

Histórico do SSL

SSL 1.0, 2.0 e 3.0

O protocolo SSL foi originalmente desenvolvido pela Netscape. A versão 1.0 nunca foi lançada publicamente; a versão 2.0 foi lançada em fevereiro de 1995, porém, continha um certo número de falhas de segurança que levaram ao projeto da versão 3.0 do SSL.

Histórico do SSL

SSL 1.0, 2.0 e 3.0

A versão 3.0, lançada em 1996, foi completamente reprojetada por Paul Kocher, em trabalho conjunto com os engenheiros das Netscape, Phil Karlton e Alan Freier. As novas versões do SSL/TLS são baseadas no SSL 3.0.

Histórico do SSL

O rascunho de 1996 da versão 3.0 foi publicada pela IETF (Internet Engineering Task Force) como um documento histórico no RFC 6101 (Request For Comments).

Histórico do SSL

TLS 1.0

Foi primeiramente definico no RFC 2246 de janeiro de 1999 como uma evolução do SSL 3.0. Segundo o RFC, "as diferenças entre este protocolo e o SSL 3.0 não são dramáticas, mas são significantes o suficiente para impedir a interoperabilidade entre TLS 1.0 e SSL 3.0."

Histórico do SSL

TLS 1.0

TLS 1.0 não inclui um mecanismo no qual a implementação TLS poderia descer a conexão para o SSL 3.0, assim enfraquecendo a segurança.

Histórico do SSL

TLS 1.1

Foi definido no RFC 4346 em abril de 2006. É uma atualização do TLS 1.0.

As principais diferenças dessa versão incluem:

Adicionada proteção contra ataques ao modo de operação CBC (encadeamento de blocos cifrados).

Histórico do SSL

TLS 1.1

O vetor de inicialização (IV) implícito foi trocado por um IV explícito;

Mudanças no tratamento dos "padding errors".

Suporte ao registro de parâmetros IANA (Internet Assigned Numbers Authority).

Histórico do SSL

TLS 1.2

Foi definidio no RFC 5246 em agosto de 2008. É baseado na especificação anterior 1.1. As maiores diferenças incluem:

A combinação do MD5-SHA1 na função pseudoaleatória (PRF) deu lugar ao SHA-256, com uma opção de usar funções pseudoaleatórias especificadas pelo pacote de criptografia;

Histórico do SSL

TLS 1.2

A combinação do MD5-SHA1 no hash mensagem Finished deu lugara ao SHA-256, com a opção de usar algoritmos hash especificados pelo pacote de criptografia. Embora o tamanho do hash na mensagem Finished ainda seja truncado para 96 bits;

Histórico do SSL

TLS 1.2

A combinação do MD5-SHA1 no elemento assinado digitalmente deu lugar a um simples hash negociado durante o handshake, o padrão é o SHA1;

Melhoras na forma como o cliente e e servidor especificam qual algoritmos de hash e assinatura eles aceitarão;

Histórico do SSL

TLS 1.2

Expansão do suporte a cifradores de criptografia autenticada, usados principalmente para Galois/Counter Mode e modo CCM da criptografia com AES;

Adicionados a definição do TLS Extensions e pacotes de criptografia AES;

Histórico do SSL

TLS 1.2

O TLS 1.2 foi futuramente definado no RFC 6176 em março de 2011 removendo retrocompatibilidade com SSL de tal forma que sessões TLS nunca negociarão o uso de SSL 2.0.

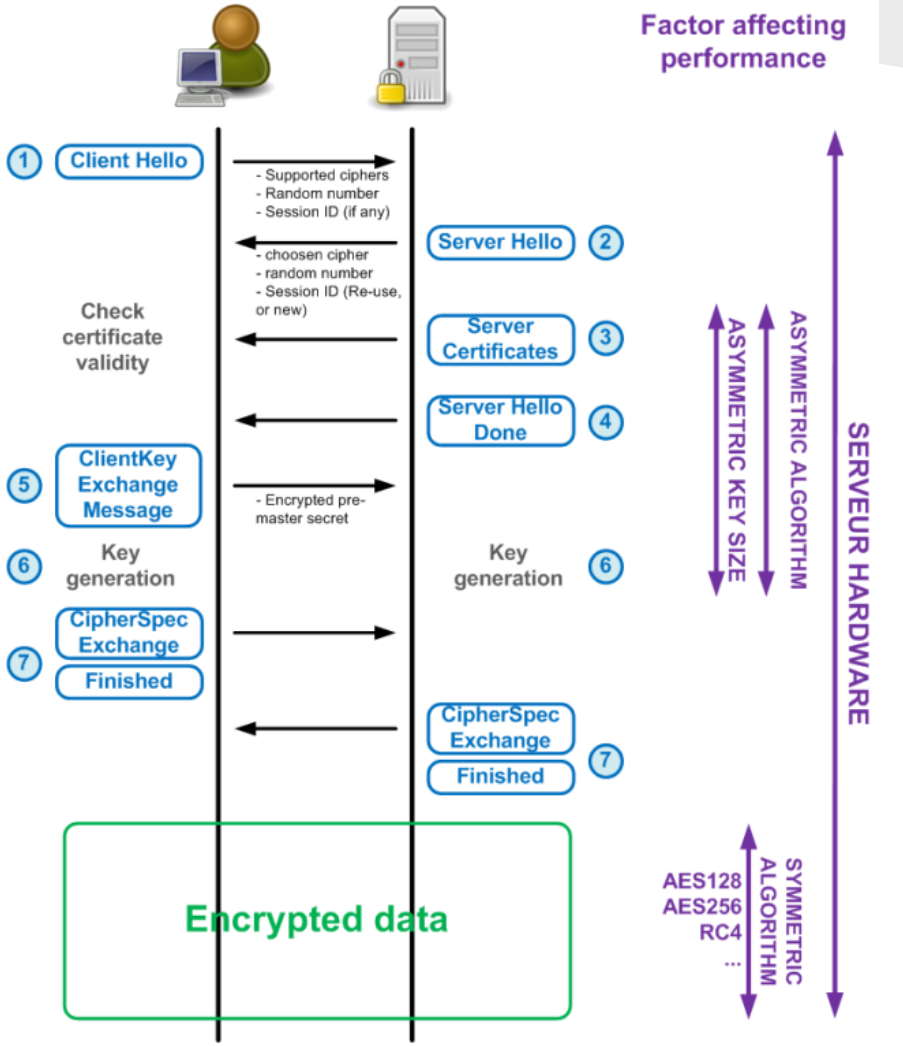
Funcionamento

O protocolo SSL/TLS se baseia na troca de registros que encapsulam o dado a ser compartilhado. Cada registro pode ser comprimido, completado e acrescentado com um código de autenticação de mensagem (MAC) ou cifrado, tudo dependendo do estado da conexão. Cada registro tem um campo content_type que especifica o protocolo de nível superior, um campo de tamanho e outro para a versão SSL/TLS.

Funcionamento

No início da conexão o SSL utiliza o protocolo Handshake (content_type = 22), trocando diferentes mensagens entre cliente e servidor.

Handshake


Factor affecting performance
1 Client Hello
- Supported ciphers
- Random number
- Session ID (if any)
Server Hello 2
- choosen cipher
- random number
- Session ID (Re-use, or new)
Check certificate validity
Server Certificates 3
Server Hello Done 4
5 ClientKey Exchange Message
- Encrypted pre-master secret
6 Key generation
Key generation 6
7 CipherSpec Exchange
Finished
CipherSpec Exchange
Finished 7
Encrypted data
ASYMMETRIC KEY SIZE
ASYMMETRIC ALGORITHM
SERVEUR HARDWARE
AES128
AES256
RC4
...
SYMMETRIC ALGORITHM

ClientHello

Versão do SSL do cliente
Estampa de tempo
Número randômico #1
ID de sessão (opcional)
Lista de pacotes de criptografia
Lista de algoritmos de compressão

ServerHello

Maior versão do SSL do servidor compatível com a versão do cliente

Estampa de tempo

Número randômico #2

ID de sessão

Pacote de criptografia escolhido

Algoritmos de compressão escolhido

Certificado (opcional)

Certificado digital do Servidor
Certificado da Autoridade Certificadora 1
Certificado da Autoridade Certificadora 2

Server Key Exchange (opcional)

Parâmetros para estabelecer a chave mestre:
RSA: chave pública para cifrar a pré-chave.
Diffie-Hellman: p, g
Assinatura (inclui os números aleatórios #1 e #2).

Requisição de Certificado (opcional)

Se a conexão for mutuamente certificada o servidor requisita o certificado do cliente e este envia o seu certificado.

ServerHelloDone

Ao receber essa mensagem o cliente verifica a autenticidade do servidor.

Client Key Exchange

RSA: Pré-chave cifrada com a chave pública do servidor.

Diffie-Hellman: gera y e envia g^y.

Change Cipher Spec

Pede para aplicar os novos parâmetros negociados.

Finished

Mensagem autenticada e cifrada, contendo o hash e o MAC de todas as mensagens do handshake.

Change Cipher Spec

Pede para aplicar os novos parâmetros negociados.

Finished

Mensagem autenticada e cifrada, contendo o hash e o MAC de todas as mensagens do handshake.

Aplicações

Normalmente encapsula HTTP, FTP, SMTP, NMTP e XMPP.

Primeiramente sobre o confiável TCP.

Pode ser usado sobre UDP e DCCP, conhecido como DTLS.

Aplicações

Proeminentemente utilizado para tráfego seguro na World Wide Web entre navegadores e servidores. Com HTTP se torna o HTTPS e tem como padrão a porta 443.

Notável adoção para comércio eletrônico e gestão de ativos.

Aplicações

Também é utilizado para criar o túnel utilizado em VPNs.

Vantagem sobre o tradicional IPSec quando utilizado em firewalls e NAT.

Desde o final da década de 90 tem sido bastante utilizado "fora do mundo dos navegadores" em aplicações cliente/servidor.

Aplicações

Também é método padrão para proteger sinalização de aplicações SIP. TLS pode ser usado para prover autenticação e criptografia da sinalização SIP associado com VoIP e outras aplicações baseadas em SIP.

SSL

Muito Obrigado!

Herivelton Coelho
Pedro Brentan
Renato da Motta Bustamante