Chapter 07: Transport-Level Security(bảo mật tầng Transport)

Ôn lại:

Secure Socket Layer(SSL) cung cấp security services giữa TCP và Ứng dụng mà sử dụng giao thức TCP.

The Internet standard version is called Transport Layer Service (TLS).

SSL/TLS cung cấp **confidentiality (bảo mật)** sử dụng **symmetric encryption(mã hóa đối xứng)** và **message integrity(toàn vẹn tin nhắn) sử dụng message authentication code(MAC).**

SSL/TLS gồm **protocol mechanisms** (kỹ thuật giao thức) to enable two TCP users to determine the security mechanisms and services they will use.(giữ bảo mật giữa TCP người dùng và dịch vụ họ dùng)

HTTPS tổng hợp giữa HTTP và SSL để hiện thực bảo mật giao tiếp giữa Web browser và Web server.

Secure Shell (SSH) cung cấp đăng nhập từ xa an toàn và tạo điều kiện thuận lợi của khách hàng / máy chủ an toàn khác.

**Web Security**

Bây giờ, Web được sử dụng trong kinh doanh, chính phủ, cá nhân. Internet & Web có lỗ hổng; gặp những đe dọa: toàn vẹn, bảo mật, từ chối dịch vụ, xac thực. Vì vậy, cần thêm vào các kỹ thuật bảo mật.

Phân loại tấn công:

Thụ động: nghe trộm (**eavesdropping**) trên đường truyền browser và server; tiếp cận các thông tin trên 1 trang web. Điều này phải được hạn chế.

Chủ động: **mạo danh (impersonating)** người dùng khác; thay đổi **(altering)** thông điệp trên đường truyền giữa client và server;thay đổi thông tin (**altering information**) trên 1 website.

Phân loại theo vị trí đe dọa: máy chủ Web, trình duyệt Web, và đường truyền mạng giữa các trình duyệt và máy chủ.

**Web Traffic Security Approaches(Phương pháp):**

Một cách là dùng **IP security(IPsec).** Lợi thế của IPsec : transparent (trong suốt) với người dùng cuối và ứng dụng; cung cấp giải pháp general-purpose. Hơn nữa, IPsec bao gồm một khả năng lọc để lưu lượng chỉ được lựa chọn cần phải chịu các chi phí IPsec processing.

Một giải pháp general-purpose khác là hiện thực bảo mật trên tầng TCP. Ví dụ quan trọng nhất của phương pháp này là **Secure Sockets Layer(SSL)** và theo tiêu chuẩn Internet Transport Layer Security (TLS). Ở cấp độ này, có hai lựa chọn thực hiện. Cho tổng quát đầy đủ, SSL (hoặc TLS) có thể được cung cấp như một phần của bộ giao thức cơ bản và do đó được minh bạch để các ứng dụng. Ngoài ra, SSL có thể được nhúng vào trong các gói cụ thể. Ví dụ, Netscape và Microsoft Explorer trình duyệt được trang bị với SSL, và hầu hết các máy chủ Web đã thực hiện các giao thức

**SSL:**

Netscape originated SSL.

Phiên bản 3 của giao thức được thiết kế với công chúng xem xét và đóng góp của ngành công nghiệp và đã được xuất bản như là một dự thảo tài liệu Internet.

Sau đó, khi một sự đồng thuận đã đạt được để nộp cho giao thức chuẩn Internet, các nhóm làm việc TLS đã được hình thành trong IETF để phát triển một **tiêu chuẩn(standard)** chung.

**SSL Architecture**

SSL được thiết kế để sử dụng TCP để cung cấp một dịch vụ đáng tin cậy end-to-end an toàn.

SSL không phải là một giao thức duy nhất mà là hai lớp giao thức.

Cung cấp hai dịch vụ:

• **Bảo mật**: The Handshake Protocol xác định một khóa bí mật chia sẻ đó là sử dụng để mã hóa thông thường trọng tải SSL.

• **Message Integrity**: The Handshake Protocol cũng định nghĩa một khóa bí mật được chia sẻ được sử dụng để tạo thành một mã xác thực thông điệp (MAC).

SSL chia ra SSL session và SSL connection:

Connection: **peer to peer relationships;**

Connection: client and server; được tạo bởi the Handshake protocol.

**SSL Record Protocol Services:**

**Confidentiality:** mã hóa với key tạo bởi Handshake Protocol; thông điệp được nén trước khi mã hóa. AES, IDEA, RC2-40, DES-40, DES, 3DES, Fortezza, RC4-40, RC4-128.

**message integrity:** MAC; similar to HMAC but with different padding.

**Change Cipher Spec Protocol:**

The Change Cipher Spec Protocol là một trong ba giao thức SSL-cụ thể mà sử dụng SSL Record Protocol, và nó là đơn giản nhất.

Mục đích duy nhất của tin nhắn này là để gây ra các trạng thái chờ đợi để được sao chép vào tình trạng hiện tại, trong đó cập nhật các bộ mật mã được sử dụng trên kết nối này.

**SSL Alert Protocol:**

Được sử dụng để chuyển tải các thông báo SSL-liên quan đến các thực thể ngang hàng.

Như với các ứng dụng khác có sử dụng SSL, các thông điệp cảnh báo được nén và mã hóa.

**SSL Handshake Protocol: ( bắt tay):**

most complex part of SSL is the Handshake Protocol.

Cho phép server và client:

• để **xác nhận lẫn nhau** và

• để **đàm phán một mật mã và thuật toán MAC** và

• **Đàm phán các khóa mật mã** được sử dụng để bảo vệ các dữ liệu được gửi trong một bản ghi SSL.

Bao gồm một loạt các tin nhắn trong giai đoạn

• Thiết lập khả năng bảo mật.

• Authentication Server và Exchange Key

• Client Authentication và Exchange Key

• Finish.

**Cryptographic tính toán**

Hai hạng mục tiếp tục được quan tâm:

việc tạo ra một bí mật tổng thể được chia sẻ bởi các phương tiện của các trao đổi khóa và

• một lần giá trị 48-byte

• tạo ra bằng cách sử dụng trao đổi khóa an toàn (RSA / Diffie Hellman) và sau đó băm thông tin

thế hệ của các thông số mật mã bí mật từ bậc thầy.

• khách hàng bí mật ghi MAC, một máy chủ viết bí mật MAC, một khách hàng ghi chính, một máy chủ ghi trọng, một khách hàng ghi IV, và một máy chủ ghi IV

• tạo ra bằng cách băm mật chủ.

**TLS**

TLS là một sáng kiến tiêu chuẩn IETF mà mục đích là để tạo ra một phiên bản tiêu chuẩn Internet của SSL

với những khác biệt nhỏ

• in record format version number

• sử dụng HMAC cho MAC

• một hàm giả ngẫu nhiên bí mật mở rộng

dựa trên HMAC sử dụng SHA-1 hoặc MD5

• có mã cảnh báo thêm

• một số thay đổi trong thuật toán mã hóa được hỗ trợ

• Những thay đổi trong các loại giấy chứng nhận và các cuộc đàm phán

• Những thay đổi trong tính toán mật mã & đệm.

**HTTPS**

HTTPS (HTTP over SSL)

• combination of HTTP & SSL/TLS to secure communications between browser & server

documented in RFC2818

no fundamental change using either SSL or TLS

use https:// URL rather than http://

• and port 443 rather than 80

encrypts

• URL, document contents, form data, cookies, HTTP headers

connection initiation

• TLS handshake then HTTP request(s)

connection closure

• have “Connection: close” in HTTP record

• TLS level exchange close\_notify alerts

• can then close TCP connection

• must handle TCP close before alert exchange sent or

completed

SH (Secure Shell)

 protocol for secure network communications

• designed to be simple & inexpensive

 SSH1 provided secure remote logon facility

• replace TELNET & other insecure schemes

• also has more general client/server capability

 SSH2 fixes a number of security flaws

 documented in RFCs 4250 through 4254

 SSH clients & servers are widely available

 method of choice for remote login/ X tunnels

**SSH Transport Layer Protocol**

 server authentication occurs at transport layer, based on server/host key pair(s)

• server authentication requires clients to know host keys in advance

 packet exchange

• establish TCP connection

• can then exchange data

 identification string exchange, algorithm negotiation, key exchange, end of key exchange, service request

• using specified packet format

SSH User Authentication Protocol

 authenticates client to server

 three message types:

• SSH\_MSG\_USERAUTH\_REQUEST

• SSH\_MSG\_USERAUTH\_FAILURE

• SSH\_MSG\_USERAUTH\_SUCCESS

 authentication methods used

• public-key, password, host-based

SSH Connection Protocol

 runs on SSH Transport Layer Protocol

 assumes secure authentication connection

 used for multiple logical channels

• SSH communications use separate channels

• either side can open with unique id number

• flow controlled

• have three stages:

 opening a channel, data transfer, closing a channel

• four types:

 session, x11, forwarded-tcpip, direct-tcpip.

**Port Forwarding**

 convert insecure TCP connection into a secure SSH connection

• SSH Transport Layer Protocol establishes a TCP

connection between SSH client & server

• client traffic redirected to local SSH, travels via tunnel, then remote SSH delivers to server

 supports two types of port forwarding

• local forwarding – hijacks selected traffic

• remote forwarding – client acts for server