Universidad Técnica Federico Santa María
Departamento de Electrónica

Secure Socket Layer protocol (SSL)

Fecha: 20 de Junio del 2000
Profesor: Agustín Gonzalez
Alumno: Felipe Figueroa

Resumen

El protocolo SSL tiene la nalidad de entregar un mecanismo práctico de seguridad, a nivel de aplicación, y de amplio uso para sistemas de comunicación cliente/servidor en internet. Este trabajo muestra un análisis técnico de la encriptación en el protocolo SSL 3.0. Se presenta una serie de pequeños defectos en el protocolo y algunos ataques activos al SSL, que sin embargo pueden ser corregidos sin modicar su estructura básica.

Introducción

El crecimiento explosivo de Internet a traído la necesidad de proteger información vulnerable enviada por esta red abierta. Es por esto que la nalidad principal del protocolo SSL es entregar privacidad y condencialidad entre dos aplicaciones en comunicación.

El protocolo se compone de dos capas. En el nivel mas bajo, ubicado sobre un protocolo de transporte como el TCP, se encuentra el protocolo SSL de registro, usado para encapsular una serie de protocolos de niveles mas altos, otorgándoles condencialidad, autenticidad y protección. Uno de estos protocolos encapsulados es el protocolo SSL de establecimiento de conexión, que permite al cliente y servidor autenticarse y negociar un algoritmo de encriptación, y claves criptográcas antes que el protocolo de la aplicación transmita o reciba el primer byte de datos. Una de las ventajas es su independencia de los protocolos de aplicación. Los protocolos de niveles mas altos se pueden montar sobre el protocolo SSL en forma transparente.

Las tres propiedades básicas del SSL son:

- La conexión es privada. Se utiliza encriptación después del establecimiento de conexión para denir una clave secreta, y se usa criptografía simétrica para encriptar los datos.

- Se puede autenticar la identidad usando criptografía asimétrica o de clave pública.

- La conexión es conable. El transporte de mensajes incluye conrmación de integridad usando una MAC[1] codicada.

Otra nalidad del protocolo SSL 3.0 es su interoperabilidad, esto es, diferentes programadores deberían ser capaces de desarrollar aplicaciones utilizando SSL 3.0, que posteriormente intercambiarán parámetros criptográcos en forma exitosa sin necesidad de conocer los códigos fuente de las demás aplicaciones.

SSL busca proveer un marco de trabajo en el cual puedan ser incorporados métodos nuevos de encriptación y claves públicas como sea necesario.

Siempre es importante una eciencia relativa en el proceso de encriptación. Las operaciones criptográcas tienden a sobrecargar la CPU, particularmente las operaciones de claves públicas. Por esta razón, el protocolo SSL ha incorporado un esquema opcional de "cache" de sesiones para reducir el número de conexiones establecidas desde cero. Adicionalmente, se ha tomado cuidado en reducir la actividad en la red.

[1] *Message Authentication Cryptography*

Capa de Registro

Esta sección considera la fortaleza criptográca del protocolo de la capa de registro, y asume que el protocolo de intercambio de claves ha establecido una sesión, claves, y parámetros de seguridad. Por supuesto es vital para la seguridad de los datos de una aplicación, contar con un protocolo de intercambio de claves conable.

Condencialidad

El protocolo SSL encripta todos los datos de la capa de aplicación con un código y una clave de sesión corta negociada por el protocolo de establecimiento de conexión. Existe una amplia variedad de algoritmos usados en modos estándar, así que las aplicaciones pueden encontrar siempre un algoritmo de encriptación que cumpla con sus requerimientos de seguridad. Hay un buen manejo de claves: se generan claves de sesión de corto término mezclando en forma aleatoria datos secretos. En una conexión se usan claves independientes en cada dirección.

El análisis de tráco tiene la intención de recuperar información condencial sobre sesiones de protección al examinar campos de paquetes no encriptados y atributos desprotegidos. Por ejemplo, al examinar las direcciones IP fuente y destino desencriptadas, o al examinar el volumen de tráco en la red, el analista puede determinar que partes están interactuando, que tipos de servicio están en uso, e incluso a veces recuperar información sobre negocios o relaciones personales. En la práctica, normalmente los usuarios consideran relativamente inofensivo este tipo de amenazas, así que el SSL no intenta detener el análisis de tráco.

Sin embargo, hay otras amenazas que conlleva el análisis de tráco. Se ha notado que examinando el largo de texto codicado puede revelar información sobre requerimientos de URL en tráco SSL o SSL encriptado. Cuando un navegador de Web se conecta a un servidor Web a través de un transporte encriptado como el SSL, el requerimiento de recepción que contiene el URL es transmitido en forma encriptada. Exactamente qué página Web fue bajada por el navegador es considerada información condencial, y por esta razón, normalmente el conocer el URL es a menudo suciente para que algún adversario pueda bajar la página Web completa. Pero el análisis de tráco puede recuperar la identidad del servidor Web, el largo del URL requerido, y el largo de los datos HTML regresados por el servidor Web. Esta ltración podría permitir a algún espía descubrir qué página Web fue accesada.

Es importante que el SSL proteja en forma segura datos condenciales incluso ante ataques activos. Por supuesto el algoritmo de encriptación debería ser invulnerable ante

ataques de texto plano o texto codicado adaptivos, pero esto no es suciente. Se ha sabido que sosticados ataques activos sobre la capa de registro pueden violar la condencialidad de un sistema incluso ante un código fuerte. Al parecer la capa de registro del SSL 3.0 resiste estos poderosos ataques.

Autenticación de Mensajes

Además de proteger la condencialidad de los datos de alguna aplicación, el SSL autentica criptográcamente las comunicaciones sensibles. En internet, cada vez es mas fácil lanzar ataques activos. Mas aún, el incentivo nanciero por explotar las vulnerabilidades de la seguridad en comunicaciones crece rápidamente. Por esto es necesaria una fuerte autenticación de mensajes.

El SSL protege la integridad de los datos de una aplicación usando una MAC (Message Authentication Cryptography o Criptografía de Autenticación de Mensajes). Los diseñadores del SSL decidieron usar HMAC, una construcción simple y rápida, que ha mostrado ser, en su formato mas reciente, óptima para máxima seguridad.

Las claves MAC en SSL contienen al menos 128 bits de entropía, incluso en modos de exportación debilitados, lo que debería entregar una excelente seguridad para estos modos e implementaciones de tipo doméstico. Se utilizan claves independientes para cada dirección de conexión en cada conexión y por cada nueva ramicación de una conexión. La elección de HMAC debería detener los ataques de análisis de criptografía. SSL no entrega servicios de no repulsión, y deliberadamente deja esto a protocolos de aplicación de niveles más altos.

El uso nativo de MAC no necesariamente detiene a un posible adversario de enviar paquetes vencidos. Los ataques de repetición son una preocupación legítima, y ya que es tan fácil protegerse contra ellos, sería una irresponsabilidad fallar en direccionar estas amenazas. El SSL protege contra ataques de repetición al incluir una secuencia de números implícita en los datos MAC. Este mecanismo también protege contra datos atrazados, reordenados o borrados. Los números de secuencia tienen 64 bits de largo y son mantenidos separados para cada dirección de cada conexión, y son renovados en cada nuevo intercambio de claves, así que no hay vulnerabilidades obvias.

Protocolo de Intercambio de Claves

Esta sección considera la seguridad del protocolo SSL de establecimiento de conexión. El diseño de un protocolo de intercambio de claves seguro ha sido un difícil esfuerzo. Hay envuetas una cantidad signicativa de complejidades, por lo que llegar a descubrir algunas

debilidades no nos debería sorprender.

El ujo de mensajes del protocolo de establecimiento de conexión del SSL 3.0 implica al cliente y servidor negociando un paquete de códigos aceptable para ambas partes, intercambiando tiempos al azar, y el cliente enviando un pre_master_secret encriptado. Entonces cada terminal deriva un master_secret del pre_master_secret y verica que su protocolo está coincidiendo al autenticar cada mensaje con el master_secret. Asumiendo que la vericación es exitosa, ambos generan claves de sesión desde el master_secret y proceden a enviar datos de aplicación protegidos criptográcamente. El SSL también incluye un protocolo de reestablecimiento de sesión mas simple, que permite generar claves de sesión renovadas a dos partes que ya han intercambiado un master_secret, e iniciar una nueva conexión con estos parámetros.

El SSL 3.0 consta de un mecanismo para prevenir que los mensajes del protocolo de establecimiento de conexión sean modicados. Todos los mensajes iniciales de éste son enviados desprotegidos. El protocolo de intercambio de claves, en vez de modicar los parámetros en uso, modica el estado de una sesión pendiente. Después que la negociación se ha completado, cada parte envía un mensaje corto de cambio de especicación de código, que simplemente alerta a la otra parte a renovar el estado de la sesión pendiente a actual. El estado de la nueva sesión se inicia desde el próximo mensaje, aunque el mensaje de cambio de especicación de código está desprotegido, o más bien dicho, esta protegido con el estado de la antigua sesión. Inmediatamente después viene el mensaje de terminación que contiene una MAC en todos los mensajes del protocolo de establecimiento de conexión codicados por el master_secret. El master_secret de 48 bytes, nunca es revelado, en cambio se generan claves de sesión desde él. Esto asegura que incluso si las claves de sesión son recuperadas, el master_secret permanecerá secreto para que los mensajes del protocolo de establecimiento de conexión sean autenticados en forma segura. El mensaje de terminación viene en sí protegido con el nuevo paquete de códigos establecido. Ninguna de las partes puede aceptar datos de aplicación hasta que un mensaje de terminación haya sido recibido y vericado desde el otro terminal.

El protocolo de establecimiento de conexión del SSL 3.0 tiene también otro defecto de diseño. Un servidor puede enviar parámetros de clave pública de corta vida, rmados bajo su clave certicada de largo término en el mensaje de intercambio de claves del servidor. Desafortunadamente la rma en los parámetros de corta vida, no protege al campo que especica qué tipo de algoritmo de intercambio de claves está en uso.

A continuación se muestra la estructura de datos mas relevante del mensaje de intercambio de clave del servidor en SSL 3.0

```
enum { rsa, dife_hellman, ... }
   KeyExchangeAlgorithm;
struct {
   opaque rsa_modulus;
   opaque rsa_exponent;
} ServerRSAParams;
struct {
   opaque dh_p;
   opaque dh_g;
   opaque dh_Ys;
} ServerDHParams;
struct {
   select (KeyExchangeAlgorithm) {
      case dife_hellman:
         ServerDHParams params;
         Signature signed_params;
      case rsa:
         ServerRSAParams params;
         Signature signed_params;
   }
} ServerKeyExchange;
```

El valor de KeyExchangeAlgorithm es sacado implícitamente desde el paquete de códigos negociado por cada terminal. El campo signed_params contiene la rma del servidor en una mezcla del campo ServerParams relevante, ya sea ServerDHParams o ServerRSAParams, de acuerdo al valor de la variable KeyExchangeAlgorithm, pero la rma no cubre el valor de KeyExchangeAlgorithm.

Cabe mencionar que una implementación cautelosa no debería ser engañada por ciertos trucos que chequeen cuidadosamente el largo del campo ServerParams. Sin embargo, las implementaciones en general no dicen nada al respecto, ya que alguna implementación estándar podría ser vulnerable.

Seguridad del master_secret

Para la seguridad del SSL, es de tremenda importancia que el master_secret se mantenga realmente secreto. Todas las claves de sesión son generadas desde el master_secret, y la protección contra interferencias con el protocolo de establecimiento de conexión SSL, recae grandemente sobre su condencialidad. En diseño de protocolos esto signica que el uso del master_secret debería ser muy limitado.

Un enemigo podría juntar cantidades ilimitadas de texto conocido para la transformación MAC codicada por el master_secret, encontrado en el mensaje de terminación. El adversario

informado abre muchas conexiones simultáneas a través de mensajes "client hello", pidiendo la reanudación de la sesión en cuestión. Para cada una de esas conexiones, el servidor tomará un tiempo aleatorio, calculará una MAC con el master_secret, y la enviará de vuelta encriptada en un mensaje de terminación. Un adversario inteligente debería dejar todas estas conexiones abiertas sin responder a los mensajes de terminación del servidor. El enviar datos incorrectos a cualquiera de las conexiones, causará una alerta fatal, lo que hace la sesión irresumible. De este modo, el oponente puede juntar grandes cantidades de texto conocido, mezclado con el master_secret. Si algún analista de criptografía descubre un ataque sobre adhoc-MAC() que use mucho texto conocido para recuperar la clave secreta, el protocolo SSL podría llegar a ser inseguro. Probablemente un protocolo de establecimiento de conexión muy robusto, podría limitar la cantidad de texto conocido que está disponible a un analista de criptografía.

En resumen, el protocolo SSL de establecimiento de conexión tiene varias vulnerabilidades de que preocuparse. Pero estas realmente pueden causar algún problema ante ataques activos. Más allá de esto, no son debilidades generalizadas ya que diferentes implementaciones pueden o no ser vulnerables. Un defecto en un protocolo no necesariamente producirá una implementación vulnerable.

Conclusiones

En general, el protocolo SSL 3.0 otorga una excelente seguridad contra espías y otro tipo de ataques pasivos. Aunque los modos de exportación debilitada ofrecen sólo una condencialidad mínima, no hay nada que el SSL pueda hacer para mejorarlo. El único cambio que se podría recomendar a la protección del SSL contra ataques pasivos, es un soporte para detener el análisis de tráco de un requerimiento de longitud.

El análisis hecho en este trabajo, revela una serie de maneras en que la robustez del protocolo SSL podría ser mejorada. Muchas de las observaciones no son inspiradas directamente por vulnerabilidades, pero aún así vale la pena considerarlas en posibles futuras versiones del SSL.

Es importante no exagerar el signicado práctico de los posibles defectos del protocolo SSL. La mayoría de las debilidades descritas, provienen de una pequeña revisión, y pueden ser corregidas sin necesidad de modicar la estructura básica del protocolo.

Aunque todavía hay que mejorar unos pocos detalles técnicos, todo el SSL 3.0 es un valioso paso hacia una práctica seguridad en comunicaciones para aplicaciones de internet.

Referencias

- *David Wagner, Bruce Schneier,*
  "Analysis of the SSL 3.0 protocol",
  April 15, 1997

- *Alan Freier, Philip Karlton, Paul Kocher,*
  "The SSL Protocol version 3.0",
  November 18, 1996

- *http://home.netscape.com/security/techbriefs/ssl.html*

- *http://www.netscape.com/eng/ssl3/*