

Otras técnicas de autenticación

Con el fin de hacer mas claras las explicaciones que siguen, se describirán brevemente las Funciones Resumen.

Funciones resumen (o de hash)

Una función resumen o de hash H es una transformación que, tomando como entrada una cadena x de bits de longitud variable, produce como salida una cadena h de bits de longitud fija ($h = H(x)$). Para que una función de este tipo pueda usarse con propósitos criptográficos, se debe cumplir una serie de requisitos:

- *la entrada puede tener cualquier longitud. Deben proveerse mecanismos para evitar el desbordamiento (overflow).

- *la salida debe ser de longitud fija, independientemente de cual fuera la longitud de la entrada.

- *para cualquier entrada, su resumen (o valor de hash) debe ser sencillo de calcular.

- *la función resumen debe ser de un "único sentido", entendiendo por este concepto que, dado $f(x)$, debe ser computacionalmente difícil encontrar un valor y (tal vez el mismo x) tal que $f(y) = f(x)$.

- *es difícil encontrar dos entradas x e y , tales que $H(x) = H(y)$ (colisiones).

Al resumen o valor de hash de un mensaje M se le llama generalmente huella digital de M . Si la salida de la función tiene una longitud de n bits, entonces existen $k = 2^n$ salidas diferentes. Las funciones resumen son también extensivamente utilizadas como parte de los mecanismos que generan números aleatorios.

Ejemplos de funciones resumen usadas en criptografía son MD2, MD4, MD5 o SHA.

Ya se ha señalado la conveniencia de utilizar criptografía de clave asimétrica para la autenticación. Pero ésta no es más que una faceta de un problema más general.

Esquemas de identificación

Son métodos mediante los que una entidad (un usuario, una máquina, etc.) puede probar su identidad a alguien distinto a sí mismo, sin revelar ningún dato propio esencial que permita que un intruso o su mismo interlocutor le suplanten. Si Pepe se autentifica a sí mismo frente a María, aunque Manolo intercepte el canal, debe ser imposible para él suplantar a Pepe en el futuro (también debe serlo para María).

Los esquemas de identificación implican que el aspirante:

- *Demuestra saber algo: palabras de paso.

- *Demuestra tener algo: una tarjeta magnética, por ejemplo.

*Muestra algo característico e indeleble: retina, ADN, ritmo de teclado, ...

*Está en un cierto sitio.

*Existe un tercero de confianza que lo avala: certificados.

El método tradicional de identificación es el uso de una palabra de paso (password). Pepe se identifica a sí mismo frente a un servidor introduciendo su palabra de paso cuando trata de acceder a su cuenta. Desgraciadamente, este esquema es inseguro. La palabra de paso puede ser obtenida mediante:

*Exposición no autorizada (robo de la palabra de paso).

*Adivinación: es un hecho que los usuarios tienden a elegir palabras de paso inadecuadas, que pueden ser fácilmente adivinadas (utilizando cracks)

*Escucha en tránsito: mediante sniffers, por ejemplo.

*Reproducción ciega (replay): un intruso que haya interceptado las comunicaciones entre Pepe y el servidor puede repetir el mensaje previo (es decir, la palabra de paso) de Pepe, de modo que gane acceso a la cuenta o al recurso del usuario interceptado.

*Alteración del verificador de la palabra de paso.

La autenticación mediante palabra de paso es un ejemplo de esquema de identificación unilateral, ya que sólo la entidad de uno de los extremos de la comunicación se identifica a sí misma, sin obtener identificación desde el otro extremo.

Un mecanismo mucho más común consiste en el uso de firmas digitales y de criptografía de clave pública usando esquemas **reto-respuesta**: Pepe envía retos a Manolo con diferentes preguntas y verifica las respuestas hasta que considera que la persona que responde a sus retos es, efectivamente, Pepe.

En uno de tales esquemas (similar, conceptualmente, al utilizado por el Handshake Protocol de SSL, que se verá mas adelante), Pepe genera un número aleatorio y se lo transmite a Manolo. Éste genera un nuevo número aleatorio y firma digitalmente un mensaje, el cual contiene ambos números aleatorios. Entonces envía el mensaje firmado, junto con su número aleatorio a Pepe. Pepe verifica la firma para asegurarse de que se está comunicando, efectivamente, con Manolo (en este caso, un único reto es suficiente). Este esquema es resistente frente a intrusos interceptando la línea. También previene la suplantación de Pepe por Manolo más adelante. Hasta este punto, el protocolo es unilateral.

Para identificarse a sí mismo, Pepe puede también firmar el mensaje que contiene ambos números aleatorios y devolver el mensaje firmado a Manolo. Manolo verifica la firma para convencerse de la identidad de Pepe. Este es un ejemplo de un esquema de identificación mutua.

Palabras de paso de uso único (One-Time Password)

Una palabra de paso de uso único actúa exactamente como su nombre indica. Se usa una única vez y no vuelve a ser empleada. Esto proporciona una gran seguridad frente a intrusos que utilicen la repetición ciega de palabras de paso (replay).

Existen varias formas en las que un mecanismo de este tipo puede ser implementado. Algunos de ellos requieren el uso de un hardware especial.

En 1981, Leslie Lamport propuso, en su célebre artículo Password Authentication with Insecure Communication, un esquema de palabras de paso de uso único.

Se asume la existencia de una función F razonablemente fácil de computar en un sentido, pero imposible de invertir de forma efectiva. Esto nos lleva a pensar en las funciones resumen* como buenas candidatas a función F . También debe asumirse que el usuario que desea identificarse posee algún secreto (una frase secreta), x .

Para permitir al usuario ingresar en el sistema cierto número de veces, n , el centro verificador calcula $F(x)$ n veces. Por ejemplo, para permitir al usuario acceder 1000 veces al sistema antes de cambiar su frase secreta, el centro verificador debe calcular $F^{1000}(x)$ y almacenar únicamente este valor.

Si el usuario quiere acceder al sistema, éste le plantea un reto con un número de sesión y un valor aleatorio (nrv , non-repeatable value). El usuario utilizará este último dato, junto con su frase secreta, para alimentar una función resumen. El usuario aplicará dicha función tantas veces como indique el número de secuencia. El resultado será la respuesta al reto.

En el ejemplo que hemos considerado, la primera vez que el usuario acceda al sistema, calculará $F^{999}(x, nrv)$. El centro verificador validará al usuario calculando:

$$F(F^{999}(x, nrv)) = F^{1000}(x, nrv)$$

Si el acceso es correcto, el valor suministrado por el usuario $F^{999}(x, nrv)$ se convierte en el nuevo valor almacenado en el centro verificador.

De este modo, se previene la suplantación en el centro servidor y la repetición ciega de palabras de paso.

Kerberos

Kerberos es un sistema de autenticación diseñado en el MIT para su utilización sobre redes inseguras, como, por ejemplo, Internet. El sistema de autenticación Kerberos fue diseñado con dos propósitos, proveer autenticación y distribuir claves. El sistema Kerberos actúa como autoridad de certificación que garantiza una relación correcta entre claves y personas.

Esta autenticación se lleva a cabo usando acreditaciones (tickets). Estas acreditaciones son sólo válidas durante un periodo de tiempo determinado y en lo que se conoce como realm (conjunto de máquinas y usuarios, generalmente, coincide con una organización).

Cada usuario (o también un proceso) es conocido como principal. Este principal conecta con el Centro de Distribución de Claves (Key Distribution Center, KDC) del realm para solicitar

la acreditación que usará para hablar con el Servidor de Concesión de Acreditaciones (Ticket Granting Server, TGS) del realm. El KDC verifica la identidad del principal y otorga la acreditación. Si esta acreditación es válida, el TGS otorga una nueva acreditación al principal para comunicarse privadamente con un servidor S situado dentro del realm (con una clave conocida por el principal y el servidor S).

Las principales debilidades de Kerberos son las siguientes:

la autenticación del propio servidor Kerberos.

Kerberos es un sistema diseñado para autenticar a usuarios finales ante diferentes servidores, no para autenticar sistemas ante otros sistemas.

la gran mayoría de los sistemas no disponen de un área segura para almacenar las claves (ya que esta deben ser almacenadas en claro en el paso inicial de obtención de una acreditación).

Medios de transporte

Multi-purpose Internet Mail Extensions (MIME)

El mecanismo básico de transporte de correo a través de Internet queda definido por SMTP (Simple Mail Transfer Protocol). El RFC 822 describe el formato de los mensajes intercambiados. Según este estándar, cada mensaje consta de dos partes:

*header (cabecera): contiene la información necesaria para que el mensaje llegue a su destino. El formato de la cabecera es definido también por el estándar y está formada por un conjunto de pares clave/valor estructurados.

*body (cuerpo o contenido): la información que debe recibir el destinatario. Esta información estará siempre en formato de texto de siete bits (US-ASCII).

MIME es un protocolo de intercambio de objetos a través de Internet. Se desarrolló inicialmente para enriquecer los intercambios de correo electrónico, limitados según el RFC 822 a ASCII de siete bits, habiéndose extendido a otros muchos protocolos. El formato MIME permite el envío de texto enriquecido (8 bits), gráficos, ficheros de audio, vídeo, ... En MIME, cada objeto de mensajería se encapsula en una especie de envoltorio que especifica tanto su semántica como el medio de codificación utilizado. La caracterización semántica hecha en la cabecera permite asociar los datos con su mecanismo de transporte (codificación) y con su significado, de forma que el remitente y el destinatario utilicen coordinadamente los datos intercambiados.

Privacy Enhanced Mail (PEM)

PEM es un sistema similar a MIME y desarrollado en paralelo con éste para crear objetos de mensajería seguros. Incluye cifrado, autenticación y gestión de claves y permite el uso de algoritmos simétricos y de clave pública. Con el desarrollo de MIME, PEM es, de alguna forma, repetitivo, por lo que se ha visto desplazado por S/MIME. De hecho, lleva más de dos años en estado de borrador.

MIME Object Security Objects (MOSS)

MOSS (también conocido como PEM-MIME) es una extensión de MIME derivada de PEM que aporta exclusivamente lo que le falta a MIME para obtener las garantías deseadas: claves, firmas digitales, certificados, etc. De este modo se consigue un intercambio seguro de objetos. De acuerdo con la propia naturaleza de MIME, es posible aplicar diferentes servicios de seguridad a cada parte del mensaje.

Este estándar ha sufrido críticas debido a sus requerimientos, que son extremadamente flexibles. Fruto de esta flexibilidad puede ocurrir que dos agentes de correo diferentes, ambos soportando MOSS, sean incompatibles. Esta flexibilidad es, en parte, una reacción a la rigidez de PEM, la cual le hacía impopular entre los usuarios.

Secure-Multipurpose Internet Mail Extensions (S/MIME)

S/MIME es un protocolo que añade firmas digitales y cifrado a los mensajes MIME. MIME, en sí mismo, no proporciona ningún servicio de seguridad. S/MIME define esos servicios, utilizando criptografía de clave pública, siguiendo la sintaxis definida en el PKCS#7.

S/MIME ha sido adoptado recientemente por un gran número de compañías como FTP Software, Qualcomm, Microsoft, Lotus, VeriSign, Netscape o Novell.

Secure HTTP (S-HTTP)

Secure HTTP es un protocolo propuesto por Enterprise Integration Technologies (EII), patrocinado por el consorcio CommerceNet y desarrollado en la actualidad por Terisa Systems. Constituye una extensión del protocolo HTTP, incorporando cabeceras MIME para aportar confidencialidad, autenticación, integridad e irrenunciabilidad de las transacciones.

Utiliza un sistema inspirado en PEM, añadiendo suficientes cabeceras a cada transacción para lograr cada uno de los objetivos propuestos. Las transacciones HTTP constan simplemente de una petición de parte del cliente que produce una respuesta del servidor. S-HTTP especifica que el cliente envíe directamente toda la información pertinente: claves, certificados, códigos de integridad, etc. (incluyendo la posibilidad de referenciar secretos compartidos obtenibles exteriormente: intercambios previos o bases de datos comunes). El servidor responde siguiendo la misma filosofía PEM. El protocolo soporta varios mecanismos criptográficos y negocia los modos y opciones de estos mecanismos.

Evaluación de S-HTTP

Las principales ventajas de S-HTTP son su flexibilidad y su integración dentro de HTML (extensiones al lenguaje similares a las introducidas periódicamente por Netscape Co. en sus navegadores).

Entre sus debilidades podemos señalar los efectos derivados de mantener la compatibilidad hacia atrás y la necesidad de implementar servidores que soporten las extensiones a HTML aportadas por el protocolo S-HTTP.

Secure Sockets Layer (SSL)

Secure Sockets Layer (SSL) es un protocolo diseñado por Netscape Communications Co., que dispone un nivel seguro de transporte entre el servicio clásico de transporte en Internet (TCP) y las aplicaciones que se comunican a través de él. Proporciona conexiones seguras sobre una red insegura como es Internet, asegurando las siguientes características:

*conexión privada: la información se cifra utilizando criptografía de clave simétrica.

*autenticación: usando criptografía de clave pública.

*integridad: la integridad de los mensajes se asegura usando firmas digitales.

Además, proporciona características adicionales:

*extensibilidad: es capaz de soportar nuevos protocolos en el futuro.

*eficiencia: al utilizar compresión, minimiza el tiempo necesario para establecer la conexión.

*compatibilidad: productos con diferentes versiones de SSL pueden interoperar entre sí.

SSL se compone de dos partes diferenciadas:

Handshake Protocol: se encarga de establecer la conexión y determinar los parámetros que se van a utilizar posteriormente (fundamentalmente se trata de establecer cual va a ser la clave simétrica que se utilizará para transmitir los datos durante esa conexión).

Record Protocol: comprime, cifra, descifra y verifica la información que se transmite. Este sistema es transparente para las aplicaciones finales, que simplemente saben que el canal se encarga de proporcionarles confidencialidad entre extremos. Por tanto, podemos situar protocolos como HTTP, FTP, NNTP o Telnet.

Modo de funcionamiento

El denominado Handshake Protocol se compone dos fases, autenticación de servidor y autenticación de cliente, no siendo obligatoria esta última. En primer lugar, el servidor, respondiendo a una petición del cliente, le envía su certificado y las preferencias en lo que a algoritmos de cifrado se refiere. En ese momento, el cliente genera una clave maestra, la cifra con la clave pública del servidor y la transmite al servidor. El servidor recobra la clave maestra y se autentica respecto al cliente devolviendo un mensaje cifrado con la clave maestra. Los datos siguientes son cifrados con claves derivadas de esta clave maestra.

En la segunda fase opcional, el servidor envía un reto al cliente. Éste se autentica respecto al servidor retornándole el reto firmado digitalmente por el cliente, así como su certificado (el cual incluye su clave pública).

Algoritmos utilizados

Una gran variedad de algoritmos criptográficos son soportados por SSL. Durante la fase de acuerdo o "handshaking", se utiliza RSA (clave pública). Después del intercambio de claves, se usan unos cuantos algoritmos, entre los que se incluyen RC2, RC4, IDEA, DES y Triple-DES. Como función resumen se usa MD5 o SHA-1. Los certificados siguen el formato X.509.

El protocolo SSL está, gracias a los esfuerzos de Netscape, ampliamente extendido. La presencia de https:// en el URL de un servidor indica se trata de un servidor "seguro" y que debe utilizarse SSL en la comunicación entre dicho servidor y cliente (navegador). Los navegadores más extendidos (Netscape Navigator y Microsoft Internet Explorer) son capaces de "hablar" SSL. Esto queda indicado (en el caso de Netscape Navigator) de la siguiente forma:

*la llave de la parte inferior izquierda del navegador aparece completa, no partida como habitualmente (en los casos del MS Internet Explorer y de Netscape Communicator es un candado cerrado el que aparece en la esquina inferior izquierda).

*aparece una línea azul en el límite superior de la línea de visualización de la pantalla del navegador.

*la información del documento alojado en el servidor seguro incluye los datos del certificado que avala al servidor seguro.

Los servidores más populares de la empresa Netscape (Commerce Server, FastTrack Server y Enterprise Server) soportan SSL, con las habituales limitaciones de exportación (clave RC4 de 40 bits para los productos vendidos fuera de EE.UU. o Canadá). El servidor más extendido a escala mundial, Apache, posee una versión SSL, Stronghold, con la ventaja añadida de que, al haber sido desarrollado fuera de los EE.UU., puede vender la versión "completa" de SSL con claves de 128 bits. El servidor de Microsoft, Internet Information Server 2.0, que viene de serie con Windows NT 4.0 no soporta SSL. Las últimas versiones (IIS 4.0) soportan plenamente el estándar (junto con protocolos propios como PCT).

Conclusión

A diferencia de S-HTTP, que es un protocolo substitutivo de HTTP, SSL extiende su soporte a otros protocolos habituales en Internet. Esta es una de las principales ventajas que aporta este último. Mientras que S-HTTP proporciona cifrado en el nivel de aplicación (en este caso WWW), SSL lo hace en el nivel de conexión, proporcionando un canal seguro en el nivel de red. Por lo demás, S-HTTP y SSL pueden convivir, utilizándose uno u otro en diferentes instantes de una transacción comercial, o incluso utilizándose simultáneamente.

El sistema es tan robusto como lo sea el menos seguro de los algoritmos que utilice. Claves públicas cortas o claves DES o RC4 de 40 bits deben utilizarse con precaución. Estos son los problemas que plantean las leyes de EE.UU.

La principal desventaja de SSL no estriba en sus fundamentos teóricos o implementación, sino, fundamentalmente, la menor protección que proporcionan las versiones exportables de los productos basados en este protocolo.

También debe tenerse especial cuidado en decidir qué autoridades de certificación y qué certificados son fiables.