Algoritmos HASH y vulnerabilidad a ataques

Mena Miranda Yerko
Universidad Mayor De San Andrés
Carrera De Informática
Simulación de Sistemas
yimm_79@yahoo.es

RESUMEN

El presente documento muestra el uso a los números de resumen o números HASH utilizados para la autentificación de documentos, la generación de estos números, los algoritmos mas usados, los problemas que posee, los ataques que se pueden realizar y las soluciones que replantea para evitar inconvenientes en su uso.

Palabras clave

Algoritmos HASH, números HASH, firma digital

1. INTRODUCCION

Los números HASH, considerados como números de resumen son obtenidos mediante la aplicación de un algoritmo de HASH. Este algoritmo crea un número en base al contenido de un grupo de bits, de tamaño uniforme (dependiendo del algoritmo utilizado), sin tomar en cuenta la cantidad de bits sobre la que se aplica el algoritmo. Este numero tiene una dependencia del contenido evaluado por el algoritmo permitiendo que si se realice un cambio en los datos, el numero HASH cambiara

Aunque el algoritmo de HASH trabaja con un esquema similar a la encriptación, no permite generar el contenido original en base a el numero HASH.

Los números HASH han tenido múltiples aplicaciones entre las cuales se tienen:

Comprobación de integridad de ficheros, usada sobre todo en el envío de mensajes, mediante la aplicación del algoritmo antes de enviar (emisor) y cuando se recibe el mensaje (receptor). En caso de coincidir los números HASH, el mensaje no ha sufrido modificación durante el envío.

Procesos de identificación de sistemas, mejorando la seguridad de identificación junto con el Login, Password. Algunos sistemas de autentificación (foros, sistemas, etc.) almacenan el numero HASH de la contraseña, permitiendo que si un intruso logra llegar a donde se guardan las contraseñas solo obtendrá el numero HASH de cada una y no podrá acceder a la aplicación.

Firma digital, utilizado por algoritmos de cifrado, obtiene un numero HASH de todo el mensaje siendo usado como un identificador o firma por parte del emisor.

Existen dos tipos de algoritmos hash; los no cifrados y los cifrados.

Los algoritmos hash cifrados o HMAC se pueden utilizar para determinar si se ha manipulado un mensaje enviado a través de un canal no seguro, siempre que el remitente y el receptor compartan una clave secreta. El remitente calcula el valor hash para los datos originales y envía el valor hash y los datos originales como un solo mensaje. El receptor actualiza el valor hash en el mensaje recibido y comprueba que el código HMAC calculado coincide con el transmitido. Cualquier cambio en los datos o en el valor hash producirá una desigualdad, ya que es necesario conocer la clave secreta para cambiar el mensaje y reproducir el valor hash correcto. Por consiguiente, si el original y los valores hash calculados coinciden, el mensaje se autentica.

Los algoritmos de HASH mas usados son:

MD5 (Message-Digest Algorithm 5 o Algoritmo de Firma de Mensajes 5), realiza un procesamiento en bloques de 512 bits, generando números de 128 bits. Usado aun en comprobación de ficheros

SHA-1 (Secure Hash Algorithm 1 o Algoritmo de Hash Seguro 1), toma como entrada mensajes con longitud máxima de 264 bits y produce un numero de 160 bits

Digital Signature Algorithm (DSA) es el estándar de United States Federal Government para firma digital. Su desarrollo se atribuye a David W. Kravitz, de la National Security Agency. Fue presentado por NIST en agosto de 1991, adoptado como estándar en 1993, y con última revisión de 2000. [12]. Es un algoritmo exclusivo de firma electrónica basado en clave pública, pero no vale para comunicaciones confidenciales.

RIPEMD-160 es un algoritmo del resumen del mensaje de 160 bits (y función criptográfica de hash) desarrollado en Europa por Hans Dobbertin, Antoon Bosselaers y Bart Preneel, y publicados primeramente en 1996. Es una versión mejorada de RIPEMD, que estaba basado sobre los principios del diseño del algoritmo MD4, y es similar en seguridad y funcionamiento al más popular SHA-1.

2. RUPTURA DE SEGURIDAD DE LOS ALGORITMOS HASH

2.1. Vulnerabilidades de los algoritmos HASH

La seguridad que ofrecen los números HASH depende del algoritmo que lo genera, siendo uno de los puntos revisar la dificultad para romper la seguridad de los algoritmos de HASH, y ver las soluciones en caso de ocurrir lo anterior.

Las características que debe poseer un algoritmo HASH deben ser:

A partir del número HASH debe ser imposible obtener el mensaje original.

Debe ser imposible encontrar dos mensajes con un numero HASH igual

El principal objetivo de los ataques a los algoritmos de HASH es obtener una colisión, o sea obtener el numero HASH de dos mensajes distintos tal que los números HASH sean iguales.

2.2. Ataques relacionados con vulnerabilidades de los algoritmos HASH

2.2.1. Ataque de cumpleaños

Basado en una paradoja que permite hallar la cantidad de personas que se necesita para encontrar al menos una pareja que tenga el mismo día de cumpleaños. Este ataque es el mas usado mediante el método de fuerza bruta (búsqueda intensiva de un dato x tal que su numero de HASH sea similar a otro dato y. Este ataque permite la búsqueda de 2 mensajes con numero HASH similares, pero sin conocer los mensajes que los generan.

Para los algoritmos MD5 y SHA-1 la cantidad de mensajes que se deben crear para hallar los mensajes con números HASH similares es alta, se necesitan 2^64 y 2^80 mensajes respectivamente

A pesar de todo cada vez que se reduce la cantidad de operaciones el algoritmo pierde un 50% de su fiabilidad. Resalta la investigación de Wang, Feng, Lai, y Yu que han logrado reducir la cantidad de mensajes para obtener la colisión en el algoritmo SHA-1 de 2^80 a 2^69 mensajes y la investigación realizada en Australia que redujo a 2^52 mensajes

2.2.2. Ataque de mensaje sin sentido

Mediante el envío de un mensaje sin sentido, tal que colisione con un mensaje con sentido, permitiendo atacar algoritmos de encriptado asimétricos de firma digital, firmando documentos sin sentido que puede ser visto por el receptor como fidedignos

2.2.3. Ataque con mensaje con sentido

En base al anterior pero se realiza con un mensaje falso que posee sentido, que es capaz de colisionar con un mensaje verdadero. Esto permite crear certificado o mensajes con autenticidad alta creando un efecto desastroso.

Este tipo de ataque es el mas problemático a enfrentar pero a la vez es el mas difícil de obtener, dado que para obtener el mensaje falso se necesita hacer 2^n mensajes que posean sentido, donde n es el numero de bits usa el algoritmo de HASH en sus resultados.

2.3 Soluciones a los problemas de los algoritmos hash

Los problemas principales que debe resolverse para evitar los ataques a los algoritmos HASH se basan en 2 cuestiones

¿Qué se gana incrementando el número de bits de salida del algoritmo?

¿Qué ataques reales son practicables?

La primera cuestión indica que a mayor cantidad de bits posea la salida del algoritmo HASH, se obtiene una menor probabilidad que un ataque de fuerza bruta (aplicar una función inversa del algoritmo de HASH 2n/2 veces donde n es el numero de bits de salida del algoritmo).Esta solución pierde eficiencia a largo plazo dado la mayor rapidez de los ordenadores.

La segunda cuestión tiene relación con la anterior, dado la gran cantidad de operación necesarias para obtener los mensajes con numero HASH iguales.

Dado el uso de los algoritmos HASH en la certificación de documentos digitales, el problema radica en los documentos que ya han sido firmados y tengan valor en el tiempo ya que al ser autentificados por estos algoritmos, es posible que mediante la búsqueda de colisiones se pueda crear un documento similar en numero HASH pero con un contenido distinto.

Por el momento la solución más práctica es el uso combinado de 2 algoritmos por ejemplo SHA-1 Y RIPEMD-160, puesto que una colisión en SHA-1 es virtualmente imposible que coincida también en RIPEMD-160.

4. EL DILEMA DE LOS ALGORITMOS HASH

Evidencia Digital, como cualquier otro tipo de evidencia, requiere identificación, colecta, una cadena de custodia, examen/análisis, y finalmente autenticación en Corte durante la presentación para verificación de hechos.

A raíz de las mejores prácticas, un hash forense es utilizado para identificación, verificación y autenticación de archivos de datos. Un hash forense es una forma de ChekSum (Verificación de suma de bits). Un ChekSum es un cálculo matemático, que en su forma más simple, suma la variedad de bits en una cadena de datos y proporciona un valor MD5 (message digest 5) y SHA-1 (Secure Hash Algorithm 1) son algoritmos más complejos de verificación. Un hash forense es el proceso de utilizar una función matemática y aplicarla dentro los datos colectados, el valor hash de resultado es un identificador único para la adquisición (recogido) de datos (similar a una secuencia de ADN o una huella digital de los datos). Cuando un algoritmo es utilizado, se calcula una

cadena de números para un archivo digital. Cualquier cambio en los datos dará lugar a un cambio en el valor de hash. Tanto MD5 y SHA-1 son algoritmos que se utilizan para los archivos de imagen forense. El proceso hash se utiliza normalmente en la adquisición de los elementos de prueba, durante la verificación de la imagen forense (copia de las pruebas), y de nuevo al final del examen para asegurar la integridad de los datos y procesamiento forense. MD5 y hash SHA-1 también se utilizan actualmente para validar la integridad de los archivos descargados en las aplicaciones de la tecnología de información, que han sido aceptadas por la comunidad científica y de consumo para confirmar que los archivos que se solicitan o descargan son los mismos y se encuentran completos.

Si los delincuentes saben crear un hash de colisión, impedirían el filtro de archivos de identificación conocidos, ¿por qué no se ponen en venta sumas MD5 preconfiguradas de los ficheros precisamente con ese fin? Por ejemplo, si tengo una extensa colección de pornografía de contrabando, ¿por qué no hacer esos archivos duplicados perfectamente inofensivo en un conjunto de archivos imagen?

Incluso si los delincuentes causan la colisión con los archivos de contrabando, porque no cambiar el contenido visual de la imagen de un ilícito, sólo MD5 impediría la identificación de la imagen.

No me sorprende en absoluto al descubrir que algunos grupos delictivos están trabajando en la creación de un hash perfectamente ajustado a las correspondientes hash de Aplicación de la ley que establece los usos para la identificación y filtrado.

Es muy poco probable en este momento porque la información de la colisión de investigación criptográfica indica que no es posible construir un valor de hash conocido utilizando esta técnica.

"Nosotros no podemos tener como objetivo obtener un valor hash, y producir un bit (significativo) de entrada a una cadena hashing para obtener un valor… estos archivos tienen que ser especialmente preparado por el atacante … Los archivos con una cantidad conocida de hash que no se han preparado de esta manera se no vulnerables. " (Stevens et al.)

Además, el "grupo delictivo" necesita dos cosas que serían difíciles de obtener, la Aplicación de la ley conocida Establece hash del archivo y todos los archivos físicos representados en las bases de datos de conjunto de hashes. Stevens et al. promedió una colisión de, cada cinco minutos, se tardaría más de 69,25 años para crear una colisión MD5 para los 14,5 millones de valores en la MD5 NIST Biblioteca Nacional de software de base de datos de hashes.

Así, en el tribunal, el fiscal tiene a su experto en el argumento de que el valor hash MD5 demuestra que el archivo en cuestión es de contrabando. Entonces el abogado defensor aporta su experto, que muestra exactamente el mismo valor hash MD5 de una imagen de un camión de bomberos y boom! Su caso fue sólo malo - a menos que haya examinado todos los archivos identificados y los que eran conocidos de contrabando.

Un valor hash MD5 no puede probar que el archivo es de contrabando, y un experto no debe opinar o concluir que el valor hash MD5 demuestra que lo es. Valores hash MD5 pueden identificar un archivo que debe ser verificado visualmente. Sí, es mucho tiempo, y siguiendo las mejores prácticas es una parte de nuestro proceso en estos casos.

Así pues, usted tiene un montón de falsificaciones dispersas entre las reales entonces debe volver a sentarse y esperar a que el perito declare.

Esto ya ocurre (hay "hijos / niños" sitios web que promueven la "adolescentes reales" y / o apenas legal). Esa es la razón por la verificación visual y recursos tales como el Centro Nacional para Niños Desaparecidos y Explotados (NCMEC) Programa de Identificación de Niños Víctimas son fundamentales para los investigadores y los examinadores de la preparación previa a las conclusiones, la presentación de informes, testimonios y el juicio.

El poder de la computación moderna tiene el potencial de permitir a los individuos descargar un programa para provocar colisiones de valor hash MD5.

5. CONCLUSIONES

El uso de los algoritmos HASH ha permitido una mejora en los sistemas de seguridad de autentificación, pero sus recientes debilidades los hacen susceptibles, aunque en menor manera que otros métodos de cifrado. El objetivo de las siguientes versiones debe ser una reducción de que un ataque a estos algoritmos sea efectivo y logre causar grandes daños o efectos devastadores.

6. REFERENCIAS

[1] http://foro.elhacker.net/criptografia/funciones_de_hash-t100025.0.html

[2] http://www.kriptopolis.org/algoritmo-hash-thunderbird

[3] http://www.rosalesuriona.com/spip.php?article466