José Emmanuel Guzmán Sánchez
Criptografía 2020-2

Análisis SHA-1

Secure Hash Algorithm 1 (SHA-1) es un algoritmo que nos permite obtener resúmenes 'únicos' de documentos. Sin importar la longitud del documento o mensaje, siempre y cuando sea menor que 2^{64} bits, la salida siempre será el resumen del mismo con un longitud de 160 bits. Es considerado seguro porque es computacionalmente inviable que dos mensaje distintos produzcan el mismo resumen, es decir, es muy poco probable que existan colisiones.

Para poder realizar el algoritmo se aplica un padding al mensaje de entrada, de manera que la longitud sea múltiplo de 512 y así poder dividirlo en bloques de esta magnitud. Este proceso se realiza agregando un "1" seguido de m "0"s y de un entero de 64 bits al final del mensaje para producir un mensaje rellenado de la longitud deseada. El entero de 64 bits es la longitud del mensaje original. El mensaje rellenado luego es procesado por el SHA-1 como n bloques de 512 bits.

Se utiliza una serie de funciones lógicas ya definidas, las cuales operan sobre 3 palabras (B, C, D) de 32 bits cada una y producen a la salida una palabra de la misma longitud.

```
f(t;B,C,D) = (B AND C) OR ((NOT B) AND D)          ( 0 <= t <= 19)

f(t;B,C,D) = B XOR C XOR D                         (20 <= t <= 39)

f(t;B,C,D) = (B AND C) OR (B AND D) OR (C AND D)   (40 <= t <= 59)

f(t;B,C,D) = B XOR C XOR D                         (60 <= t <= 79).
```

De igual manera, utiliza constantes ya definidas dadas en valores hexadecimales, las cuales guardan una relación con las funciones.

```
K(t) = 5A827999          ( 0 <= t <= 19)

K(t) = 6ED9EBA1          (20 <= t <= 39)

K(t) = 8F1BBCDC          (40 <= t <= 59)

K(t) = CA62C1D6          (60 <= t <= 79).
```

Existen distintos métodos para implementar este algoritmo y todos producen la misma salida dado el mismo mensaje de entrada, sin embargo difieren el uno del otro de acuerdo a las necesidades del sistema. Algunos ocupan más memoria utilizando vectores auxiliares para facilitar algunos cálculos o búsquedas y otros evitan utilizar memoria extra, pero consumen más tiempo computando el proceso.

Conclusiones

Este algoritmo es un poco similar a MD5, en el sentido en que ocupa funciones ya definidas y vectores de inicialización, sin embargo, es más complejo. A pesar de todo esto, sigue siendo susceptible a ataques por fuerza bruta. En la actualidad es un algoritmo que ya no se usa porque está roto, por lo que ahora se considera inseguro. Los principales navegadores empresas tecnológicas importantes; como Microsoft, Google, Apple y Mozilla han dejado de aceptar los certificados SSL SHA-1 desde 2017.

Si se encuentra una debilidad en una función hash que permite que dos archivos tengan el mismo resumen, se considera que la función está rota, ya que las huellas digitales generadas con ella se pueden falsificar y dejan de ser confiables. Con esto, los atacantes podrían crear una actualización de software con malware que sería aceptada y ejecutada por un mecanismo de actualización que valida las actualizaciones mediante la verificación de firmas digitales.

Debido a esto, los algoritmos actuales se enfocan más en aceptar llaves cada vez más largas, para que el costo computacional de romperlos mediante fuerza bruta lo hagan prácticamente inviable. Con todo esto, el desarrollar e implementar este tipo de algoritmos también conllevan un costo ya sea en el rendimiento, tiempo o espacio requerido.