

La muerte de MD5 (y algunos certificados SSL)

CADENA DE CONFIANZA ROTA

Algunos investigadores pretenden comprometer MD5 en un esfuerzo por convencer a la gente para que dejen de usarlo. Explicamos cómo funcionó el ataque y qué significa para nosotros. **POR KURT SEIFRIED**

El algoritmo *Message Digest 5* (MD5 de forma abreviada) es una función hash criptográfica de un solo sentido. En términos simples, toma la entrada, la destroza y genera un valor de 128 bits (expresado normalmente como un número hexadecimal de 32 caracteres, del tipo `76ffd163bd23504cf873a9c027b2ed`). La misma entrada (*password* por ejemplo), siempre generará la misma salida (por ejemplo, `5f4dcc3b5aa765d61d8327deb882cf99`). Así que, ¿Por qué usar MD5? Cuando los datos se firman criptográficamente (como email o certificados SSL), resulta mucho más eficiente suscribir una firma criptográfica de los datos que del bloque completo (128 bits de datos contra un kilobyte o más para un certificado SSL).

Mucha gente usa MD5. Por ejemplo, son muchas las distribuciones que lo utilizan por defecto para hashear valores de contraseñas en el fichero `/etc/shadow`, numerosas autoridades de certificados SSL lo soportan y muchos proveedores de aplica-

ciones lo prefieren a algoritmos fuertes como SHA-1 o SHA-256 (un algoritmo de hasheado funcionalmente similar a MD5).

La Compensación

Como cualquier problema de seguridad, un continuo de selecciones oscila entre una combinación de “barato, fácil, inseguro y computacionalmente económico” a “caro, difícil, seguro y computacionalmente caro”. En el caso de MD5, falla en algún lugar en el medio, no tanto por causa de una selección consciente para economizar, sino en gran parte debido a su edad (fue inventado en 1991).

El mayor defecto de MD5 es su limitado tamaño de hash: 128 bits, significativamente más pequeño que otros algoritmos de hasheado como SHA-1 (160 bits) o SHA-256 (256 bits). Este tamaño limitado permite a un atacante realizar lo que se conoce como “ataque de cumpleaños”. En términos criptográficos, dicho tipo de ataques ocurren cuando dos entradas diferentes (por ejemplo, dos diferentes aunque peticiones de certificados SSL válidamente

formadas) tienen la misma salida después de ser pasadas a través de una función hash como MD5. Porque MD5 sólo tiene 2 elevado a 128 posibles salidas, y existen obviamente muchas más entradas (por ejemplo, 100 caracteres ASCII estándar representan 2 elevado a 800 entradas posibles) [1]. Incluso algo tan simple como una fecha y un número de serie pueden representar fácilmente unas 2 elevado a 128 entradas potenciales.

En realidad, lo único que previene a alguien de realizar un ataque a MD5 es la cantidad de energía computacional necesaria y la resistencia del algoritmo ante varios tipos de ataques.

Desafortunadamente, se encontraron algunas debilidades en MD5 (algunas se remontan a 1993), y su potencia computacional se abarató más rápidamente de lo esperado, incluso teniendo en cuenta la ley de Moore. Así que, armados con unas cuantas debilidades conocidas en MD5, un grupo de investigadores pretenden atacarlo y comprometerlo de tal manera que se demuestre finalmente su debilidad de forma concluyente (esperando convencer a todo el mundo para que deje de usarlo).

Uno de los usos públicos más comunes de MD5 es la firma de certificados SSL; un pequeño grupo de autoridades de certificación (tales como *Thawte*, *RapidSSL*, *RSA* y *VeriSign Japan*) aún usan MD5, haciéndolos vulnerables a este ataque. Entonces sólo hacía falta que un atacante creara dos peticiones de certificado: una petición estándar y legítima para un sitio web seguro, y otra para un certificado con la firma autorizada permitiendo usarla para crear certificados firmados a voluntad.

¿Cómo Funciona el Ataque?

En dos palabras, los investigadores encontraron una autoridad de certificación que expedía certificados de manera que permitía a los atacantes controlar los datos colo-

cados en el certificado por la autoridad de certificación. No es bueno tener dos certificados con firmas MD5 similares cuando la autoridad añade una marca de tiempo y número de serie aleatorio, o lo que es lo mismo, cambia la firma MD5 del certificado. La autoridad de certificaciones vulnerable utilizó números de serie secuenciales (por ejemplo, 1001, 1002, 1003) y marcas de tiempo de exactamente seis segundos más tarde del momento en que el usuario presentó la petición de certificado para su página web.

En este momento, lo único que los investigadores tuvieron que hacer fue encontrar hardware suficientemente barato como para calcular un par de certificados en una cantidad de tiempo razonable.

Por suerte para ellos, la PlayStation contiene un chip procesador especializado denominado *Cell* especialmente adaptado para calcular un ataque cumpleaños, y con la friolera de 200 máquinas (unos 80,000 dólares como precio de venta al por menor), los investigadores pudieron calcular los datos iniciales necesarios como para encontrar un juego de parejas de certificados en 10 horas. Fueron necesarios más cálculos para generar los certificados, que se hicieron en un sistema quad core (o dicho de otro modo, una máquina no muy cara).

Al final los investigadores pudieron llevar a cabo un ataque exitoso que les dio un certificado válido para la firma de otros certificados. Por suerte para todos, como eran buenos chicos, fijaron las fechas de los certificados a 2004, así que ya habían expirado y formulado un aviso cuando los encontraron.

¿Qué Significa para Nosotros?

A pesar de que este ataque requiere una inversión relativamente modesta (de aproximadamente 100,000 dólares en hardware), la sofisticación tecnológica necesaria es bastante elevada. Además, sólo unas cuantas autoridades de certificación se vieron afectadas por el problema, ya que la vasta mayoría dejó de usar MD5 hace algunos años (cuando alguien encuentra una debilidad teórica en un sistema de seguridad, no suele tardar mucho en aparecer un exploit funcional).

Aunque este tipo de ataques es el santo grial de los chicos malos de la web (utilizándolo, se hacen pasar por nuestro banco o tienda online), es poco probable que un

atacante cree y use un certificado con firma para hacerse pasar por sitios web. La razón principal es que hay maneras mucho más sencillas de imitar un sitio web seguro.

Certificados SSL Falsos

Lo malo es que los atacantes pueden conseguir un certificado SSL para un sitio arbitrario de la forma más simple: Sencillamente comprándolo. En un caso, alguien pudo comprar uno para Mozilla.org de un distribuidor SSL que no verificó que el personal tenía permitida la obtención de certificados, ni siquiera si era afiliado a Mozilla.org.

En otros casos, los atacantes han podido obtener un certificado "Issued in minutes" ("Expedido en minutos") (a citar, *RapidSSL.com*) con peticiones falsas, faxando pedidos con membretes con la apariencia oficial de organizaciones para las que se expedían los certificados SSL. Básicamente, reclaman verificar nuestra información de forma segura en unos cuantos minutos (en realidad, una simple petición de información WHOIS para nuestro dominio y contactos de correo listados, dándoles la oportunidad de cancelar la orden de certificado). Con autoridades de certificados *CCL* vendiendo certificados a, virtualmente, cualquiera que los solicite, con un descuido mínimo, el sistema puede ser víctima del abuso de los atacantes fácilmente.

Cómo Protegernos

Por desgracia, no se puede hacer mucho por nuestra protección. Incluso habilitando la comprobación de la revocación de nuestro certificado en nuestro navegador web, si un atacante usa el método MD5 para crear un certificado de autoridad falso, puede simplemente omitir la información de revocación del certificado (¡lo que significa que nuestro navegador no puede comprobar si ha sido revocado o no!). Deshabilitando los certificados raíz de las autorizaciones que aún soportan MD5 se descifraría un gran número de sitios web, algunos de los cuales queremos usar (lo cual es en gran parte porque, hasta ahora, Firefox no ha bloqueado el uso de la autorización Comodo SSL).

Si quisiéramos hacerlo nosotros mismos, las instrucciones serían: Ir a la pestaña de opciones *Advanced | Encryption* en Firefox y pulsar en *View Certificates*, seleccionando luego *Authorities*. Seguidamente buscamos el certificado que deseamos des-

habilitar (manualmente, porque no existe función de búsqueda) y lo seleccionamos. Por último, sólo resta seleccionar *Edit* y desmarcar el cuadro *This certificate can identify web sites*.

En el futuro, los sitios web que usan esta autoridad de certificación para conseguir sus certificados del sitio web se presentarán como no firmados mediante una autorización de confianza, y conseguiremos que Firefox avise. El proceso es igual de complicado, o más, para otros navegadores web. Y para más inri, la mayoría de las autoridades de certificación en nuestro navegador web no poseen información descriptiva. Algunos no tienen ni siquiera sitios web válidos (porque han salido del negocio y vendido sus certificados firmados a otras compañías). Los navegadores web podrían hacer importantes mejoras en este área.

Conclusión

Lo bueno de todo esto es que las organizaciones responsables de los estándares técnicos tales como MD5 y SHA-1 no se han relajado. La NIST (*American National Institute of Standards and Technology*) ha organizado un concurso para desarrollar un nuevo algoritmo de hash lo suficientemente bueno como para usarlo durante algunas décadas de uso [3]. Los proveedores de navegadores web tampoco se quedan de brazos cruzados. La llegada del nuevo *Extended Verification* certifica lugares de control mucho más estrictos sobre cómo se emiten los certificados y a quién (aunque uno añadiría que esto debería haberse hecho siempre).

Por desgracia, sin educación, la mayoría de los usuarios no tendrán posibilidad de distinguir entre un sitio web con un certificado "estándar" y otro con *Extended Verification*, aunque en la mayoría de los casos, el navegador coloca el nombre de la compañía sobre un fondo verde en la barra de direcciones. ■

RECURSOS

- [1] Creando un certificado CA malicioso: <http://www.phreedom.org/research/rogue-ca/>
- [2] mozilla.dev.tech.crypto: http://groups.google.com/group/mozilla.dev.tech.crypto/browse_thread/thread/9c0cc829204487bf?pli=1
- [3] Competición de algoritmo hash criptográfico: <http://csrc.nist.gov/groups/ST/hash/sha-3/index.html>