HUECOS DE SEGURIDAD EN ALGORITMOS SSL (HEARTBLEED)

Quintero, Francisco. Díaz, Jhon.
ingfcoquintero@hotmail.com, jhonhdp@hotmail.com
Universidad Piloto de Colombia
Bogotá, Colombia

Abstract— The open source SSL version (OpenSSL) uses a software library to encrypt communications between the browser and the server that hosts the service that is focusing. It affects versions from 1.0.1 to 1.0.1f, where the attacker through a development obtained confidential information of its users. This vulnerability is exploited by sending a malformed heartbeat request in order to obtain the response of the victim, allowing attackers to view sensitive data, which puts at risk the confidentiality of the victim. This affects the sites that use those SSL versions and whose final aim is the consumer of that service.

Although vulnerability can be corrected from the vendor who provides the service, the end user must be aware of the impact of this kind of security holes and follow some recommendations described in this article.

Index Terms—Heartbeat, Heartbleed, HTTPS, Security Holes, OpenSSL, Computer Security, SSL

Resumen—La versión SSL de código abierto (OpenSSL) utiliza una librería de software para cifrar las comunicaciones entre el navegador y el servidor que aloja el servicio que se está prestando. Afecta las versiones 1.0.1 y 1.0.1f, donde el atacante mediante un desarrollo obtiene información confidencial de sus usuarios. Esta vulnerabilidad se explota mediante el envío de una solicitud de un *heartbeat* mal formado con el fin de obtener la respuesta de la víctima, permitiendo a los atacantes leer datos sensibles, lo cual pone en peligro la confidencialidad de la víctima. Esto afecta a los sitios que utilizan esas versiones de SSL y cuyo objetivo final es el consumidor de ese servicio. Aunque la vulnerabilidad puede ser corregida desde el proveedor quien presta el servicio, el usuario debe ser consciente del impacto de esta clase de huecos de seguridad y seguir algunas recomendaciones descritas en este artículo.

Índice de Términos—Heartbeat, Heartbleed, HTTPS, Huecos de seguridad, OpenSSL, Seguridad Informática, SSL.

I. INTRODUCCIÓN

La información es uno de los activos más valiosos de toda organización, grupo o persona. Para proteger este recurso se han implementado a través del tiempo medidas de control para restringir el acceso a los datos que se consideran importantes. A la par de quienes implementan métodos para garantizar mayores niveles de seguridad existen personas que dedican parte de su tiempo y esfuerzo a encontrar maneras de llegar a esos datos y sacar provecho, casi siempre de manera indebida o ilegal.

En el medio se encuentran herramientas como SSL que buscan que las transacciones de datos que se efectúan en internet no puedan ser violentadas para obtener la información que allí viaja. Una de las versiones de SSL más difundidas y usadas en el mundo es la creada a partir de un proyecto colaborativo llamado OpenSSL, en la cual recientemente se descubrió un error dentro de su código que permite que un atacante recopile los datos que el servidor está procesando en ese momento y cuyo contenido pueden ser números de cuentas bancarias, contraseñas u otra información sensible.

En este artículo se pretende mostrar cuál es la vulnerabilidad antes mencionada, cómo funciona y los aspectos más relevantes a tener en cuenta para corregir dicho error y reducir su impacto.

II. SSL Y HEARTBLEED

SSL (*Secure Sockets Layer* o Capa de conexión segura) es un protocolo criptográfico que permite mantener un nivel de seguridad de las comunicaciones que se efectúan en una red de

datos, dentro de las cuales se encuentra Internet. Este protocolo es en esencia un conjunto de reglas establecidas para emplear técnicas de criptografía sobre un mensaje definido.

El uso de criptografía busca que dicho mensaje, que viaja desde un emisor a un destinatario específico, se encuentre cifrado o codificado con el fin de que su contenido sea incoherente si éste es interceptado por un receptor que no está autorizado. Lo anterior permite que la información sea confidencial al garantizar su acceso solo al personal autorizado, que sea íntegra al poder corregir y completar los datos, que el mensaje pueda ser asociado o vinculado a las personas o sistemas que participen en la comunicación y que se pueda verificar la identidad del comunicador [1].

SSL se utiliza inicialmente para verificar la identidad de un servidor mediante un sistema de certificado digital. Estos servidores emplean por lo general conexiones HTTPS (*Hypertext Transfer Protocol Secure* o Protocolo seguro de transferencia de hipertexto) sobre las cuales se transfiere información sensible como datos personales, usuarios o contraseñas de aplicaciones que están disponibles en una red. Las tiendas online, sitios web bancarios o redes sociales son ejemplos de sistemas que se valen de SSL para generar transacciones seguras.

Dentro de un programa, compuesto por una considerable cantidad de líneas de código escrito por seres humanos, puede ocurrir que una parte se encuentre mal diseñada o que presente un grado de complejidad tan alto que sea difícil predecir su comportamiento en todos los escenarios. Si este fragmento no altera la funcionalidad del programa es posible que sea detectado y corregido mucho tiempo después de ser implementado, ya sea por desconocimiento del error o por falta de recursos [2]. En ese momento se dice que el sistema presenta una vulnerabilidad o hueco de seguridad que, de ser detectado por alguien malintencionado, sería aprovechado para sustraer información, realizar fraudes, instalar programas dañinos o destruir el sistema.

Existe en la actualidad una versión libre de SSL denominada OpenSSL, que es utilizada por una gran cantidad de sitios web para transferencia de datos de manera segura. Cuenta con una funcionalidad llamada *Heartbeat* (latido de corazón) introducida en diciembre de 2011 y adoptada para su uso en marzo de 2012 [3]. El objetivo de *Heartbeat* es determinar si el servidor aún está activo solicitándole la devolución de un mensaje enviado por el cliente. Junto con este se incluye una etiqueta que le indica al servidor cuál es la longitud del mensaje.

Un ejemplo de lo anterior es suponer que el cliente envía al servidor un mensaje con la palabra “papaya” y una etiqueta con la longitud (6 letras). Si el servidor se encuentra activo le retornará al cliente el mismo mensaje, es decir, la palabra “papaya”. Ahora se planteará un escenario ligeramente diferente: el cliente de nuevo envía al servidor el mensaje con la palabra “papaya” pero en esta ocasión la etiqueta de la longitud ha sido modificada, pasando de 6 a 500 letras. El servidor se encuentra activo y retorna el mensaje con la palabra “papaya” más 494 letras adicionales.

Estas letras adicionales en la vida real corresponden a porciones aleatorias de la memoria del servidor y pueden contener información sensible. No se puede elegir el fragmento de memoria que el servidor retornará, pero al intentarlo múltiples veces se pueden llegar a obtener datos importantes [4]. Este uso indebido de la funcionalidad *Heartbeat* es llamada *Heartbleed* (hemorragia de corazón) y fue reportada como una vulnerabilidad de OpenSSL en abril de 2014 [3].

Los puntos más destacados de esta vulnerabilidad o *bug* son los siguientes [5]:

- Lleva un poco más de dos años dentro del código de OpenSSL. Las versiones inseguras van desde la 1.0.1 a la 1.0.1f. La versión 1.0.1g contiene la corrección del error.

- Debido a su amplio uso pudo haber afectado a una gran cantidad de servidores y aplicaciones, tales

como Amazon, Google, Yahoo, Facebook o Twitter. Ya que el admitir este tipo de fallos puede afectar seriamente la reputación de una empresa de esta envergadura es posible que la vulnerabilidad no sea reconocida abiertamente. También puede ocurrir que los responsables de los servidores desconozcan la existencia del fallo y que esto conlleve a evitar alertar a los usuarios finales.

• *Heartbleed* permite obtener fragmentos de memoria del servidor en bloques de hasta 64 Kb. Con la información que se obtiene allí es posible suplantar el servidor en cuestión o extraer datos que en ese momento estén siendo procesados. Un hecho importante es que esta captura indebida de información, si se desea, no deja rastros.

• El fallo puede afectar a otros dispositivos tales como teléfonos o televisores inteligentes (en el caso de los smartphones se ha detectado este hueco de seguridad en aquellos que emplean el sistema operativo Android 4.1.1 o *Jelly Bean*), enrutadores u otros similares que tengan acceso a Internet y que deban realizar transacciones de modo seguro. Adicionalmente se evidenció su uso en la evasión de la seguridad de las Redes Privadas Virtuales (o VPN por sus siglas en inglés), que son ampliamente utilizadas en el entorno empresarial para restringir el acceso a las redes corporativas.

• Aunque el error se atribuye a una omisión humana, se comenta en Internet que el bug pudo ser deliberadamente puesto allí para fines de espionaje, lo cual apunta a escándalos tales como el de la Agencia Nacional de Seguridad Estadounidense (NSA).

• A pesar de que muchos sitios web desactivaron el fallo o implementaron acciones correctivas en el término de días después de ser publicado, no se conoce con certeza desde cuándo los atacantes potenciales conocían sobre esta vulnerabilidad y cuánto provecho sacaron de la misma [6].

III. Recomendaciones para remover el error

Antes de aplicar una solución para este hueco de seguridad se debe analizar en cuál de estos tres frentes se encuentra la persona afectada: si es usuario de un sitio seguro (HTTPS), si es un proveedor de un sitio seguro o servidor o si se tiene un dispositivo que puede estar vulnerado.

En el primer escenario lo recomendable es primero verificar si la página a la cual se está accediendo es vulnerable. Páginas como http://diagnostico.seguridad.unam.mx/, http://tif.mcafee.com/heartbleedtest o https://lastpass.com/heartbleed/ pueden ser de ayuda en este paso. Si el resultado de la validación es que el sitio web es vulnerable no se recomienda conectarse a dicha página, pues los datos de usuario y contraseña que se ingresan para acceder muy posiblemente serán robados. Si ocurre el caso contrario, es decir que la página no es vulnerable, lo mejor es realizar el cambio de contraseñas lo más pronto posible. En este punto es importante destacar que los cambios de esta clase de datos deben hacerse directamente en el sitio web y no mediante enlaces que puedan llegar al correo electrónico, debido a que podrían estar direccionados a páginas web falsas que se usan para, de igual manera, robar la contraseña ingresada. Esta clase de acciones se denominan ataques de *phishing* [5].

Otra recomendación importante es no almacenar las contraseñas en los navegadores web que se empleen en el día a día y seguir sugerencias de creación de *passwords* fuertes en las que se incluyen el uso de mayúsculas y minúsculas, números y caracteres especiales. Adicionalmente, si el servidor lo permite, se sugiere utilizar mecanismos de seguridad extra para acceder al mismo servicio como por ejemplo ingresar el usuario, contraseña y un código de autenticación enviado a través de un mensaje de texto al teléfono celular.

Para el segundo escenario (cuando se tiene un servidor que provee conexiones seguras por HTTPS) se debe realizar lo siguiente:

• Identificar la versión de OpenSSL con el que el

servidor cuenta. Se debe recordar que las versiones vulnerables van desde la 1.0.1 a la 1.0.1f.

• Revocar los certificados digitales del servidor, cuya función principal es la de proporcionarle una identificación digital ante la red.

• Instalar una versión de OpenSSL ya corregida.

• Solicitar e instalar un nuevo grupo de certificados digitales.

• Solicitar a los clientes o usuarios que acceden al servidor el cambio de sus contraseñas.

Por último se encuentra el escenario en el que se posee un dispositivo que puede estar afectado. Este es el caso más complejo, pues se depende del proveedor de dicho hardware para saber si es vulnerable y también de la cadena de empresas (desde el fabricante hasta quien lo integra en el dispositivo e incluso el mismo proveedor) para que se realice una actualización del software. La sugerencia es mantenerse informado sobre actualizaciones que sean publicadas y seguir las recomendaciones para su instalación. En el ámbito empresarial aparte de lo anterior se recomienda una coordinación constante con el proveedor del dispositivo para aplicar los correctivos necesarios.

IV. Conclusiones

Dado el gran impacto que *Heartbleed* ha causado en miles de sitios web seguros que emplean OpenSSL, los ojos del mundo han recaído sobre este proyecto colaborativo, dentro del cual se argumenta que si bien esta clase de errores no son malintencionados se pueden deber a la poca financiación o estímulo que reciben los desarrolladores, lo a su vez que conlleva a tener poca disposición de tiempo y recursos para revisar exhaustivamente los programas y detectar vulnerabilidades como esta. Por esta razón y dado que muchas grandes empresas utilizan programas de código abierto tipo OpenSSL se realizó un acuerdo para donar sumas considerables de dinero con el fin de crear un grupo que financie las mejoras en estas iniciativas de software [6].

Tarde o temprano un error en el código fuente de un programa determinado será detectado. No obstante, si no se presta la atención debida a esta clase de incidentes cabe una alta probabilidad de que personas malintencionadas hayan detectado el problema y mantengan esa información como un "arma secreta" para robar información, ingresar de manera irregular o incrustar software malicioso para generar un bien individual o colectivo.

Heartbleed no será la primera ni la última vulnerabilidad que se descubra. En la red la noción de lo que hoy se considera seguro y confiable puede cambiar el día mañana de manera inesperada. Para ello se debe llevar un control de las vulnerabilidades existentes y estar atento a revisar hasta qué punto se registra afectación. Esto también conlleva a que los usuarios deben replantear algunas prácticas comunes con el manejo de los datos sensibles o seguros, para que constantemente éstos sean reforzados.

Finalmente, no hay que ocultar que toda red tiene peligros y problemas tal como los tienen todos los sistemas y herramientas desarrolladas por el hombre. Lo importante es continuar con la labor de descubrirlos, aislarlos y corregirlos, así como hacer responsables a quienes los conocen, explotan y ocultan con deliberación.

Referencias

[1] Wikipedia. (2014, Jul 6). Criptografía [Online]. Available: http://es.wikipedia.org/wiki/Criptografía

[2] F. Ferri-Benedetti. (2013, Ene 13). ¿Qué es un agujero de seguridad? [Online]. Available: http://articulos.softonic.com/que-es-una-vulnerabilidad

[3] Codenomicon. (2014, Abr 29). Heartbleed Bug [Online]. Available: http://heartbleed.com

[4] M. Nordhoff. (2014, Abr 8). How exactly does the OpenSSL TLS heartbeat (Heartbleed) exploit work? [Online]. Available: http://security.stackexchange.com/questions/55116/how-exactly-does-the-openssl-tls-heartbeat-heartbleed-exploit-work/55117#55117

[5] F Acero. (2014, Abr 22). Algunas recomendaciones críticas sobre el fallo de OpenSSL (Heartbleed) [Online].

Available: http://www.kriptopolis.com/recomendaciones-heartbleed

[6] Wikipedia. (2014, Jul 6). Heartbleed [Online]. Available: http://es.wikipedia.org/wiki/Heartbleed