EVALUACIÓN DESDE LA ÓPTICA DE LA COMPUTACIÓN FORENSE DEL BUG OPENSSL - HEARTBLEED

Evaluation from the perspective of forensic computing of the bug OpenSSL - Heartbleed

JOHN ALEXANDER RICO FRANCO*

Recibido: 1 de abril de 2015. Aceptado: 11 de junio de 2015

Resumen

En abril de 2014, se desvelo por parte de la comunidad especialista en seguridad informática, un fallo que afectaba a una de las librerías criptográficas primordiales en la protección de datos en Internet tal como lo es OpenSSL y debido a la popularidad de esta librería se estipula que este fallo conocido como Heartbleed, llego posiblemente a afectar al 66% de servidores de servicios web implementados en toda la Internet, esto debido al hecho de la implementación implícita de OpenSSL en aplicaciones libres tan importantes y que manejan datos sensibles de los usuarios, tales como lo son Apache, Dropbox, Gmail, Minecraft, MySQL, OpenVPN y Ubuntu[1].

El presente documento busca presentar los conceptos y conclusiones que han sido resultado de un proyecto de investigación basado en estudiar algunos de los más interesantes aspectos concernientes a la falencia Heartbleed, desde su identificación, características, técnicas de aprovechamiento por parte de los delincuentes informáticos y medidas prácticas de mitigación; buscando así lograr dar una visión panorámica de esta catastrófica vulnerabilidad, que aunque esta ha sido ampliamente discutida y documentada en toda Europa, Asia y Estados Unidos, cabe resaltar que en Colombia y en varios países de sur y centroamérica esta alerta paso bastante desapercibida por la gran mayoría de los usuarios de dichos servicios comprometidos; por ende otra finalidad de este documento es aparte de presentar al bug Heartbleed es el de generar conciencia en cualquier profesional de la ingeniería de sistemas y a cualquier lector interesado, de que siempre existe la posibilidad de la existencia de otros fallos de igual o mayor magnitud sobre la protección de información en la Internet, puesto que una de las principales ventajas que tienen los delincuentes es la desinformación por parte de las personas sin altos conocimientos en sistemas y que utilizan la totalidad de los beneficios aportados por los distintos servicios desplegados en la Internet y que por ende muy posiblemente fueron víctimas directas en este desafortunado incidente computacional.

Palabras clave: extensión Hearbeat, fallo heartbleed, OpenSSL, kali linux, nmap, metasploit, seguridad informatica, servidores web, robo de información.

Abstract

In April of 2014 the community of computer security specialists centralized on the study of vulnerabilities over the Internet find out a catastrophic bug in OpenSSL library, this is one of the most used cryptographic libraries for the data protection over TCP/IP networks; this bug was called Heartbleed and it's estimated that error harm over the 66% of the live web servers implemented in the Internet, adove the active time of the Heartbleed, and all this happened because the implicit use of the OpenSSL over critical applications like Apache, Dropbox, Gmail, Minecraft, MySQL, OpenVPN, Ubuntu and YouTube.

* Ingeniero de Sistemas - Especialista en Seguridad de Redes de la Universidad Católica de Colombia, con más de 7 años de experiencia como consultor independiente en proyectos referentes a temas de seguridad informática, criptografía y realización de pruebas de calidad de software. Catedrático Universitario y Docente Investigador del Grupo de Investigación y Desarrollo de Ingeniería de Sistemas (G.I.D.I.S) de la Corporación Universitaria Republicana. Correo electrónico: johnricof@gmail.com

Advertencia: La información exhibida en el presente artículo es presenta da exclusivamente con fines educativos y preventivos; por ende el investigador y la institución educativa que este representa se deslindan de cualquier responsabilidad ligada al hecho de que el lector de este documento utilice de manera inapropiada o delictiva la información y/o herramientas a continuación descritas.

The following paper intends to exhibit the concepts and conclusions which are the result of a research project of the most interesting aspects concerning the Heartbleed bug, this study covers the identification of the error, their characteristics, harvesting techniques used by the hackers in the active time of the bug and some mitigation measures; all this aims to generate a local overview of this major failure because in Europe, Asia and the United States this flaw was highly documented and socialized but in Colombia and several other countries in Central and South America this warning went unnoticed by the vast majority of users of those influenced web services and it´s because of this scene I don't want only submit to the facts of the Heartbleed bug, I intend revive the discussion of that sensitive issue and raise awareness of the reader about the possibility of the existence of other problems of equal or greater magnitude in the protection of sensitive data in networks TCP/IP and so reduce one of the greatest tools in the arsenal of the informatic offenders which is the misinformation and ignorance on these sensitive issues by millions of potential victims all over the Internet.

Key words: hHearbeat extension, heartbleed bug, OpenSSL, kali linux, nmap, metasploit, information security, web servers, loss of information.

I. Introducción

A inicios del 2014, se detecto por parte de un sector de investigación dedicado a profundizar sobre temas de seguridad informática, una vulnerabilidad de alto impacto en la librería criptográfica libre OpenSSL[2] y que generaba un abismo de seguridad al ser implementada de manera nativa con cualquier servidor web que utilizara dicha librería para la protección de datos compartidos en un ambiente tan inseguro como lo es Internet; pero esta falla llego a ser tan desastrosa fue por el hecho de afectar de manera directa a cualquier implementación web basada en el reconocido servidor Apache[3]. Esta falla de seguridad en la protección de información sensible salvaguardada bajo la librería OpenSSL fue llamada Heartbleed y es considerada por la comunidad de expertos en seguridad informática como una de las vulnerabilidades mas funestas en la protección de datos en Internet, no solo por su alta propagación debido al hecho de la confianza absoluta que se tenía sobre OpenSSL al momento de concebir procesos de comunicaciones seguras entre clientes y servidores web, por lo cual hacia que muchos sitios o servicios web reconocidos fundamentaran de manera incondicional su proceso de protección de información bajo esta librería; sino que también era bastante sencillo el aprovechamiento de esta vulnerabilidad0 por parte de los delincuentes informáticos conocedores de esta.

Y aunque en la historia de OpenSSL se han presentado varios incidentes de seguridad significativos, el bug Heartbleed ha sido el que más repercusiones ha representado; esto debido al hecho de que esta falencia le permitía a un delincuente informático poder extraer y leer en texto-plano datos sensibles almacenados en la memoria del servidor web transgredido, en los cuales se podía encontrar nombres de usuario, contraseñas, llaves criptográficas descartadas o utilizadas previamente entre otros datos ambicionados por este tipo de personajes al margen de la ley.

Ya en la actualidad, la gran mayoría de empresas o servicios basados en Internet que se vieron afectados por dicho bug, han realizado las tareas apropiadas para su detección y eliminación, tales como la actualización de sus servidores web y por ende adoptar de manera implícita versiones corregidas de OpenSSL; hecho el cual no implicaba que la información compartida con sus clientes fuera automáticamente resguardada, esto debido a que en la ventana de tiempo entre la salida de la versión de OpenSSL vulnerable hasta la detección del fallo, que fue de casi dos años, los clientes y el servidor afectado pudieron compartir información sensible que pudo ser comprometida por parte de un atacante y que muy probablemente se debía actualizar por parte del cliente sus datos sensibles de tipo acceso a servicios web como lo son los logins y passwords, puesto que aunque el canal de comunicación ya se encentraba protegido, no se sabe cuanta información se llego a filtrar y a recolectar por parte de dichos delincuentes en la ventana activa del bug y que muy posiblemente estos datos aun se encuentren vigentes y utilizados por los delincuentes para realizar sus actos delictivos.

Pero se ha presentado una brecha de interés que se quisiera mitigar con la realización de este documento y es el hecho de que en Estados Unidos, Europa y demás países del « primer mundo», se han realizado grandes campañas para dar información sobre la existencia de la falla Heartbleed, sus consecuencias y medidas de mitigación para personas con niveles intermedio a básico en el uso de dichos servicios en línea (como lo es la gran mayoría de dichos usuarios),

pero en países latinoamericanos esta alarma paso casi desapercibida por no decir invisiblemente, hecho el cual preocupa de manera alarmante puesto que en la actualidad nosotros, el pueblo latino somos grandes consumidores de dichos servicios y no estábamos totalmente consientes de que nuestros datos sensibles compartidos en las redes sociales, entidades bancarias en línea y otras compañías con presencia activa en internet, puedan estar siendo aprovechadas de manera delictiva por parte de terceros malintencionados. Por ende la intención del presente artículo es poder presentar a cualquier profesional en seguridad informática o cualquier lector interesado una exposición de los hechos que llevaron a la materialización de ataques de robo de información bajo el aprovechamiento del la falencia Heartbleed, cuáles fueron los servidores web, productos y demás entes que se vieron afectados por dicha vulnerabilidad, como los delincuentes ejecutaban sus ataques al aprovechándose del bug presente en la extensión Heartbeat de OpenSSL y algunas medidas de mitigación utilizadas en su momento por las grandes compañías afectadas; todo esto en búsqueda de generar conciencia en el lector sobre el hecho de que no existe un concepto de seguridad total en ningún ambiente computacional (por mas mecanismos de protección implementados) y que aunque las vulnerabilidades de estos no se han visto documentadas y aprovechadas inmediatamente a la luz pública, no quiere decir que estas no existan y que no estén siendo aprovechadas por los delincuentes informáticos actualmente.

II. Contexto

En abril 07 de 2014, los encargados del proyecto OpenSSL informaron a la opinión publica el descubrimiento de una vulnerabilidad de alto nivel, a la cual llamaron Heartbleed debido a que era un fallo de programación presente en la extensión Heartbeat en su librería principal de cifrado de datos; esta falla le permitía a cualquier persona poder descargar secciones de la memoria del servidor web comprometido, permitiéndole al delincuente poder filtrar información sensible que se encontrara almacenada en esta. En la siguiente sección se buscara presentar que es la librería OpenSSL y el funcionamiento adecuado de la extensión Heartbeat [1].

2.1. Librería OpenSSL:

La librería OpenSSL radica en un conjunto de aplicativos de administración y bibliotecas de implementación de procesos criptográficos, por ende permite aplicar protocolos de cifrado tales como SSL (Secure Sockets Layer) y TLS (Transport Layer Security) detallados en los RFCs 6101 y 5246 respectivamente; esta es una librería de software abierto escrita en el lenguaje de programación C. Su desarrollo es completamente impulsado por el trabajo de desarrolladores voluntarios y es libre para ser utilizada en ambientes tanto comerciales como libres bajo las licencias de uso OpenSSL y SSLeay, lo cual permite que esta se encuentre disponible para una gran mayoría de sistemas operativos libres basados en Unix, de los cuales se incluyen los sistemas Linux, Mac OS, Solaris, entre otros [2, 3].

2.2. Extensión Heartbeat:

El protocolo Heartbeat es utilizado en la librería OpenSSL para negociar y monitorear la disponibilidad de un recurso web; su funcionamiento básico consta en generar una señal periódica entre clientes y servidores para evaluar su normal comportamiento en una transmisión de datos o simplemente para sincronizar otros aspectos de comunicación entre estos; esencialmente este proceso inicia con en el envío de señales de tipo Heartbeat en intervalos de tiempo entre los equipos actores del proceso de comunicación de datos, si se recibe la señal de manera idónea, el proceso de comunicación sigue activo pero si no se tiene ninguna respuesta por parte del equipo receptor o esta es errónea, la comunicación se finaliza debido a la posibilidad de que este equipo no se encuentre disponible o muy posiblemente esté comprometido para continuar la comunicación segura de datos. Esta extensión fue presentada en febrero de 2012 bajo el RFC 6520 y adicionada a la librería OpenSSL en diciembre 31 de 2011 y lanzada con el bug al público en la versión OpenSSL 1.0.1 en marzo 14 del 2012 [4].

Ya profundizando, la extensión Heartbeat consta del manejo de dos tipos de mensajes básicos:

- HeartbeatRequest.
- HeartbeatResponse

Los mensajes HeartbeatRequest y Heartbeat Response consisten en un campo de un byte para el tipo de mensaje Heartbeat implementado, otro campo de dos bytes que almacena la longitud de los datos a extraer, posteriormente hay otro campo donde se empaquetara los datos legítimamente requeridos y por ultimo un campo de 16 bytes donde se almacenan datos de tipo relleno provenientes de la memoria del servidor; la estructura básica de estos mensajes Heartbeat es la siguiente [4]:

```
Struct    {
HeartbeatMessageType type;
uint16 payload_length;
opaque payload[HeartbeatMessage.payload_
length];
opaque padding[padding_length];
}HeartbeatMessage;
```

En donde:

- *HeartbeatMessageType:* Define el tipo de mensaje implementado, puede ser Heartbeat_ Request o Heartbeat_Response.

- *Payload_Length:* Indicador del número total de caracteres a recopilar.

- *Payload:* Sector de almacenamiento donde se recopilara la información valida solicitada.

- *Padding:* Variable de captura de datos de relleno para completar el mensaje Heartbeat. Estos datos son extraídos directamente del buffer memoria del equipo emisor y este contenido debe ser, en condiciones ideales, ignorado por parte del receptor del mensaje.

  Ya en la implementación del servicio de evaluación de la comunicación de datos, el protocolo funciona según la respuesta del receptor de los mensajes de tipo Heartbeat [5]:

- Respuesta esperada: El inicializador del protocolo envía un mensaje de tipo HeartbeatRequest hacia el receptor en espera de que este responda con un mensaje HeartbeatResponse; si el destinatario envía la respuesta bajo los parámetros del emisor, se ha logrado ejecutar de manera apropiada el proceso de Heartbeat, permitiendo que siga la comunicación constante entre las dos partes, esta funcionalidad es llamada « keep-alive» puesto que permite mantener la conexión en constante evaluación de las partes implicadas.

- Sin respuesta o respuesta errónea: Uno de los actores del proceso de comunicación envía un Heartbeat Request hacia el otro ente, buscando que este mande en respuesta un mensaje HeartbeatResponse. Si el ente inicializador no recibe una respuesta por parte de su contraparte o la respuesta no es apropiada, el emisor vuelve a enviar el mensaje HeartbeatRequest y si después de que se han enviado un número determinado de mensajes de solicitud de evaluación de conexión y el receptor no responde, se finaliza la comunicación de datos puesto que muy posiblemente el receptor ya no se encuentra disponible o se encuentra comprometido.

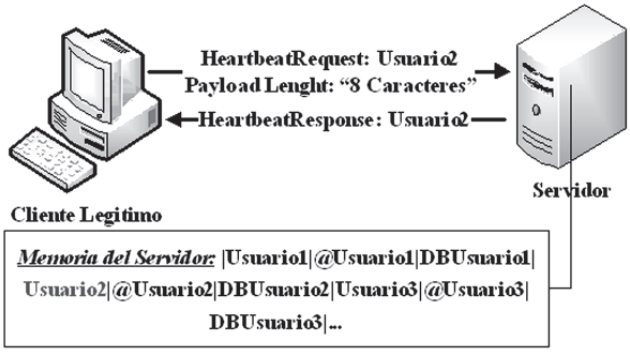


Figura 1. Funcionamiento apropiado de la extensión Heartbeat.

III. Fallo Heartbleed

Como ya se pudo apreciar en la sección anterior, el protocolo Heartbeat es muy práctico al momento de realizar verificaciones de procesos de comunicación de datos en ambientes protegidos; pero existía una falla en el momento de imple-mentar este protocolo, esta radicaba en el hecho de que al momento de gestionar cualquier mensaje de tipo HeartbeatResponse para verificar si la transmisión de datos segura aun se encuentra activa (keep-alive), un usuario malintencionado podía modificar el valor almacenado campo de Payload_ Length, permitiéndole así al atacante poder solici-

tarle al servidor desprotegido una cantidad mínima de información habilitada y extraer una cantidad máxima de datos aleatorios del buffer de datos. Esta vulnerabilidad se fundamentaba en el hecho de que el protocolo Heartbeat incrustado en la librería OpenSSL, no hacia una verificación para poder evaluar si la cantidad de datos solicitados por el HeartbeatResponse correspondía a el valor real de los datos que eran enviados por el servidor, esto quiere decir que se podía inducir al error por parte del servidor aprovechándose de una incongruencia al momento de gestionar los mensajes Heartbeat, ya que aparte de solicitar la cantidad de datos validos a extraer del servidor, también se solicita indicar cuantos son los caracteres de los que consta dicha información; así que si la cantidad de datos solicitados legítimamente al servidor era por ejemplo de 8 caracteres pero se indicaba en el mensaje Heartbeat que era de 16 caracteres, el protocolo lo que hacía era verificar de cuantos caracteres era la información exigida (que para nuestro ejemplo son 16) y gestionaba el mensaje Heartbeat con los 8 caracteres propios de la información solicitada y pasaba a saturar al campo Payload con los siguientes 8 caracteres almacenados en la memoria del servidor, para así lograr cumplir con la cuota de datos indicados en Payload_Length; todo esto permitía a un delincuente capturar de manera ilícita toda la información sensible que pudiera estar alojada en la memoria del servidor al momento del aprovechamiento de esta vulnerabilidad [5, 6].

Este bug le permitía a cualquier atacante informático poder extraer 64kb de información aleatoria almacenada en el buffer de memoria del servidor web vulnerable en formato de texto-plano, así que existía la posibilidad de filtración de datos sensibles de los usuarios que interactuaran con dicho servidor inseguro, tales como passwords o nombres de usuario y en el peor escenario posible, el delincuente podía encontrar datos como llaves privadas de encriptación, certificados digitales o otra información utilizada por el servidor para proteger su interacción con sus clientes y/o con otros servidores; pero se debe tener en cuenta que el atacante no podía seleccionar los datos robados al servidor, ya que solamente se podía acceder a la información almacenada en el buffer de memoria del equipo expuesto en un momento particular, así que la estrategia aplicada por estos era el generar la mayor cantidad de mensajes Heartbeat modificados y recolectar toda la información posible para su posterior análisis en búsqueda de datos de alto interés.

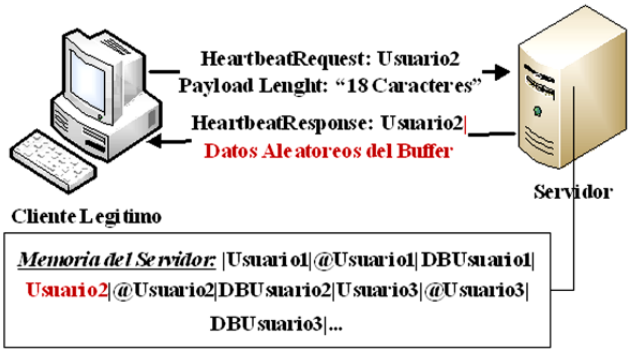


Figura 2. Aprovechamiento de la vulnerabilidad Heartbleed.

3.1 Dispersión del Bug Heartbleed:

Desde un inicio, la vulnerabilidad Heartbleed únicamente afecto a las versiones OpenSSL 1.0.1 a la 1.0.1f, las cuales estuvieron disponibles para todo el mundo y activas en implementaciones web de alto impacto entre el inicio de 2012 hasta principios del 2014. Pero este fallo no fue exclusivo en ambientes web clásicos de tipo cliente servidor, también se vieron afectados diferentes productos tales como muchas distribuciones de sistemas operativos tipo Linux, elementos de red y dispositivos móviles que utilizaran la librería OpenSSL para la protección de sus datos y que fueron lanzados en esa ventana de inseguridad de dos años. Este error fue detectado y solucionado a partir de la versión de OpenSSL 1.0.1g que salió al público en abril de 2014 [7].

3.1.1. Servidores y servicios comprometidos:

Tal como se expuso anteriormente, no solamente fueron afectados los servidores web que aplicaban como mecanismo de seguridad a la librería OpenSSL, sino que también varios servidores de correo, de bases de datos, entre otros se vieron vulnerados; a continuación se presentan algunos de los servidores que de manera colateral eran atacados por los delincuentes informáticos aprovechándose del bug Heartbleed:

- Apache.
- Cliente de Bitcoin.
- Cyrus.
- Google GWS.
- MySQL.
- Nginx.

- OpenLDAP.
- OpenVPN.
- Postfix.
- PostgreSQL.
- Qmail.
- Sendmail.
- Stunnel.
- Tomcat.
- Zimbra.

Y en cuanto a los servicios web que fueron perturbados y que informaron sobre este inconveniente, los más relevantes fueron [8]:

- Dropbox.

- Facebook.

  Google: El equipo técnico de Google apenas se detecto y divulgo la existencia del bug Heartbleed, realizo los procesos típicos de parcheo de sus servidores web estratégicos como primera medida de mitigación, logrando así fortificar a servicios tales como YouTube y Gmail; posteriormente informaron a los usuarios sobre la afectación de este fallo sobre sus servicios y que no era rotundamente obligatorio modificar sus datos de acceso a servicios, pero indicaban que existía la posibilidad de que sus datos sensibles pudieron llegar a verse comprometidos en el lapso de tiempo en el cual el bug estuvo activo y que si se tenían dudas, lo mejor era realizar la modificación de sus logins y passwords utilizados en todas las aplicaciones de Google.

- Instagram: En un comunicado la empresa informo que aunque no se había evidenciado ataques de aprovechamiento de la vulnerabilidad implícita en la extensión Heartbeat sobre alguna de sus cuentas, recomendaron a sus clientes actualizar sus passwords como medida de precaución.

- Pinterest: Este servicio fue claramente afectado, debido a la implementación de una versión de OpenSSL vulnerable, así que realizaron los procesos de parcheo de la librería en sus servidores y pasaron a comunicarse con los posibles usuarios afectados vía mail y les solicitaron modificar sus datos de ingreso a esta red social.

- Minecraft: Apenas se enteraron de la existencia del fallo y sus consecuencias, el equipo técnico de Mojang AB detuvieron de manera inmediata a todos los servidores implicados en el funcionamiento de este importante juego online e iniciaron los procesos de actualización de estos, para asi instalar la versión corregida de la librería OpenSSL y al igual que los casos anteriores, pasaron a informar a sus clientes él porque de la suspensión temporal de sus servicios y los invitaban a modificar sus credenciales de acceso a sus servidores de juego.

- Yahoo: Varios de los servicios principales de la empresa Yahoo se vieron comprometidos por este fallo, entre los que se resaltan:

  - Flickr.
  - Tumblr.
  - Yahoo Mail.
  - Yahoo Sports.
  - Yahoo Tech.

- Wikipedia.

- Wordpress.

3.1.2. Elementos de hardware vulnerados:

Esta falla de seguridad llego a ser tan sorprendente que no solo llego a afectar a servicios web sino que también logro trasgredir a varios dispositivos de hardware, ya que existen varios de estos que manejan de manera embebida la funcionalidad de Heartbeat; los más destacados fueron [5 9]:

- Todos los dispositivos inteligentes Android, en su versión Jelly Bean (4.1.1).

- Routers de marca Cisco y Juniper.

- Sistemas VoIP de iPECS.

- Multifuncionales WiFi de las marcas Brother, Dell, Lexmark y Hewlett-Packard.

- Elementos de red de la marca Barracuda Networks y Fortinet.

- Elementos de seguridad de la referencia SonicWALL de Dell.

- Firewalls pfSense y WatchGuard.
- Sistemas NAS (Network-Attached Storage) de las marcas D-Link, LaCie, QNAP, Western Digital, entre otros.

3.1.3. Sistemas operativos afectados:

En el campo de los sistemas operativos, debido a la libre implementación de la librería OpenSSL para la protección criptográfica de datos compartidos entre clientes y servidores en ambientes de red TCP/IP, se vieron afectadas por esta vulnerabilidad algunas de las distribuciones más importantes de Linux, tales como [10]:

- CentOS 6.5 (OpenSSL 1.0.1e).
- Fedora 18 (OpenSSL 1.0.1e).
- FreeBSD 10 (OpenSSL 1.0.1e).
- NetBSD 5.0.2 (OpenSSl 1.0.1e).
- OpenBSD 5.3 (OpenSSL 1.0.1c).
- OpenBSD 5.4 (OpenSSL 1.0.1c).
- OpenSUSE 12.2 (OpenSSL 1.0.1c).
- Ubuntu 12.04.4 LTS (OpenSSL 1.0.1).

3.2. Mecanismo de aprovechamiento del bug:

En la presente sección se mostrara de manera práctica como un atacante informático lograba aprovecharse del fallo Heartbleed; para esto se presenta un escenario de ataque entre un equipo servidor victima montado sobre una distribución vulnerable Ubuntu[4] 12.04.4 LTS (Precise Pangolin) y un equipo atacante Kali Linux[5] 1.1.0; el diagrama básico de red del escenario descrito sería el siguiente [11]:

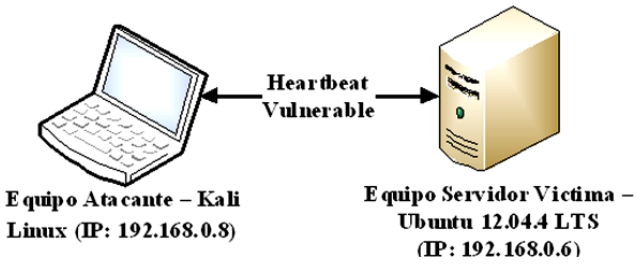


Figura 3. Escenario de demostración de ataque a la vulnerabilidad Heartbleed.

Antes que nada se verifica la versión de OpenSSL instalada en el equipo víctima:

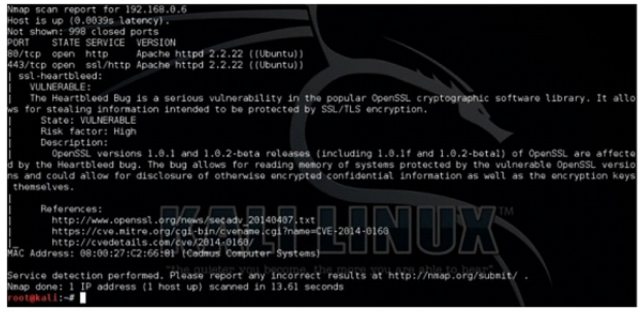
```
john@john-VirtualBox:~$ sudo su
[sudo] password for john:
root@john-VirtualBox:/home/john# openssl
OpenSSL> version
OpenSSL 1.0.1 14 Mar 2012
OpenSSL>
```

Se puede observar que la versión OpenSSL instalada es la 1.0.1 de marzo 2012, la cual fue la primera versión de esta librería en implementar la extensión Heartbeat vulnerable. Posteriormente con el uso de Nmap[6] desde el equipo atacante se hace un análisis del equipo víctima [12]:

```
Nmap scan report for 192.168.0.6
Host is up (0.00038s latency).
Not shown: 998 closed ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https
MAC Address: 08:00:27:C2:66:81 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 0.85 seconds
root@kali:~#
```

Podemos apreciar que el equipo victima tiene activo el puerto 443, que es el utilizado por la versión vulnerable de Heartbeat y que muy posiblemente este implementando de manera insegura la librería OpenSSL, así que como paso siguiente, el victimario procede a realizar una evaluación más profunda con Nmap de esta «potencial» victima; para esto se utilizo la sentencia **nmap -sV —script=heartbleed <Direccion IP de la victima>** en donde **-sV** permite que el la sentencia se ejecute sin inconvenientes aun apuntando a puertos poco convencionales que soportan SSL y **—script=heartbleed** selecciona el script de Nmap diseñado exclusivamente para la evaluación del fallo Heartbleed:

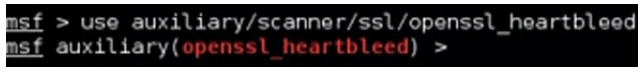



```
State: VULNERABLE
Risk factor: High
```

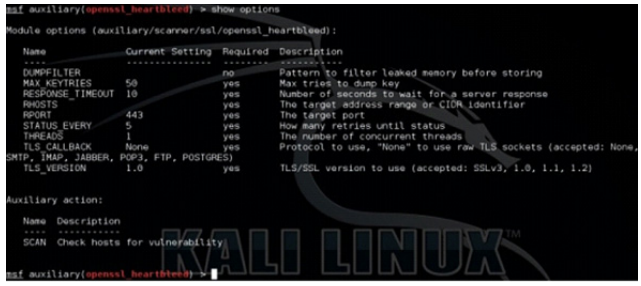
Claramente se puede observar que el equipo víctima es vulnerable a ataques de aprovechamiento

del fallo Heartbleed y que su nivel de riesgo es alto; así que al encontrar este equipo asequible para el robo de datos, el delincuente pasa a ejecutar el Metasploit Framework[7] para usarlo como herramienta de ataque y busca que módulos tiene disponibles para la explotación del error en la extensión Heartbeat [13]:

El Metasploit Framework posee dos módulos enfocados para el usufructo de la falencia Heartbleed, el primero es **auxiliary/scanner/ssl/openssl_ heartbleed** y el **auxiliary/ server/openssl_heartbleed_client_memory**; el primero está estructurado para robar información de manera remota al servidor desprotegido y el segundo es utilizado para exponer los datos almacenados en la memoria del cliente. Para efectos prácticos de exhibición de la explotación de la vulnerabilidad y buscando presentar tal como era aprovechada por parte de los delincuentes informáticos en la realidad, se presentara la ejecución del primer modulo de ataque, puesto que este es el que permitía la extracción de datos sensibles del servidor y es allí donde se podía encontrar datos como llaves de cifrado utilizadas en el protocolo SSL, nombres de usuario y/o contraseñas de los clientes legítimos del servidor, o valores de las cookies y/o de tokens de sesión; que por obvias razones, estos datos son de alto interés para estos usuarios malignos.

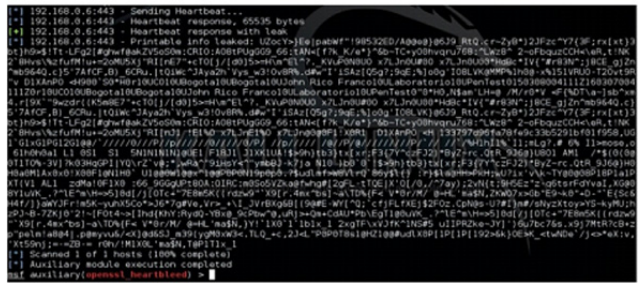


Subsiguientemente se revisan cuales son las opciones de configuración del modulo **auxiliary/scanner/ssl/openssl_ heartbleed** para el robo de datos almacenados en la memoria del servidor:



Para el uso correcto del modulo de asalto seleccionado solamente se deben configurar dos parámetros, el primero es **verbose** para asegurarnos de que se cargue el plugin de ejecución del ataque con todos los comentarios posibles y el segundo es el **rhosts** para así darle al modulo la dirección IP del equipo a vulnerar:

Y a continuación se ejecuta el ataque:



Aunque en la mayoría de veces la memoria del servidor puede estar cargada con datos aleatorios sin ningún valor para el atacante, siempre estaba la posibilidad de llegar a encontrar información de alto interés, como en este caso; si se presta atención a los datos presentados por el modulo de ataque podemos ver que se capturaron algunos datos sensibles:

Entre la información adquirida ilícitamente, los datos de mayor interés fueron:

- Nombre de la unidad organizacional a la cual pertenece el servidor (OU): Laboratorio.
- Nombre del servidor (CN): Pentest.
- Nombre del país en el cual se encuentra el servidor (C): CO.
- Nombre de la provincia y de la ciudad en la cual está radicado el servidor (ST y L): Bogotá.
- Nombre de la organización dueña del servidor o de su encargado (O): John Rico Franco.

3.3. Estrategias de mitigación de los efectos de Heartbleed:

La primera respuesta por parte de los administradores de los servidores desprotegidos fue la de actualizar inmediatamente la versión OpenSSL vulnerable por la versión 1.0.1g para así detener cualquier comunicación « protegida» pero vulnerada colateralmente por la utilización de la extensión imperfecta Heartbeat y como paso siguiente debieron reiniciar todos sus servicios web, para así garantizar la implementación de nuevos valores en los tokens y/o cookies de sesión en el servidor y también permitir la generación de nuevas llaves de cifrado SSL. En referencia al parche de actualización de la librería OpenSSL, este lo que hace es agregar una verificación que descarta cualquier mensaje de tipo Heartbeat cuya longitud de caracteres a extraer del servidor sea diferente a los datos extraídos y almacenados en dicho mensaje en su campo de carga de datos.

Otra medida que fue muy implementada y que se considera una buena práctica en la protección de datos en ambientes de red TCP/IP es la instalación de plugins de alerta en los navegadores de equipos clientes que se vieron afectados por dicha vulnerabilidad, permitiendo así que a cualquier usuario que intente realizar un proceso de comunicación « segura» con un servidor vulnerable será informado inmediatamente sobre este problema antes de que se incurra en el error.

IV. Conclusiones

Debido al alto impacto forjado por esta vulnerabilidad, se ha generado un halo de desconfianza frente a errores aun no detectados en distintos servicios de protección de datos a nivel de redes TCP/IP y es por esto que empresas como Amazon, Microsoft y entre otros han empezado a unir esfuerzos para prevenir futuros casos similares, mediante el estudio de otros proyectos críticos de software libre en búsqueda y eliminación de otras posibles falencias en temas de seguridad que pueden llegar a estar camufladas desde los mismos pilares de una tecnología en tan alto estado de evolución como lo son las telecomunicaciones basadas en la Internet.

Pero este concepto no solo debe ser adoptado por las empresas, sino que también los usuarios básicos de los servicios alojados en Internet deben tomar conciencia y reflexionar sobre todas las posibles falencias en temas de seguridad que aun no han sido detectadas por las entidades responsables de salvaguardar los datos que fluyen a través de un medio tan inseguro como lo es la Internet y que posiblemente ya estén siendo explotadas por los delincuentes informáticos.

Por último, se debe tener muy en cuenta que la vulnerabilidad Heartbleed alcanzo a estar altamente distribuida, completamente activa y anónima por casi dos años. Y aunque la gran mayoría de servicios web de alto impacto que se vieron implicados ya se encuentran reparados y robustecidos, no se puede saber a cabalidad cuántos de estos servicios o páginas web fueron vulnerados por delincuentes informáticos robando información sensible en el pasado; por ende lo invito a usted estimado lector a que cambie sus passwords en la totalidad de sitios web que alberguen sus datos personales y/o financieros e invite a sus allegados a hacer lo mismo, en búsqueda de evitar posibles brechas de seguridad personal y reducir la posibilidad de que

cualquier atacante pueda volver a utilizar cualquier dato de ingreso previamente robado en la bonanza de aprovechamiento del agujero de seguridad aportado por el bug Heartbleed y en si cualquier otro descuido en la protección de su información sensible.

Referencias

[1] H. Kelly; « *The 'Heartbleed' security flaw that affects most of the Internet*»; Documento WEB; [http://edition.cnn.com/2014/04/08/tech/web/heartbleed-openssl/].

[2] A. Freier; P. Karlton y P. Kocher; « *The Secure Sockets Layer (SSL) Protocol*»; RFC 6101; Documento WEB; [https://tools.ietf.org/html/rfc6101]; 2011.

[3] T. Dierks y E. Rescorla; « *The Transport Layer Security (TLS) Protocol*»; RFC 5246; Documento WEB; [https://tools.ietf.org/html/rfc5246]; 2008.

[4] R. Seggelmann; M. Tuexen y M. Williams; « *Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension*»; RFC 6520; Documento WEB; [https://tools.ietf.org/html/ rfc6520]; 2012.

[5] B. Chandra; « *A technical view of the OpenSSL Heartbleed vulnerability*»; Documento WEB - PDF; [http://ibm.biz/dwsecurity].

[6] T. Mpofu, N. Elisa y N. Gati; « *The Heartbleed Bug: An Open Secure Sockets Layer Vulnerability*»; Documento WEB - PDF; [http://www.ijsr.net/archive/v3i6/MDIwMTQ0ODk=.pdf]

[7] Iupati Tumaalii; « *Heartbleed – What's the big fuss?*»; Documento WEB - PDF; [http://www.aarnet.edu.au/images/uploads/resources/AARNet_Whitepaper_Heartbleed_06_2014.pdf].

[8] Varios; « *The Heartbleed Hit List: The Passwords You Need to Change Right Now*»; Documento WEB; [http://mashable.com/2014/04/09/heartbleed-bug-websites -affected/].

[9] J. Pagliery; « *Heartbleed bug affects gadgets everywhere*»; Documento WEB; [http://money. cnn.com/2014/04/11/technology/security/heartbleed-gear/].

[10] Accuvant Labs; « *Heartbleed Bug Advisory (CVE-2014-0160)*»; Documento WEB - PDF; [http://accuvantstorage.blob.core.windows.net/web/file/2016b4dc040c49ee991b5721e0dd62b3/HeartBleed-Bug-CVE-2014-0160-release.pdf].

[11] D. Dieterle; « *Basic Security Testing with Kali Linux*»; Editorial: CreateSpace Independent Publishing Platform; 2014.

[12] G. Lyon; « *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*»; Editorial: Nmap Project; 2009.

[13] D. Kennedy, J. O'Gorman y D. Kearns; « *Metasploit: The Penetration Tester's Guide*»; Editorial: No Starch Press; 2011.

Bibliografía

[1] A. Johns; « *Mastering Wireless Penetration Testing for Highly-Secured Environments*»; Editorial: Packt Publishing; 2015.

[2] D. Stuttard y M. Pinto; « *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws*»; Editorial: Wiley; 2011.

[3] G. Weidman; « *Penetration Testing: A Hands-On Introduction to Hacking*»; Editorial: No Starch Press; 2014.

[4] I. Ristic; « *Bulletproof SSL and TLS: Understanding and Deploying SSL/TLS and PKI to Secure Servers and Web Applications*»; Editorial: Feisty Duck; 2014.

[5] P. Kim; « *The Hacker Playbook: Practical Guide To Penetration Testing*»; Editorial: CreateSpace Independent Publishing Platform; 2014.

[6] T. Wilhelm; « *Professional Penetration Testing: Creating and Learning in a Hacking Lab*»; Editorial: Syngress; 2013.

Infografía

[1] J. Pagliery; «*Don't assume you're safe from Heartbleed*»; Documento WEB; [http://money.cnn.com/2014/04/24/technology/security/heartbleed-security/].

[2] M. Ward; « *Heartbleed bug creates confusion online*»; Documento WEB; [http://www.bbc.com/news/technology-26971363].

[3] S. Curtis; « *'Heartbleed' bug in web technology threatens user data*»; Documento WEB; [http://www.telegraph.co.uk/technology/internet-security/10754169/Heartbleed-bug-in-web-technology-threatens-user-data.html].

[4] U.S. Department of homeland security; « *'NCCIC – Heartbleed OpenSSL Vulnerability*»; Documento WEB - PDF; [https://www.us-cert.gov/sites/ default/files/publications/Heartbleed%20Open SSL%20Vulnerability_0.pdf].0