# The Heartbleed Bug

Librado Santibanez

ICTN 4040

Dr. Phil Lunsford

April 9, 2015

**Abstract**

A confounding computer bug called “Heartbleed” is causing major security worries across the internet. Heartbleed affects many things, including web servers, routers that connect office networks to the internet, mobile apps and VPNs (Virtual Private Network). It has been estimated that 60 percent of secure web sites that are using OpenSSL are affected. In addition, Heartbleed cannot be traced. In many cases, online access to web sites were shut down for several days until it could be patched and upgrade for Heartbleed. What is Heartbleed bug? How does it work? What does it affect? Who created the bug and when? What do we need to know? How can we fix it? Overall, these are the most frequently asked questions as a client and server.

**Introduction**

The OpenSSL (Open Secure Sockets Layer) is used to provide a secure platform for transactions that happen over the internet. Transactions such as online shopping, emails and social media are carried out on the internet through the OpenSSL and other platforms which provide a security [1]. Furthermore, on March 14, 2012, the erroneous code was released with an official OpenSSL update, where it made its way out into the internet and hundreds of thousands of popular websites.

In March 2014, the catastrophic vulnerability in OpenSSL, was found as some researchers called the Heartbleed allows attackers to read sensitive memory from vulnerable servers, potentially including OpenSSL an open-source implementation of SSL and TLS protocols. The core library includes tools for generating RSA private keys and Certificate Signing Request. In addition, it’s estimated that 66% of web servers use

OpenSSL, which help to protect client data by encryption when clients use their login credentials and other private data. In this research, it analyzes of the impact of the vulnerability. Also using extensive scanning, we assess who was vulnerable, and who is still vulnerable. The purposes of the project involved coding bug fixes and new features for OpenSSL, a protocol that anyone can update.

**Background**

In April 7, 2014 The Heartbleed Bug was independently discovered by a team of security engineers (Riku, Antti, and Matti, 2014) at Codenomicon and Neel Mehta of Google Security, who first reported it to the OpenSSL team. The security engineers did not have an idea of the vulnerability until the team found heartbleed bug while improving the Safeguard features. This was the city Codenomicon's Defense security testing tools and reported this bug to the NCSC-FI for vulnerability coordination and reporting to OpenSSL team [3].

In addition, Bloomberg (2014) accused the U.S National Security Agency (NSA) of knowing the Heartbleed Bug for the last two years. Although, the report says the NSA was using it to gain information instead of disclosing it to the OpenSSL developer. After the NSA declining to comment to report of knowing about the Heartbleed Bug, NSA also denied that they were aware of Heartbleed Bug until the vulnerability was made public by the private security engineering of google. Overall, the questions remain about whether anyone from the NSA or U.S government might have exploited the code for their benefits before published to the public

The protocols SSL and TLS are widely used by server software to facilitate secure connections for different purposes. As non- technical reader, a few key terms to understand are: SSL (secure socket layer) a cryptographic protocol that put the S in the “HTTP”, TLS (transport layer security) a protocol that ensures privacy between communication and their users on the internet such as server and client communication and DTLS (datagram transport layer security) provides integrity, authentication and confidentiality. The Heartbleed Bug is not a virus, it’s not a worm or a malicious code, and it has nothing to do with the Man-in-the-Middle, but it’s a simple programming mistake. However, the Heartbleed Bug is a serious vulnerability in the most popular OpenSSL cryptographic software library. This software allows anyone with little knowledge to steal the information such as the names and passwords of the users and the actual content protected, under normal conditions, by the SSL/TLS encryption used to secure the internet. In addition, the code of the Heartbleed Bug is available to the public and there are several sites that have tutorials to teach the use of the software, therefore this vulnerability is most critical.

The purposes of the SSL/TLS is to provide communication security and privacy over the internet for applications such as web, email, VPNs and social media [3]. Smartphones are the best practical example of client side attack, which lead to Blackberry (Z10) products to be vulnerable to Heartbleed Bug, in contrast of Apple’s iOS devices are not affected by OpenSSL. There are other devices affected by Heartbleed such as; IP Phones, Routers, Medical Devices and Smart TV sets. In addition, about 34 percent of Android devices run on version 4.1.x of the mobile OS, which according to Google millions of Android smartphones never, or only rarely,

receive available updates that patch dangerous security defects. For that reason, Android users should download Heartbleed Detector, a free application developed by Lookout. **Figure 1**. Below provides an example of the Heartbleed Detector application.

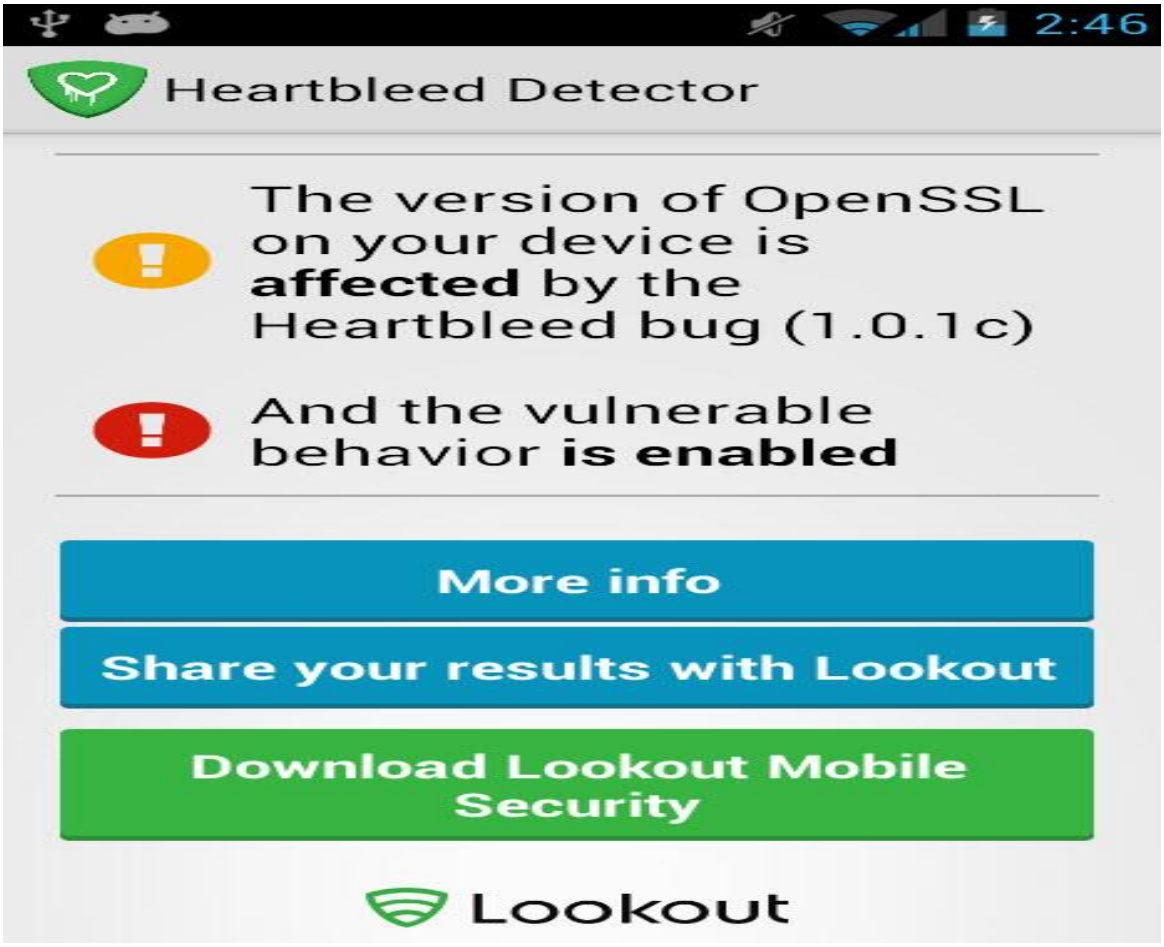


**Figure 1. Hearbleed Detector Application**

The Heartbleed Bug attack works in several steps: First, the attacker creates a custom Heartbleed. Second, the packet is transmitted to vulnerable OpenSSL web server. Third web server processes packet. Fourth, the code grabs up 64KB of extra memory and hopes of capturing something sensitive from memory. Fifth, web server responds by sending a packet back which knowingly includes this extra sensitive data. Sixth, attacker analyzes packets to see if there is anything interesting, if not reruns attack to

capture more memory. Lastly, if web server's certificates private key is captured, it can be used to decrypt current and historical user data and credentials. Overall, is not complex to use the Heartbleed software. As mentioned before, any Heartbleed based attacks are not traceable, due that the problem has existed for the past 2 years without the knowledge of the public. Most server operators use a vulnerable method of the OpenSSL versions 1.0.1 – 1.0.1f and likely don't have enough logs/monitoring to determine whether a site was compromised [4].

The Heartbleed bug reflects one of the most catastrophic vulnerabilities during the OpenSSL history for several reasons: it allowed attackers to retrieve private information and user data, it was easy to exploit and HTTPS and other TLS services have become increasingly popular by the resulting in more affected services [2]. In addition, Stephen Solis-Reyes 19 year-old from Canada was arrested for exploiting the Heartbleed Bug to attack the website of the Canada Revenue Agency. As result, of the attack, Mr. Solis-Reyes had stolen 900 social insurances numbers (Elsevier, 2014). According, to Ivan Ristic, director of engineering at Qualys, has claimed that the percentage of websites vulnerable to the flaw had dropped from 25 percent since the bug was discovered [6]. "Assistant Research Scientist Dave Levin and Assistant Professor of Electrical and Computer Engineering Tudor Dumitras were part of a team that analyzed the most popular websites in the United States-more than one million sites were examined-to better understand the extent to which systems administrators followed specific protocols to fix the problem"(NewsRx, 2014) [7].

Who and what caused Heartbleed Bug? To answer the question, I have displayed 2 graphics with the bad code and the good code. The programmer Robin

Seggelmann, a 31 year old based in Germany, submitted the code. The purposes of the software was to enable a function called “Heartbeat” in OpenSSL. This software package was to be used by nearly half of all web server to enforce the connections. “In one of the new features, unfortunately, I missed validating a variable containing a length” (Seggelmann, 2012). In addition, the code went undetected by several code reviewers and everyone else for over two years. The graphics below shows the c-language code for the Heartbeat message in the OpenSSL source code. In the first graphic, it shows the data structure and the length of the message is given as payload_length.

As it shows below in the graphic 1, the incoming data contains a payload length “payload” which the mistake of the code is that it trust the request without bounds checks. OpenSSL then allocates a buffer for its response, and copies “payload” data byres from the pointer “pl” into it. As result, there’s no “if statement” to make sure that there are actually “payload” bytes in data, or that this is in bounds. Since, there is no “if statement” the attacker gets a 64KB of data in length from main memory. When the attacker gets the 64KB of data the connection is no longer secure between servers and computers [4].

As stated by (Balon, 2014) “Think of the server’s memory like your mind,“ “Whatever is going through your mind right now, what you see in front of you, my words to you, whatever else you’re thinking about -- maybe a hotdog pops in there -- all that information can be scraped off the server, through the Heartbleed bug,” (Balon, 2014) [5].

On the other hand, Graphic 2 shows the correct code with the “if statement” placed in the correct place. However, by making the correction of the code it does not guarantee that you server is secure and is no longer vulnerable. In order, to have a secure server or routers the security technician must take the following actions; upgrade your server to the latest version of OpenSSL, reissue and then revoke all certificates used with the vulnerable version of OpenSSL, and upgrade your security patches. As social media and online shopping user such as; Facebook, Google, eBay, Instagram and other sites that required user credentials may have to change your password if you haven’t change within the past 6 months.

**Graphic 1**

```
/* Read type and payload length first */

hbtype = *p++;

n2s(p, payload);

pl = p;
```

**Graphic 2**

```
hbtype = *p++;

n2s(p, payload);

if (1 + 2 + payload + 16 > s->s3->rrec.length)

    return 0; /* silently discard per RFC 6520 sec. 4 */

pl = p;
```

**Graphic 1 and 2 shows the Heartbleed code**

There are many ways to test if a website has a version of OpenSSL with this bug using the following links, Heartbleed test (https://filippo.io/Heartbleed/), Qualys SSL Labs (https://www.ssllabs.com/ssltest/), LastPass Heartbleed checker

(https://lastpass.com/heartbleed/) or you can also download a Python program from OpenSSL Heartbleed PoC with STARTTLS support (https://gist.github.com/takeshixx/10107280), to be able to check the status of the server you current use or were using and see if it returns memory outside the Heartbleed buffer, but as mentioned before it cannot be traceable.

**Conclusion**

The research, it analyzed numerous aspects of the recent OpenSSL Heatbleed vulnerabilities, including who was vulnerable by providing several tools to check their vulnerable, the impact of Heartbleed can cause on non-secure servers. Overall, Heartbleed Bug has named one of the worst vulnerabilities all over the internet. In addition, there are servers that remain vulnerable. As mentioned before Heartbleed Bug is not a virus, but there are around 66 percent of web servers use OpenSSL. As security technician it's important to be aware of this software and to use the different methods to test the servers and to find a solution. In concluding, the Heartbleed Bug is the most critical software that has been around over the internet. Any attacker can use Heartbleed Bug to hack: smartphones, IP phones, smart TV set, routers and any device that use OpenSSL. The questions remain unanswered of who knew about the Heartbleed. However, Heartbleed Bug cannot be traceable.

## References

[1]. Thabiso Peter Mpofu, Noe Elisa, Nicholaus Gati, Thabiso. *The Heartbleed Bug: An Open Secure Sockets Layer Vulnerability* (n.d.): n. pag. *The Heartbleed Bug*. Web. <http://ijsr.net/archive/v3i6/MDIwMTQ0ODk%3D.pdf>.

[2]. Gulia, Preeti, and Zakir S. Chillar. "A New Approach to Generate and Optimize Test Cases for UML State Diagram Using Genetic Algorithm."*SIGSOFT Softw. Eng. Notes ACM SIGSOFT Software Engineering Notes*37.3 (2012): 1. *The Matter of Heartbleed*. Web. <https://jhalderm.com/pub/papers/heartbleed-imc14.pdf>.

[3]. "The Heartbleed Bug." *Heartbleed Bug*. CC0, n.d. Web. 09 Apr. 2015. <http://heartbleed.com/>.

[4]. Santibanez. “PowerPoint Presentation.” *The Heartbleed Bug.* (2015).

[5]. Christopher. "Heartbleed Bug Simplified: The Internet's Illegitimate Child Is Not Through With Us Yet." *International Business Times*. N.p., 15 Apr. 2014. Web. 09 Apr. 2015. <http://www.ibtimes.com/heartbleed-bug-simplified-internets-illegitimate-child-not-through-us-yet-1572013>.

[6]. Elsevier. "This Is the 360 Link Sidebar Helper Frame - Use This to Find Other Links to This Content or Links to Additional Library Resources."*ECU Libraries Article Linker*. Ltd, (2015). Web. 09 Apr. 2015**.***

<http://jw3mh2cm6n.search.serialssolutions.com/?ctx_ver=Z39.88-2004&ctx_enc=info%3Aofi%2Fenc%3AUTF-

8&rfr_id=info%3Asid%2Fsummon.serialssolutions.com&rft_val_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Ajournal&rft.genre=article&rft.atitle=Heartbleed%2Bbug%2Bleads%2Bto%2Bforking%2Band%2Bfunding&rft.jtitle=Network%2BSecurity&rft.au=Anonymous&rft.date=2014-05-01&rft.pub=Elsevier%2BBV&rft.issn=1353-4858&rft.volume=2014&rft.issue=5&rft.spage=1&rft_id=info%3Adoi%2F10.1016%2FS1353-4858%2814%2970045-5&rft.externalDocID=3402133691¶mdict=en-US>.

[7]. University of Maryland; cybersecurity experts discover lapses in heartbleed bug fix. (2014). *NewsRx Health & Science,* Retrieved from * http://search.proquest.com.jproxy.lib.ecu.edu/docview/1626397458?accountid=10639