Heartbleed Report

CVE-2014-0160

By Timur Ozkul #999072

1. Introduction

On the first of April 2014, a computer bug was discovered which grabbed the attention of the world's media. Rising to fame, unlike most other bugs and was the center stage in the news world. Along with its infamous reputation, it had the corresponding name, Heartbleed. The main reason why it gained such attention was the sheer magnitude of the affected websites. During the time of the disclosure of the bug, it was said that 17% of the world's websites were vulnerable to the bug, as stated by Netcraft. A month after, there were still 800,00 websites still vulnerable. Shodan listed on the sixth of July 2017 that around 180,000 internet-connected devices were still vulnerable and, as of the eleventh of July 2019, 144,000 internet-connected devices. All types of big brands were affected, including Google, Imgur, OKCupid, Eventbrite, Yahoo, Flickr, and even the FBI's website [DZ14] [AHH14].

2. Technical Overview

Heartbleed, also known under the more technical name CVE-2014-0160 is exploited through the flaw in the 1.0.1 version of the OpenSSL encryption software. This software is open-source software that allows for communication over the SSL protocol.

2.1 SSL protocol

The SSL protocol is also known as the Secure Sockets Layer, is a cryptographic security protocol. Along with its successor, TLS, also known as Transport Layer Security, is used to provide data integrity and privacy through encryption and hashing. The encryption makes certain that no one can read or tamper the data in transit. If the data would not be encrypted, people could simply capture that data easily through tools like Wireshark. Wireshark allows for packet sniffing, which is basically capturing the data that's in transit. For example, with Wifi, the signal is broadcast over the air, and anyone can capture the data with the right tools. The main use of TLS and SSL are in web browsers to protect the connection to the web servers. Other common use cases are in emails (SMTP/POP3), instant messaging (XMPP), VoIP, VPN, FTP. Most of these TCP-based protocols are based on the OpenSSL library. Since this protocol provides security, the letter S is appended to the corresponding protocol, for example, HTTPS, FTPS, SMTPS, to exemplify the use of this protocol [Pr19].

2.2 OpenSSL

OpenSSL is the software library written in the programming language C to enable the SSL/TLS protocol. Apache and Nginx are the two most common services run on web servers that use OpenSSL to encrypt websites. Contributing to well over most existing web servers. The version that was affected was 1.0.1 of OpenSSL, which was released in March 2012 [Ka19]. The flaw, more specifically, was in the lack of verification of the heartbeat feature. This heartbeat configuration must be enabled, meaning it's not a default set up. If this were not the case, a considerable amount of more services would have been vulnerable. This is also where it gets its name, Heartbleed.

2.3 Heartbeat Feature

The heartbeat feature of the OpenSSL library provides a method of checking if there is a client or server on the other line of the connection. This is relevant because some router, for instance, will drop the connection if the session is idle for too long. So this feature allows for a TLS session to be up running even though no data has gone through it in a while. This is done through what is called a heartbeat request. If the sender of the heartbeats has not received any responses back for some time, it will assume the connection has failed and will try to set up a new connection, which computationally is more costly. The request consists of three main parts: a short random message, the number of characters in the message, and a request of acknowledgment. The end client simply parrots back the message [Le15][Hu14].

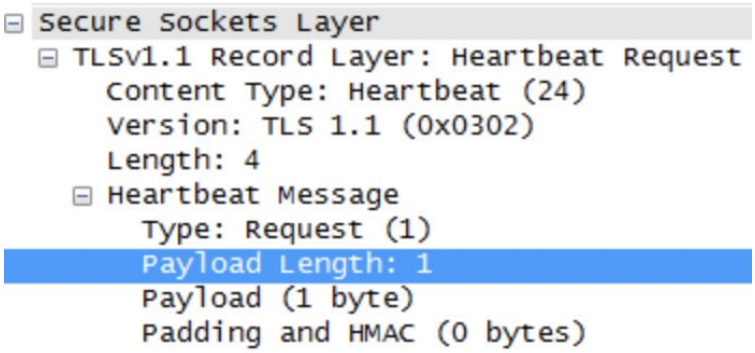

Figure 1, Heartbeat Request captured through Wireshark

2.4 Heartbleed

The security flaw of the OpenSSL is that the reported length to the end client is never checked. When the Heartbleed request is being crafted, it requires the length of the message being sent. One can simply say the message is 64kb in length but have a 1kb message. The end client then allocates a region of physical memory called the memory buffer, to store that apparent 64kb message [Fr17]. The memory buffer is a region of memory used for storing temporary data. It doesn't organize the data in any way, it just stores data in batches. So memory buffer might contain a bunch of non-related data that might include sensitive data such as credentials, keys, cookies, etc [Ma19]. Even though this memory is allocated, it's only filled with 1kb instead of the declared 64kb. When the message is then parroted back to the one making the Heartbeat request, it sends back the 1kb message and also an additional 63kb of data in that memory buffer. One can perform this attack several times and get different random data each time.

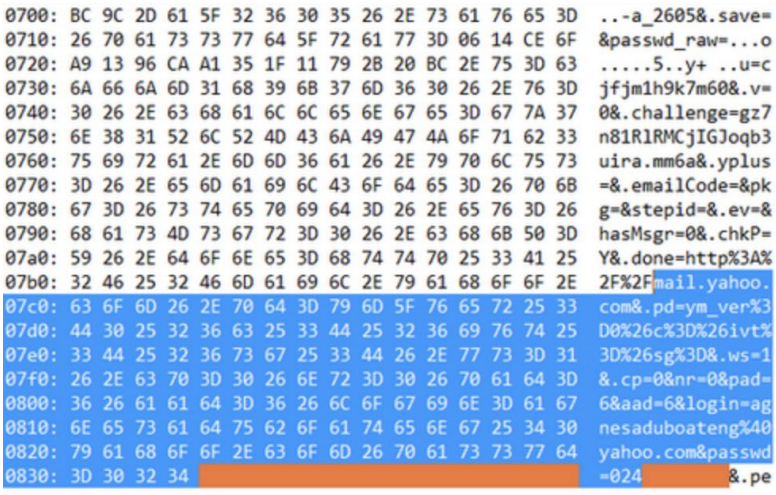

Figure 2, example of memory overflow revealing sensitive data

2.5 Heartbleed Code

The cause of the Heartbleed can pin down to one line of code of the OpenSSL.

```
memcpy(bp, pl, payload);
```

Memcpy copies data to memory, the **bp** parameter is the location of the memory, and **pl** is where it's being copied from, and the third parameter the **payload** is the length of data being copied. There is no verification made that the length of **pl** is equal to that of the **payload** [Fr17].

3. The Cause

The flaw in code was added by a German person named Robin Seggelmann in March 2012. He had told the Sydney Morning Herald. "In one of the new features, unfortunately, I missed validating a variable containing a length." The submission of the code was reviewed by OpenSSL developers, and no one else had noticed it either [Le15]. It is important to mention that developers contributing to OpenSSL were not being paid. They were contributing their own free time to contribute to the code base.

It's difficult to pin down the first attack, but the first record that resembled Heartbleed attacks dates to November 2013 [Br14]. The discovery was made by the Google Security research group called Codenomicon in April 2014, who then helped quickly spread the awareness of the vulnerability [Le15].

4. The Impact

Taking under consideration that a vast use case of OpenSSL only a mere 600,000 websites vulnerable, according to McAfee. In general, the announcement of vulnerabilities is quite hazardous since it also informs malicious actors. Then it becomes a race to get a patch through as fast as possible. Even when patches are available, it is commonplace that organizations don't update immediately, and it is also that organizations don't update many months after.

A day after the announcement from Codenomicon about the vulnerability, there was an attack on the Canadian Revenue Agency, where 900 Social Insurance Numbers were stolen, according to BBC. IBM received 300,000 attacks in 24 hours after four days of the announcement. Thereafter there were also several known successful exploits of the bug.

Another famous one was In August 2014, from alleged Chinese hackers had stolen 4.5 million confidential records of patients from the second-biggest for-profit U.S. hospital chain in the United States.

5. Conclusion

The programming society has glamorized open source and its importance for many years. This vulnerability could have possibly never been found if it was not publicly available code for anyone to be able to read. The fact that OpenSSL was a contribution of many individuals and not the service of an organization lead to lenient behavior. One could argue if the code was based on an organization they would have a more rigorous process annd could have prevented such a mistake.

Nevertheless, this attack was an eye-opener and led to a monetary contribution to the core infrastructure of the OpenSSL project [Le15]. Unfortunately, even six years after the attack, there are still servers and systems out there that are still affected in accordance with Shodan. This tends to be the general theme in the cybersecurity scene with all vulnerabilities. Considering the countless vulnerabilities that come out every year, the need for more training and resources is required for programming in a more security-conscious manner.

Bibliography

[Br14] Sanford-Brown. The Heartbleed impact so far. Retrieved 1 March 2020, from https://www.sanfordbrown.edu/Student-Life/blog/October-2014/Heartbleed-Impact-So-Far

[Fr17] Josh Fruhlinger. What is the Heartbleed bug, how does it work and how was it fixed? Retrieved 1 March 2020, from https://www.csoonline.com/article/3223203/what-is-the-heartbleed-bug-how-does-it-work-and-how-was-it-fixed.html

[He14] Alex Hern. The Guardian. Heartbleed: Hundreds of thousands of servers at risk from catastrophic. Retrieved 27 February 2020, from https://www.theguardian.com/technology/2014/apr/08/heartbleed-bug-puts-encryption-at-risk-for-hundreds-of-thousands-of-servers

[HeH14] Alex Hern. Heartbleed: don't rush to update passwords, security experts warn. Retrieved 27 February 2020, from https://www.theguardian.com/technology/2014/apr/09/heartbleed-dont-rush-to-update-passwords-security-experts-warn

[Hu14] Troy Hunt. Everything you need to know about the Heartbleed SSL bug. Retrieved 27 February 2020, from https://www.troyhunt.com/everything-you-need-to-know-about3/

[Ka19] Marty Kalin. Getting started with OpenSSL: Cryptography basics. Retrieved 28 February 2020, from https://opensource.com/article/19/6/cryptography-basics-openssl-part-1

[Ku14] Mohit Kumar. The Hacker news. HeartBleed Bug Explained. Retrieved 27 February 2020, from https://thehackernews.com/2014/04/heartbleed-bug-explained-10-most.html

[Le15] Timothy B. Lee. The Heartbleed Bug, explained. Retrieved 29 February 2020, from https://www.vox.com/2014/6/19/18076318/heartbleed

[Ma19] Gilad Maayan. Five years later, Heartbleed vulnerability still unpatched Retrieved 1 March 2020, from https://blog.malwarebytes.com/exploits-and-vulnerabilities/2019/09/everything-you-need-to-know-about-the-heartbleed-vulnerability/

[Pr19] Agathoklis Prodromou. TLS Security 1: What Is SSL/TLS. Retrieved 28 February 2020, from https://www.acunetix.com/blog/articles/tls-security-what-is-tls-ssl-part-1/

[Za14] Durumeric Zakir. Heartbleed. Retrieved 27 February 2020, from https://en.wikipedia.org/wiki/Heartbleed