IJCSE International Journal of Computer Science and Engineering Open Access
Research Paper Volume-2, Issue-5 E-ISSN: 2347-2693

Heartbleed Bug: AnOpenSSL Heartbeat Vulnerability

Siddharth Gujrathi[1]

Dept. of Computer Engineering, University of Pune, India, sidh.gujrathi@gmail.com

www.ijcaonline.org

Received: 17/04/2014 Revised: 14 /05/2014 Accepted: 22/05/ 2014 Published: 30 /05/2014

Abstract— Due to exponential growth of Internet, Internet user privacy and data integrity is main concern for developers and service providers over Internet. Keeping this aim in mind Internet has adopted web encryption technique called OpenSSL, which gives a way to secure data and user privacy over Internet. Most of security sensitive application on Internet such as Internet banking, eCommerce, eGoverment has adopted this new technique which gives them a trustworthy way to connect them with their users. OpenSSL is only security mechanism having such reliability of work and has used from long time. But researchers have disclosed a serious vulnerability in this standard Web encryption software known as "Heartbleed", the bug can give hackers access to personal data like credit card numbers, usernames, passwords, and, perhaps most importantly, cryptographic keys—which can allow hackers to impersonate or monitor a server. This research paper provides a detail working of Heartbleed bug and how this bug can affect your online privacy and data integrity.

Keywords/Index Term-OpenSSL; Heartbleed bug; Vulnearbility; Web Encryption; Encryption;heartbeat vulnearbillity;OpenSSL Vulnearability

I. INTRODUCTION

We are living in the world of Internet where everyone is connected with each other somehow using Internet. In this last decade we have seen some tremendous changes and growth of Internet and their usage, revolutionary growth of mobile device and their successor technologies gives every human being power to connect to the grid (Internet). In this era of Internet human can do their any job using Internet it might be shopping, billing, social connections or something else, everything is on grid now a days.

Now a day's Internet is not only tool to get information, there are very less number of peoples who use traditional ways of shopping, banking, information sharing etc. They are more likely to use Internet for their daily task. But while using service like banking, ecommerce etc. user transmits its most valuable information over Internet such as bank credentials, username passwords, personal information and more.

So, the question here comes is, what determine the security of this data over Internet? Or how is it possible to stay secure and use billions of services available on the Internet ? Actually user don't have to worry about these questions. Internet itself take care of all this problems/queries. Services over Internet uses a special type of software/protocol/technique called "OpenSSL" to secure data transmission over Internet.

OpenSSL is open source project adopted by Internet for secure transmission of data over Internet. OpenSSL works seamless and unknowingly. user has not to worry about working of OpenSSL while using Internet. OpenSSL can be viewed as way to the secure tunnel where data can be protected all the time. In this research paper detail technical aspects of OpenSSL are covered.

Corresponding Author: *Siddharth Gujrathi, sidh.gujrathi@gmail.com*

While using OpenSSL form more than decade problem has been arrived when researchers have disclosed a serious vulnerability in this standard Web encryption software known as "Heartbleed", the bug can give hackers access to personal data like credit card numbers, usernames, passwords, and, perhaps most importantly, cryptographic keys—which can allow hackers to impersonate or monitor a server.

After the most used and trusted encryption technique has been vulnerable, secure data of any Internet user could be in the hand of criminal or person having no rights to have that data.

Paper Statement: This Research Paper provides the detail analysis and study of working of Heartbleed bug that can help Internet user to understand and prevent their self from this bug.

Purpose and Motivation : Even after disclosure of "Heartbleed" bug many of you are not aware of this vulnerability and unknowingly this bug becomes part of the everyday life of almost every Internet user. Heartbleed affects almost every secure web services including Google, Yahoo, Facebook, Twitter [4] and many more services which has been in use almost every individual's daily life.

As we all knows, Internet is the primary source of information whether a student using it for the assignments, a teacher using it for papers, an engineer using it for programs, a businessman using it for decision making or a scientist using it for new researches.

So, the purpose of research is how can we prevent our self from this vulnerability and understand basics of how our data is protected by OpenSSL.

II. OPENSSL AND HOW IT WORKS

OpenSSL is an open-source implementation of SSL and TLS protocol [5]. SSL and TLS are transport layer protocol which are mainly involved in end-to-end security over the Internet.

A transport layer protocol provides end-to-end security services for application that use reliable transport protocol such as TCP (Transmission Control Protocol)[2]. The Idea is to provide security services for transactions on Internet. For example, when customer shops online, the following security services are desired :

1. The customer needs to be sure that server (website) belongs to the actual vendor, not an imposter. The customer doesn't want to give an imposer his/her credit card numbers. Likewise, the vendor needs to authenticate the customer.
2. The customer and vendor need to be sure that the contents of the message are not modified during transition (message integrity)
3. The customer and vendor need to be sure that imposter doesn't intercept sensitive information such as credit card numbers.[2]

And to provide this type of security two protocols are dominant today are Secure Socket Layer (SSL) Protocol and The Transport Layer Security (TLS). Technically SSL that came first and TLS is more like successor to SSL. TLS is IETF(Internet Engineering Task Force) standard version of SSL.

For example when you see letters *https://* in your web browser next to a lock icon as shown in fig 2.1

Fig 2.1 use of *https://* in google.com

that means the web page your seeing is in encrypted form, in particular that means you are using SSL/TLS protocol to safeguard your information while using that site. Any information transmitting through this website that might be password or any information is in encrypted form and OpenSSL is just an implementation of these protocols.

To transmute and receive data from this secure site OpenSSL uses session, in this session user and server is involved. In particular there is an extension to TLS protocol known as "Heartbeat" [1]. And what heartbeat extension is allows you to do is keep TLS session up and running even though no data has gone through in a while by sending special request message known as heartbeat request.

So, heartbeat request sends from one computer to another, and basically this request includes some data as "payload" and "size" size of payload as shown in fig 2.2.

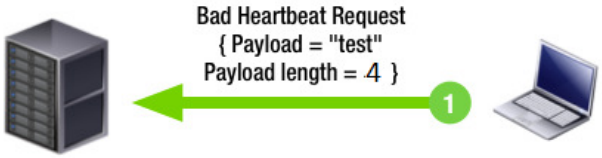

Fig. 2.2 Heartbeat request from client to server.

The Computer that responding to heartbeat request will actually contents the same payload information and also little bit of padding as shown in fig 2.3

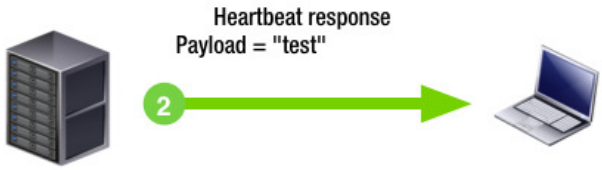

Fig. 2.3 Heartbeat Respond from server to client

In this way a secure session between user and server keeps alive and secure users data using OpenSSL.

From this section I assumed that you understand the basic mechanism of OpenSSL protocol and how it works, now in next section I will be confront you with actual problem i.e. OpenSSL vulnerability "Heartbleed" and how does it works.

III. HEARTBEAT VULNEARBILLITY

The Heartbleed Bug is a serious vulnerability in the popular OpenSSL cryptographic software library. This weakness allows stealing the information protected, under normal conditions, by the SSL/TLS encryption used to secure the Internet.[6]

In OpenSSL implementation version 1.01 and in some of beta releases 1.02 Beta there is highly critical programming mistake is present that can lead to an attacker learning your confidential data.

And the system that running these vulnerable versions of OpenSSL can be attack quite easily. The actual flaw of OpenSSL found in implementation of heartbeat Request/Respond module of TLS. By manipulating these heartbeat request anyone on the Internet can read the memory of the systems protected by the vulnerable versions of the OpenSSL software. This compromises the secret keys used to identify the service providers and to encrypt the traffic, the names and passwords of the users and the actual content. This allows attackers to eavesdrop on communications, steal data directly from the services and users and to impersonate services and users.[6]

The actual technical implementation of this attack/bug is explained in next section.

IV. IMPLEMENTATION OF HEARTBLEED BUG

The vulnerability in encryption software OpenSSL was discovered by Google researcher Neel Mehta and the security firm Codenomicon. They gave the bug—officially known as CVE-2014-0160—the appropriately evocative and frightening name Heartbleed.

In particular Heartbleed bug is actually a programming mistake of one of OpenSSL module which is heartbeat Request/Respond module. As we have seen working of this module in Section II, heartbeat request is use to keep secure session up and running even if no data has been transmitted in while by sending a heartbeat message.

Attacker takes advantage of this heartbeat request message, they craft request such that server receiving this request forced to send addition data from server which might be credentials or any important data. Let's see in detail how this attack works.

Heartbeat Request Message have following parameters in his request package :

1. Payload : some information (it can be anything simple as "test")
2. Size : it has size of payload with it to notify server/computer whom we are sending.

So, attack is held in following sequence ,

1. Attacker first need to know that target machine should be running.
2. Then attacker crafts a special Heartbeat request message which has payload and fake size of payload shown in fig 4.1

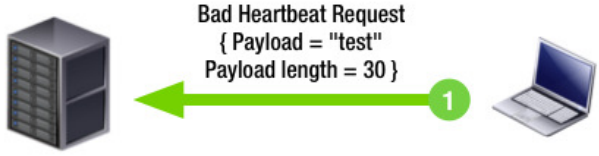

Fig 4.1 Custom Crafted Malicious Heartbeat Request Message

Here, attacker crafted a request message with payload **"test"** and give its size as **"30 Bytes"** and sends to the target machine.

3. Target receives heartbeat request message and create response message, now here is the trick. I would like to point out some background information here.

   Every server running OpenSSL has its own Heap memory to store some data in it. This data mainly consist of data that has been received in secure session such as password, usernames, credit card numbers or encryption keys.

4. While crafting response message it checks for payload size given in request message. in our case it is **"30 Bytes",** so server creates response with same payload i.e. **"test"** which founds to be only 4 bytes in size.
5. So to complete remaining **26 bytes** target machine unknowingly pads 26 bytes of data from its heap memory along with payload as shown in fig 4.2

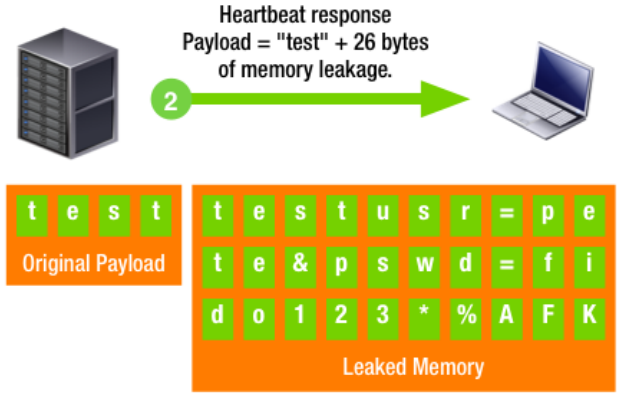

Fig 4.2 Server Response message with Leaked Server Data.

This leaked data from server might be username, password, credit card Number or encryption keys.

If attacker can manage to get history session from server then these leaked keys can be useful to decrypt all information from store history session which can be dangerous than any information leaked from server.

V. PROTECT YOUSELF FROM HEARTBLEED BUG

First and for most to avoid this Heartbleed bug is awareness about vulnerability. The more you know about this bug more you can protect yourself from it.

Common methods to prevent yourself from this bug are as follows :

1. First you need to know about the web service and websites that are affected by this bug. you can find this information on **http://visual.ly/major-sites-affected-heartbleed** [4]. To check which services you use are affected use **http://lastpass.com/heartbleed.**

2. Find out which services has patched the bug.

3. If website has not patched bug, leave your account as is whether logged it or not. Cause if you do any activity logged in or logged out it may lead to exploit to attacker.

4. Change your credentials on website that has been patched bug.

VI. CONCLUSION

In the end, I would like to conclude that awareness and study of vulnerability is the only key to protect yourself from it. The heartbeat vulnerability is most crucial and serious vulnerability found in decade so far so; I research and write this research paper to aware about the vulnerability.

VII. SCOPE FOR FURTHER RESEARCH

As a further dimension in future, I will try to study more important security essentials and security vulnerability to spread the world.

And as Heartbleed concern I will study further for source code of OpenSSL for vulnerabilities and way of improvement.

ACKNOWLEDGMENT

I thank God Almighty for the successful completion of my research and study. I am also grateful to all other respected research in the field of Security and member of the faculty Computer Engineering Department for their cooperation. Finally, I wish to thank all my dear friends, for their whole-hearted cooperation and encouragement

REFERENCES

[1] John Viega, Matt Messier, Pravir Chandra, "Network Security with OpenSSL: Cryptography for Secure Communications" O'Really Medi Inc., First Edition, pp. 21-22, 2002.
[2] Behrouz A Forouzan, "Data Communication and Networking", The McGraw-Hills Companies, Fourth Edition, pp. 1008-1014
[3] "Growth of Internet ", http://www.socialmarketingforum.net/2012/09/the-Internet-explosive-growth-and-changes/, 2013
[4] "Heartbleed affected Sites", http://visual.ly/major-sites-affected-heartbleed, 2014
[5] "Definition", http://en.wikipedia.org/wiki/OpenSSL, 2014
[6] "Heartbleed intro", http://heartbleed.com/, 2014

AUTHOR'S PROFILE

Siddharth Gujrathi : He is Programming enthusiastic and computer security researcher. He is currently surving as CEO of Codiliffe Software solutions Nashik and is persuining his Bachelors of Engineering from Sandip Institue of Engineering and Management (University of Pune), India. He also completed his diploma in Computer Engineering in 2010 from MSBTE.