Heartbleed OpenSSL Vulnerability: a Forensic Case Study at Medical School

Han Wu
Office of Research, New Jersey Medical School,
Rutgers, The State University of New Jersey

Heartbleed vulnerability in OpenSSL was released to public that remote attacker may get sensitive data, possibly including user authentication credentials and secret keys, through incorrect memory handling in the TLS heartbeat extension. By choosing this topic, the case study I am doing is real, including the case analysis, procedures and findings. I am using word Institution to refer my employer in this paper, as my employer is a public institution instead of a for-profit company. The institution has been notified to take further actions and procedures including internal audit including server inventory audit and risk assessment. Initial internal audit has been completed in a short timeframe and user communities are kept updated. Further Phase 2 work is still ongoing and is not completed yet as external auditors is involved and the senior management and corporate office of information technology are taking the lead.

Keywords: Heartbleed, Vulnerability, IT Audit, SSL

For correspondence contact: Han Wu, Office of Research, New Jersey Medical School, Rutgers, The State University of New Jersey, 185 S. Orange Ave., MSBC690, Newark, NJ
07103.
E-mail: hw289@njms.rutgers.edu

1. INTRODUCTION

Heartbleed vulnerability in OpenSSL could allow remote attacker to get sensitive data, possibly including user authentication credentials and secret keys, through incorrect memory handling in the TLS heartbeat extension [1, 2].

Here there are some updates regarding Heatbleed in the real world [21]:

- On April 19th, Healthcare.gov users asked to reset passwords following Heartbleed bug.
- On April 16th, the first Heartbleed hacker has been arrested.
- On April 14th, the cause of theft of 900 Canadian tax ID numbers is Heartbleed.

1.1 Origin of Heartbleed Bug

The naming of Heartbleed is based on Heartbeat, while the Heartbeat is an Extension for the Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) protocols, it was proposed as a standard in February 2012 by RFC 6520[5, 15]. It provides a way to test and keep alive secure communication links without the need to renegotiate the connection each time.

In 2011, one of the RFC's authors, Robin Seggelmann implemented the Heartbeat Extension for OpenSSL, OpenSSL failed to notice a bug in Seggelmann's implementation, and introduced the flawed code into OpenSSL's source code repository on December 31, 2011[16, 17]. The vulnerable code was adopted into widespread use with the release of OpenSSL version 1.0.1 on March 14, 2012. Heartbeat support was enabled by default, causing affected versions to be vulnerable by default [3, 18, 19].

Bug is in the OpenSSL's implementation of the TLS/DTLS (transport layer security protocols) heartbeat extension (RFC6520). When it is exploited it leads to the leak of memory contents from the server to the client and from the client to the server.

1.2 Descriptions

This critical flaw in OpenSSL versions 1.0.1 up to 1.0.1f allows an attacker to retrieve private memory of an application that uses the vulnerable OpenSSL library in chunks of 64k at a time. Note that the exploit code is publicly available for this vulnerability that an attacker can repeatedly leverage the vulnerability to retrieve as many 64k chunks of memory as are necessary to retrieve the intended secrets. The sensitive information that may be retrieved using this vulnerability includes [3, 5] primary key material contains secret keys, secondary key material contains user names and passwords used by vulnerable services, protected content contains sensitive data used by vulnerable services, and collateral contains memory addresses and content that can be leveraged to bypass exploit mitigations.

This bug was independently discovered by a team of security engineers including Riku, Antti and Matti [3] at Codenomicon and Neel Mehta of Google Security, who first reported it to the OpenSSL team. Codenomicon team found heartbleed bug while improving the SafeGuard feature in Codenomicon's Defensics security testing tools and reported this bug to the The National Cyber Security Centre Finland (NCSC-FI) for vulnerability coordination and reporting to OpenSSL team. On April 7th, 2014, National Vulnerability Database (NVD) of NIST released a Vulnerability Summary for CVE-2014-0160[4]. CVE-2014-0160 is the official reference to this Heartbleed bug.

1.3 About The Institution

My employer is a flagship public research institution which consists of several campuses from north to south of this state. I am working in the Health Sciences campus where the major medical school and affiliated hospital are located.

Due to the nature and the mission of medical school and hospital, we have to maintain the compliances including 21CRF11, FISMA and HIPPA. Our IT at Health Sciences campus consists of 4 divisions:

1. Enterprise IT, which is serving the cyber infrastructures of the institution.
2. Educational IT, which is serving the advanced, professional trainings in the field of biomedical and health sciences to meet the education mission.

3. Research IT, which is serving our talented researchers' need to meet the research mission.

4. Clinical IT, which is supporting the clinical practice to meet the clinical mission.

2. ANALYSIS

Actually this is my first time to hear about Heartbeat extension after the unveiling of this Heartbleed security incident. As an all stack developer, I am not a fulltime server administrator. Every time when I deal with OpenSSL, I just need to access the server and run the shell command (openssl) to generate the private key and Generating a Certificate Signing Request (CSR), then what I should do is to communicate with certificate authority, strictly speaking, indirectly, as we have designated institutional certificate administrator whom usually we talk to.

However, from this case, I read so many articles. I learned that this time this vulnerability is different from the other bugs which may come and go and are fixed by new versions, this bug has left large amount of private keys and other secrets exposed to the Internet. Considering the long exposure which is lasting almost two years, ease of exploitation and attacks leaving no trace this exposure should be taken seriously.

As reported[3, 5, 6], this bug is not a design flaw in SSL/TLS protocol specification, instead, it is the implementation problem that now we know it is the programming mistake in popular OpenSSL library that provides cryptographic services such as SSL/TLS to the applications

Encryption is used to protect secrets that may harm your privacy or security if they leak. In order to coordinate recovery from this bug we have classified the compromised secrets to four categories [3].

Primary key material: the encryption keys. Leaked secret keys allow the attacker to decrypt any past and future traffic to the protected services and to impersonate the service. Any protection given by the encryption and the signatures in the X.509 certificates can be bypassed.

Secondary key material: e.g. the user credentials including user names and passwords used in the vulnerable services. Recovery from this leak requires owners of the service first to restore trust to the service according to steps described above. After this users can start changing their passwords and possible encryption keys according to the instructions from the owners of the services that have been compromised. All session keys and session cookies should be invalidated and considered compromised.

Leaked protected content: This is the actual content handled by the vulnerable services. It may be personal or financial details, private communication such as emails or instant messages, documents or anything seen worth protecting by encryption. For our clinical applications, PHI data contains patients' identity information are at risk.

Leaked collateral: may contain technical details such as memory addresses and security measures such as canaries used to protect against overflow attacks. These have only contemporary value and will lose their value to the attacker when OpenSSL has been upgraded to a fixed version.

As reported [3], NCSC-FI took up the task of verifying this immediately after the report of the bug, analyzing it further and reaching out to the authors of OpenSSL, software, operating system and appliance vendors, which were potentially affected. However, this vulnerability had been found and details released independently by others before this work was completed. Vendors should be notifying their users and service providers. Internet service providers should be notifying their end users where and when potential action is required.

2.1 Versions of the OpenSSL Affected

Status of different versions[5, 9]:

1. OpenSSL 1.0.1 through 1.0.1f (inclusive) are vulnerable
2. OpenSSL 1.0.1g is NOT vulnerable
3. OpenSSL 1.0.0 branch is NOT vulnerable
4. OpenSSL 0.9.8 branch is NOT vulnerable

Bug was introduced to OpenSSL in December 2011 and has been out in the wild since OpenSSL release 1.0.1 on 14th of March 2012. OpenSSL 1.0.1g released on 7th of April 2014 fixes the bug.

2.2 Server OS Distributions Affected

Some operating system distributions that have shipped with potentially vulnerable OpenSSL version:

1. Debian Wheezy (stable), OpenSSL 1.0.1e-2+deb7u4
2. Ubuntu 12.04.4 LTS, OpenSSL 1.0.1-4ubuntu5.11
3. CentOS 6.5, OpenSSL 1.0.1e-15
4. Fedora 18, OpenSSL 1.0.1e-4
5. OpenBSD 5.3 (OpenSSL 1.0.1c 10 May 2012) and 5.4 (OpenSSL 1.0.1c 10 May 2012)
6. FreeBSD 10.0 - OpenSSL 1.0.1e 11 Feb 2013
7. NetBSD 5.0.2 (OpenSSL 1.0.1e)
8. OpenSUSE 12.2 (OpenSSL 1.0.1c)

2.3 Server OS That Are Not Vulnerable

1. Debian Squeeze (oldstable), OpenSSL 0.9.8o-4squeeze14
2. SUSE Linux Enterprise Server
3. FreeBSD 8.4 - OpenSSL 0.9.8y 5 Feb 2013
4. FreeBSD 9.2 - OpenSSL 0.9.8y 5 Feb 2013
5. FreeBSD 10.0p1 - OpenSSL 1.0.1g (At 8 Apr 18:27:46 2014 UTC)
6. FreeBSD Ports - OpenSSL 1.0.1g (At 7 Apr 21:46:40 2014 UTC)

3. AUDIT PROCEDURES

3.1 Internal Audit

The corporate Office of Information Technology send announcement as soon as CVE-2014-0160 was released and the institution was also notified by state cybersecurity office. Then all server groups in the 4 units mentioned above are notified immediately. Then the audit and assessment processes have been carried out among departmental-level IT, School-level IT and the central side (Corporate IT).

Departmental IT or unit computing services exist in some schools as some larger departments or units have resources to maintain their own departmental level IT team, this level of IT team may have only one all-around technical staff or more. Those departmental server administrators can initiate their own assessment tasks.

School-level IT office has its own dedicated server team overseeing and maintaining the school's own data center and all the server resources.

Corporate IT as the institution's centralized IT office who is overseeing the infrastructure and its multiple data centers located in different locations in the state, some are self-owned, while some are contracted with outside vendors.

As an all-around technical staff developing and managing the school's research computing services, my internal audit and security assessment work has been overseen and is coordinated by the IT office of the school, corporate information technology of the institution and institution security office.

3.1.1 Inventory the environment

1. Create a list of every server or service I am managing that offers encrypted network access via SSL or TLS. I developed several web applications which are serving cancer center, clinical research unit, medical school and etc., all web server and database servers are Linux based: 7 Ubuntu servers, 3 CentOS servers and only 1 Deibian server.

2. Non-web services: email, SSH, LDAP, and anything else offering connectivity to users on the network.

3. 3rd party services contract with (cloud services, outsourced services, etc.)

4. For each item in the inventory determine if it is vulnerable.

5. Detect prior Heartbleed exploit

6. It is very hard to detect if someone has exploited this against our services, as Exploitation of this bug does not leave any trace of anything abnormal happening to the logs. However, we still pull out all the recent Apache and MySQL, PostgreSQL logs from the server and run analysis. During Phase 1, we only analyzed recent 6 months of logs, more will continue during next phases when time and resources allowed.

Find out if the server is vulnerable to the risk[6]

Run the command:

```
openssl version
```

to get the version number of openssl. If the command shows e.g.:

```
openssl version
OpenSSL 1.0.1e 11 Feb 2013
```

Then the server might be vulnerable as the version is 1.0.1 and is below 1.0.1g. But some Linux distributions patch packages; If the server uses a 0.9.8 release like it is used on Debian squeeze, then the server is not vulnerable as the heartbeat function has been implemented in OpenSSL 1.0.1 and later versions only.

Run shell command:

```
openssl version
OpenSSL 0.9.8o 01 Jun 2010
```

Fix the vulnerability

To fix the vulnerability, install the latest updates for my server.

Table I. Server and respective command

Server	command
Debian	apt-get update apt-get upgrade
Ubuntu	apt-get update apt-get upgrade
Fedora	yum update
CentOS	yum update
OpenSuSE	zypper update

Then restart all services that use OpenSSL, if you want to be absolutely sure that you did not miss a service, and then restart the whole server by running "reboot" on the shell. After you installed the Linux updates, check if the openssl package has been upgraded correctly.

Check the package on Debian and Ubuntu [6], and the output is listed in Appendix A:

```
dpkg-query -l 'openssl'
```

For Fedora and CentOS, use this command to find the installed package name, Appendix B lists the packages:

```
rpm -qa | grep openssl
```

External tools used for internal audit

1. Calculate the environmental score of OpenSSL Heartbeat Extension Vulnerability

2. Vulnerability testing services

There are tests available to verify if you successfully closed the security hole in your Server. The test can be found here:

1. Heartbleed Scanner by Italian cryptologist Filippo Valsorda[10]

2. Heartbleed Vulnerability Test by Cyberoam[11]

3. Critical Watch Free Online Heartbleed Tester[12]

4. Heartbleed Server Scanner by Rehmann[13]

Since our institution is using Comodo certificate, I also use COMODO SSL Analyzer [7, 14] to run the vulnerability scan. All scan results shows negative which means No Vulnerability.

3.2 External Audit

The external audit procedures are coordinated by both corporate OIT and institution security office, I am not involved in this loop as this time. However, given my previous experience when I was on a certain technology advisory committee, we contracted with PwC as our external auditors.

Also we have regional FBI computer forensics lab in local that we contract with. The servers hosting PHI data need further 3rd party assessment.

4. FINDINGS

Table II. Audit Result of the Servers

Server	**OpenSSL**	**Vulnerability**	**numbers**
Linux Ubuntu	0.9.8	no	2
CentOS	1.0.0e	no	3
Debian	1.0.0c	no	1
Linux Ubuntu	1.0.0e	no	5

1) Vulnerable services found have to be patched as soon as patches become available

2) Not all vendors have patches yet.

3) New SSL key have to be created and new SSL certificate have to be obtained for our vulnerable services identified. Our corporate OIT offers Comodo certificates for free.

4) There is a significant risk that our existing SSL key has already been compromised and copied; it MUST be replaced.

5) Issue a certificate revocation for the old SSL certificate with our Certificate Authority vendor to help prevent attackers from successfully impersonating our service and compromising our users in the future.

6) Notifying our end-users the audit progress and result.

The issue comes because there's (apparently) no way of knowing if the server has been compromised using the heartbleed vulnerability - which might theoretically have included exposing your private key as part of the data read from memory[8]. And that would enable an attacker to set up a machine to intercept traffic intended for your server, and decrypt it. Recovery from this leak requires patching the vulnerability, revocation of the compromised keys and reissuing and redistributing new keys. Even doing all this will still leave any traffic intercepted by the attacker in the past still vulnerable to decryption.

TLS certificates are used to verify the website is who they say it is. Most of us use third party certificate authority so our users' browser can automatically check our credential through the third party. This is one of the main reasons we fork out the money. Getting the TLS certificate to work will be transparent to the user, so is changing the TLS cert.

Of course, if your certificate authority is compromised then there is a chance that your TLS cert is compromised as well. If that is the case they will notify you to change the cert _and_ you will be affected regardless of which web server you are using. Still, the process of switching to the new cert is transparent to your user.

There is no total of 64 kilobytes limitation to the attack, that limit applies only to a single heartbeat. Attacker can either keep reconnecting or during an active TLS connection keep requesting arbitrary number of 64 kilobyte chunks of memory content until enough secrets are revealed.

Even though the actual code fix may appear trivial, OpenSSL team is the expert in fixing it properly so fixed version 1.0.1g or newer should be used. If this is not possible software developers can recompile OpenSSL with the handshake removed from the code by compile time option -DOPENSSL_NO_HEARTBEATS.

5. CONCLUSIONS

The phase 1 internal audit is successfully done within a short timeframe which is within 2 business days. We took actions to minimize the security risk, e.g. Patch OpenSSL 1.0.1g, Patch All Major Linux Distributions. It is really a good opportunity to upgrade security strength of the secret keys used. The school IT office has collected assessment result, survey responses from all departmental IT teams, after full review, then those records and logs were forward to corporate IT for final review and comparison with the audit that has been accomplished by themselves (the OIT). And during this time, our user communities are kept updated, regular alert and email distributions are being sent via mailing list.

On April 22nd, 2014, the corporate Office of Information Technology notified the whole community regarding the Heartbleed vulnerability and explained that the community would be notified in groups to change their password over the next few weeks. The password changes had to be spread out by group due to load to the system. OIT states that they have no evidence at this time that any user accounts have been compromised, but changing account password is strongly recommended, especially if certain users have access to restricted data. We are also reminded to update password on mobile devices such as the smart phone, iPad, etc. And it is best not to use the same password for all websites.

However, on the other hand, the external audit in Phase 2 takes much longer time. In the time I am concluding my work and am writing this page, Phase 2 work is still ongoing and is not completed yet as external auditors is involved and the senior management and corporate office of information technology are taking the lead, initial meetings are still being scheduled for the time being.

Heartbleed vulnerability proves that people need to rethink how open source software should be used concerning Open source vs Security, and what can be done to prevent this from happening in future. We got this lesson that we must learn to find these inevitable human mistakes sooner.

This incident does remind us to support the development effort of opensource software you trust your privacy to. As the Heartbleed bug has revealed, this essential tool lacks of support; the team in charge of the open source protocol is severely understaffed and underpaid. It is reported that only two persons have been primarily responsible for OpenSSL for more than a decade [20].

APPENDIX

A. OUTPUT ON CORRECTLY PATCHED DEBIAN 7 SERVER

```
dpkg-query -l 'openssl'

Desired=Unknown/Install/Remove/Purge/Hold
```

```
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name            Version          Architecture   Description
+++-==================-===============-==============-===========================================
ii  openssl         1.0.1e-2+deb7u5  amd64          Secure Socket Layer (SSL) binary and related.
```

B. LINKS WITH RELEASE NOTES OF THE FIXED VERSIONS

Debian: http://www.debian.org/security/2014/dsa-2896

Ubuntu: http://www.ubuntu.com/usn/usn-2165-1/

Fedora: https://lists.fedoraproject.org/pipermail/announce/2014-April/003206.html

CentOS: http://lists.centos.org/pipermail/centos-announce/2014-April/020249.html

REFERENCES

[1] Vulnerability Note VU#720951: OpenSSL TLS heartbeat extension read overflow discloses sensitive information, http://www.kb.cert.org/vuls/id/720951, last accessed April 14th, 2014

[2]Security Advisory: Heartbeat overflow issue, https://www.openssl.org/news/secadv_20140407.txt, last accessed April 14th, 2014

[3]The Heartbleed Bug, http://heartbleed.com/, last accessed April 14th, 2014

[4]Vulnerability Summary for CVE-2014-0160, http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-0160, last accessed April 14th, 2014

[5]Heartbleed, http://en.wikipedia.org/wiki/Heartbleed, last accessed April 19th, 2014

[6]How to find out if your server is affected from Openssl Heartbleed vulnerability (CVE-2014-0160) and how to fix that, http://www.howtoforge.com/find_out_if_server_is_affected_from_openssl_heartbleed_vulnerability_cve-2014-0160_and_how_to_fix, last accessed April 15th, 2014

[7]Comodo Advises Customers and Partners to Patch Systems to Run the Latest Version of OpenSSL in Light of 'Heartbleed' Vulnerability, http://www.comodo.com/news/press_releases/2014/04/comodo-advises-to-run-latest-version-of-openssl.html, last accessed April 14th, 2014

[8]How to Detect a Prior Heartbleed Exploit, http://www.riverbed.com/blogs/Retroactively-detecting-a-prior-Heartbleed-exploitation-from-stored-packets-using-a-BPF-expression.html, last accessed April 15th, 2014

[9]Cisco Security Advisory: OpenSSL Heartbeat Extension Vulnerability in Multiple Cisco Products, http://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20140409-heartbleed, last accessed April 18th, 2014

[10] Heartbleed Scanner by Italian cryptologist Filippo Valsorda, http://filippo.io/Heartbleed, last accessed April 15th, 2014

[11]Heartbleed Vulnerability Test by Cyberoam, http://csc.cyberoam.com/cyberoamsupport/webpages/webcat/2014-0160.jsp, last accessed April 15th, 2014

[12]Critical Watch Free Online Heartbleed Tester, http://heartbleed.criticalwatch.com/, last accessed April 15th, 2014

[13]Heartbleed Server Scanner by Rehmann, http://rehmann.co/projects/heartbeat, last accessed April 15th, 2014

[14]COMODO SSL Analyzer, https://sslanalyzer.comodoca.com/, last accessed April 22nd, 2014

[15]RFC6520, https://datatracker.ietf.org/doc/rfc6520/, last accessed April 15th, 2014

[16] Grubb, Ben (April 11, 2014). "Man who introduced serious 'Heartbleed' security flaw denies he inserted it deliberately".The Sydney Morning Herald, last accessed April 20th, 2014

[17] "Meet the man who created the bug that almost broke the Internet". Globe and Mail. Last accessed April 11th, 2014.

[18]Goodin, Dan (April 8, 2014). "Critical crypto bug in OpenSSL opens two-thirds of the Web to eavesdropping". Ars Technica., last accessed April 15th, 2014

[19]Hagai Bar-El (April 9, 2014). "OpenSSL "Heartbleed" bug: what's at risk on the server and what is not"., last accessed April 15th, 2014

[20]http://www.theverge.com/2014/4/27/5658368/two-men-are-tasked-with-taking-care-of-openssl/in/5371655, last accessed April 20th, 2014

[21]Heartbleed: the bug that put the internet on high alert, http://www.theverge.com/2014/4/12/5607614/heartbleed-openssl-bug-storystream, last accessed April 28th, 2014