Heartbleed

Did the Internet bleed to death?

How long has it been out?

- OpenSSL 1.0.1 came out March 14 2012
- OpenSSL 1.0.1g came out April 7 2014

That’s over two years of vulnerability!

How Widespread Is It?

- OpenSSL 1.0.1 - 1.0.1f
- And probably more SSL because…
- It's an IETF approved RFC!
- http://tools.ietf.org/html/rfc6520
- Basically a rewrite of RFC520 ;)

RFC6520

Lets take a look…

```
struct {
    HeartbeatMessageType type;
    uint16 payload_length;
    opaque payload[HeartbeatMessage.payload_length];
    opaque padding[padding_length];
 } HeartbeatMessage;
```

RFC6520

You read that right…

uint16 payload_length;

A heartbeat needs 65536 bytes of payload?!

RFC6520

- Not only is there 65536 bytes of payload, but the client can set the length of the payload response!
- OpenSSL trusts unknown internet users?!
- Who needs verified security with protocols like SSL/TLS!

So what?

- User gets to specify the size buffer returned
- It can be large, but what harm can 64KB do?
- Memory comes from the kernel cleaned up
- Not a NULL dereference or overflow either

If only OpenSSL left memory to the OS...

OpenSSL Knows Best!

- Has it’s own memory management
- Completely ignores host OS protections!
- Of course, most standard systems would probably have leaked keys anyway
- Except… OpenBSD would’ve caught this!

Operating System Memory

- Majority of OS's don't return free(3)'d memory to kernel until program exits
- Speed-up: it's cheaper to just give back memory the program free(3)'d if it asks for more
- OpenBSD returns free(3)'d memory to the kernel almost immediately to catch use-after-free bugs

Let’s Ask Someone Else

To quote Ted Unangst:

> What if malloc's "G" option were turned on? You know, assuming the
> subset of the worlds' programs you use is good enough to run with that.

No. OpenSSL has exploit mitigation countermeasures to make sure it’s exploitable

http://marc.info/?l=openbsd-misc&m=139698531410614&w=2

Let’s Ask Someone Else

Or Bob Beck:

Heartbleed was really not the final straw.

Their Malloc replacement layer was the final straw for us.

It never frees memory. (tools can't spot bugs)

It uses LIFO recycling (use after 'free'? no problem)

Includes a Debugging malloc that sends private info to logs, always there.

Includes the ability to replace the malloc/free at runtime (eek!)

In a nutshell, This was a very effective exploit mitigation technique countermeasure. It could not have been designed better to make an attack like heartbleed both hard to detect, and have dire consequences. OpenBSD's memory allocator, Valgrind and Coverity do not notice the memory issues.

OpenSSL memory

- Allocates a giant pool of memory at start

- Has it’s own alloc() and free() functions
- So now memory beyond allocated amount can be accessed without fear of crashing
- Enter RFC6520...

Heartbeat

So now let’s look at the actual Heartbeat protocol…

Heartbeat

- User sends message of ‘cat’ with payload length of 3 bytes
- Server responds with ‘cat’ 3 bytes
- User sends message of ‘cat’ with payload length of 65536 bytes…
- Server responds with ‘cat’ + 65533 more bytes of memory from OpenSSL’s pool

But what leaks?

- Cloudflare claims you cannot get private keys
- There’s no easy way to specify what memory you wish to retrieve
- It’s only 64KB

It must just be people fear mongering...

WRONG!

- Cloudflare’s competition only took 9 hours for attackers to get private keys!
- There is NO WAY to detect attacks
- Basically cost-free to brute force, you can do it over 2.5 million times and just keep going
- It can even be reversed to get client keys!

Solutions?

- Revoke ALL keys that were on servers connected to Internet
- Yes, that includes Root CA’s if OpenSSL managed the Root CA key on an Internet connected machine
- Patch! Patch! Patch!
- Reissue all new certificates

Fixes

- Each OS provides patches versions
- Compile your own from source (1.0.1g or later)
- Recompile vulnerable versions with Heartbeat functionality disabled

Fallout

- LibreSSL
- Start of some cleanup for OpenSSL
- Privilege separated private key management
- Not much else…

Obligatory XKCD

https://xkcd.com/1354/

Surprisingly accurate explanation!

Vulnerable BSD Systems

FreeBSD 8, 9, 10

OpenBSD 5.4, 5.5

NetBSD 5.1, 5.2, 6.0, 6.1

Exploits

- http://www.exploit-db.com/exploits/32791/
- http://www.exploit-db.com/exploits/32998/
- http://packetstormsecurity.com/files/126102/BleedOut1.0.0.10.zip
- http://www.exploit-db.com/exploits/32764/
- http://packetstormsecurity.com/files/126069/heartbleed-altered.py.txt

Exploits

- Packet Storm Security has at least 10
- Exploit DB has at least 4

Useful links

- http://heartbleed.com/
- https://www.openssl.org/news/secadv_20140407.txt
- http://ftp.openbsd.org/pub/OpenBSD/patches/5.5/common/002_openssl.patch.sig
- http://www.freebsd.org/security/advisories/FreeBSD-SA-14:06.openssl.asc
- http://ftp.netbsd.org/pub/NetBSD/security/advisories/NetBSD-SA2014-004.txt.asc

Donate!

Help the LibreSSL Project and donate!

http://www.libressl.org/

Questions?

Homepage: http://purebsd.net/

Email: jason@purebsd.net