Heartbleed, Technical Overview

Bruno Pujos

May 13, 2014

Plan

OpenSSL

- The Open Source toolkit for SSL/TLS
- Begin in 1998
- SSL/TLS
- Heartbeat - CVE-2014-0160
  - Heartbeat extension (RFC6520)
  - Release on 14/03/12 (OpenSSL 1.0.1)
  - Patch on 07/04/14 (OpenSSL 1.0.1g)

Plan

Read Overrun

A missing bounds check in the handling of the TLS heartbeat extension can be used to reveal up to 64k of memory to a connected client or server.

Only 1.0.1 and 1.0.2-beta releases of OpenSSL are affected including 1.0.1f and 1.0.2-beta1.

Thanks for Neel Mehta of Google Security for discovering this bug and to Adam Langley <agl@chromium.org> and Bodo Moeller <bmoeller@acm.org> for preparing the fix.

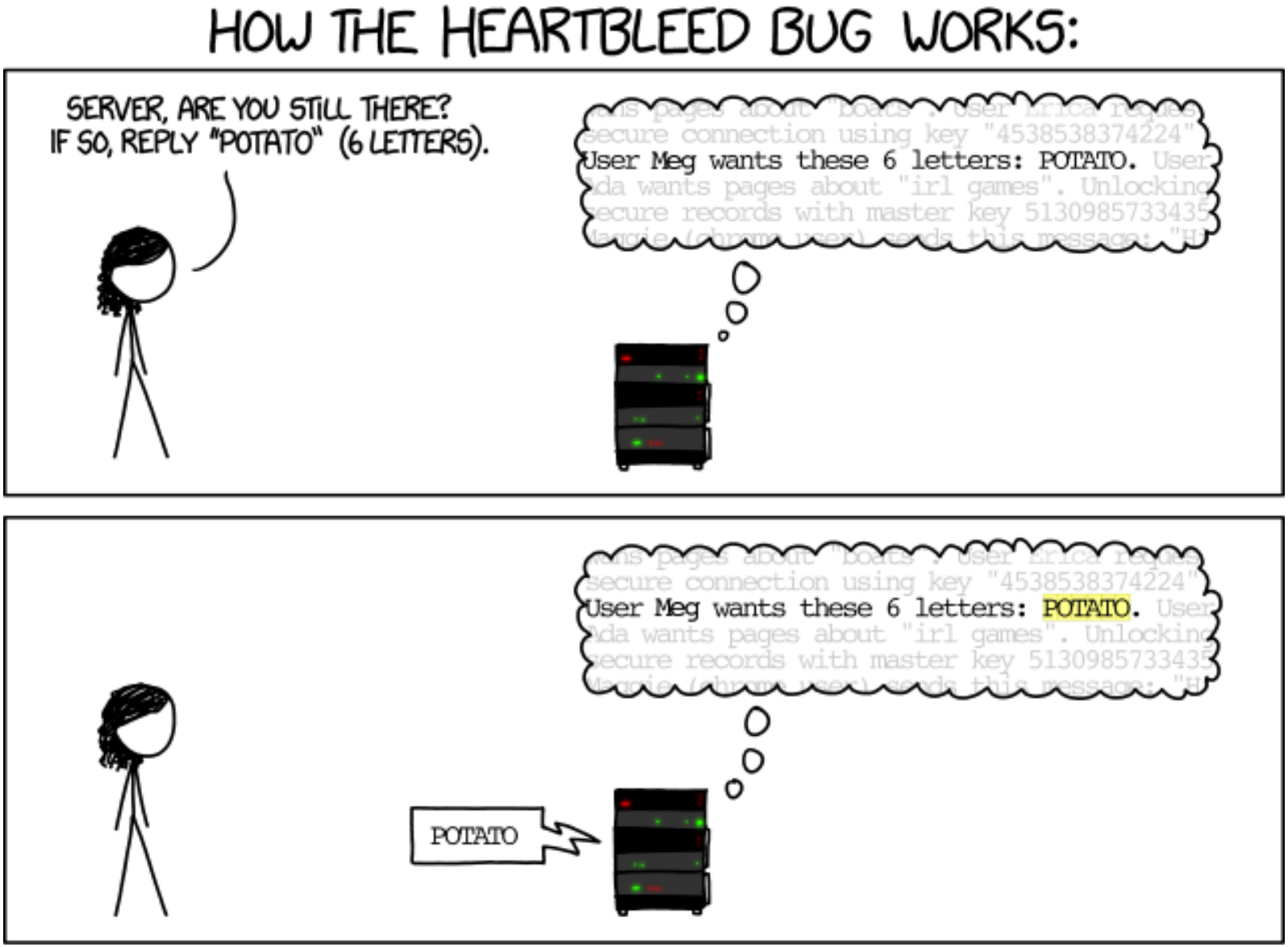
Heartbeat

- Heartbeat Extension (RFC6520)
- for TLS and DTLS (Datagram Transport Layer Security)
- design to do PMTUD (path maximum transmission unit discovery)
- 2 message types:
  - HeartbeatRequest
  - HeartbeatResponse
- Work in both way (with client attacking and with server attacking)

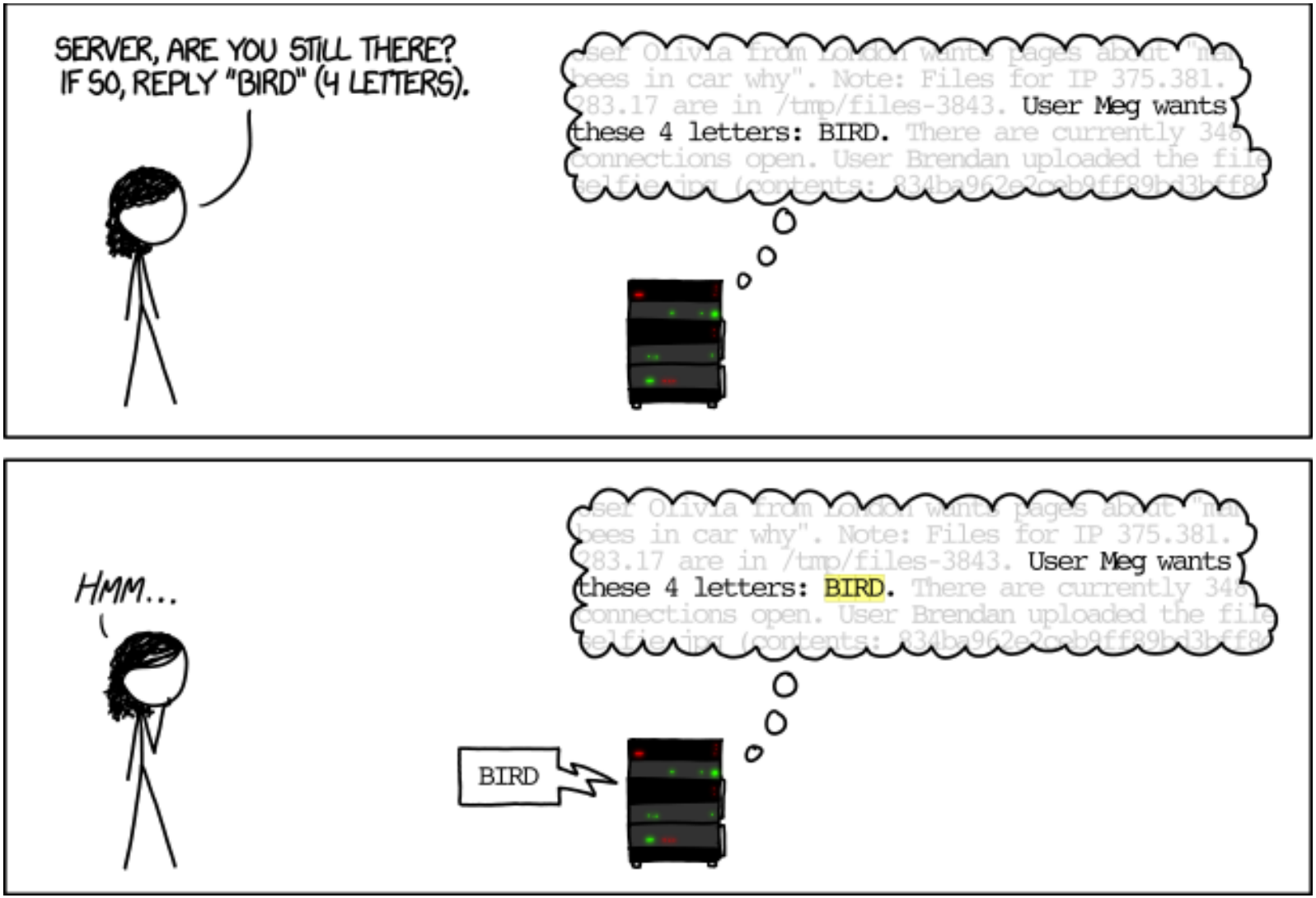
HeartbeatMessage structure

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

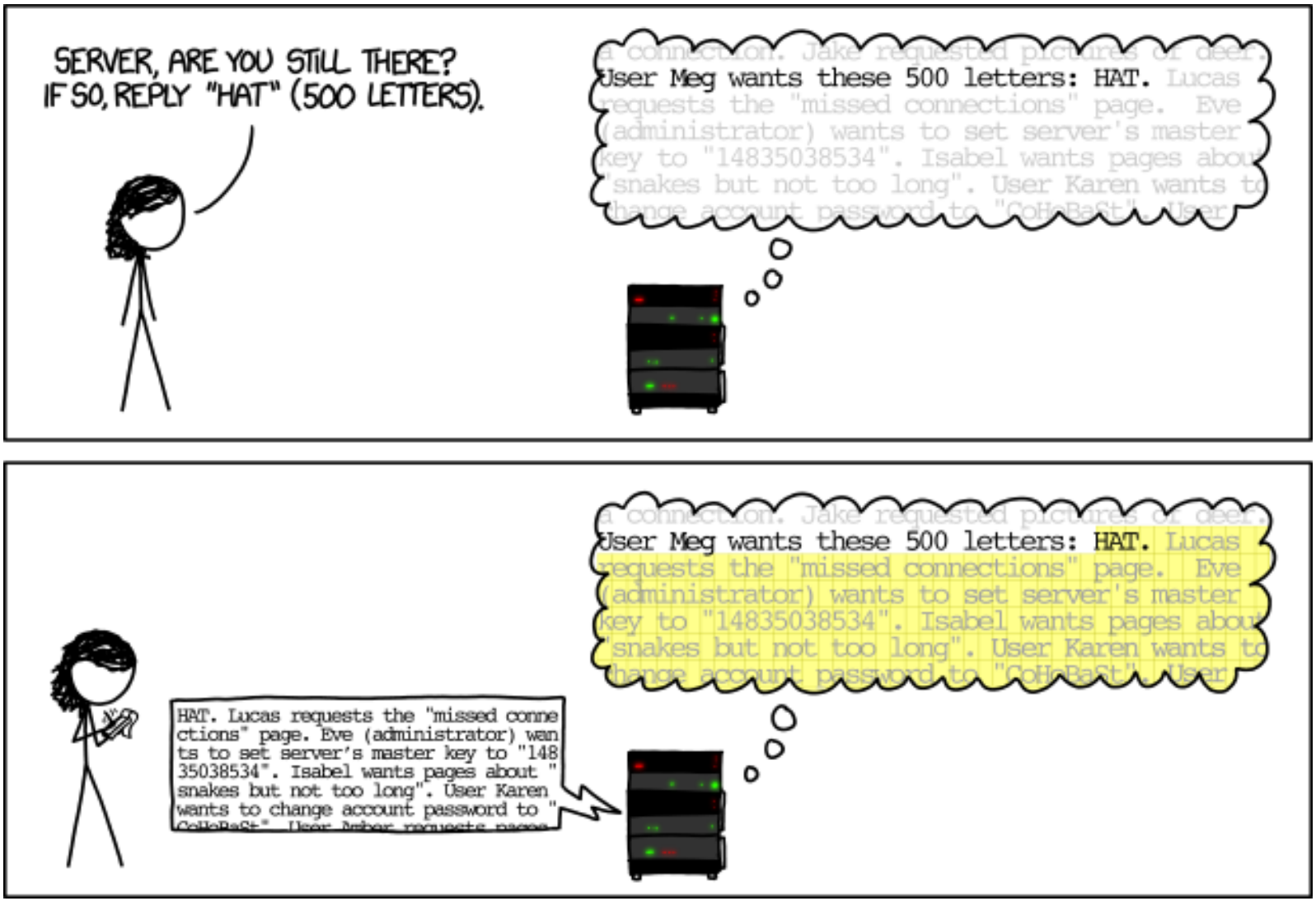
XKCD explanation



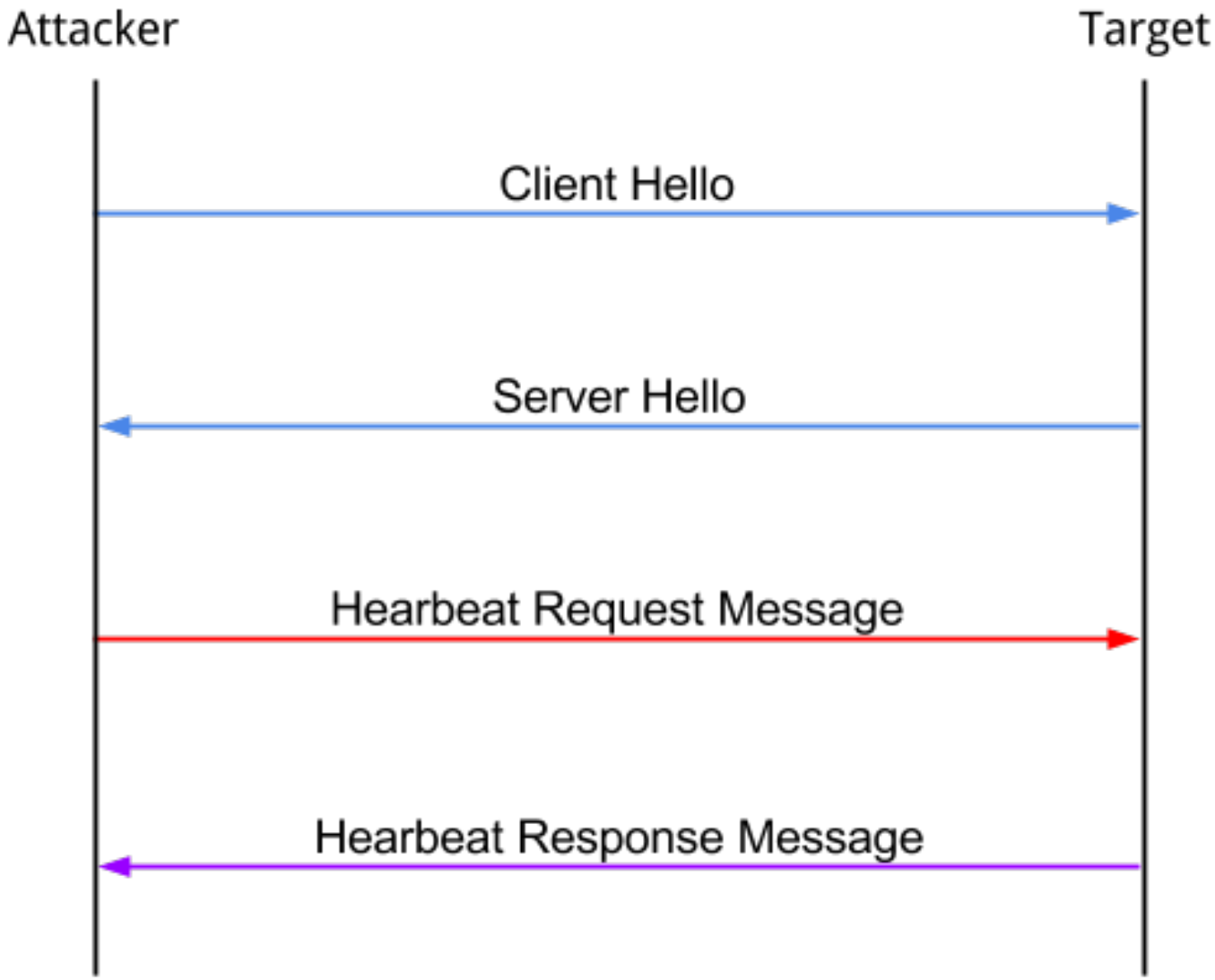
XKCD explanation



XKCD explanation



Heartbleed



Plan

Risk ?

- HTTP Headers
- Private Key
- Possibly anything from the server memory...

Private Key

- It's possible but unlikely to have directly the RSA Private Key as it in the memory
- For encrypting a message OpenSSL use the struct RSA:

RSA struct

```
struct rsa_st
{
int pad;
long version;
const RSA_METHOD *meth;
ENGINE *engine;
BIGNUM *n;
BIGNUM *e;
BIGNUM *d;
BIGNUM *p;
BIGNUM *q;
BIGNUM *dmp1;
BIGNUM *dmq1;
BIGNUM *iqmp;
CRYPTO_EX_DATA ex_data;
 ...};
```

RSA

- p and q prime numbers
- $n = pq$ modulus
- $\varphi(n) = n - (p + q - 1)$
- e (public key exponent), $1 < e < \varphi(n)$ and $gcd(e, \varphi(n)) = 1$
- d (private key exponent), $d \equiv e^{-1} (mod\ \ \varphi(n))$
- n and e are the public key, n and d are the private
- d, p, q, $\varphi(n)$, must be private
- if we get p or q it's a win

Leaking p

- p is a prime number and divide the modulus
- from the public key we have the size of p
- scan the memory for a number which divide the modulus

Why not found earlier ?

- OpenSSL code is extremely complex
- Static analysis of pointers, function pointers. . . is complex
- Over-read vulnerability is not the most common things to look for
- OpenSSL has is own memory allocation routines

Plan

Conclusion

- ”Everything” is impact
- Patch is not enough
- Firefox, Chrome, Opera don’t use OpenSSL except on Android
- Lot’s of tools will find it now
- Really interesting articles and research after this
- Lot’s of review in OpenSSL since that

Bibliography

- http://git.openssl.org/gitweb/?p=openssl.git;a=commitdiff;h=96db902
- http://tools.ietf.org/html/rfc6520
- http://heartbleed.com/
- https://github.com/einaros/heartbleed-tools
- https://hacking.ventures/rsa-keys-in-heartbleed-memory/
- http://www.lightbluetouchpaper.org/2014/04/25/heartbleed-and-rsa-private-keys/
- http://blog.trailofbits.com/
- http://www.leviathansecurity.com/blog/leviathans-mandatory-heartbleed-blog-entry/
- http://www.dwheeler.com/essays/heartbleed.html
- https://filippo.io/Heartbleed/
- ...