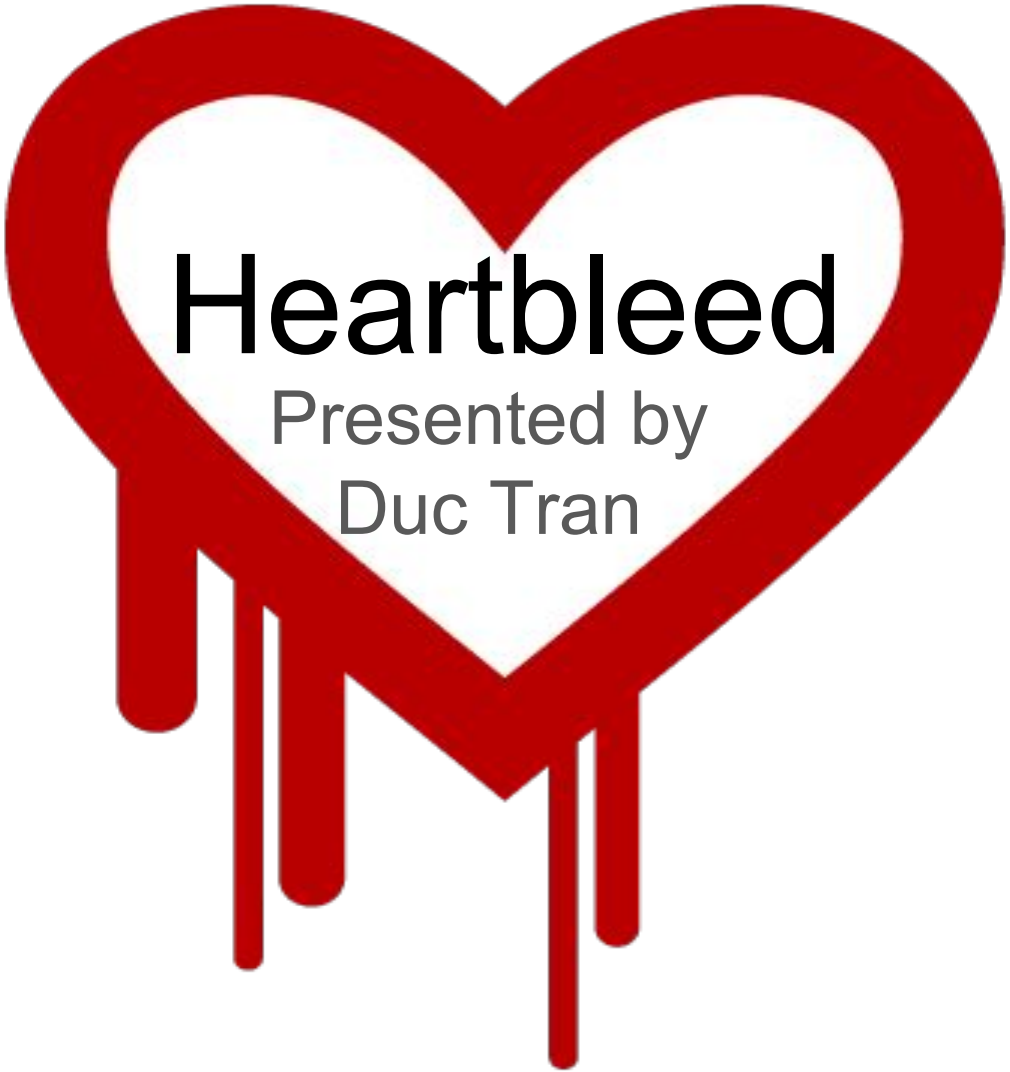
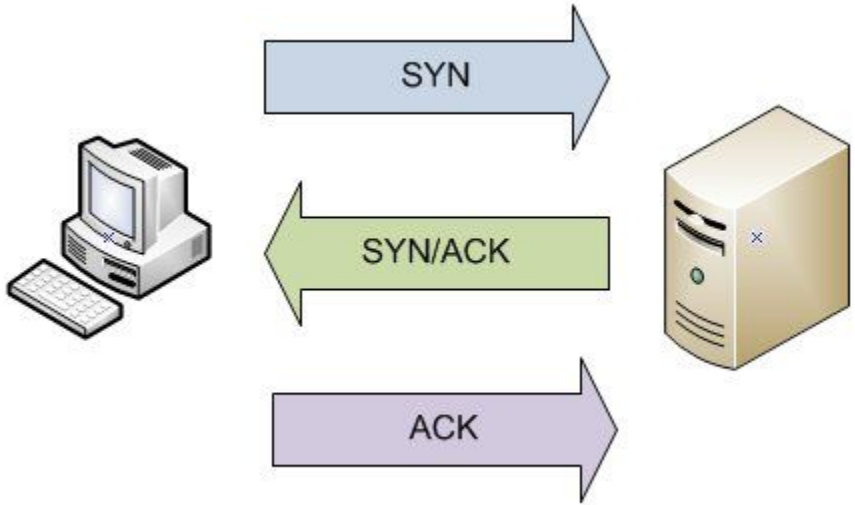

Heartbleed
Presented by
Duc Tran

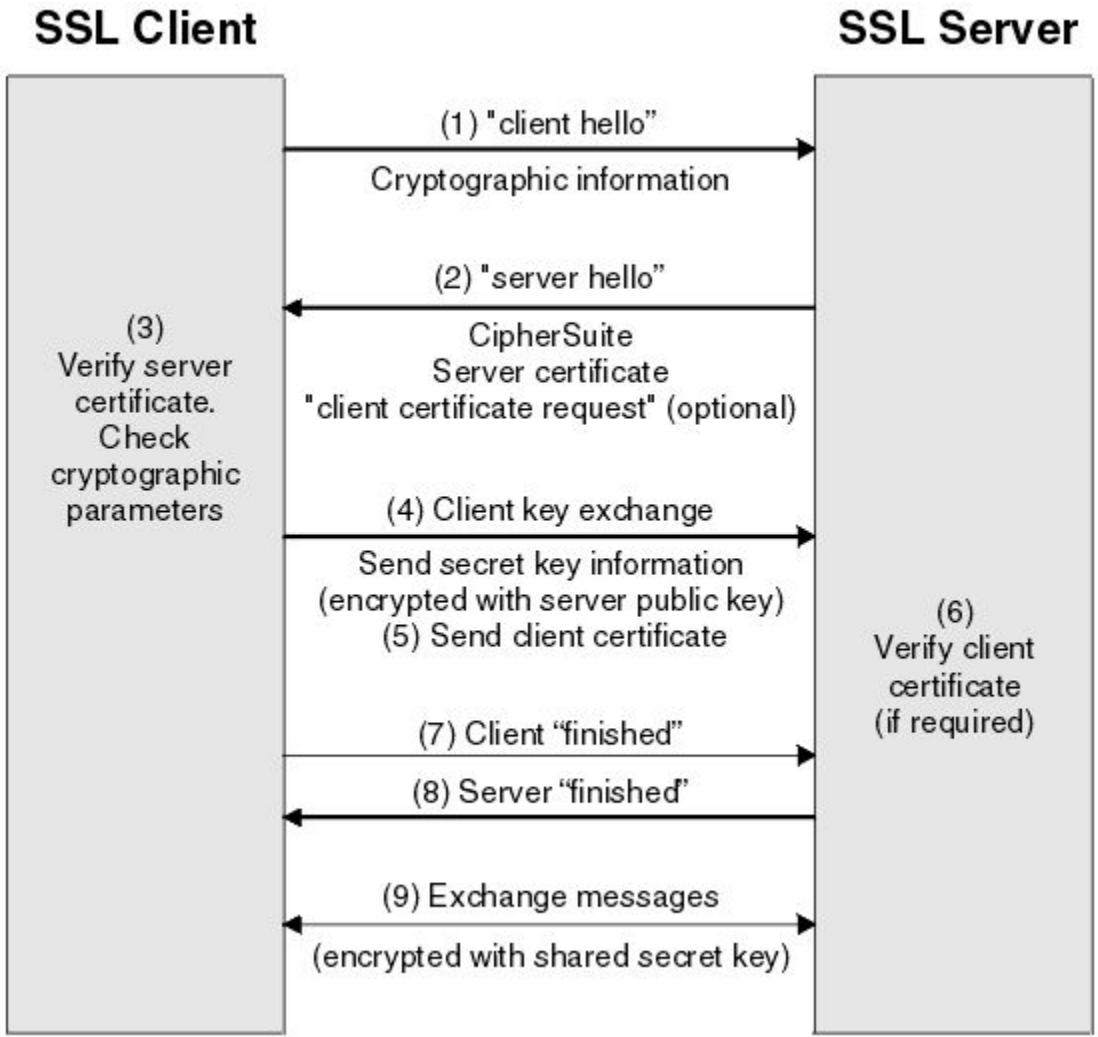
Agenda

- Background
  - TLS
  - OpenSSL
  - TLS Heartbeat Extension
- The Hearbleed Bug
- Who’s Vulnerable
- Demo
- Why it’s bad
- Protections

Background

- What is Transport Layer Security (TLS)?
  - Formerly known as Secure Socket Layer (SSL)
  - Cryptographic Protocols for encrypted communication over a network
- Initial Three-Way Handshake



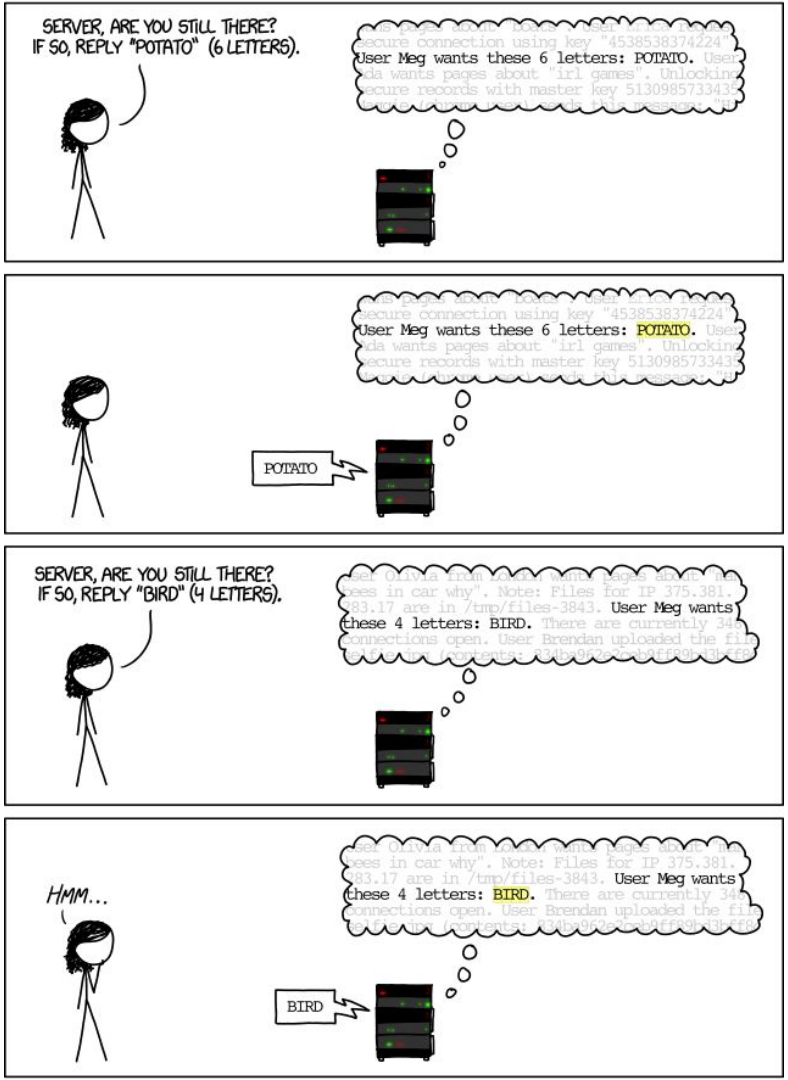

SSL Client
SSL Server
(1) "client hello"
Cryptographic information
(2) "server hello"
CipherSuite
Server certificate
"client certificate request" (optional)
(3)
Verify server
certificate.
Check
cryptographic
parameters
(4) Client key exchange
Send secret key information
(encrypted with server public key)
(5) Send client certificate
(6)
Verify client
certificate
(if required)
(7) Client "finished"
(8) Server "finished"
(9) Exchange messages
(encrypted with shared secret key)

Background

- What is OpenSSL?
  - “OpenSSL is an open source project that provides a robust, commercial-grade, and full-featured toolkit for the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols. It is also a general-purpose cryptography library.” - openssl.org
  - Used for secure connections for:
    - Web
    - Email
    - VPN
    - Messaging Services
    - Certificates
  - Most popular open source cryptographic library and TLS implementation on the internet

Background

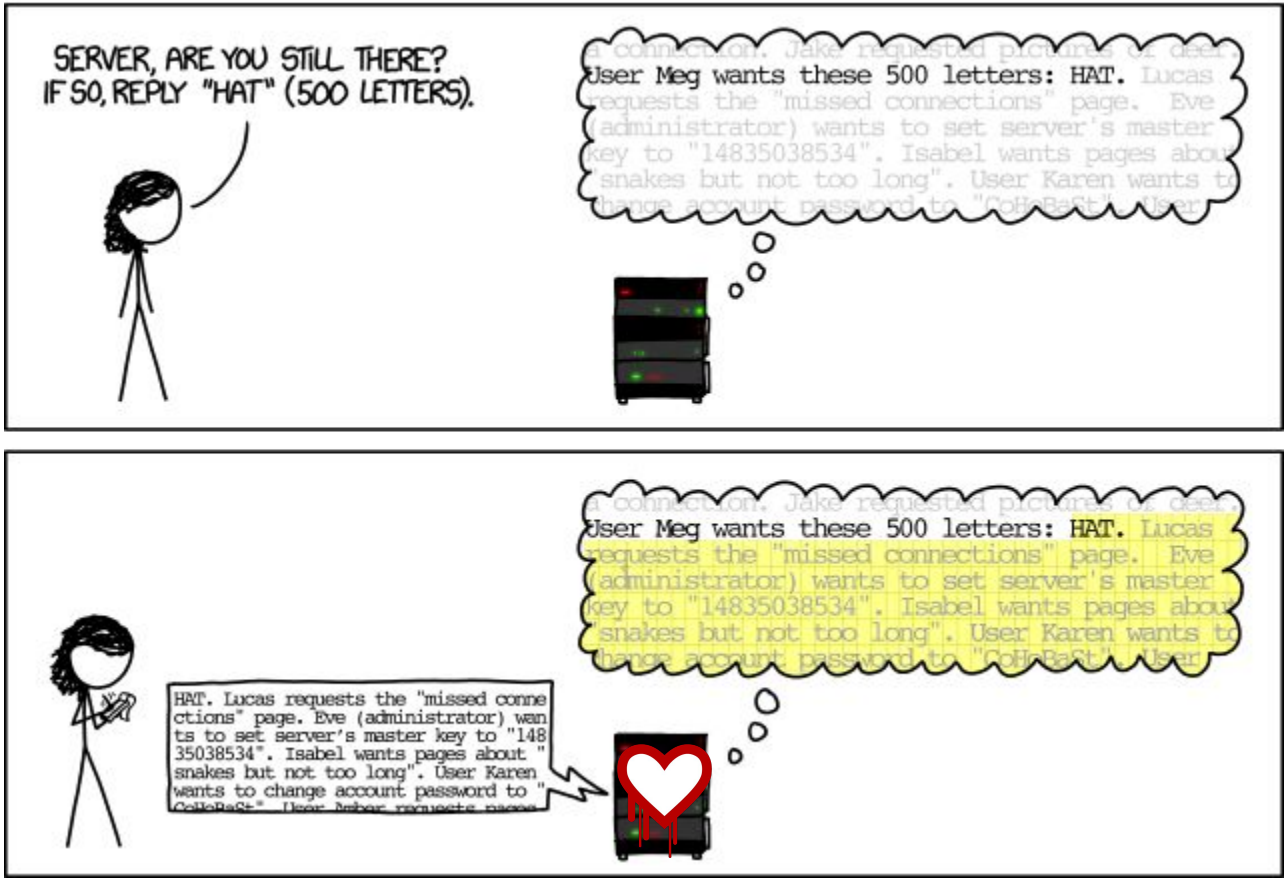
- TLS Heartbeat Extension
  - RFC 6520
  - Provides a protocol for TLS to allow the usage of the Keep-Alive functionality without continuous data transfer
  - Heartbeat Request
    - Payload
    - Payload Length
  - Heartbeat Response
    - Responds with the exact Payload that was sent
- Two Main Purposes:
  - Make sure connection does not close
  - Make sure peers are alive


SERVER, ARE YOU STILL THERE? IF SO, REPLY "POTATO" (6 LETTERS).
User Meg wants these 6 letters: POTATO.
POTATO
SERVER, ARE YOU STILL THERE? IF SO, REPLY "BIRD" (4 LETTERS).
these 4 letters: BIRD.
HMM...
BIRD

The Heartbleed Bug (CVE-2014-0160)

- Heartbleed Bug is a flaw in the implemented TLS Heartbeat Extension
  - Not a Vulnerability of TLS/SSL
- Publicly disclosed in April of 2014
- No Bounds Checking for the Heartbeat messages
  - Allows for Buffer Over-Read
- Allows for stealing information:
  - Session ID
  - Private Keys
  - Passwords
  - Usernames
  - E-mails
  - more…...

HOW THE HEARTBLEED BUG WORKS:



OpenSSL Git Logs

Original Code -->

Checks for Empty Payload -->

Makes sure payload length is not too large -->

Another check for the Heartbeat message -->

```
diff --git a/ssl/d1_both.c b/ssl/d1_both.c
index 7a5596a..2e8cf68 100644 (file)
--- a/ssl/d1_both.c
+++ b/ssl/d1_both.c
@@ -1459,26 +1459,36 @@ dtls1_process_heartbeat(SSL *s)
        unsigned int payload;
        unsigned int padding = 16; /* Use minimum padding */

-       /* Read type and payload length first */
-       hbtype = *p++;
-       n2s(p, payload);
-       pl = p;
-
        if (s->msg_callback)
                s->msg_callback(0, s->version, TLS1_RT_HEARTBEAT,
                        &s->s3->rrec.data[0], s->s3->rrec.length,
                        s, s->msg_callback_arg);

+       /* Read type and payload length first */
+       if (1 + 2 + 16 > s->s3->rrec.length)
+               return 0; /* silently discard */
+       hbtype = *p++;
+       n2s(p, payload);
+       if (1 + 2 + payload + 16 > s->s3->rrec.length)
+               return 0; /* silently discard per RFC 6520 sec. 4 */
+       pl = p;
+
        if (hbtype == TLS1_HB_REQUEST)
                {
                unsigned char *buffer, *bp;
+               unsigned int write_length = 1 /* heartbeat type */ +
+                                           2 /* heartbeat length */ +
+                                           payload + padding;
                int r;

+               if (write_length > SSL3_RT_MAX_PLAIN_LENGTH)
+                       return 0;
+
                /* Allocate memory for the response, size is 1 byte
                 * message type, plus 2 bytes payload length, plus
                 * payload, plus padding
                 */
-               buffer = OPENSSL_malloc(1 + 2 + payload + padding);
+               buffer = OPENSSL_malloc(write_length);
                bp = buffer;
```

Who’s Vulnerable

OpenSSL versions:

- 1.0.1 [14 March 2012]
- 1.0.1a
- 1.0.1b
- 1.0.1c
- 1.0.1d
- 1.0.1e
- 1.0.1f
- 1.0.1g [07 April 2014 - Heartbleed Patch]

Shodan.io Links

Shodan is a search engine for Internet Connected Devices

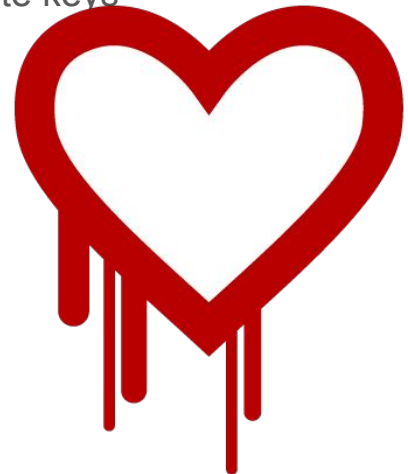
We can use it to look for servers using vulnerable versions of OpenSSL

- https://www.shodan.io/search?query=OpenSSL+1.0.1a+port%3A%22443%22
- https://www.shodan.io/search?query=OpenSSL+1.0.1a+port%3A%228443%22
- https://www.shodan.io/search?query=OpenSSL+1.0.1b+port%3A%22443%22
- https://www.shodan.io/search?query=OpenSSL+1.0.1c+port%3A%22443%22
- https://www.shodan.io/search?query=OpenSSL+1.0.1d+port%3A%22443%22
- https://www.shodan.io/search?query=OpenSSL+1.0.1e++port%3A%22443%22&page=5
- https://www.shodan.io/search?query=OpenSSL+1.0.1f+port%3A%22443%22

Demo

Why Heartbleed was bad

- Exposed large amount of private keys, secrets, and critical information
- Attack was relatively easy and left no trace
- Hundred of thousands of servers were vulnerable
- Certificate Renewal and Revocation
  - 30,000 of the 500,000+ possible compromised X.509 certificates by April 11, 2014
  - 43% by May 9, 2014 …… 7% reissued with potentially compromised private keys
- OpenSSL vulnerable to Heartbleed for a long time
  - March 2012 - April 2014



Protection from Heartbleed

Update OpenSSL to version 1.0.1g or greater!

If cannot update OpenSSL version, recompile OpenSSL with compile time option: `-DOPENSSL_NO_HEARTBEATS`

Questions?

References

OpenSSL

- https://openssl.org/

TLS Heartbeat Extension

- https://tools.ietf.org/html/rfc6520

Heartbleed

- http://heartbleed.com/
- https://git.openssl.org/gitweb/?p=openssl.git;a=commitdiff;h=96db9023b881d7cd9f379b0c154650d6c108e9a3
- https://jhalderm.com/pub/papers/heartbleed-imc14.pdf
- https://xkcd.com/1354/

Demo

- https://alexandreborgesbrazil.files.wordpress.com/2014/04/hearbleed_attack_version_a_1.pdf
- https://gist.github.com/akenn/10159084