Heartbleed

RFC6520 defines SSL Heartbeats - What are they?

1. SSL Heartbeats are used to keep a connection alive without the need to constantly renegotiate the SSL session.

2. Used in MTU path discovery

Why is this a problem? Heartbeat requests can be sent WITHOUT authentication to the server.

Open SSL Heartbleed bug
What exactly is bleeding?

- The bug allows attackers to grab 64K chunks of memory contents near the SSL heartbeat on a vulnerable host.
- It is random chunks of data in this memory space – ASLR helps in this situation
- Attack can be repeated many times to grab different random chunks of data
- 64k does seem like much - but it is!

Memory disclosure: what exactly can an attacker get?

- 1. Private crypto keys - the keys to the kingdom, or at least the server.
- 2. Usernames and Passwords
- 3. Session identifiers
- 4. Private data – data payloads
- 5. Meta data for the SSL session, programming structures may defeat other exploit protections.
TLS – Bleeding the Heart

**How the Heartbleed Bug Works:**

1. **Step 1:** Eve impersonates a trusted entity.
2. **Step 2:** Eve encrypts data using a vulnerable protocol.
3. **Step 3:** Eve waits for Heartbleed to occur.

**User Meg wants these 500 letters: HAC.**

**User wonders:**

- Eve requests the "missed connections" page.
- Eve (administrator) wants to set server’s master key to "14850859584".
- Isabel wants pages about snakes but not too long.
- User Karen wants to change account password to "expensive".

**User wonders:**

- User Meg wonders.
- User Megan wonders.
- User Karen wonders.
- User Karen wonders.
TLS - Heartbeat

typedef struct ss13_record_t {
  int type; /* type of record */
  unsigned int length; /* How many bytes available */
  unsigned int off;  /* read/write offset into 'b */
  unsigned char *data; /* pointer to the record data */
  unsigned char *input; /* where the decode bytes are */
  unsigned char comp; /* only used with decompress */
  unsigned long epoch; /* epoch number, needed by D */
  unsigned char seq_num[8]; /* sequence number, handed */
} SS13_RECORD;

TLS – Fixing the Bleed

/* Read type and applied length first */
if (p != 1 + 3 + strrev(length).length) {
  return (-1); silently discarded */
  hsize = (p); /*
  copy back to */
  if (strrev(length).length) {
    return (-1); silently discarded per RFC and ass. x */
  } p = p + 3;
}
if (type == TLS_RSA_PKEY) {
  unsigned char *key = key_length - 1; /* heartbeat type */
  p = p + 1;
}
}
Thank you!