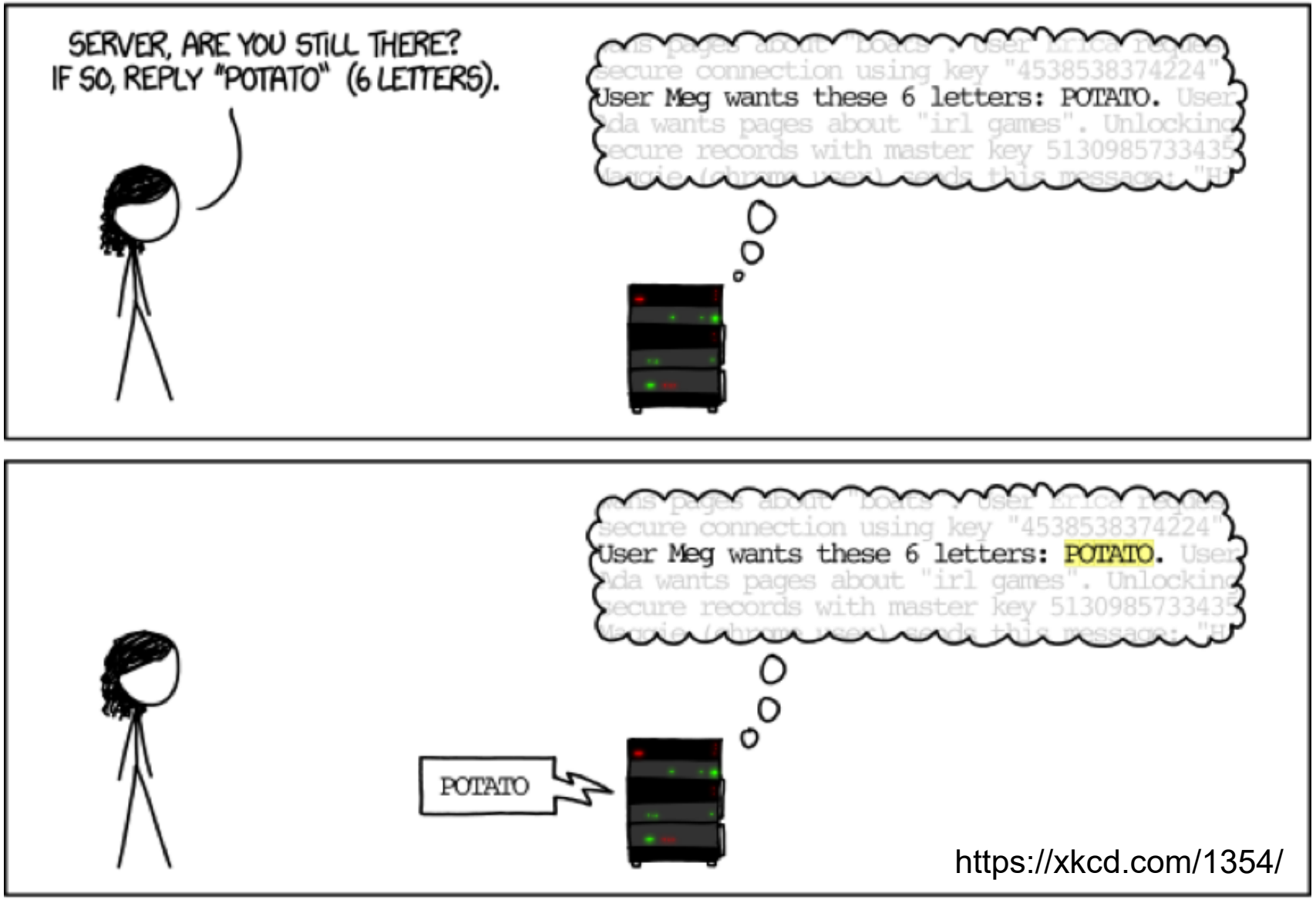
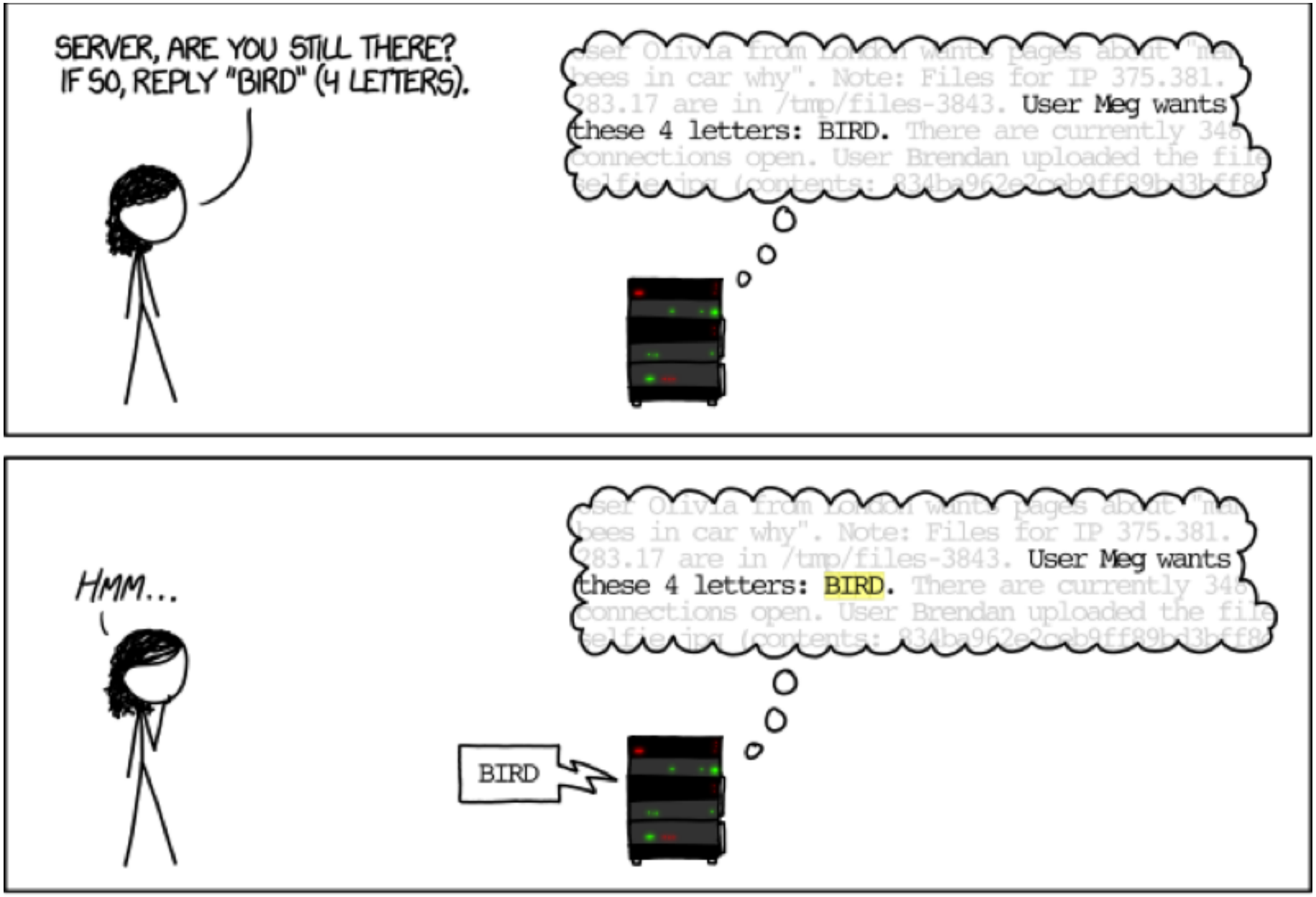
Case study: Heartbleed

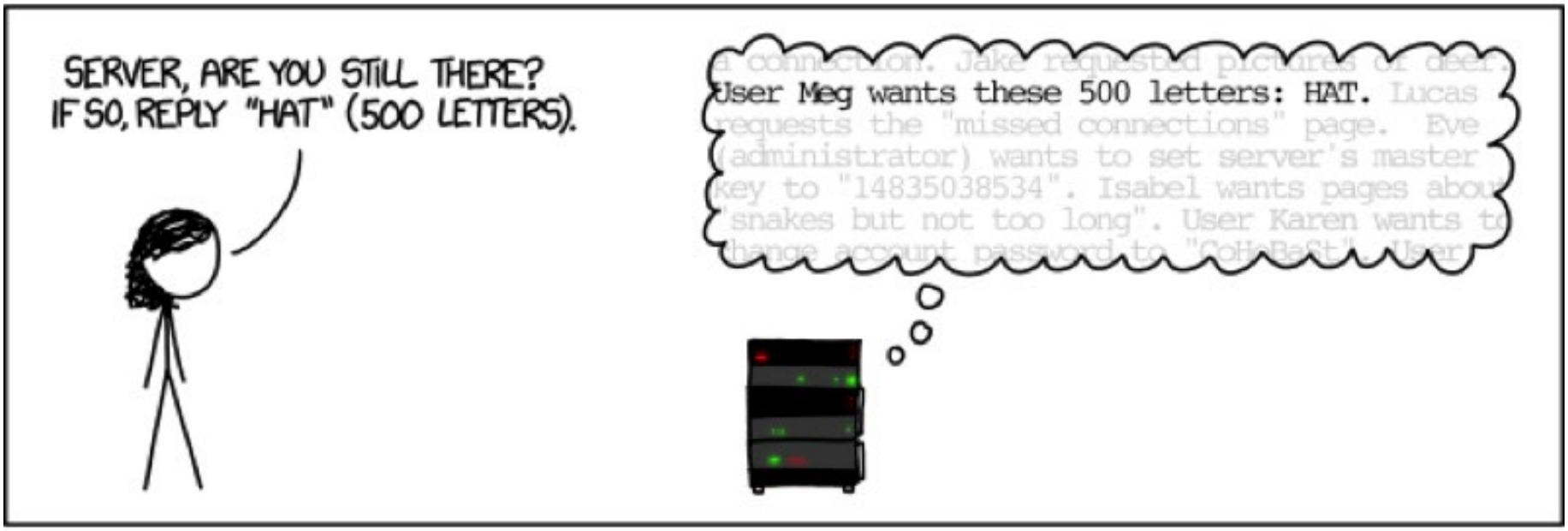
- SSL is the main protocol for secure (encrypted) online communication

- Heartbleed was a vulnerability in the most popular SSL server

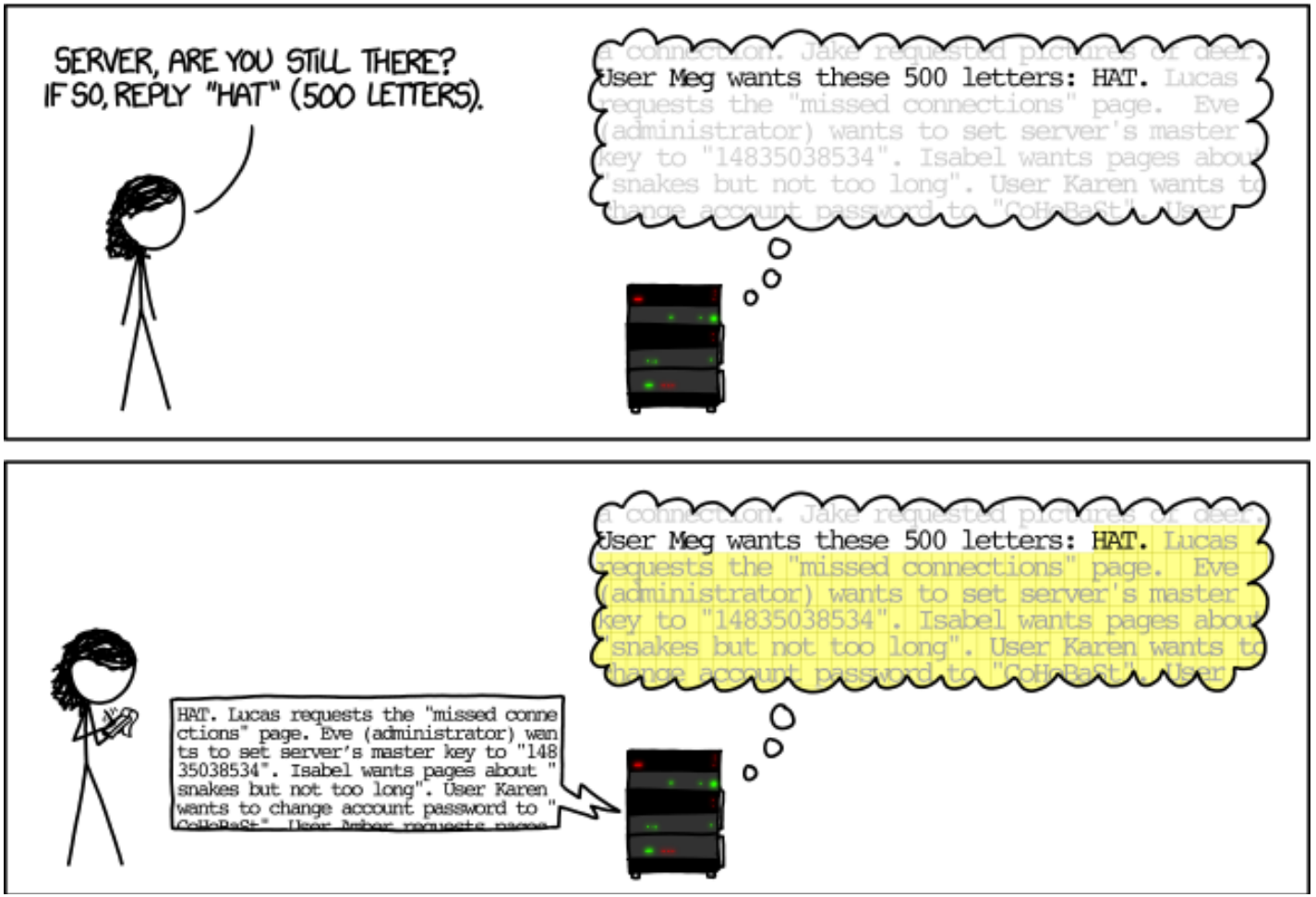
HOW THE HEARTBLEED BUG WORKS:




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


SERVER, ARE YOU STILL THERE? IF SO, REPLY "HAT" (500 LETTERS).
User Meg wants these 500 letters: HAT.

Heartbleed:
A Closer Look at Buffer Read Overflow



Case study: Heartbleed

- SSL is the main protocol for secure (encrypted) online communication
- Malformed packet allows you to see server memory
  - Passwords, keys, emails, visitor logs …..
- Fix: Don’t let the user tell you how much data to send back!
  - This is a *design* flaw

Heartbleed:
A Closer Look at Buffer Read Overflow

- Read Overflow: A bug that permits reading past the end of a buffer.

Read integer

Read message

Echo back (partial) message

```
int main() {
 char buf[100], *p;

 while (1) {
  p = fgets(buf,sizeof(buf),stdin);
  len = atoi(p);
  p = fgets(buf,sizeof(buf),stdin);
  for (i=0; i<len; i++) {
   if (!iscntrl(buf[i]))
    putchar(buf[i]);
   else putchar('.');
  }
  printf("\n");
 }
 ...
```

len may exceed actual message length!

Heartbleed:
A Closer Look at Buffer Read Overflow

- Sample Output:

```
% ./echo-server
24
every good boy does fine
ECHO: |every good boy does fine|
10
hello there
ECHO: |hello ther|
25
hello
ECHO: |hello..here..y does fine.|
```

OK: input length < buffer size

BAD: length > size !

leaked data