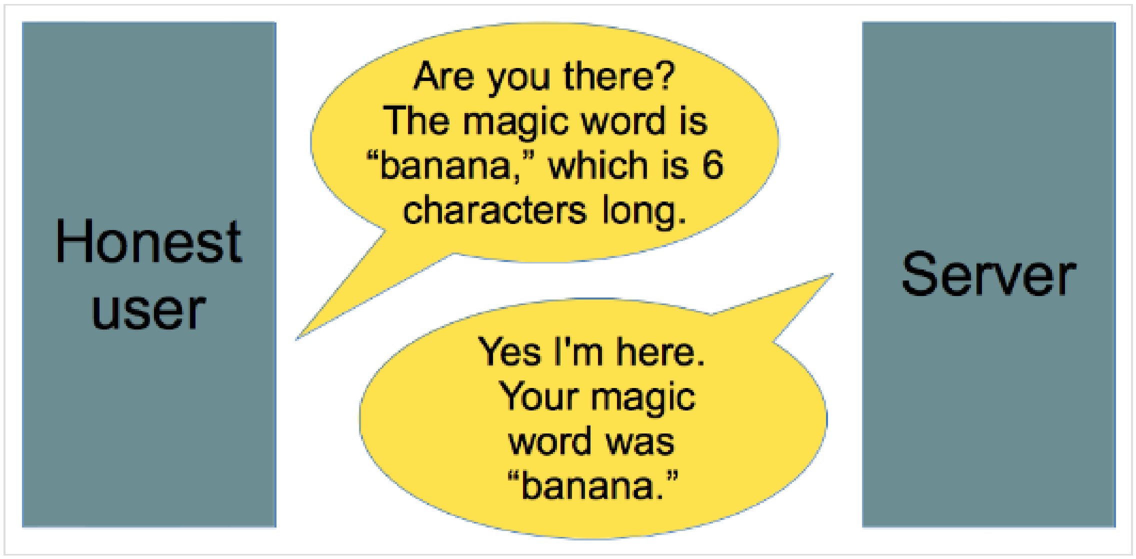
How does the heartbleed attack work?

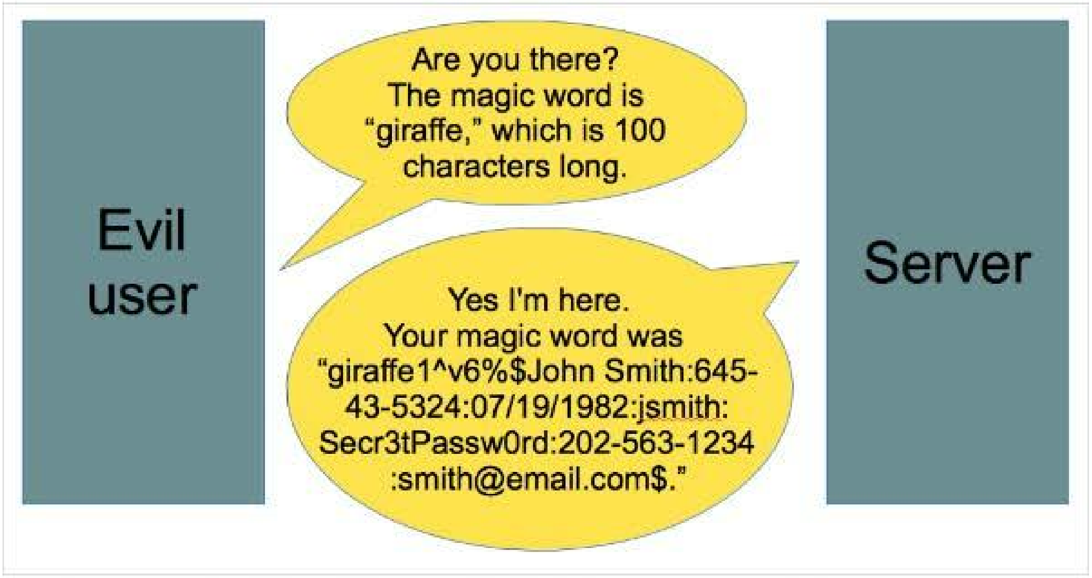
The SSL standard includes a "heartbeat" option, which provides a way for a computer at one end of the SSL connection to double-check that there's still someone at the other end of the line. This feature is useful because some internet routers will drop a connection if it's idle for too long. In a nutshell, the heartbeat protocol works like this:


Honest user
Are you there? The magic word is "banana," which is 6 characters long.
Server
Yes I'm here. Your magic word was "banana."

The heartbeat message has three parts: a request for acknowledgement, a short, randomly-chosen message (in this case, "banana"), and the number of characters in that message. The server is simply supposed to acknowledge having received the request and parrot back the message.

The heartbeat message has three parts: a request for acknowledgement, a short, randomly-chosen message (in this case, "banana"), and the number of characters in that message. The server is simply supposed to acknowledge having received the request and parrot back the message.

The Heartbleed attack takes advantage of the fact that the server can be too trusting. When someone tells it that the message has 6 characters, the server automatically sends back 6 characters in response. A malicious user can take take advantage of the server's gullibility:


Evil user
Are you there? The magic word is "giraffe," which is 100 characters long.
Yes I'm here. Your magic word was "giraffe1^v6%$John Smith:645-43-5324:07/19/1982:jsmith: Secr3tPassw0rd:202-563-1234 :smith@email.com$."
Server

Obviously, the word "giraffe" isn't 100 characters long. But the server doesn't bother to check before sending back its response, so it sends back 100 characters. Specifically, it sends back the 7-character word "giraffe" followed by whichever 93 characters happen to be stored after the word "giraffe" in the server's memory. Computers often store information in a haphazard order in an effort to pack them into its memory as tightly as possible, so there's no telling what information might be returned. In this case, the bit of memory after the word "giraffe" contained sensitive personal information belonging to user John Smith.

In the real Heartbleed attack, the attacker doesn't just ask for 100 characters. The attacker can ask for around 64,000 characters of plain text. And it doesn't just ask once, it can send malicious heartbeat messages over and over again, allowing the attacker to get back different fragments of the server's memory each time. In the process, it can gain a wealth of data that was never intended to be available to the public.

The fix for this problem is easy: the server just needs to be less trusting. Rather than blindly sending back as much data as is requested, the server needs to check that it's not being asked to send back more characters than it received in the first place. That's exactly what OpenSSL's fix for the Heartbleed Bug does.

Here is the online link for this page.

http://www.vox.com/cards/heartbleed/how-does-the-heartbleed-attack-work#E5357695