lab8

Name: 黎诗龙

SID: 11811407

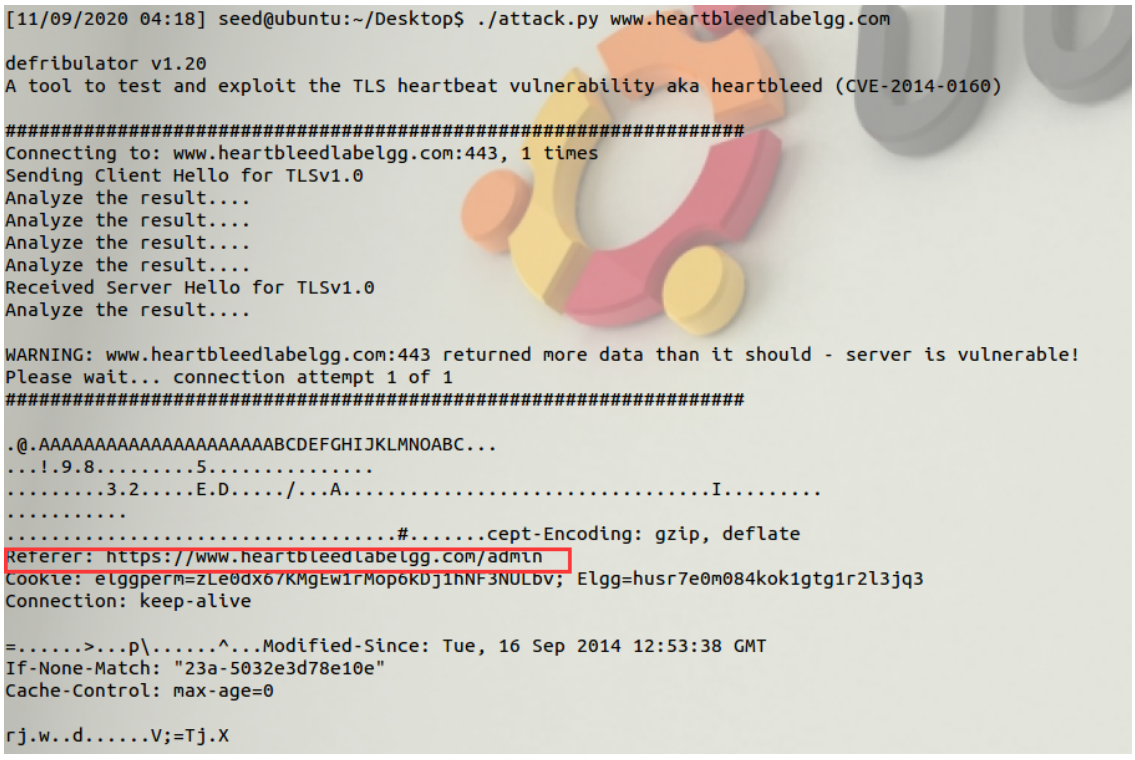
Task1

- user name and password

`username = admin`, `password=seedelgg`.

- User's activity.

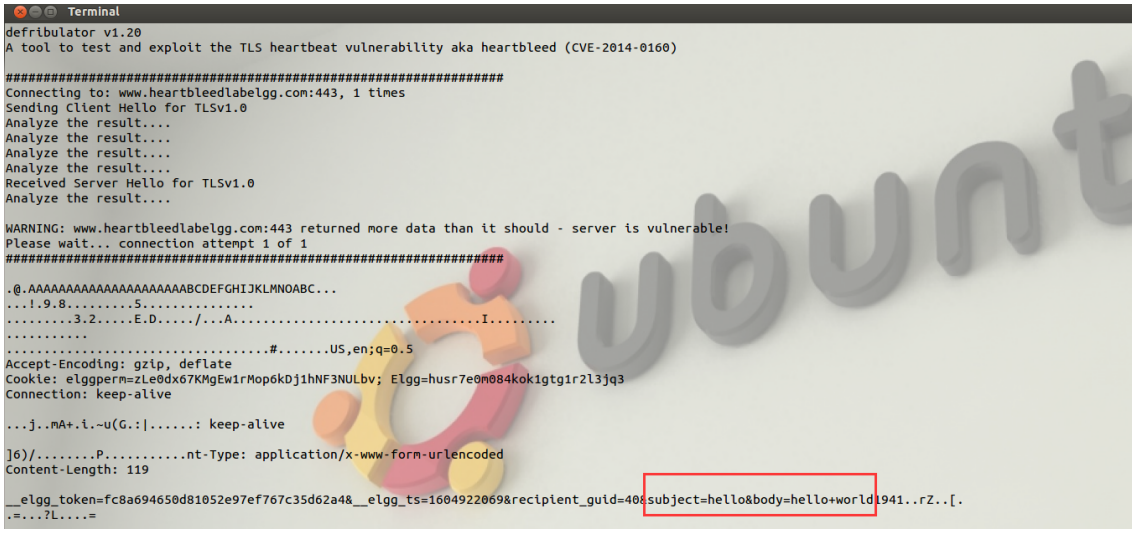
  Here shows that I browse the `URL=https://www.heartbleedlabelgg.com/admin`



  Here shows that

- The exact content of the private message

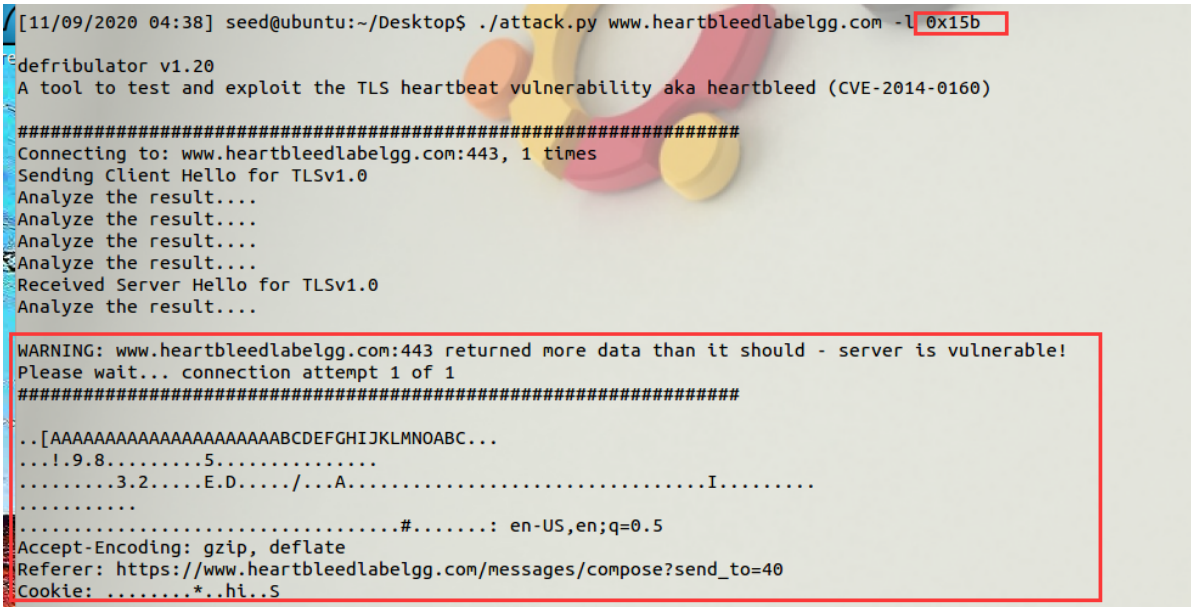
  Here the terminal shows that I send a `subject=hello`, and `body=hello world`.



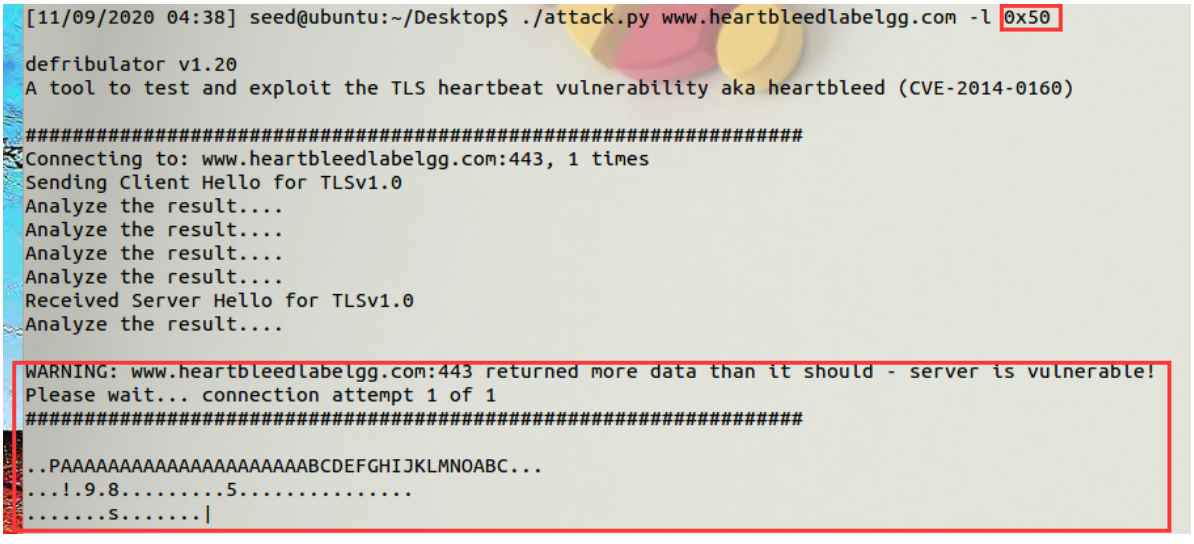
Task2

Question 2.1

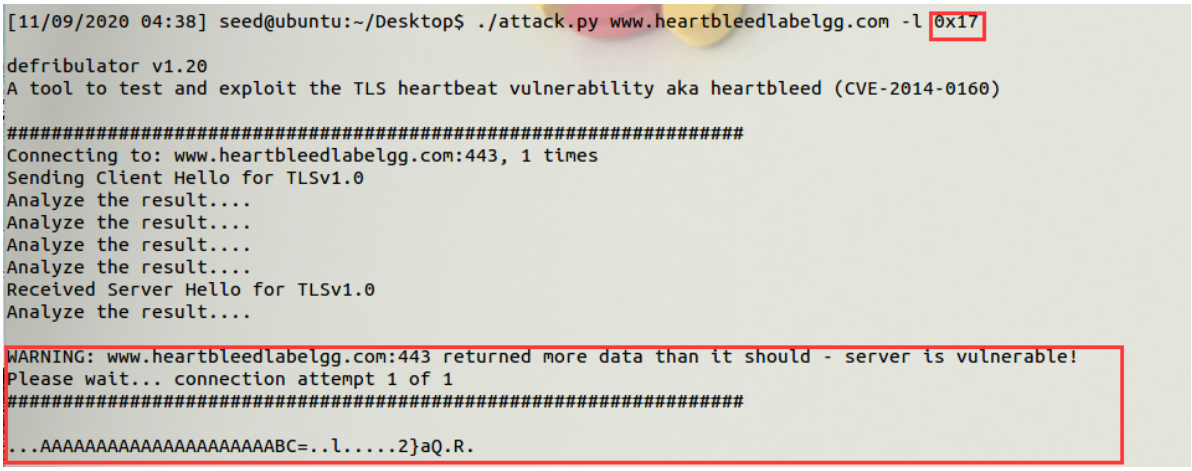
Here is length of `0x15b`.



Here is length of `0x50`.



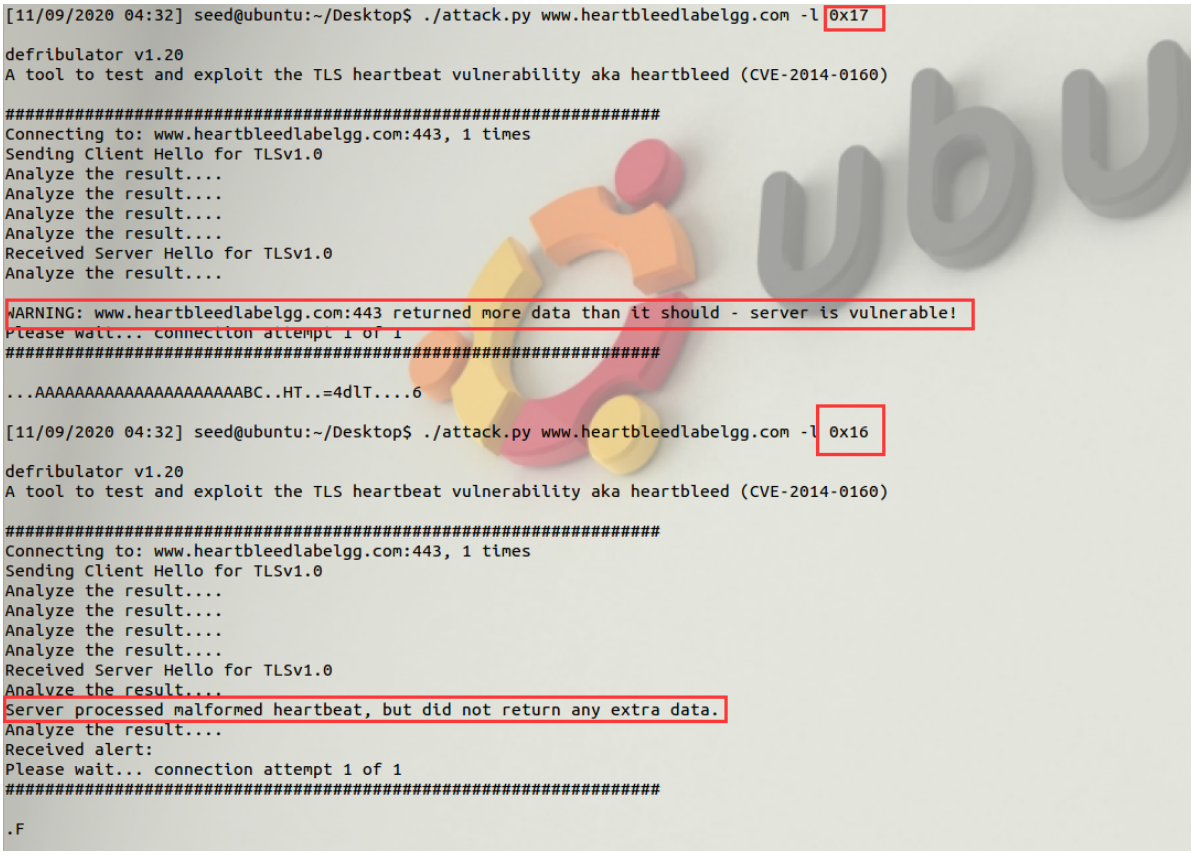
Here is length of `0x17`.



From the pictures above we see that as the length variable decreases, the response packet content length decreases.

Question 2.2

The boundary length is `0x16`.



Task3

3.1

It doesn't work.

3.2

In the surrounding of `hbtype = *p++;`, it misses the boundary checking during the buffer copy.

Solution:

Just check the boundary checking of the length next to `hbtype=*p++`, if the length is more than the packet length, then end this process.

The code is in the lecture ppt.

I think Alice and Bob's comments make sense.

The Heartbleed bug got its start from improper input validation in the OpenSSL implementation of the TLS Heartbeat extension. Due to the missing bounds check on the length and payload fields in Heartbeat requests, coupled with trusting the data received from other machines, the responding machine mistakenly sends back its own memory data. [Refer to https://www.synopsys.com/blogs/software-security/heartbleed-bug/].

I think Eva's comment is not so reasonable, because the length in the packet may be used in another standard of communications defined previously. If it were removed, then many standards might be modified.