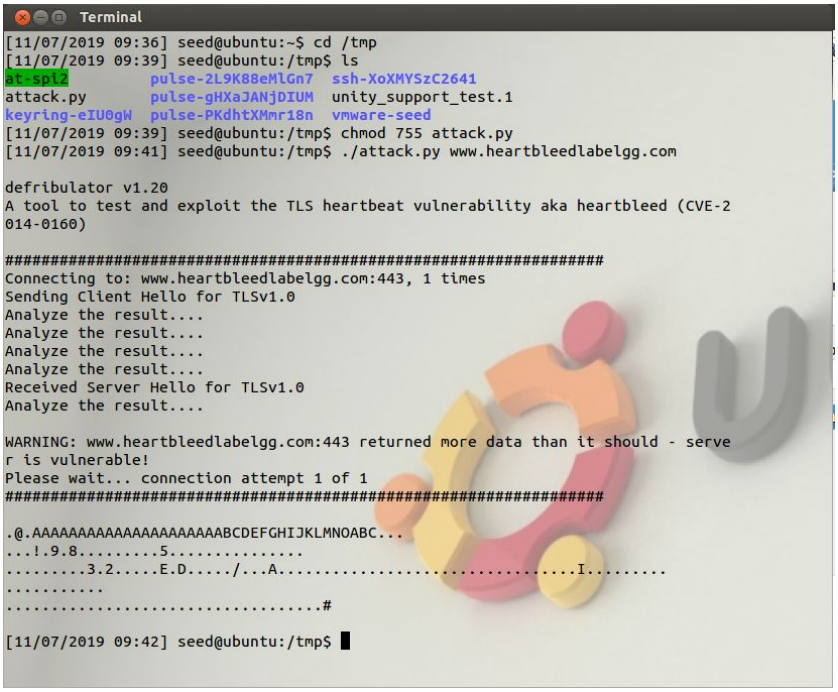
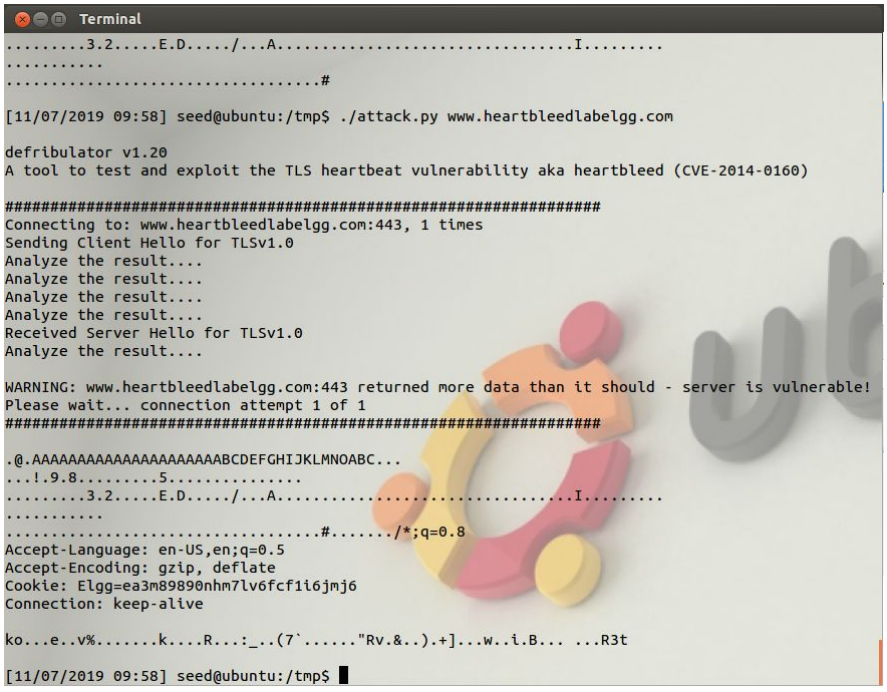
Aaron Cheung
CSC 154
Lab 4 Heartbleed Report



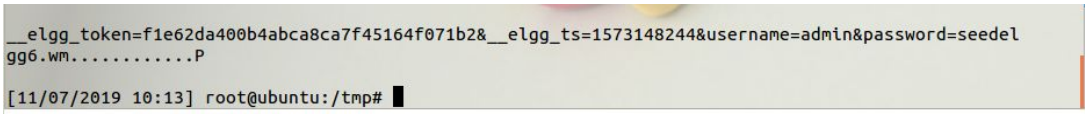
1) The first run of the heartbleed attack. We firstly had to do chmod 755 attack.py to change the permissions of the downloaded file before executing the program on www.heartbleedlabelgg.com.



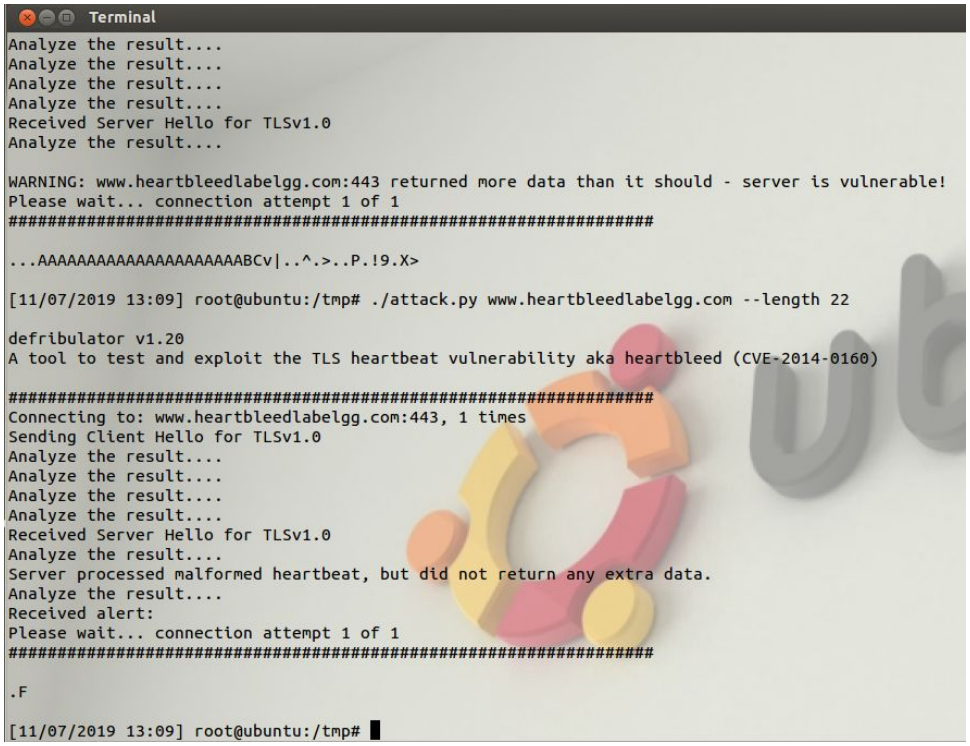
2)
After running the program *numerous* times, I finally got something different. Under the lengthy random string of characters we can see the accept-language, accept-encoding, cookie, and connection.

3)

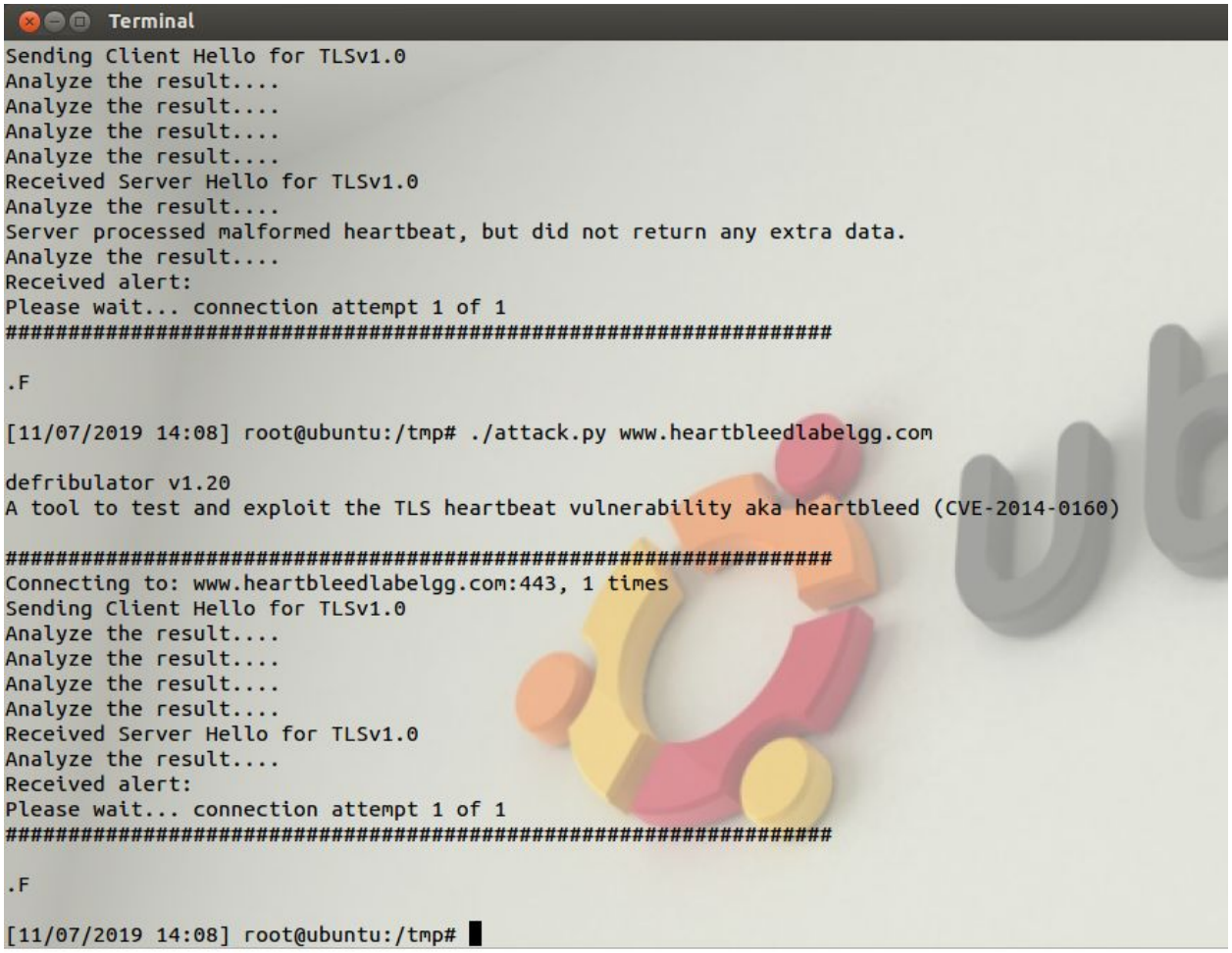
Here we can see a private message at the very bottom of the execution. They are the lines under “content length”.



4) We manage to get the username and password



5) We play around with the --length parameter and find that the server processed malformed heartbeat, but it did not return any extra data, rather than the usual message saying the server returned more data than it should. The length parameter used was 22, and when used with 23, the latter message is displayed. This means that the “threshold” for the length parameter appears to be 22. From this, I can conclude that the more the length decreases, the less data is returned.



6) After updating the victim machine, the heartbleed attack seems to no longer work. This must be because when updating the victim machine, the OpenSSL also got updated to fix the heartbleed bug.

```
38
39      // copy payload
40      memcpy(bp, pl, payload); /* pl is the pointer which
41                           * points to the beginning
42                         * of the payload content */
43
44      bp += payload;
```

7) This are the lines of code that bugs out OpenSSL. The fix is as follows:

```
if (1 + 2 + 16 > s -> s3 -> relent)
        return 0;

hbtype = *p++;

n2s(p, payload);

if (1 + 2 + payload + 16 > s-> s3 -> rrec.length)
        return 0;

p1 = p;
```

The second if statement is a bounds check to make sure that the length of the request matches the actual length.