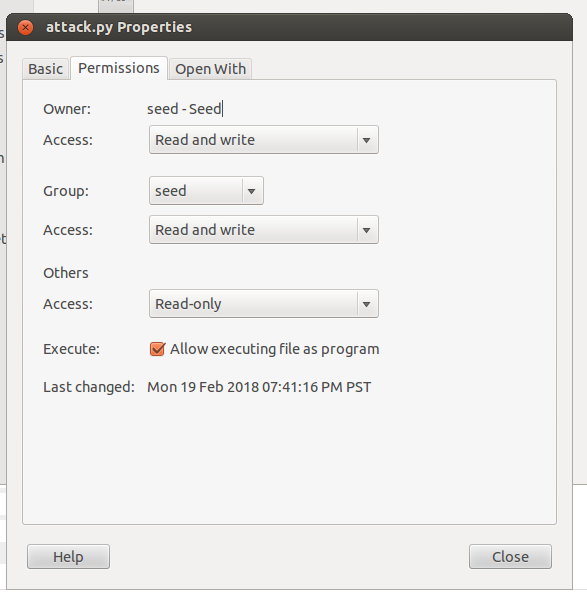
Daniel Oliveros

Garrett Bogart

Spring 2018 – Independent Study

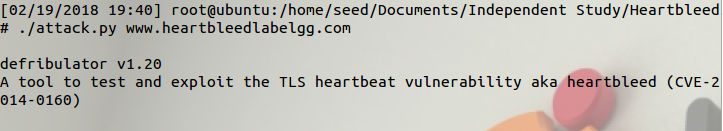
Heartbleed Vulnerability Lab

In order to get the attack file in a place where we could run it, we first needed to check the box in its properties named “Allow executing File as program”

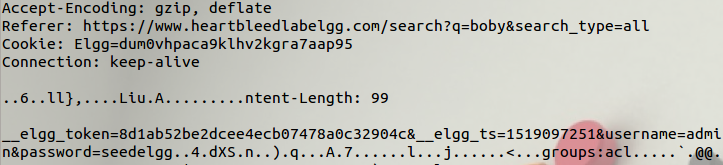


**Task 1:**

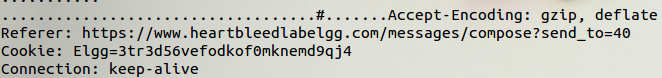
We ran the attack program multiple times to gather the information.



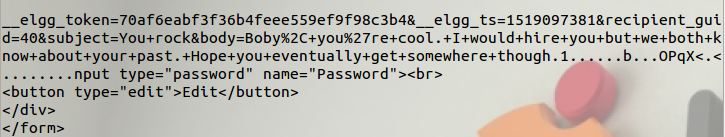
It can be seen here that we have the username and password of the admin



This image shows that the user sent a message to user 40. We also have the cookie of the user



This is the message that was sent,



**Task 2:**

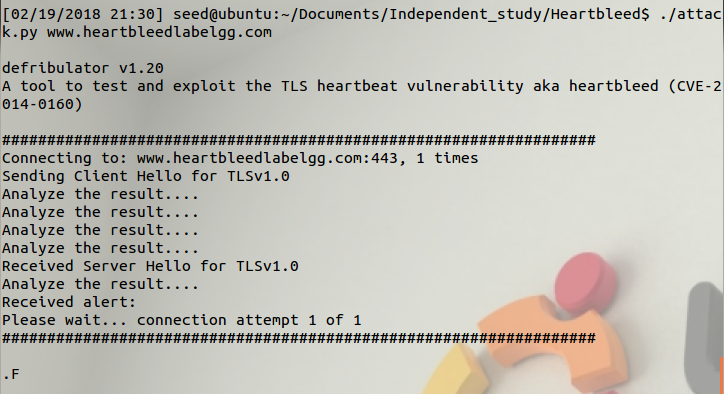
**2.1.** As the length variable decreases, the cap of how much information we can get from our request decreases as well.

**2.2.**



We tested the boundary length various times, it would usually still return a decent amount of data. Once we hit a length of 22, the output was always the same, just a “.F”. We interpreted this as the point past the boundary length, meaning that the boundary length for this attack is 23.

**Task 3.1.**



**Task 3.2**

****

To prevent the heartbleed attack we only have to add a single if statement to check the payload size. The s-s3->rrec.length is the total amount of bytes of the request packet. The 1+2+payload+16 is a check that adds the type field, length field, payload size, padding size. If the declared payload size is larger than the request packet. If it is the declared payload is asking for more information than it needs so the packet is discarded.

**Issues:**

1. Running this with two virtual machines was maddening. The two machines shared the same ip addresses so, when machine A went to look for the server it found itself rather than machine B.

**Recommendations**

1. For task 3.2 it is very important to understand how packets are made and how large their size should be.