**Lab2**

**Lab Environment**

1. Use **SEEDUbuntu12.04** VM for both **victim** and **attacker** machines.
2. Set both VMs to the **same NAT-Network** in VirtualBox.

**Lab Tasks**

1. **Launch the HeartBleed Attack**

On the **victim VM**:

* Open https://www.heartbleedlabelgg.com.
* Log in as admin:
  + Username: admin
  + Password: seedelgg
* Add **Boby** as a friend.
* Send Boby a **private message**

On the **attacker VM**:

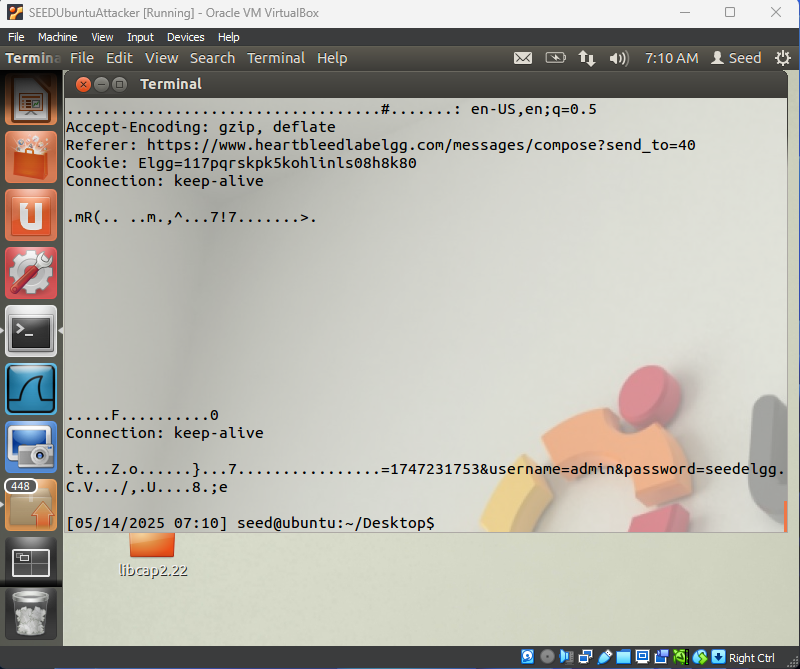
1. Edit /etc/hosts:

<Victim\_VM\_IP> www.heartbleedlabelgg.com

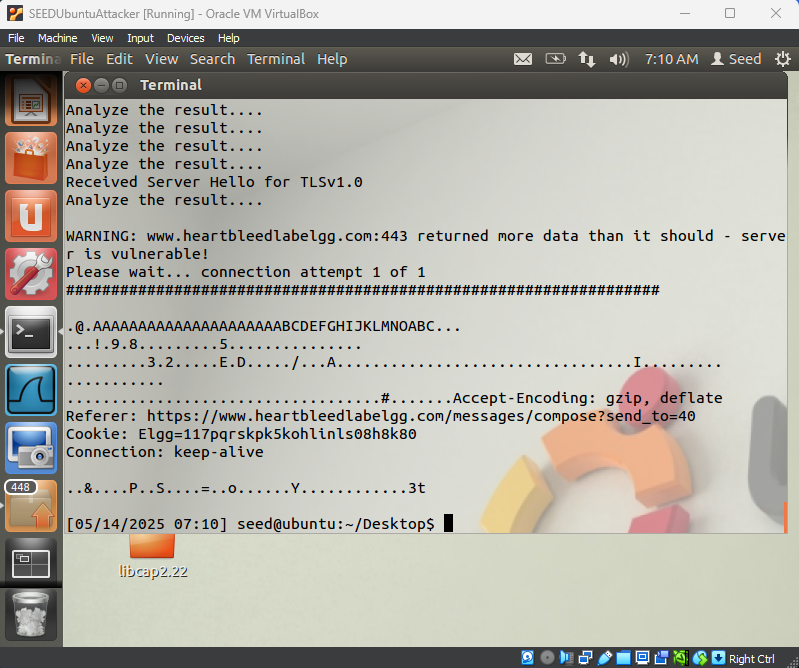
1. Download attack.py from the lab site (or your course portal).
2. Run the attack several times:

./attack.py [www.heartbleedlabelgg.com](http://www.heartbleedlabelgg.com)

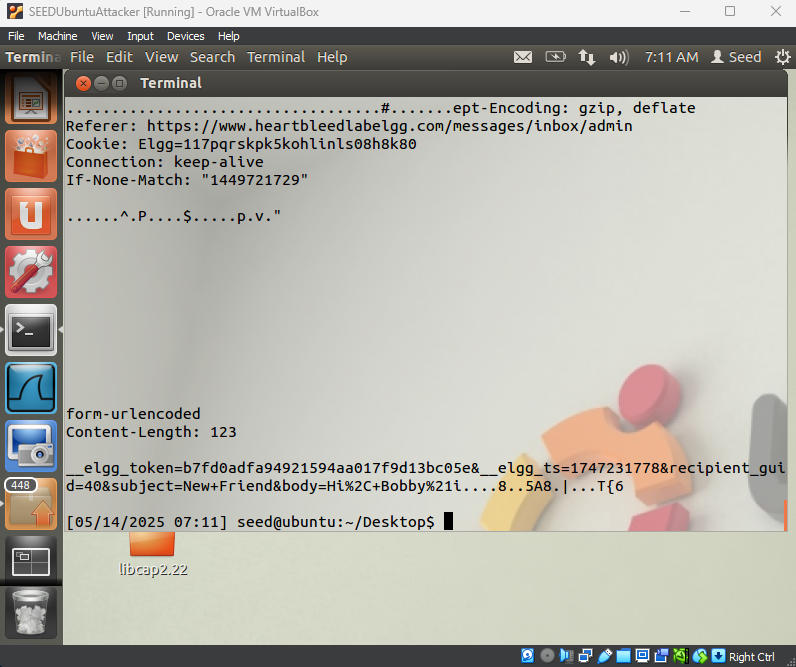
* User name and password.



* User’s activity (what the user has done).



* The exact content of the private message.



1. **Find the Cause of the HeartBleed Vulnerability**
   1. As the length variable decreases, what kind of difference can you observe?

The **amount of leaked data decreases** with smaller lengths, and eventually, no extra data is leaked when the length matches the real payload size.

* 1. 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."

./attack.py www.heartbleedlabelgg.com --length 22

1. Countermeasure and Bug fix
   1. Try yout attack again after you have updated the OpenSSL library. Please describe your observations.

The attack will **fail to return any leaked memory**.

* 1. How to fix the Heartbleed Bug ?

memcpy(bp, pl, payload); // payload might be much larger than actual payload data

This is the **core of the vulnerability**. When the server receives a Heartbeat request, it reads a payload length from the packet and uses it to copy data from memory into the response — **without checking whether the specified length matches the actual size of the payload**. As a result, it may copy not only the payload but **additional unintended memory content**.

It can be fixed by checking if the payload length is smaller or equal to the actual payload length. If not, then reject the request. Basically, the solution is to ignore Heartbeat Request messages that ask for more data than their payload need.

* **Alice is correct**: The root cause is **missing bounds checking**.
* **Bob is also correct**: The server **blindly trusts input from client**, a form of **input validation failure**.
* **Eva is incorrect**: Deleting the length field breaks protocol compliance. Length fields are needed — we must validate them, not remove them.