

MSSV: 19127102

Họ tên: Võ Hoàng Gia Bảo

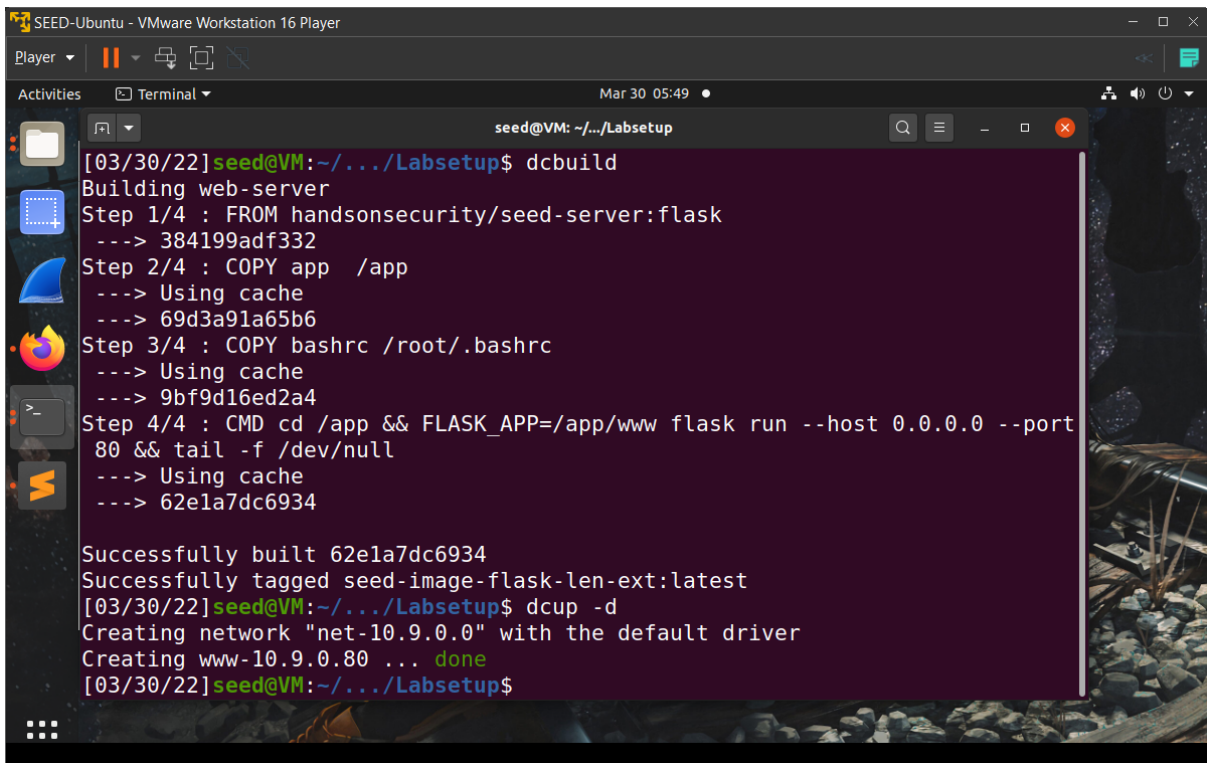
Task 1

Mở terminal và trở vào folder Labsetup

Sử dụng lệnh:

dcbuild

dcup -d



```
SEED-Ubuntu - VMware Workstation 16 Player
Player
Activities Terminal
seed@VM: ~/.../Labsetup
[03/30/22]seed@VM:~/.../Labsetup$ dcbuild
Building web-server
Step 1/4 : FROM handsonsecurity/seed-server:flask
---> 384199adf332
Step 2/4 : COPY app /app
---> Using cache
---> 69d3a91a65b6
Step 3/4 : COPY bashrc /root/.bashrc
---> Using cache
---> 9bf9d16ed2a4
Step 4/4 : CMD cd /app && FLASK_APP=/app/www flask run --host 0.0.0.0 --port
80 && tail -f /dev/null
---> Using cache
---> 62e1a7dc6934

Successfully built 62e1a7dc6934
Successfully tagged seed-image-flask-len-ext:latest
[03/30/22]seed@VM:~/.../Labsetup$ dcup -d
Creating network "net-10.9.0.0" with the default driver
Creating www-10.9.0.80 ... done
[03/30/22]seed@VM:~/.../Labsetup$
```

Mở terminal mới

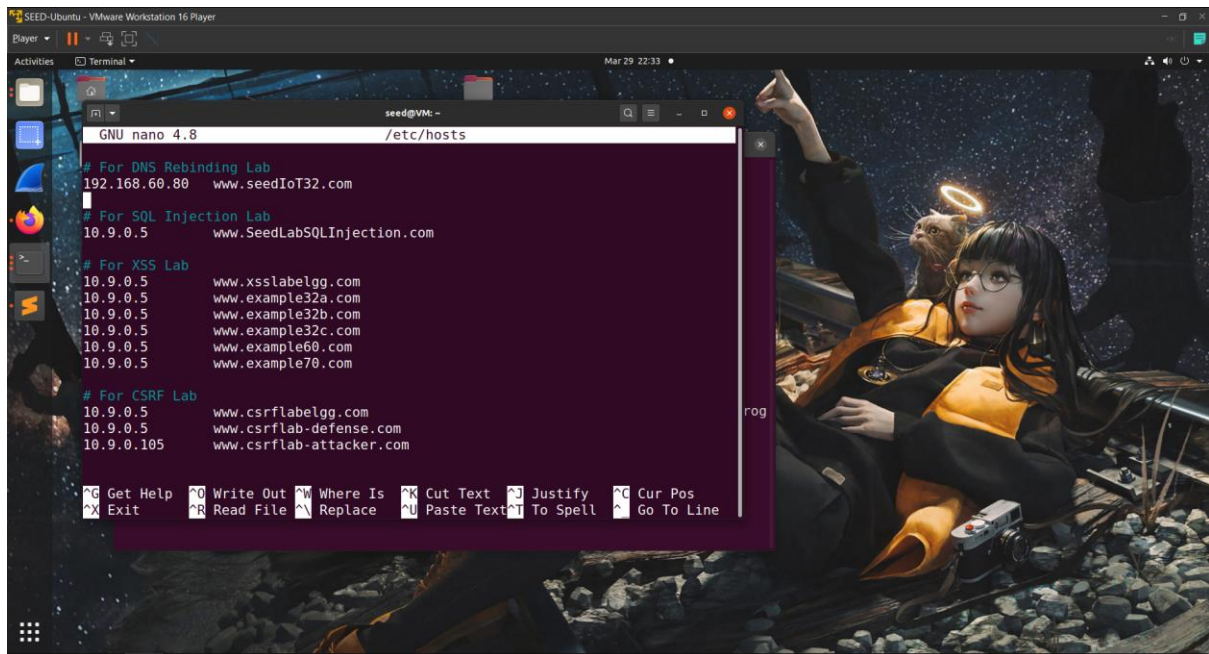
Sử dụng lệnh:

sudo nano /etc/hosts

Thêm vào dòng:

10.9.0.80 www.seedlab-hashlen.com

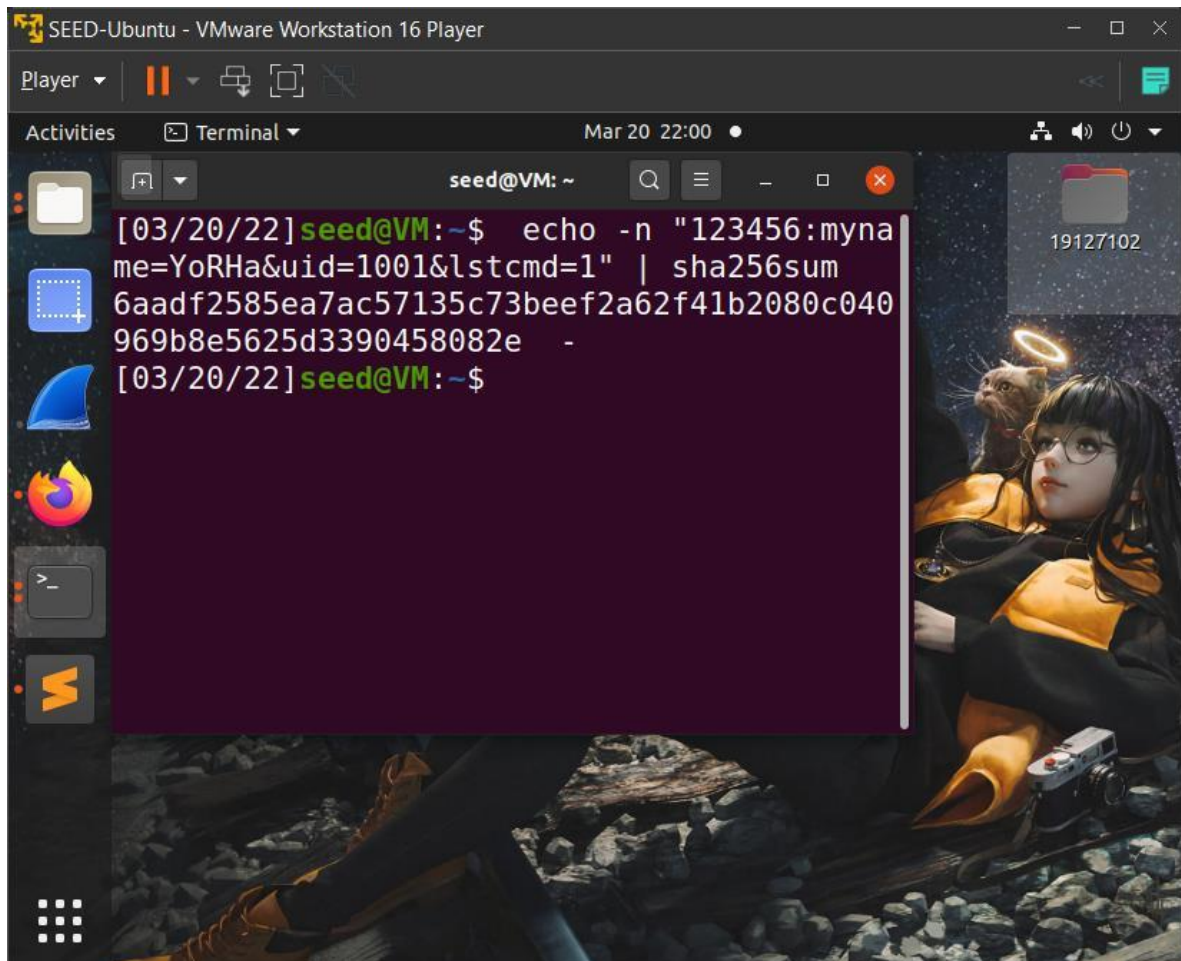
Nhấn Ctrl-X và enter để lưu thay đổi



Mở terminal mới

Sử dụng lệnh

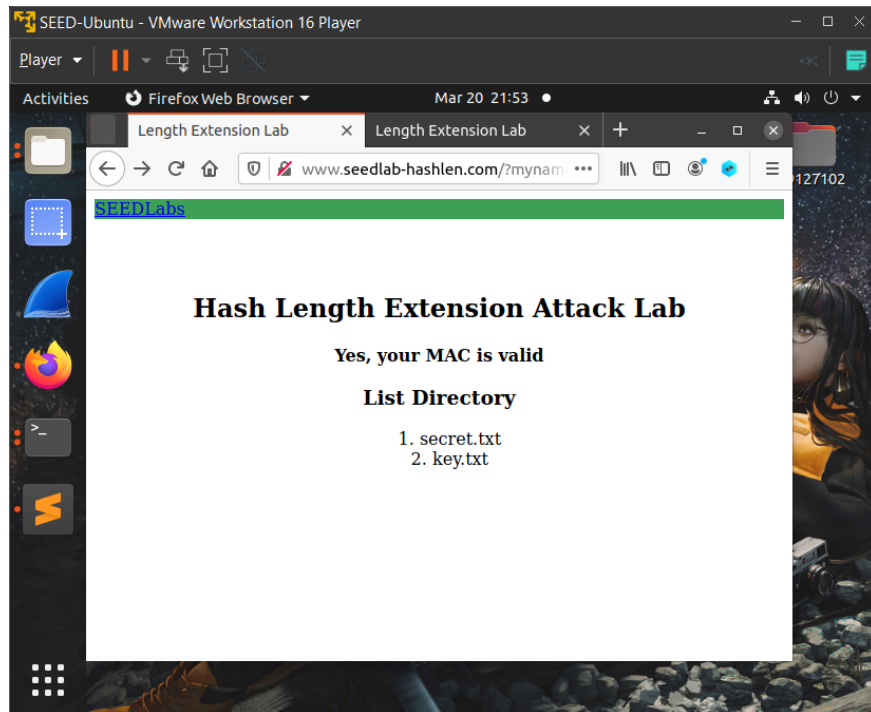
```
echo -n "123456:myname=YorHa&uid=1001&lstcmd=1" | sha256sum
```



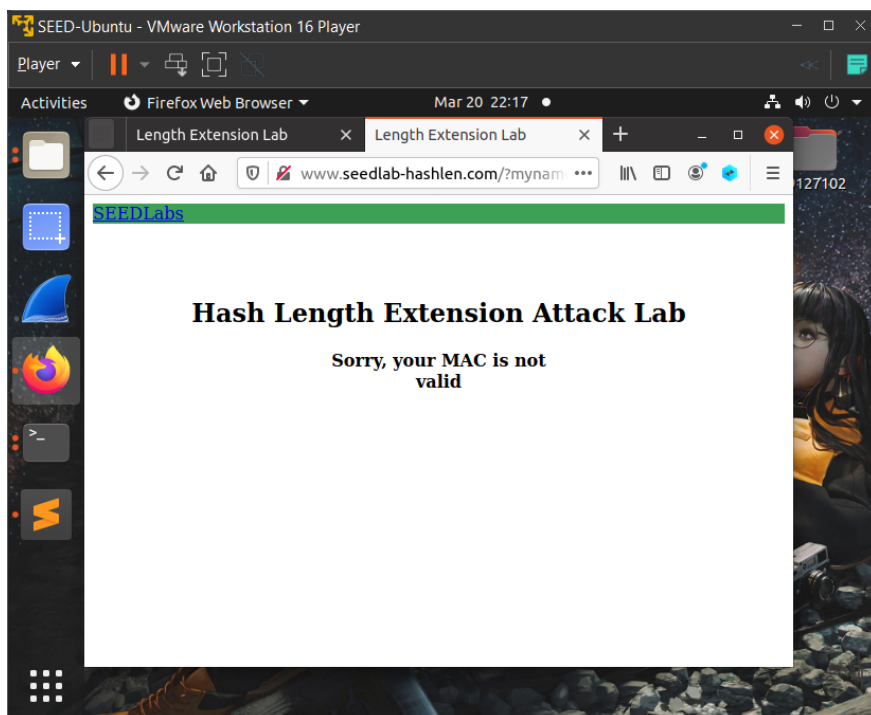
Vào browser nhập đường dẫn:

<http://www.seedlabhashlen.com/?myname=YoRHa&uid=1001&lscmd=1&mac=6aadf2585ea7ac57135c73beef2a62f41b2080c040969b8e5625d3390458082e>

Kết quả:



Nếu thêm “&download=secret.txt”



Task 2

123456:myname=YoRHa&uid=1001&lscmd=1

$64 - 37 = 27$

$37 * 8 = 296 = 128$

"123456:myname=YoRHa&uid=1001&lscmd=1"

"\x80"

"\x00\x00\x00\x00\x00\x00\x00\x00\x00"

"\x00\x00\x00\x00\x00\x00\x00\x00\x00"

"\x00\x00\x00\x00\x01\x28"

\x80\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x01\x28

%80%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%
00%00%00%00%00%00%00%00%01%28

Task 3

Chỉnh sửa source code length_ext.c đã cho theo địa chỉ MAC đã tính ở task 1

```
c.h[0] = htobe32(0x6aadf258);
```

```
c.h[1] = htobe32(0x5ea7ac57);
```

```
c.h[2] = htobe32(0x135c73be);
```

```
c.h[3] = htobe32(0xef2a62f4);
```

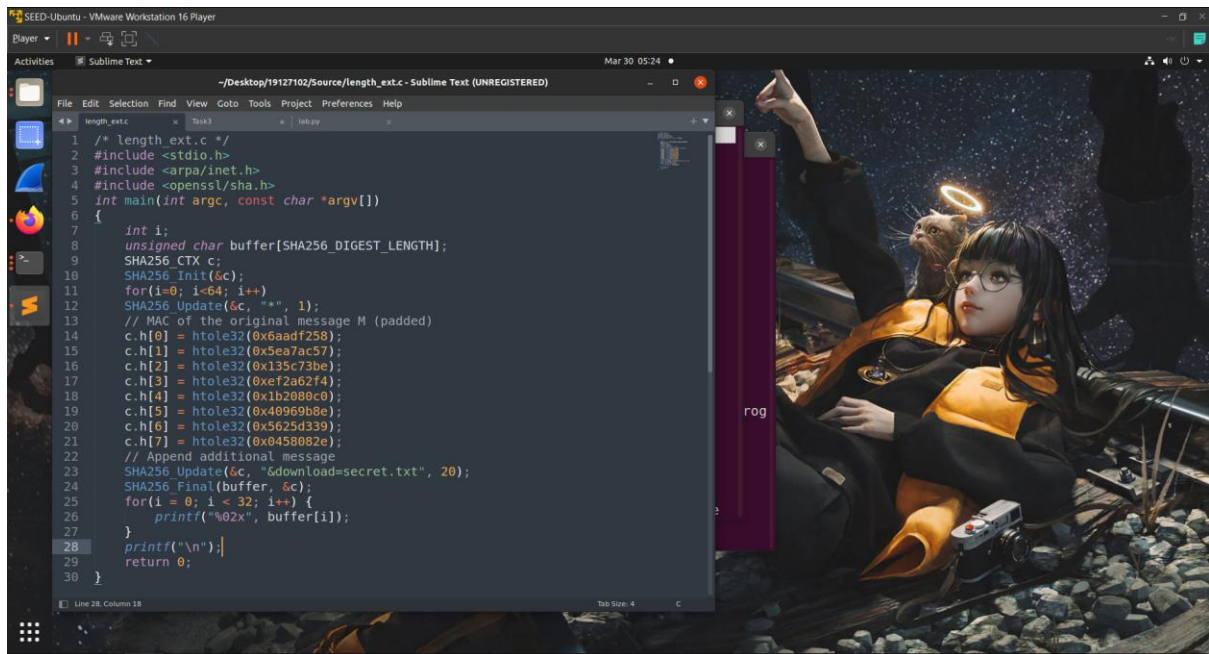
```
c.h[4] = htobe32(0x1b2080c0);
```

```
c.h[5] = htobe32(0x40969b8e);
```

```
c.h[6] = htobe32(0x5625d339);
```

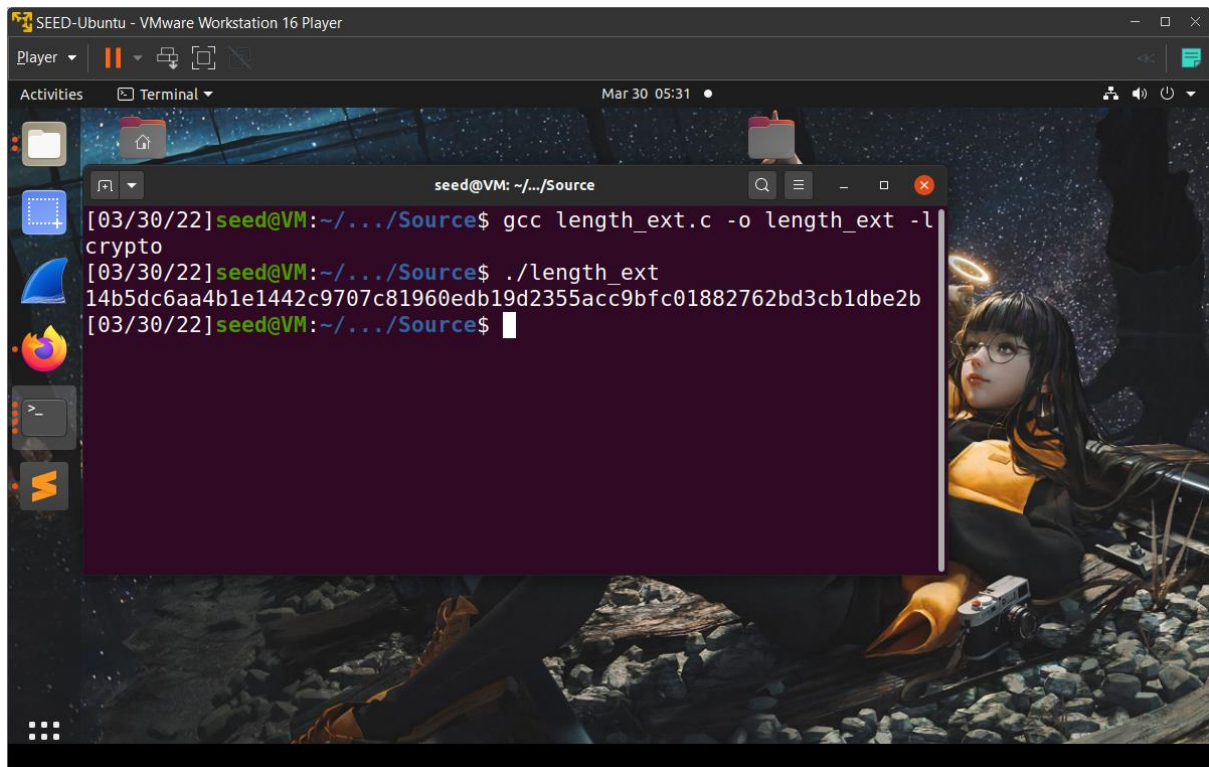
```
c.h[7] = htobe32(0x0458082e);
```

```
SHA256_Update(&c, "&download=secret.txt", 20);
```

The screenshot shows a Sublime Text editor window titled "length_ext.c - Sublime Text (UNREGISTERED)". The code is a C program that calculates a SHA256 MAC for a message. It includes headers for stdio, arpa/inet, and openssl/sha.h. The main function takes an argument, pads it, and calculates the MAC using SHA256_Init, SHA256_Update, and SHA256_Final. The MAC is then printed as a hexadecimal string.

```
1 /* length_ext.c */
2 #include <stdio.h>
3 #include <arpa/inet.h>
4 #include <openssl/sha.h>
5 int main(int argc, const char *argv[])
6 {
7     int i;
8     unsigned char buffer[SHA256_DIGEST_LENGTH];
9     SHA256_CTX c;
10    SHA256_Init(&c);
11    for(i=0; i<64; i++)
12        SHA256_Update(&c, " ", 1);
13    // MAC of the original message M (padded)
14    c.h[0] = htonl32(0x6aadf258);
15    c.h[1] = htonl32(0x5ea7ac57);
16    c.h[2] = htonl32(0x135c73be);
17    c.h[3] = htonl32(0xef2a62f4);
18    c.h[4] = htonl32(0xb2080c0);
19    c.h[5] = htonl32(0x40969b8e);
20    c.h[6] = htonl32(0x5625d339);
21    c.h[7] = htonl32(0x0458082e);
22    // Append additional message
23    SHA256_Update(&c, "download=secret.txt", 20);
24    SHA256_Final(buffer, &c);
25    for(i = 0; i < 32; i++) {
26        printf("%02x", buffer[i]);
27    }
28    printf("\n");
29    return 0;
30 }
```

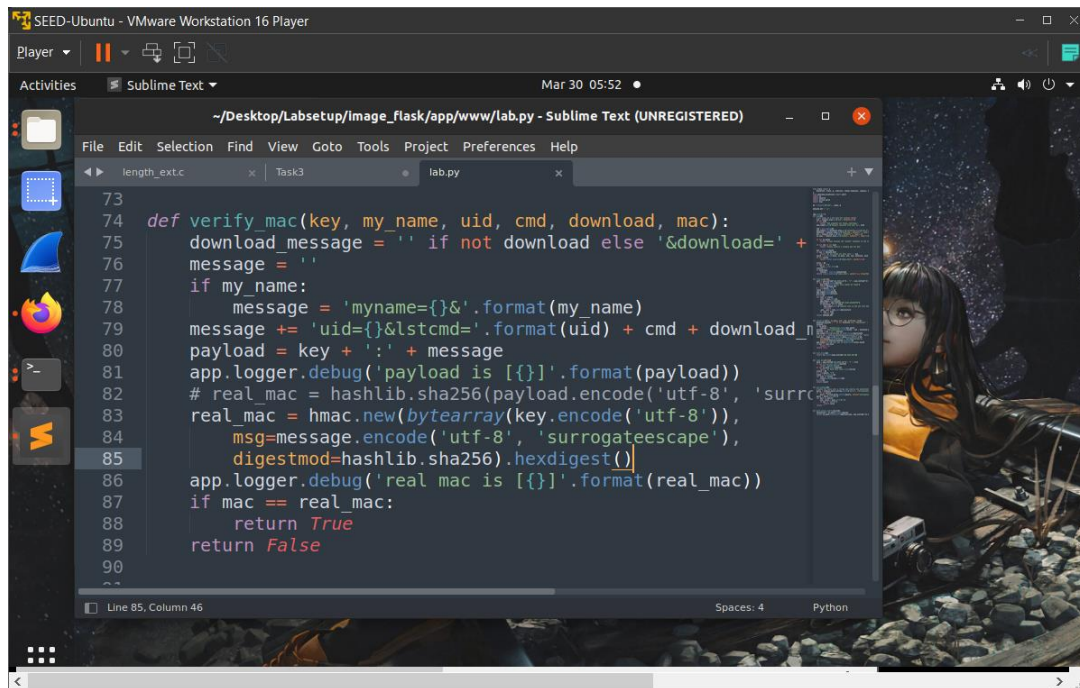


The screenshot shows a terminal window in a VM with the prompt "seed@VM: ~/Source". The user has compiled the program "length_ext.c" into "length_ext" using "gcc" and then executed it. The output is a 64-character hexadecimal string representing the SHA256 MAC.

```
seed@VM: ~/Source
[03/30/22] seed@VM: ~/Source$ gcc length_ext.c -o length_ext -lcrypto
[03/30/22] seed@VM: ~/Source$ ./length_ext
14b5dc6aa4b1e1442c9707c81960edb19d2355acc9bfc01882762bd3cb1dbe2b
[03/30/22] seed@VM: ~/Source$
```

Vào browser nhập đường dẫn:

http://www.seedlab-hashlen.com/?myname=YoRHa&uid=1001&lstcmd=1%80%01%28&download=secret.txt&mac=14b5dc6aa4b1e1442c9707c81960edb19d2355acc9bfc01882762bd3cb1dbe2b



Dùng container, build và khởi động lại.

Mở terminal và trở vào folder Labsetup

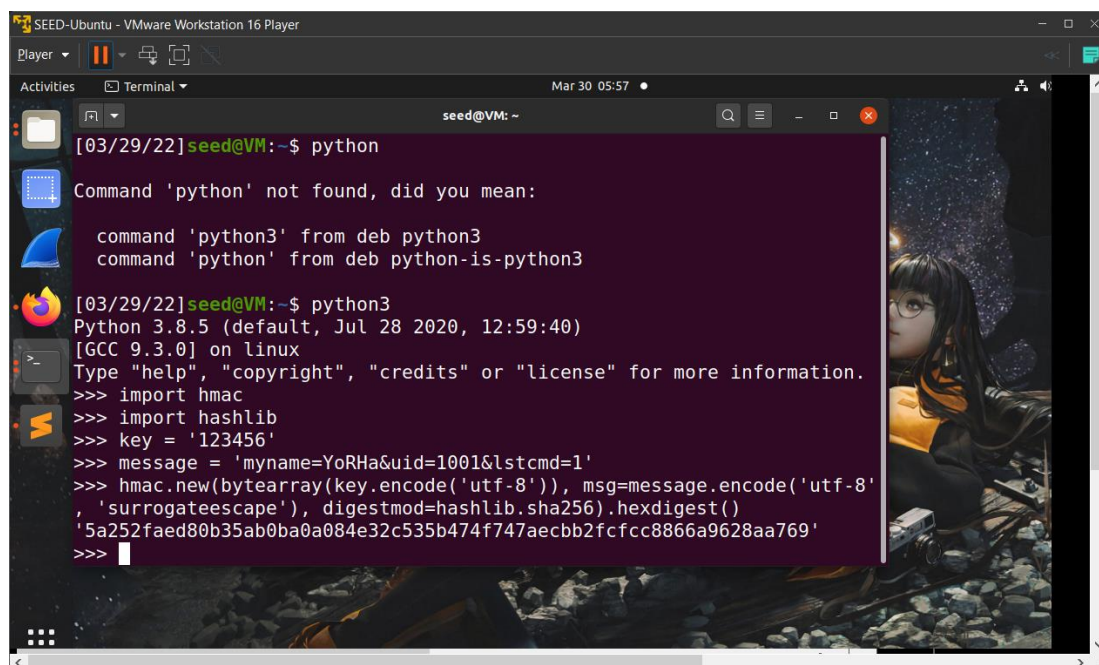
Sử dụng lệnh:

dcdown

dcbuild

dcup -d

Tính HMAC:

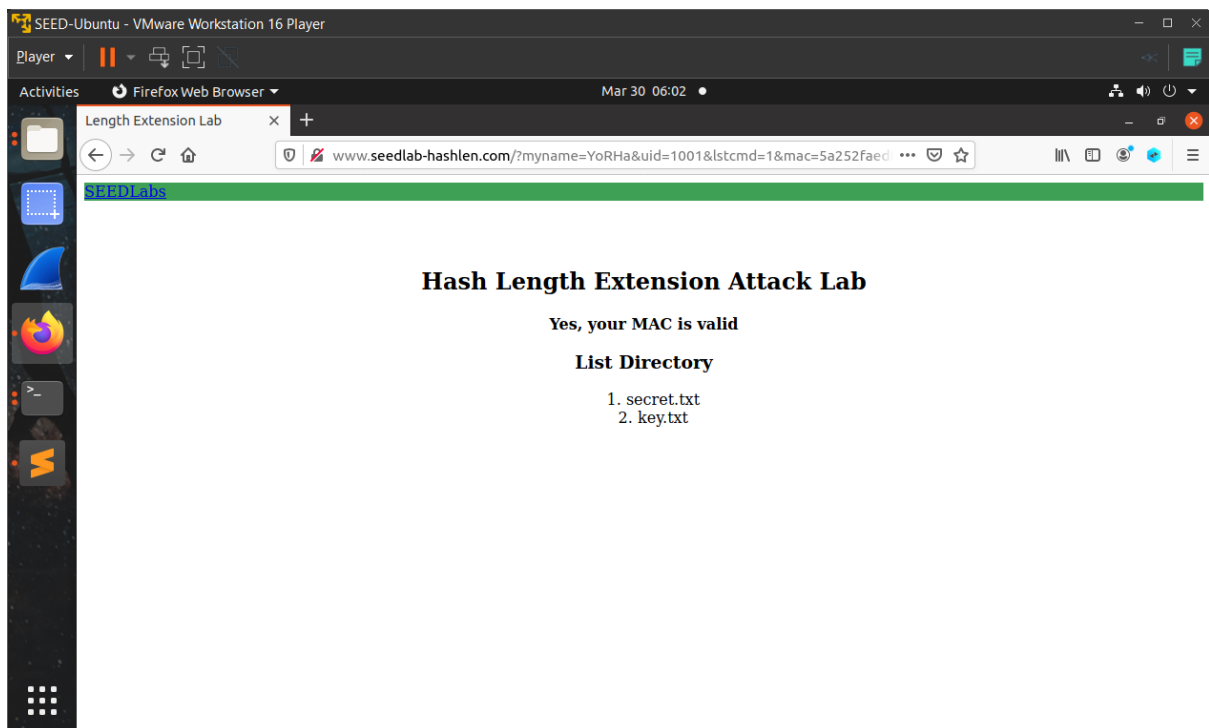


Thực hiện lại task 1 khi đã có HMAC

Vào browser nhập đường dẫn với HMAC:

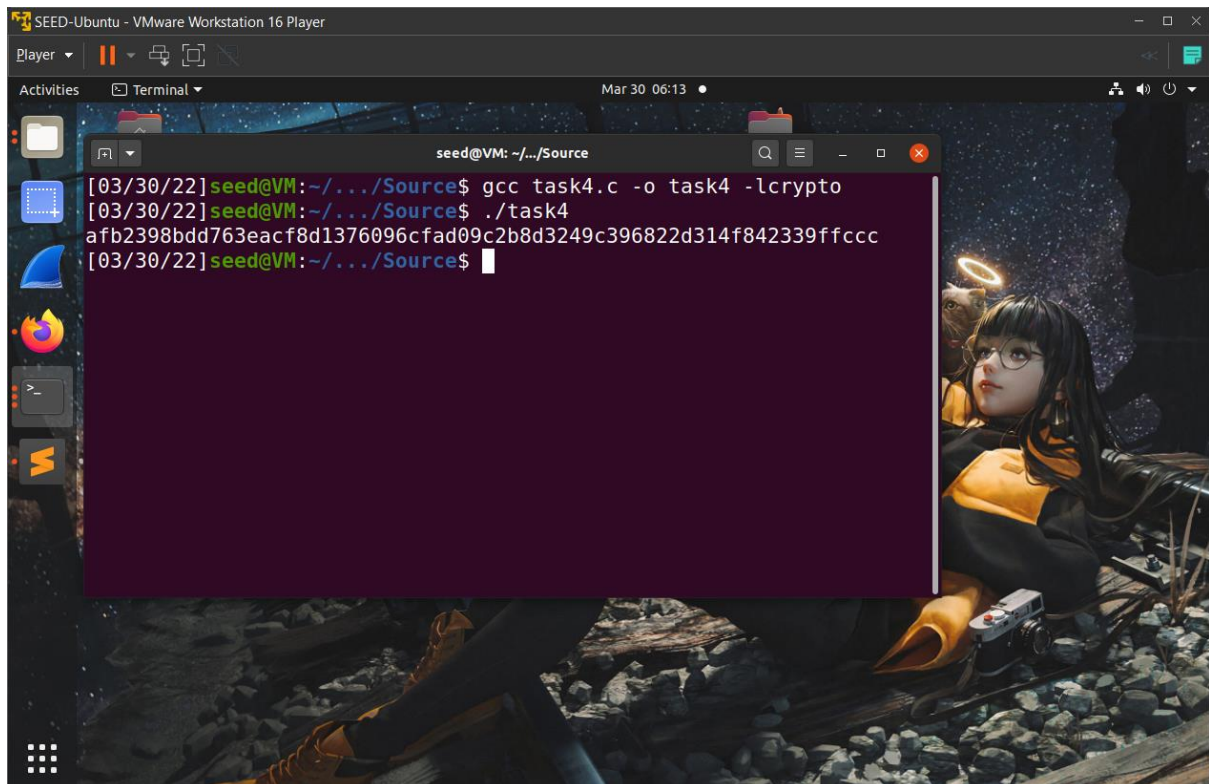
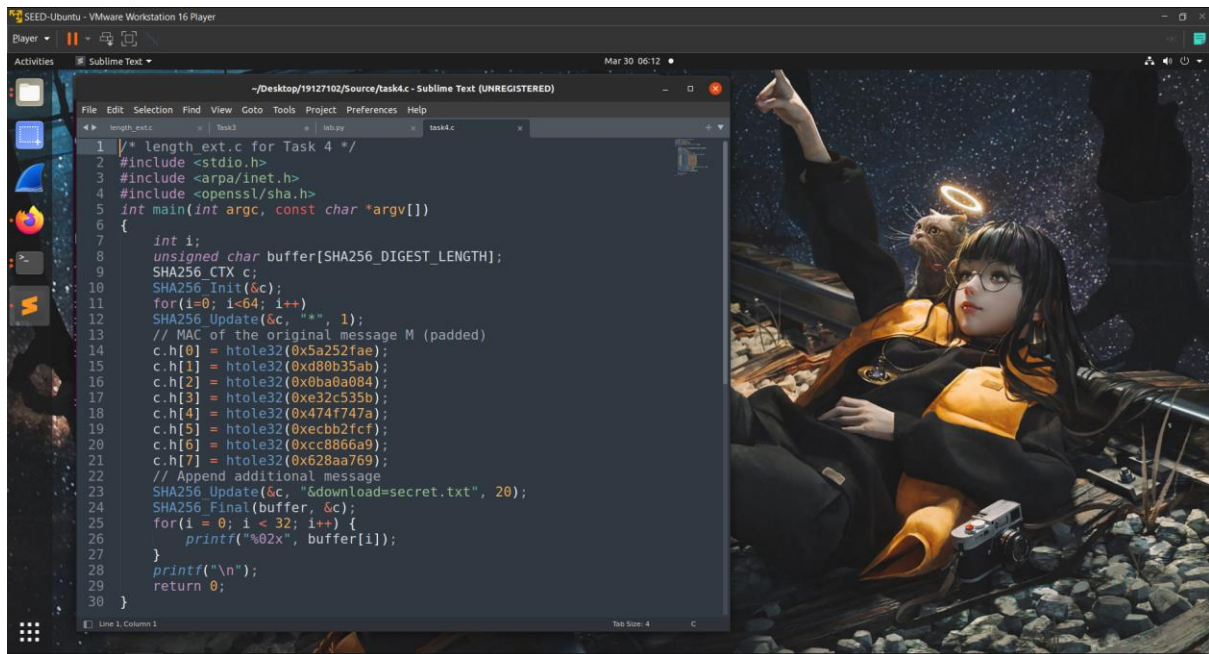
<http://www.seedlab-hashlen.com/?myname=YoRHa&uid=1001&lscmd=1&mac=5a252faed80b35ab0ba0a084e32c535b474f747aecbb2fcfcc8866a9628aa769>

Kết quả:



Thực hiện lại task 2 và 3

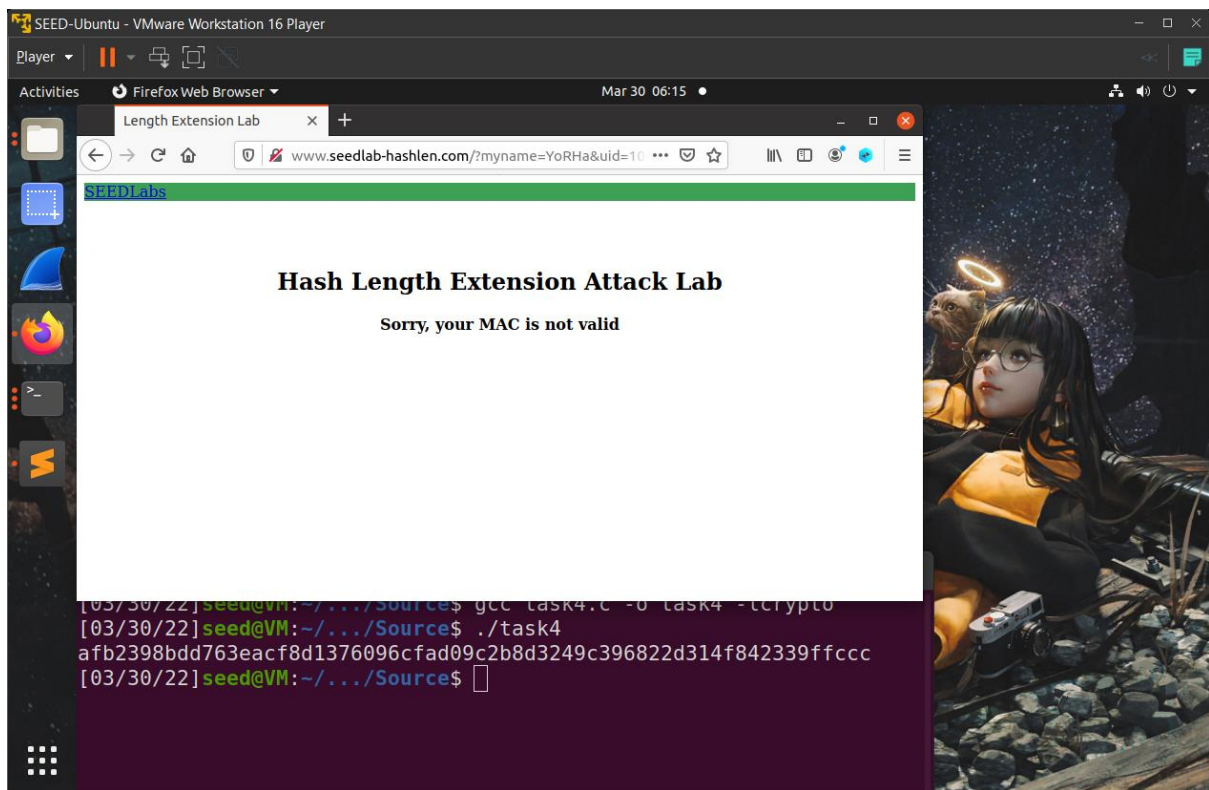
Duplicate file length_ext.c, đặt tên là task4.c và chỉnh sửa code theo HMAC



Vào browser nhập đường dẫn:

<http://www.seedlab-hashlen.com/?myname=YoRHa&uid=1001&lstcmd=1%80%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%01%28&download=secret.txt&mac=afb2398bdd763eacf8d1376096cfad09c2b8d3249c396822d314f842339ffccc>

Kết quả: **Thất bại**



Giải thích:

$$\text{MAC} = \text{hash}(\text{key} + \text{hash}(\text{key} + \text{message}))$$

Do là vì nó được hash thành tin nhắn hai lần nên key không dễ bị extension attack. Do có tính tuần tự nên ta phải có hash trong thì mới tính tiếp được hash ngoài rồi mới có MAC của nguyên cái message, gọi là hash lồng. Ta đã chỉnh cho server thành hàm HMAC thay vì MAC như ban đầu nên hacker không thể trực tiếp tạo cái MAC extension từ cái MAC mà server cho là valid. Cho nên việc tấn công từ bên ngoài hash là vô nghĩa khi mà hash trong vẫn không thay đổi và server sẽ cho nó là invalid.