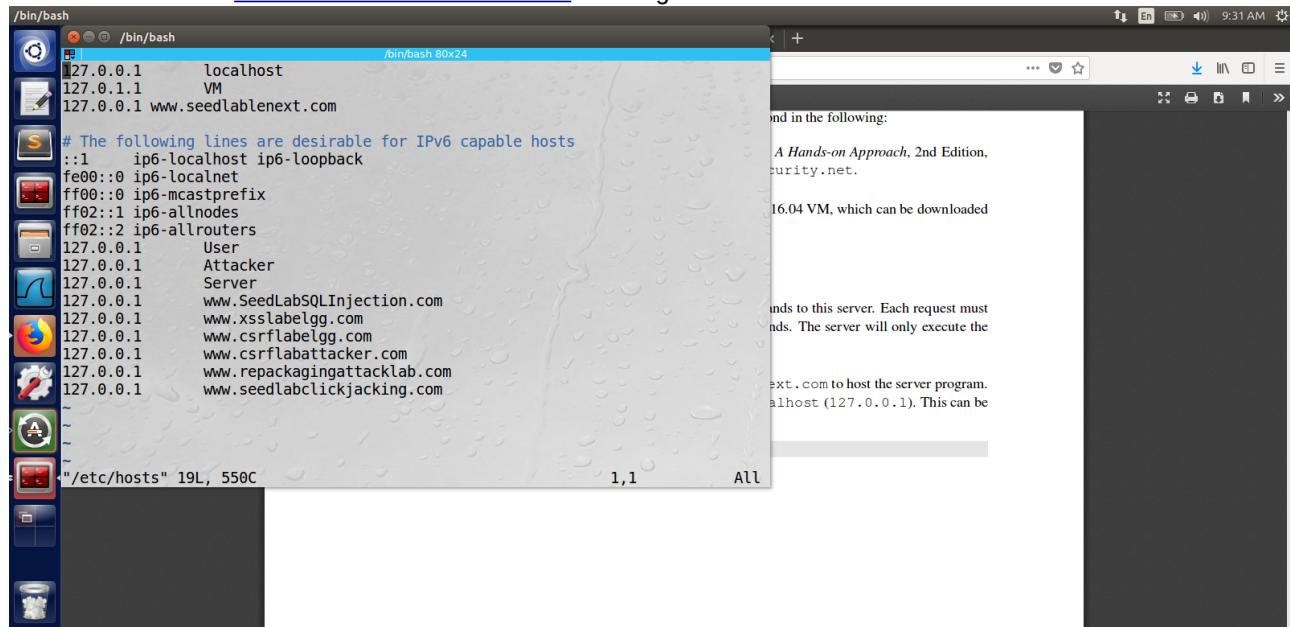


## LAB 5

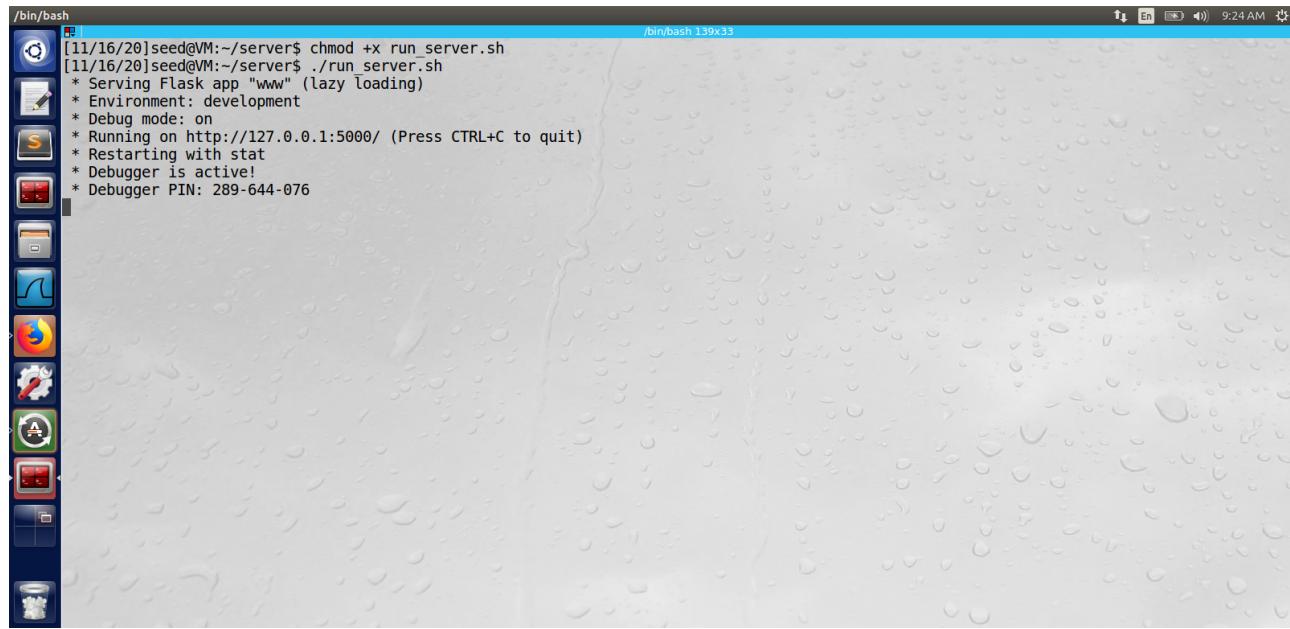
## Hash Length Extension Attack Lab

## Task 1

```
add 127.0.0.1 www.seedlablenext.com change /etc/host file
```



run the server



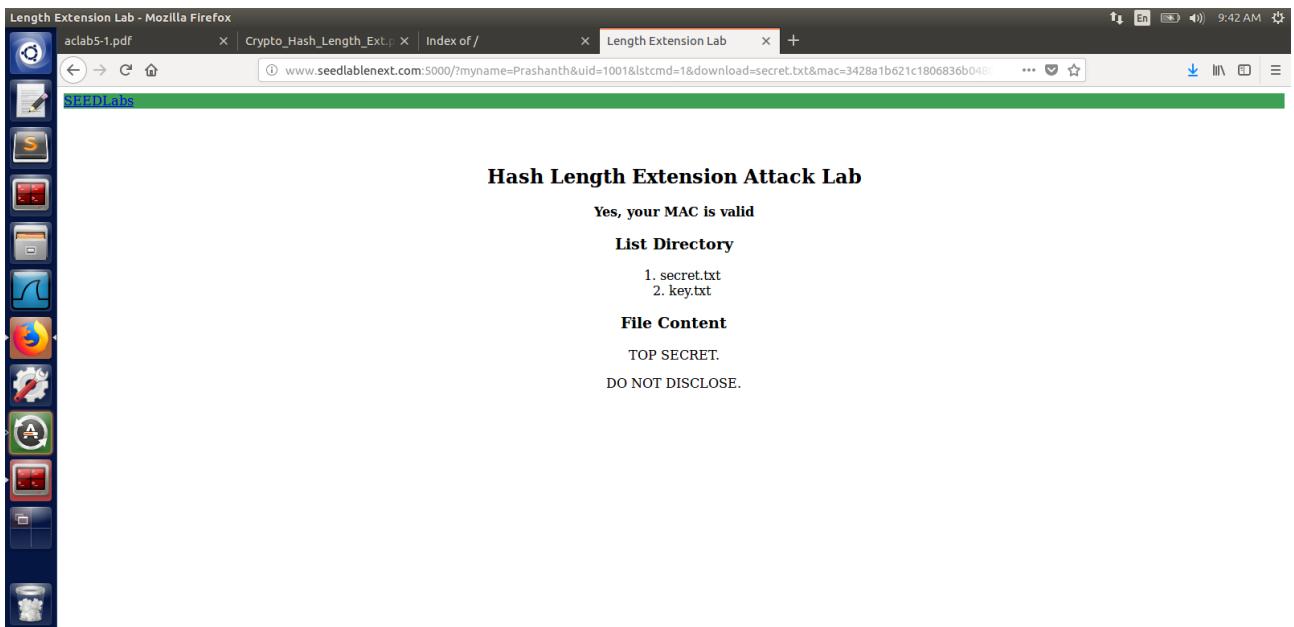
create mac

```
/bin/bash
[11/16/20]seed@VM:~$ echo -n "12345:myname=Prashanth&uid=1001&lstcmd=1" | sha256
sum
5acc614636a134fa6dbece433cfb80f93fd5c7ebd460dbfdb08483bec7ccafb2
[11/16/20]seed@VM:~$
```

web browser request

The screenshot shows a Linux desktop environment with a terminal window and a Firefox browser window. The terminal window at the top left displays a command-line session where a MAC (Message Authentication Code) is generated using the SHA-256 hash function. The command used is `echo -n "12345:myname=Prashanth&uid=1001&lstcmd=1" | sha256sum`, resulting in the output `5acc614636a134fa6dbece433cfb80f93fd5c7ebd460dbfdb08483bec7ccafb2`. The Firefox browser window in the center shows a web page titled "Length Extension Lab" from "SeedLabs". The page content includes a success message ("Yes, your MAC is valid"), a "List Directory" section showing files "secret.txt" and "key.txt", and a URL in the address bar: [www.seedlabnext.com:5000/?myname=Prashanth&uid=1001&lstcmd=1&mac=5acc614636a134fa6dbece433cfb80f93fd5c7ebd460dbfdb08483bec7ccafb2](http://www.seedlabnext.com:5000/?myname=Prashanth&uid=1001&lstcmd=1&mac=5acc614636a134fa6dbece433cfb80f93fd5c7ebd460dbfdb08483bec7ccafb2). A vertical application menu on the left lists various tools and applications.

mac for download=secret.txt



creating padding

```
[11/16/20]seed@VM:~$ python
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> s = "12345:myname=Prashanth&uid=1001&lstcmd=1"
>>> len(s)
40
>>> padding = 64 - 40
>>> padding
24
>>> message_length = 40*8
>>> hex(message_length)
'0x140'
>>> 
```

```
    "\x00\x00\x00"
    "\x00\x00\x00\x00\x00\x00\x00\xB0"
    "Extra message",
    64+13);
SHA256_Final(buffer, &c);

for(i = 0; i < 32; i++) {
    printf("%02x", buffer[i]);
} 
```

we can see we need 0x140 is the message lenght and 24 size of padding is required.

run calculate\_mac

A screenshot of a Linux desktop environment. A terminal window titled '/bin/bash' is open, showing the command-line interface. The terminal window has a title bar with the path '/bin/bash' and the date/time '[11/16/20]'. The main area of the terminal shows the following command and its output:

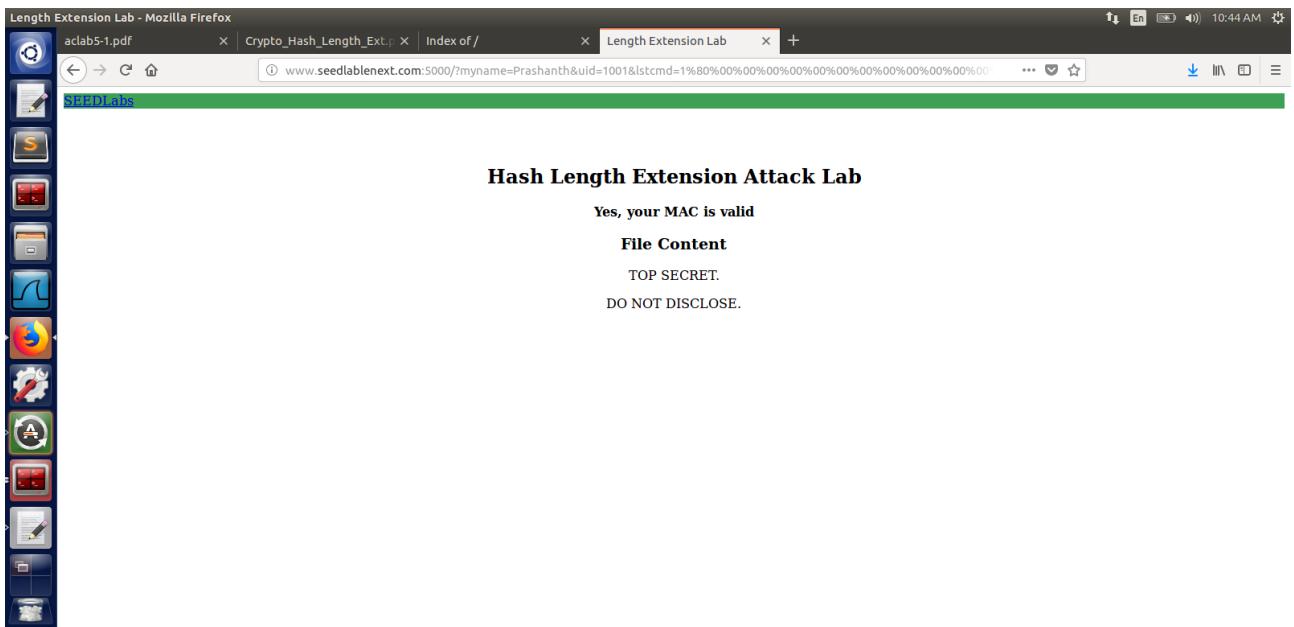
```
[11/16/20]seed@VM:~/Downloads$ gcc calculate_mac.c -o calculate_mac -lcrypto  
[11/16/20]seed@VM:~/Downloads$ ./calculate_mac  
eb71f88b08909fa9fe582c994a6f620b739045287104bf44fad9a2d0e28d6bf3  
[11/16/20]seed@VM:~/Downloads$
```

modify the file to create our mac  
we get

A screenshot of a Linux desktop environment. A terminal window titled '/bin/bash' is open, showing the command-line interface. The terminal window has a title bar with the path '/bin/bash' and the date/time '[11/16/20]'. The main area of the terminal shows the following command and its output:

```
[11/16/20]seed@VM:~/Downloads$ gcc task3.c -o task3 -lcrypto  
[11/16/20]seed@VM:~/Downloads$ ./task3  
f4d6d11a4e4b42c9ee874fc56896a4058b6e3a5d5ffa4cdf2c705ef88620d68  
[11/16/20]seed@VM:~/Downloads$
```

in browser



#### Task4: The lenght extenstion Attack

check sha256sum for "This is a test message"



complie and execute the lenght\_ext.c file

A screenshot of a Linux desktop environment. The desktop has a light blue background with a subtle water droplet texture. A docked application bar on the left contains icons for various applications like a terminal, file manager, and web browser. A terminal window is open at the top, showing a command-line session where a C program named 'length\_ext.c' is compiled and run. The output shows a long string of characters followed by a hexdigest. The system tray in the top right corner shows icons for battery, signal strength, and the date and time (10:49 AM).

modify the lenght\_ext.c file to create our extention file

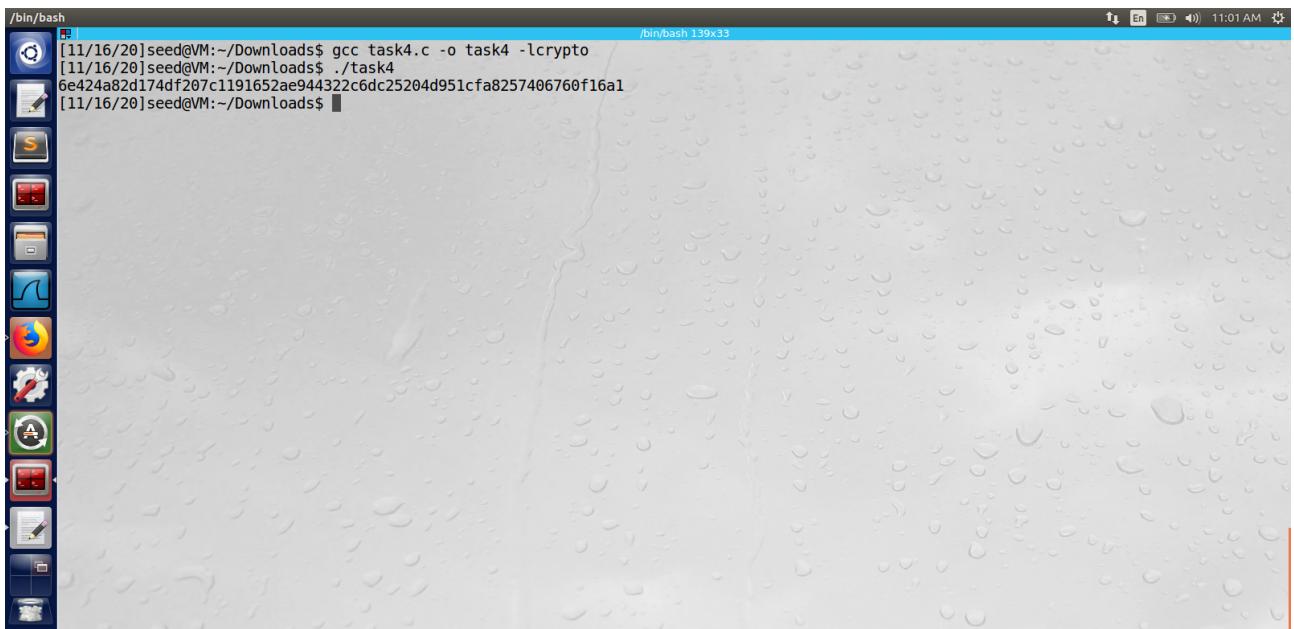
The screenshot shows a Gedit window with two tabs: "length\_ext.c" and "calculate\_mac.c". The "length\_ext.c" tab contains a C program that calculates the SHA-256 MAC of a message. It includes headers for stdio.h, rpa/inet.h, and openssl/sha.h. The main function initializes a SHA256\_CTX, performs 64 iterations of SHA256\_Update with an asterisk as the message, and then appends an extra message. The "calculate\_mac.c" tab is currently active and displays the same code. The status bar at the bottom right shows "Ln 23, Col 5".

```
/*length_ext.c*/
#include <stdio.h>
#include <rpa/inet.h>
#include <openssl/sha.h>
int main(int argc, const char *argv[])
{
    int i;
    unsigned char buffer[SHA256_DIGEST_LENGTH];
    SHA256_CTX c;
    SHA256_Init(&c);
    for (i = 0; i < 64; i++)
        SHA256_Update(&c, "*", 1);

    // MAC of the original message M(padded)
    //5acc614636a134fa0dbce433cfb80f93fd5c7ebd460dbfdb08483bec7ccafb2
    c.h[0] = htonl32(0x5acc6140);
    c.h[1] = htonl32(0x36a134fa);
    c.h[2] = htonl32(0x6dbece43);
    c.h[3] = htonl32(0x3cfb80f9);
    c.h[4] = htonl32(0x3fd5c7eb);
    c.h[5] = htonl32(0xd460dbfd);
    c.h[6] = htonl32(0xb0b8483be);
    c.h[7] = htonl32(0xc7ccafb2);

    // Append additional message
    SHA256_Update(&c, "Extra message", 13);
    SHA256_Final(buffer, &c);
    for (i = 0; i < 32; i++)
    {
        printf("%02X", buffer[i]);
    }
    printf("\n");
    return 0;
}
```

complie and exceute



using this mac in browser

The screenshot shows a Firefox browser window with the following tabs:

- Length Extension Lab - Mozilla Firefox
- aclab5-1.pdf
- Crypto\_Hash\_Length\_Ext.p
- Index of /
- Length Extension Lab
- + (New Tab)

The main content area displays the results of a hash length extension attack:

### Hash Length Extension Attack Lab

Yes, your MAC is valid

**File Content**

```
1001:123456
1002:983abe
1004:98zjxc
1005:xcuijk
```

## Task5 Attack Mitigation using HMAC

changing sha256 to hmac ... in source file

lab.py (~/server/www) - gedit

length\_ext.c

lab.py

```
calculate_mac.c
length_ext.c
lab.py
```

Save

```
        continue
    _uid, _key = line.split(delimiter)
    if _uid == uid:
        return _key
    return INVALID_KEY

def verify_mac(key, my_name, uid, cmd, download, mac):
    download_message = '' if not download else '&download=' + download
    message = ''
    if my_name:
        message = 'myname={}'.format(my_name)
    message += 'uid={}&lstcmd={}'.format(uid) + cmd + download_message
    payload = key + ';' + message
    app.logger.debug('payload is {}'.format(payload))
    real_mac = hmac.new(bytarray(key.encode('utf-8')), msg=message.encode('utf-8'),
                        digestmod=hashlib.sha256).hexdigest()
    app.logger.debug('real mac is {}'.format(real_mac))
    if mac == real_mac:
        return True
    return False

def list_files():
    return os.listdir(app.config['LAB_HOME_DIR'])

def read_file(file):
    path = app.config['LAB_HOME_DIR'] + '/' + file
    if not path_access_control(path):
        return "Access Denied"
    if not os.path.exists(path):
        return "No Such File {}".format(file)
    result = []
    f = open(path, 'r')
    lines = f.readlines()
    for line in lines:
        result.append(line.strip())
    f.close()
Saving file '/home/seed/server/www/lab.py'...
```

create a mac using hmac

/bin/bash

/bin/bash 139x33

```
[11/16/20]seed@VM:~$ python
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import hmac
>>> import hashlib
>>> key = "12345"
>>> message= "lstcmd=1"
>>> hmac.new(bytarray(key.encode('utf-8')), msg=message.encode('utf-8',
File "<stdin>", line 1
    hmac.new(bytarray(key.encode('utf-8')), msg=message.encode('utf-8',
SyntaxError: invalid syntax
>>> 'surrogateescape'), digestmod=hashlib.sha256).hexdigest()
KeyboardInterrupt
>>> hmac.new(bytarray(key.encode("utf-8")), msg=message.encode("utf-8",
.. "surrogateescape"), digestmod=hashlib.sha256).hexdigest()
'd2d9f36a1edf5eeb557f47cd76f25abad4db0b238b5ece8de470842f4104207b'
>>>
```

using the mac in browser

