

## DOCUMENTATION – mailer.c

This file is the documentation to the C program “mailer.c” in [yuvrajmalhi/mailer \(github.com\)](https://github.com/yuvrajmalhi/mailer)

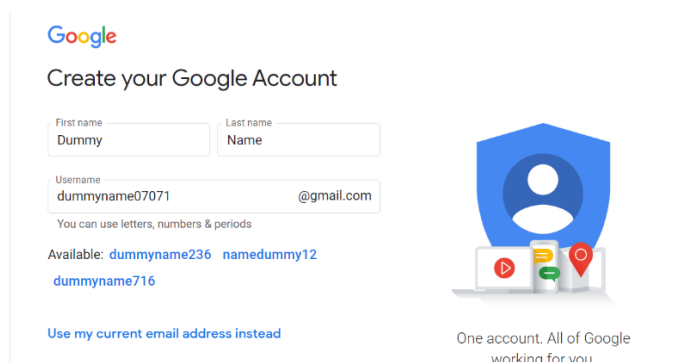
The program sends mails from any linux distro having a curl library, thus making it very general purpose and reliable. The user interface has been made very functional yet simple. Thus making this program a much faster way to send mails in one go rather than opening a browser, logging in, composing, typing etc.

### ONE TIME SETUP

Before any demonstration or functioning of code can be shown, it is important to change some settings in the sender's email account to allow 3<sup>rd</sup> party applications (like Linux-curl API) to manage emails. This can cause some security issues, but since my code is short and open sourced, one can easily verify if some harmful commands are being run. It is not the case.

Now, follow the following steps:-

Step 1. I created a new Gmail account. Skip this if an old account is to be used.



My ID -> [dummyname07071@gmail.com](mailto:dummyname07071@gmail.com)

My Password -> 07Night!

Step 2. On your Linux system (I use Kali), download “mailer.c” from [yuvrajmalhi/mailer \(github.com\)](https://github.com/yuvrajmalhi/mailer)

```
root@kali:~/Desktop/Yuvraj_Malhi# ls
mailer.c
```

Step 3. Compile the program “mailer.c” using the command:

```
gcc mailer.c -lcurl
```

If any error is seen then cURL library is not installed. Otherwise, on successful compilation, a new file “a.out” will be created.

```
root@kali:~/Desktop/Yuvraj_Malhi# gcc mailer.c -lcurl
root@kali:~/Desktop/Yuvraj_Malhi# ls
a.out mailer.c
```

Step 4. Run the command:

```
./a.out
```

Then one by one fill in all the details as asked.

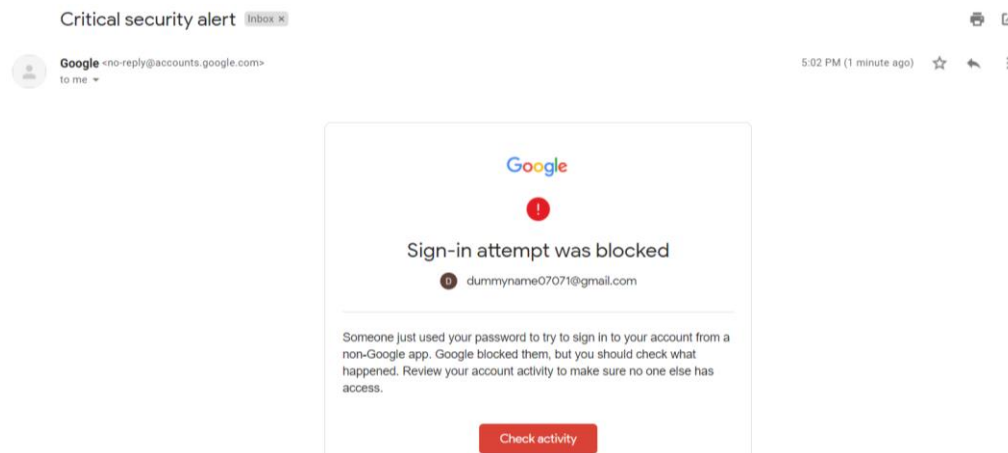
```
root@kali:~/Desktop/Yuvraj_Malhi# ./a.out
Enter the following details:
Your mail (xyz@abc.com) - dummyname07071@gmail.com
Your mail type (gmail/yahoo/office) - gmail
Your password (Mr. XYZ) - 07Night!
Your Name (Mr. XYZ) - UV
Receiver's mail (pqr@uvw.com) - 
CC mail (efg@mno.com) - 
CC name (Mr. EFG) - 
Subject - Try
Body - 
This is my email from Kali.
```

Press Enter.

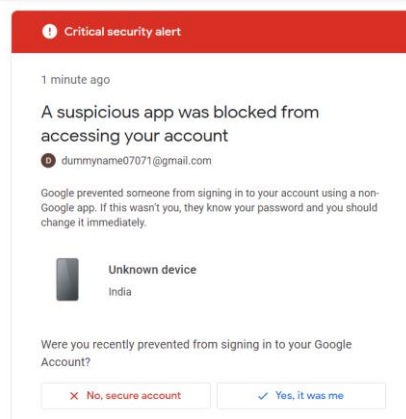
```
CC mail (efg@mno.com) - 
CC name (Mr. EFG) - 
Subject - Try
Body - 
This is my email from Kali.
curl_easy_perform() failed: Login denied
```

An error will be show. This is because gmail has not yet authorized linux-curl API to allow sending emails from this particular account. To fix this:

Step 5. Login to the account, and notice that such an email must have been received.

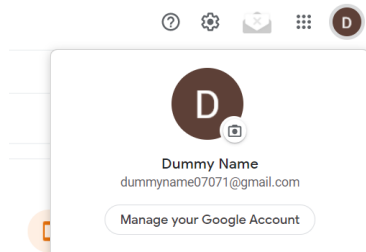


Click on “check activity” and then click “Yes, it was me”

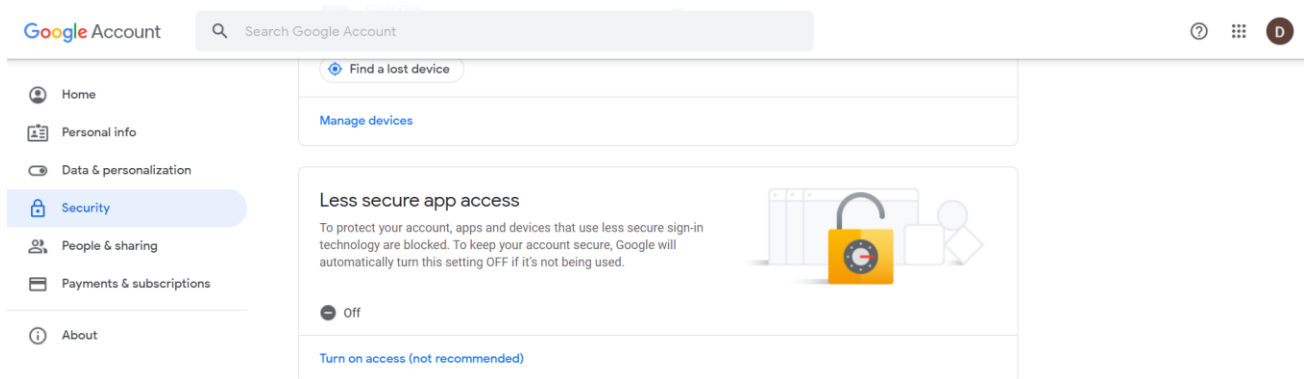


Step 6. (This may not be required for outlook/ office365 accounts).

In the gmail account, click on your profile image > “Manage your Google account”

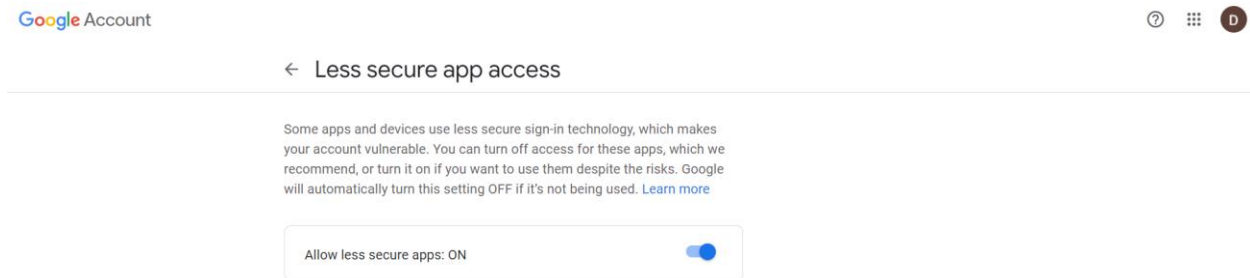


Scroll over to the security section.



Now, change the “Off” to “On” in the “Less secure app access” tab.

As I said, my code is open source, so no worries on that end.



We are all set.

Now, maximum of 300 mails (or 100 depending on whether the account is gmail, yahoo, office) per day can be sent without any hassle. Here’s an example:

To run the code, run

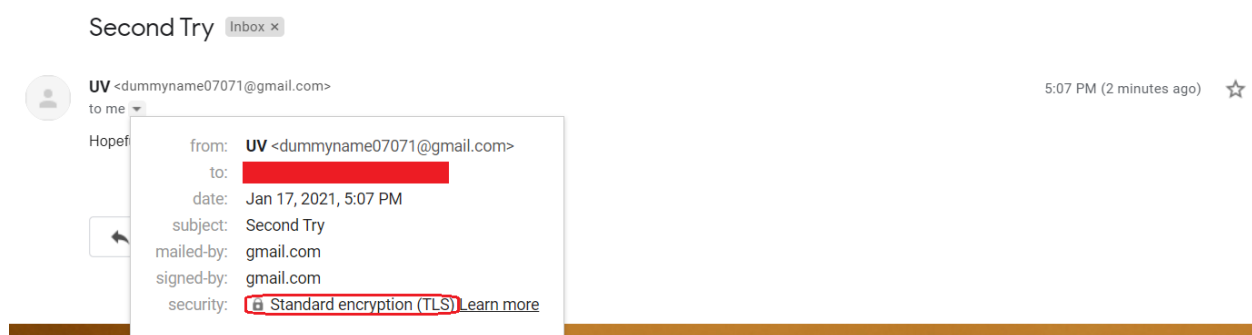
```
./a.out
```

On the same location in the linux system as in Step 4

```
root@kali:~/Desktop/Yuvraj_Malhi# ./a.out
Enter the following details:
Your mail      (xyz@abc.com)      - dummyname07071@gmail.com
Your mail type (gmail/yahoo/office) - gmail
Your password  (07Night!)         - 07Night!
Your Name      (Mr. XYZ)          - UV
Receiver's mail (pqr@uvw.com)      - yuvrajmalhi30@gmail.com
CC mail        (efg@mno.com)       -
CC name        (Mr. EFG)          -
Subject        - Second Try
Body -
Hopefully this will work
```

This time the code terminates perfectly, without any error.

Check in receiver's email, and the mail will have been perfectly received.



Also note that the email is TLS encrypted, so no worries on privacy end either!

## WORKING

In the code "mailer.c", the main function is ridiculously simple and easy to use"; just scan all the data (Name, email, passwords etc) and pass them all to a mailer() function which does the main job of sending the email.

```
int main()
{
    char your_mail[50],mail_type[20],password [30] ,your_name[30];
    char to[50],cc[50],cc_name[30] ,subject[20], body[100];

    printf("Enter the following details:\n");
    printf("Your mail      (xyz@abc.com)      - "); scanf("%[^\n]s",your_mail); getchar();
    printf("Your mail type (gmail/yahoo/office) - "); scanf("%[^\n]s",mail_type); getchar();
    printf("Your password  (07Night!)         - "); scanf("%[^\n]s", password); getchar();
    printf("Your Name      (Mr. XYZ)          - "); scanf("%[^\n]s",your_name); getchar();
    printf("Receiver's mail (pqr@uvw.com)      - "); scanf("%[^\n]s", to); getchar();
    printf("CC mail        (efg@mno.com)       - "); scanf("%[^\n]s", cc); getchar();
    printf("CC name        (Mr. EFG)          - "); scanf("%[^\n]s", cc_name); getchar();
    printf("Subject        - "); scanf("%[^\n]s", subject); getchar();
    printf("Body - \n"); scanf("%[^\n]s", body); getchar();

    mailer(your_mail,mail_type,password,your_name,to,cc,cc_name,subject,body);
    return 1;
}
```

This has been done to make the code extremely modular so it can be incorporated anywhere else.

The function mailer() has the prototype:

```
void mailer(char* YOUR_MAIL, char* MAIL_TYPE, char* PASSWORD, char* YOUR_NAME, char *TO, char* CC, char* CC_NAME, char* SUBJECT, char* BODY)
```

YOUR\_MAIL -> Sender's email ID

MAIL\_TYPE -> gmail / yahoo / office / outlook

As of now only 4 major emails can be used, but to extend it to other emails, in Line 87 -

```
char MAIL_TYPES[4][30]={ "gmail", "outlook", "office", "yahoo" };  
char SERVERS[4][30] = {"smtp.gmail.com", "smtp-mail.outlook.com", "smtp://smtp.office365.com:587", "smtp://smtp.mail.yahoo.com:587"};
```

Create a shortform for that type of email (like AOL) in MAIL\_TYPES[] array and

add the corresponding SMTP server address (like smtp.aol....) in SERVERS[] array

Be sure to increase the loop iteration limit in line 91.

**IMPORTANT NOTE:** Some other mails like yahoo may not work because of their updated privacy policy to not allow 3<sup>rd</sup> party application to send emails.

PASSWORD -> Sender's email password. Incorrect password would give "login denied error"

YOUR NAME -> Doesn't need to be correct, (but it does show up in the receiver's inbox)

TO -> Receiver's email. Incorrect email here would not give an error, but email won't be sent.

CC -> CC's email ID. If left black or if only single digit is passed (like "\*" or "1") then no CC is sent.

CC\_NAME -> Again, doesn't need to be correct, but shows the name in inbox.

SUBJECT -> Subject of email.

BODY -> Body of email. (Currently, it can be only text, file upload feature has not been added).

**Note** - In case, while sending the email, an error is being received, and if SMTP protocol is known, then uncomment Line 147 to get detail of where error is being produced. Very helpful.

**Note** - Curl SMTP stores the state of email sending procedure in the variable, this is the variable "res" in my program. It may be used by the programmer as per his/her wish. It can be returned from function too if the functionality of the program is to be extended.