



PSG College of Technology

Ethical Hacking Package - **AMONG US BOX**



Aparnaa T (21PC04)

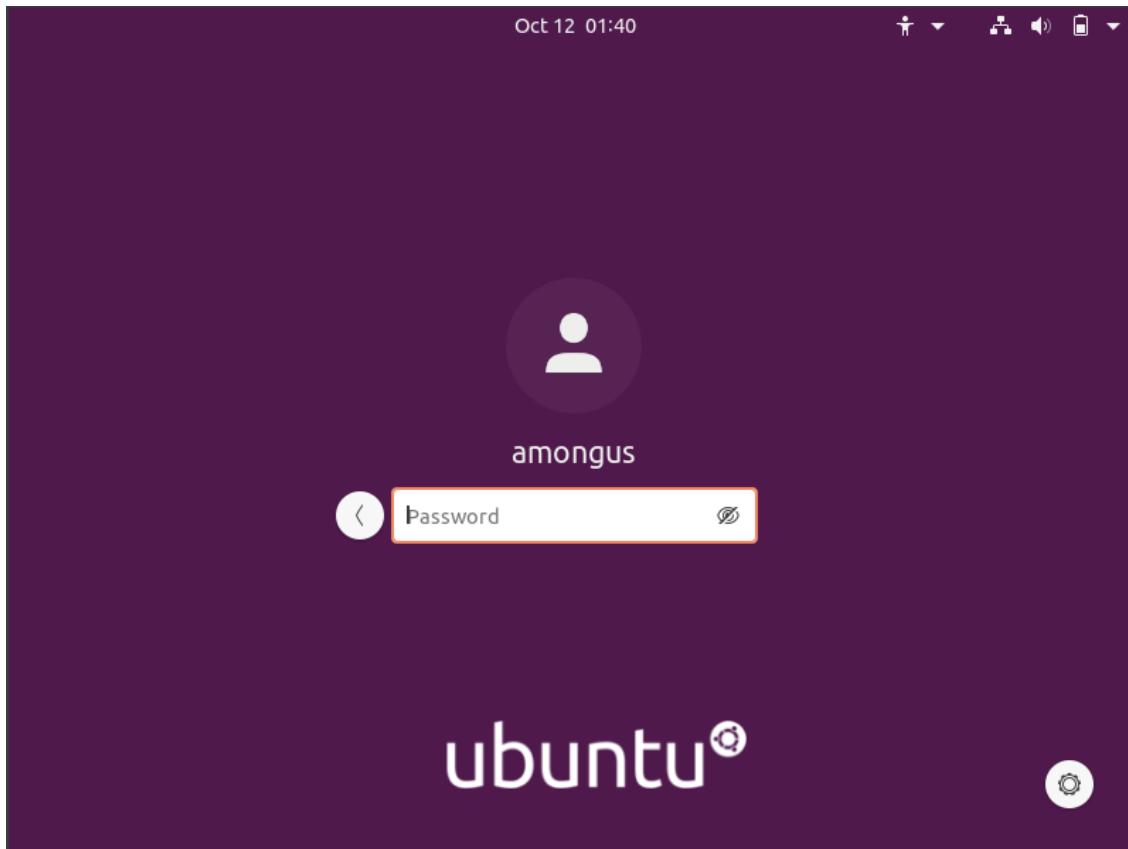
Samyukthaa B (21PC28)

Vishnupriya R(21PC39)

Software used :

OS - Ubuntu

Version - 20.04



When we open the virtual machine, an account named amongus already exists, but we are unaware of the password.

Enumeration :

We create a new NAT network and attach a kali machine and the ubuntu machine to the newly created NAT network, in order to gather information related to the ubuntu machine without logging in.

Run the command **sudo arp-scan –localnet**

```
(kali㉿kali)-[~]
$ sudo arp-scan --localnet
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 08:00:27:cb:7e:f5, IPv4: 10.0.2.15
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
10.0.2.1      52:54:00:12:35:00      (Unknown: locally administered)
10.0.2.2      52:54:00:12:35:00      (Unknown: locally administered)
10.0.2.3      08:00:27:cf:11:64      (Unknown)
10.0.2.4      08:00:27:42:5f:38      (Unknown)

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.966 seconds (130.21 hosts/sec). 4 responded
```

We get 4 IPs.

We run Nmap scan on each IP and try to look for any open ports or anything unusual.

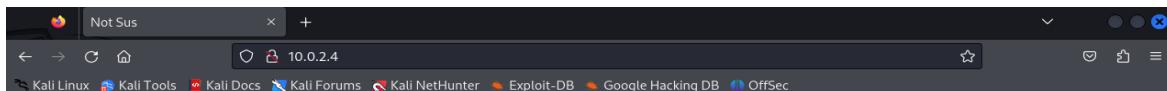
```
(kali㉿kali)-[~]
$ nmap 10.0.2.4
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-11 16:19 EDT
Nmap scan report for 10.0.2.4
Host is up (0.0033s latency).
Not shown: 968 filtered tcp ports (no-response), 30 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 3.85 seconds
```

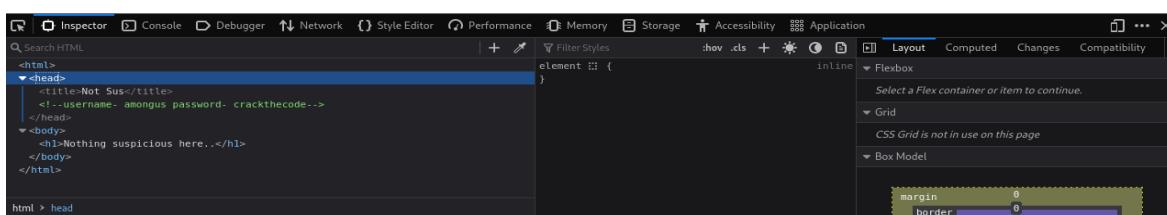
Only the last IP has FTP and HTTP ports open.

So this could be the IP of the target machine.

Using the obtained IP, we can see if <http://10.0.2.4> has anything of use.



Nothing suspicious here..



```
<html>
  <head>
    <title>Not Sus</title>
    <!--username- amongus password- crackthecode-->
  </head>
  <body>
    <h1>Nothing suspicious here..</h1>
  </body>
</html>
```

Under inspect, we find the username and password for the target machine.

Username - amongus
Password - crackthecode

Exploitation-

Now, we can use these credentials to see what the ftp port holds.

```
(kali㉿kali)-[~]
$ ftp 10.0.2.4
Connected to 10.0.2.4.
220 (vsFTPd 3.0.5)
Name (10.0.2.4:kali): amongus
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -al
229 Entering Extended Passive Mode (|||46463|)
150 Here comes the directory listing.
dr-xr-xr-x    3 65534      65534        4096 Oct 12 01:31 .
dr-xr-xr-x    3 65534      65534        4096 Oct 12 01:31 ..
drwxr-xr-x   2 1000      1000 "the quiet you become, the more yo
226 Directory send OK.
ftp> cd files
250 Directory successfully changed.
```

```
ftp> ls
229 Entering Extended Passive Mode (|||42458|)
150 Here comes the directory listing.
-rw-r--r--    1 0          0           22 Oct 12 01:25 sus.txt
226 Directory send OK.
ftp> get sus.txt
local: sus.txt remote: sus.txt
229 Entering Extended Passive Mode (|||41745|)
150 Opening BINARY mode data connection for sus.txt (22 bytes).
100% [*****] 22          2.64 KiB/s   00:00 ETA
226 Transfer complete.
22 bytes received in 00:00 (1.99 KiB/s)
ftp> exit
221 Goodbye.

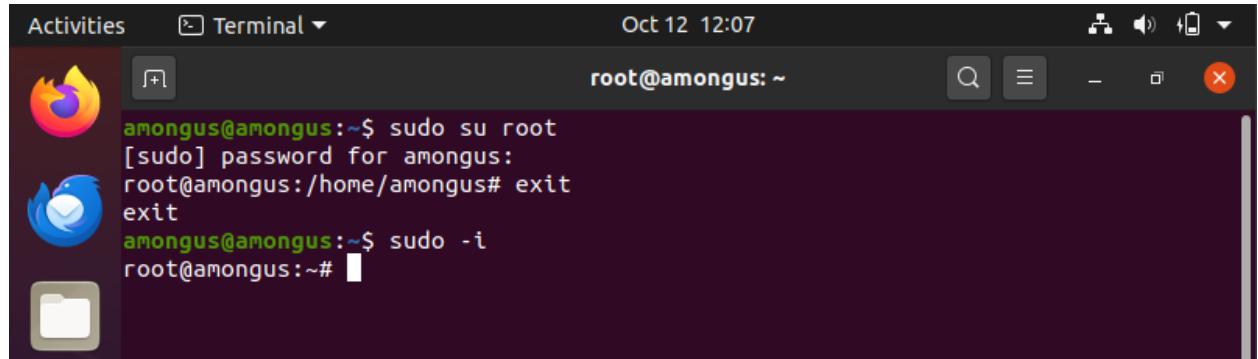
[(kali㉿kali)-~]
└─$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  sus.txt  Templates  Videos

[(kali㉿kali)-~]
└─$ cat sus.txt
AmongUs{FTP_FTW_1726}
```

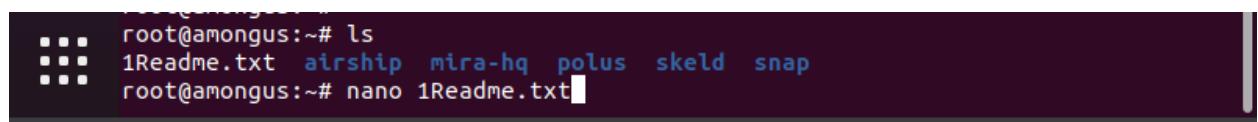
We've successfully found our first flag- **AmongUs{FTP_FTW_1726}**

Now, we can log into ubuntu using these credentials too.

Enter into root user using the command : “sudo su root” or “sudo -i”



View all the folders present in the root directory. Start with 1Readme.txt



1Readme.txt contains the 1st flag.

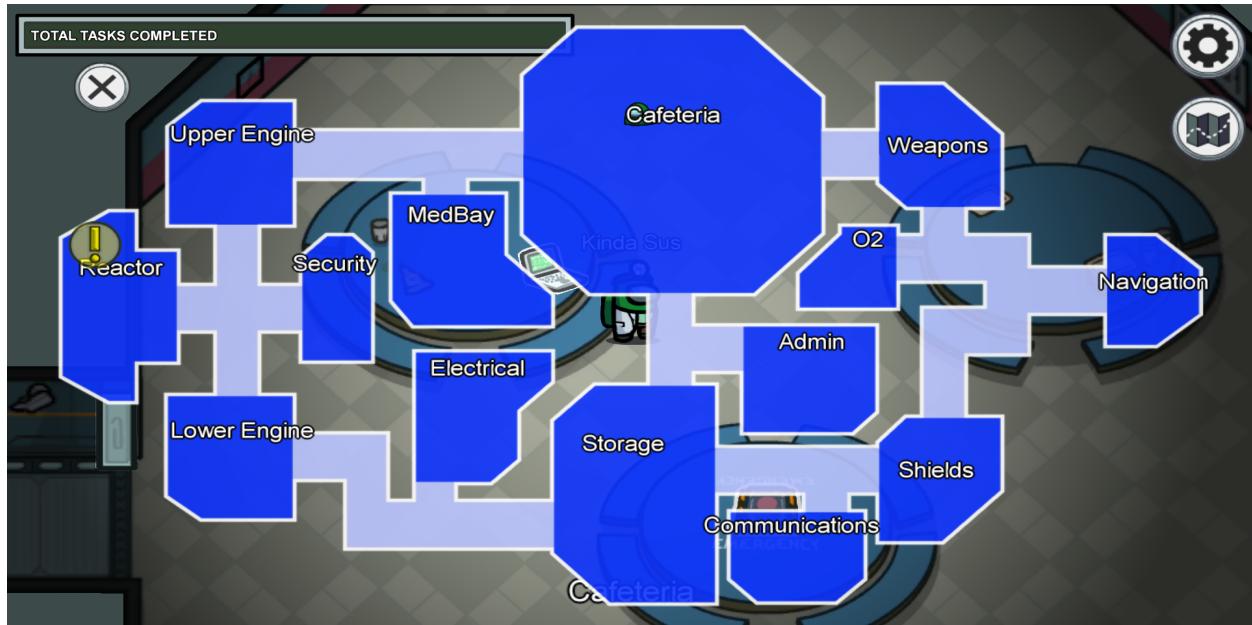
```
GNU nano 4.8                                1Readme.txt
3 2 1...
AmongUs{y0u_ar3_th3_1mp0st3r}
Find and kill all the crewmates from the 4 maps!
```



Flag 2 - **AmongUs{y0r_ar3_th3_1mp0st3r}**

Task : Check out all the rooms and complete the tasks.

Room 1 : Skeld



Entering **skeld** following files can be found.

```
* root@amongus: ~# cd skeld
root@amongus:~/skeld# ls
1Readme.txt admin.room.txt Cafeteria.room.txt 02.room.txt shields.room.txt
root@amongus:~/skeld# nano 1Readme.txt
root@amongus:~/skeld#
```

1Readme.txt will contain the directions to be followed to find the flags.

Each flags will be connected. Keep following the flags to find the next location.

1Readme.txt

```
Activities Terminal Oct 18 17:47
root@amongus: ~/skeld
GNU nano 4.8
Task: Kill WhiteHead
I heard he's doing a task in the Cafeteria!
```

That's a clue. So we need to start with Cafeteria

Command : cd Cafeteria.room.txt

A screenshot of a Linux desktop environment. At the top, there's a header bar with "Activities", a "Terminal" icon, the date and time "Oct 18 17:49", and system icons for volume, battery, and screen rotation. Below the header is a dock with icons for a browser (Firefox), a terminal, and a file manager. A window titled "root@amongus: ~/skeld" is open in the center. The terminal window has a dark background. It shows the user is in the "nano" text editor version 4.8, editing a file named "Cafeteria.room.txt". The text in the file is: "6#COXB2s#"@q[Z.CL9>;D)#pV". There are standard window controls (minimize, maximize, close) at the top right of the terminal window.

Use online decodes to decode the flag.

Link used - <https://gchq.github.io/CyberChef/>

The screenshot shows the CyberChef extension in a browser window. The main interface has a green header bar with the text "Last build: 3 months ago - Version 10 is here! Read about the new features here". On the left, there's a sidebar with various tools like "Operations", "magic", "Magic", "Image Brightness / Contrast", "Detect File Type", and "Scan for Embedded Files". Below that is a "Favourites" section with a star icon. The main workspace is titled "Recipe" and contains a "Magic" step. The "Magic" step has a dropdown menu set to "Depth 3" and an "Intensive mode" checkbox. Below it are "Extensive language support" and "Crib (known plaintext string)" options. The "Input" field contains the hex string "6#COXB2s#@q[Z.CL9>;D]#pV". The "Output" section shows two rows of results. The first row is for "From_Base65('!-u')", with the result snippet "MongoDB\ch3ck_\\$admin" and properties: Matching ops: From Base65, Valid UTF8, Entropy: 4.02. The second row is for "6#COXB2s#@q[Z.CL9>;D]#pV", with the result snippet "6#COXB2s#@q[Z.CL9>;D]#pV" and properties: Matching ops: From Base65, Valid UTF8, Entropy: 4.37.

Decode flag - AmongUs{ch3ck @dm1n}

The flag directs to the next location and it is admin.

Now open admin.roon.txt file using the command **nano admin.room.txt**

Use the same decoder to decode the file

The decoded content gives an hint that the flag is encode using RC4 algorithm.

Recipe (click to load)	Result snippet	Properties
Decode_text('IBM EBCDIC International (500)')	RC4 s st v-t y t. Ts t yts yt t (uts t t). A y ut s su t t t+ut us st 8-t ut s ut w...	Possible languages: Romanian Entropy: 2.76
Decode_text('IBM EBCDIC US-Canada (37)')	RC4 s st v-t y t. Ts t yts yt t (uts t t). A y ut s su t t t+ut us st 8-t ut s ut w...	Possible languages: Romanian Entropy: 2.76
Decode_text('IBM EBCDIC Multilingual/ROECE (Latin 2) (870)')	RC4 s st v-t y t. Ts t yts yt t (uts t t). A y ut s su t t t+ut us st 8-t ut s ut w...	Possible languages: Romanian Entropy: 2.76
Decode_text('IBM EBCDIC Greek Modern (875)')	RC4 s st v-t y t. Ts t yts yt t (uts t t). A y ut s su t t t+ut us st 8-t ut s ut w...	Possible languages: Romanian Entropy: 2.76

The decoded content gives an hint that the flag is encode using RC4 algorithm.

Decrypted flag - **AmongUs{9o_t0_5h13ld5}**

Now it directs the user to the next location shields.

Command : nano shields.room.txt

In shields.room.txt

```
GNU nano 4.8
root@amongus: ~/skeld
shields.room.txt
Task : Open and decrypt the flag 2914F510263E70D06F46547F0685D114373739E777246A7EC84B16E53FB21270.
hint : we need key to open a room.
```

Online decrypter

Google search results for "des decoder and decrypter". The search bar shows the query. Below it, the "All" tab is selected, followed by other categories like Images, Videos, Books, Shopping, More, and Tools. The results page indicates about 10,50,000 results found in 0.34 seconds. A red link at the top suggests searching for "des decoder and decryption". The first result is from DevGlan, titled "Triple DES Encryption and Decryption Online Tool", which is described as a free online tool for Triple DES encryption and decryption using ECB and CBC modes.

des decoder and decrypter

All Images Videos Books Shopping More Tools

About 10,50,000 results (0.34 seconds)

Search instead for: des decoder and **decryption**

DevGlan https://www.devglan.com › triple-des-encrypt-decrypt

Triple DES Encryption and Decryption Online Tool

Triple DES encryption and decryption online tool for free. It is an DES calculator that performs encryption and decryption of text in ECB and CBC mode.

Triple DES Online Decryption

Enter text to be Decrypted

```
2914F510263E70D06F46547F0685D114373739E7  
77246A7EC84B16E53FB21270
```

Input Text Format: Base64 Hex

Select Mode

ECB

Enter Secret Key

```
shieldsshieldsshieldsshieldsshieldsshields
```

Decrypt

Triple DES Decrypted Output (**Base64**):

```
QWlvbmdVc3tEM0BkYjBkeWYwdW5kQE8yfQ=  
=
```

Decode to Plain Text

```
AmongUs{D3@db0dyf0und@O2}
```

Flag - **AmongUs{D3@db0dyf0und@O2}**

Now it directs the user to the next location O2



The screenshot shows a terminal window titled "root@amongus:/skeld". It displays the command "root@amongus:~\$ nano 02.room.txt" and the contents of the file "02.room.txt" which is a large block of base64 encoded text. The terminal interface includes standard Linux-style navigation keys and a status bar at the bottom.

Decrypted content - PEOPLE SAY THIS PLACE IS HAUNTED AND THERE ARE
TOTALLY 47 LOWER GOST :) HERE. DO U SEE ANY? IF NO TRY FINDING THE
MEANING OF

1FB7E64F55EBDDDD755F85C76A4BE5B7ECAEA8188A842E30C16C9080D889C5A
3

Decrypted flag - **AmongUs{D3@db0y_R3p0rted}**

Room 2 - Mira-HQ



Entering **mira-hq** map, the following files and directories can be found.

```
root@amongus:~# ls
1Readme.txt airship mira-hq polus skeld snap
root@amongus:~# cd mira-hq
root@amongus:~/mira-hq# ls
launchpad.txt reactor.room.gcode welcome.mirahq.png
root@amongus:~/mira-hq#
```

Welcome.mirahq.png

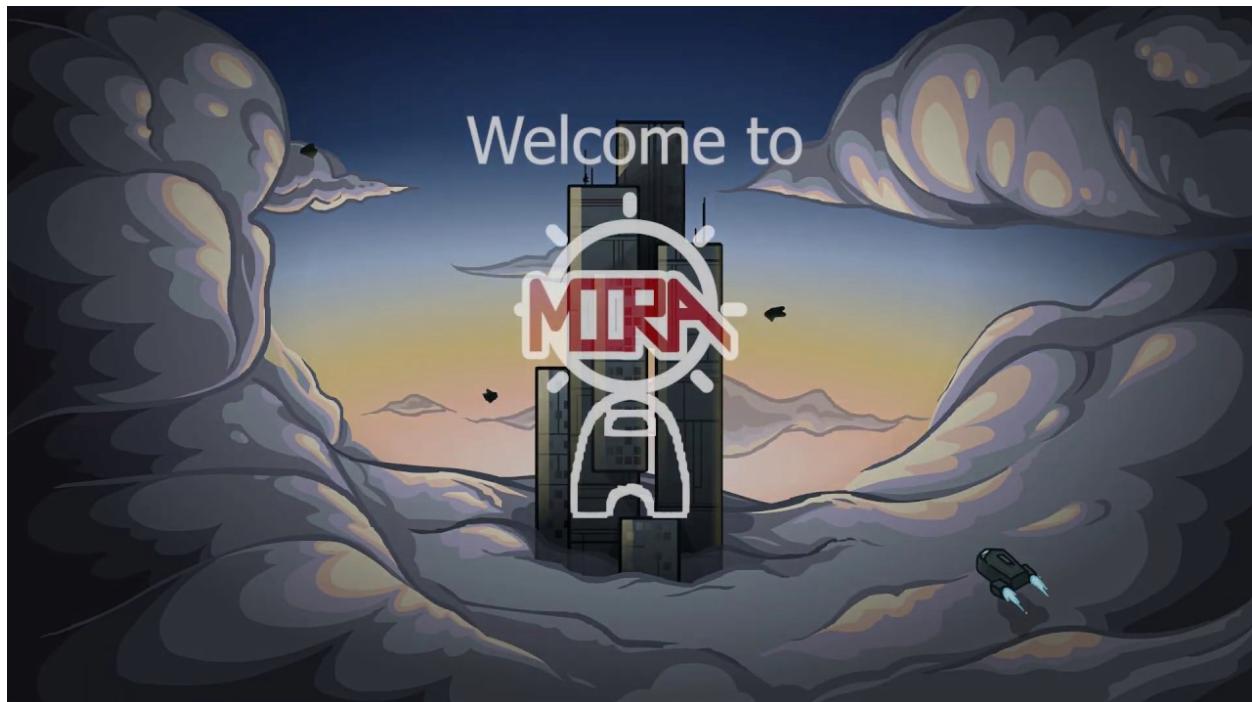
The png file can be moved or downloaded from the terminal to the home directory.

The command to move the .png file to home directory is :

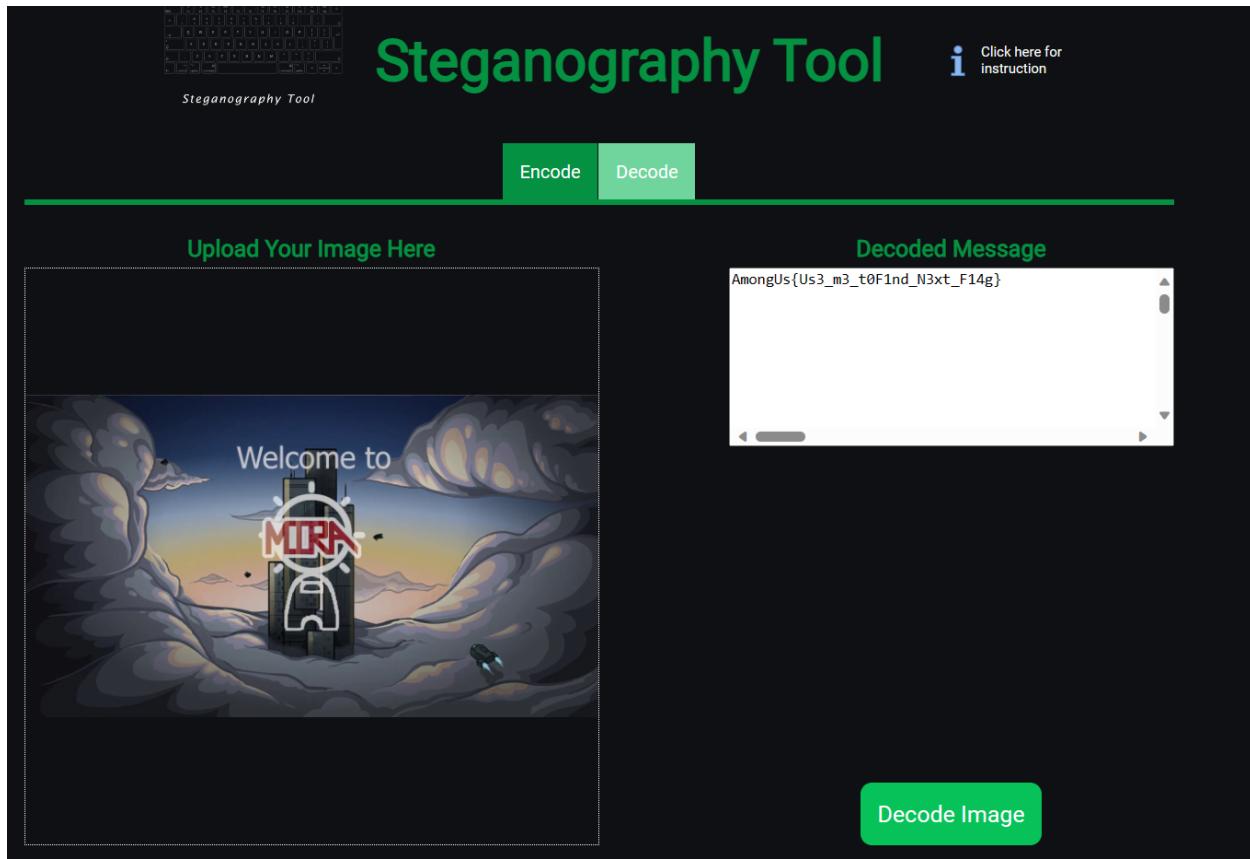
```
mv welcome.mirahq.png /home/Downloads
```

The image file will be found in the downloads directory.

The image file is



The Flag is hidden in this image. Using online tool the file can be uploaded and the flag can be retrieved.



Decrypted flag - **AmongUs{Us3_m3_t0F1nd_N3xt_F14g}**

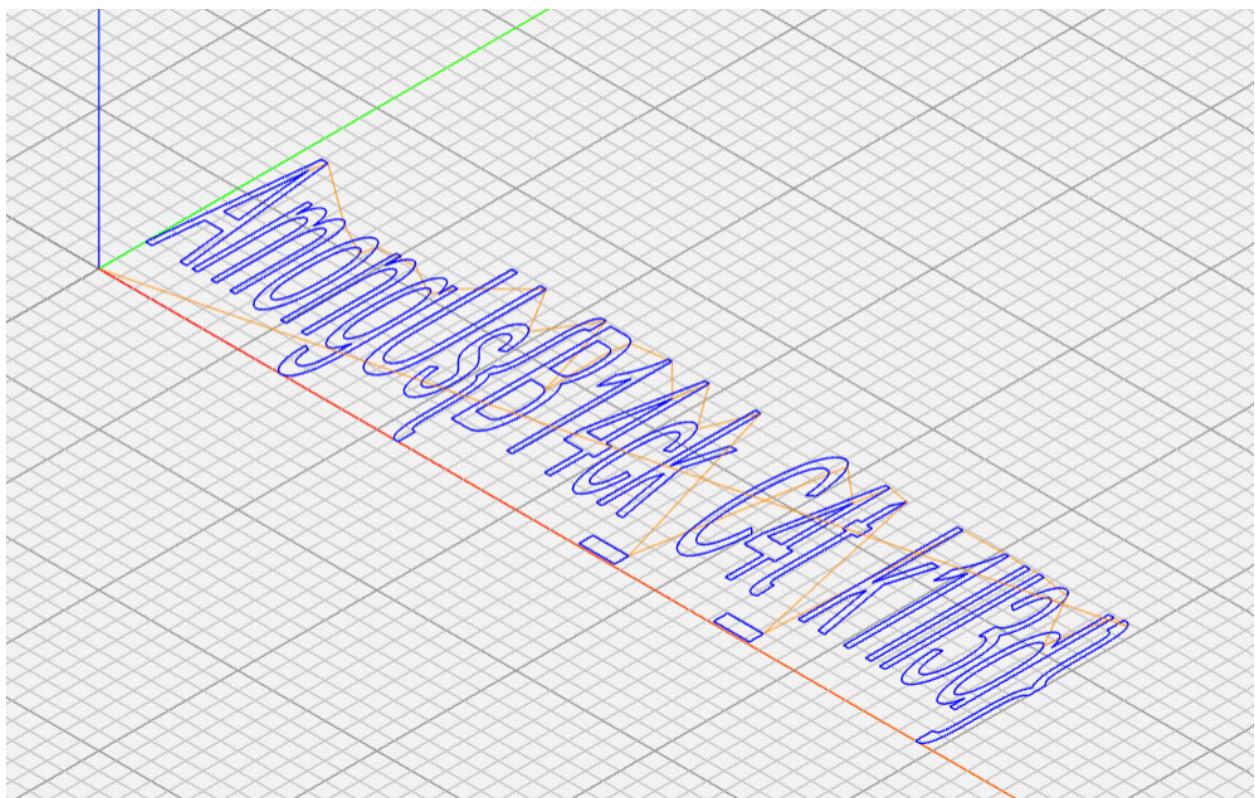
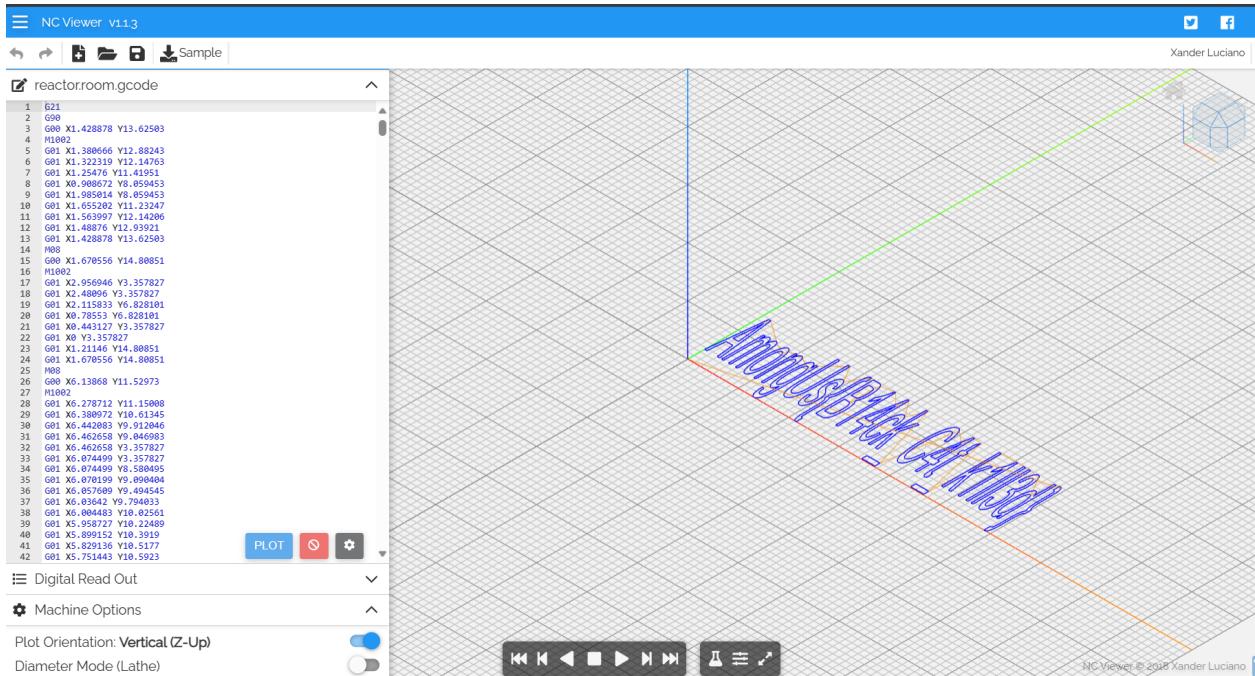
reactor.room.gcode

The file contains gcode which needs to be decoded using online tool.
Using nc viewer the gcode file can be opened. It plots a graph containing flag in it.

Content inside reactor.room.gcode

```
G00 X1.670556 Y14.80851
M1002
G01 X2.956946 Y3.357827
G01 X2.48096 Y3.357827
G01 X2.115833 Y6.828101
G01 X0.78553 Y6.828101
G01 X0.443127 Y3.357827
G01 X0 Y3.357827
G01 X1.21146 Y14.80851
G01 X1.670556 Y14.80851
M08
G00 X6.13868 Y11.52973
M1002
G01 X6.278712 Y11.15008
G01 X6.380972 Y10.61345
G01 X6.442083 Y9.912046
G01 X6.462658 Y9.046983
G01 X6.462658 Y3.357827
G01 X6.074499 Y3.357827
G01 X6.074499 Y8.580495
G01 X6.070199 Y9.090404
G01 X6.057609 Y9.494545
G01 X6.03642 Y9.794033
G01 X6.004483 Y10.02561
G01 X5.958727 Y10.22489
G01 X5.899152 Y10.3919
G01 X5.829136 Y10.5177
G01 X5.751443 Y10.5923
G01 X5.665766 Y10.6179
G01 X5.51345 Y10.55333
G01 X5.37956 Y10.35961
G01 X5.263788 Y10.03674
G01 X5.175039 Y9.571365
G01 X5.121606 Y8.950123
G01 X5.103795 Y8.174127
G01 X5.103795 Y3.357827
G01 X4.715636 Y3.357827
G01 X4.715636 Y8.744155
G01 X4.705195 Y9.316411
G01 X4.673565 Y9.785127
G01 X4.621054 Y10.14919
G01 X4.544589 Y10.40971
G01 X4.4411 Y10.56558
G01 X4.311202 Y10.6179
```

Copying and pasting the content in this file in the tool will give the result.

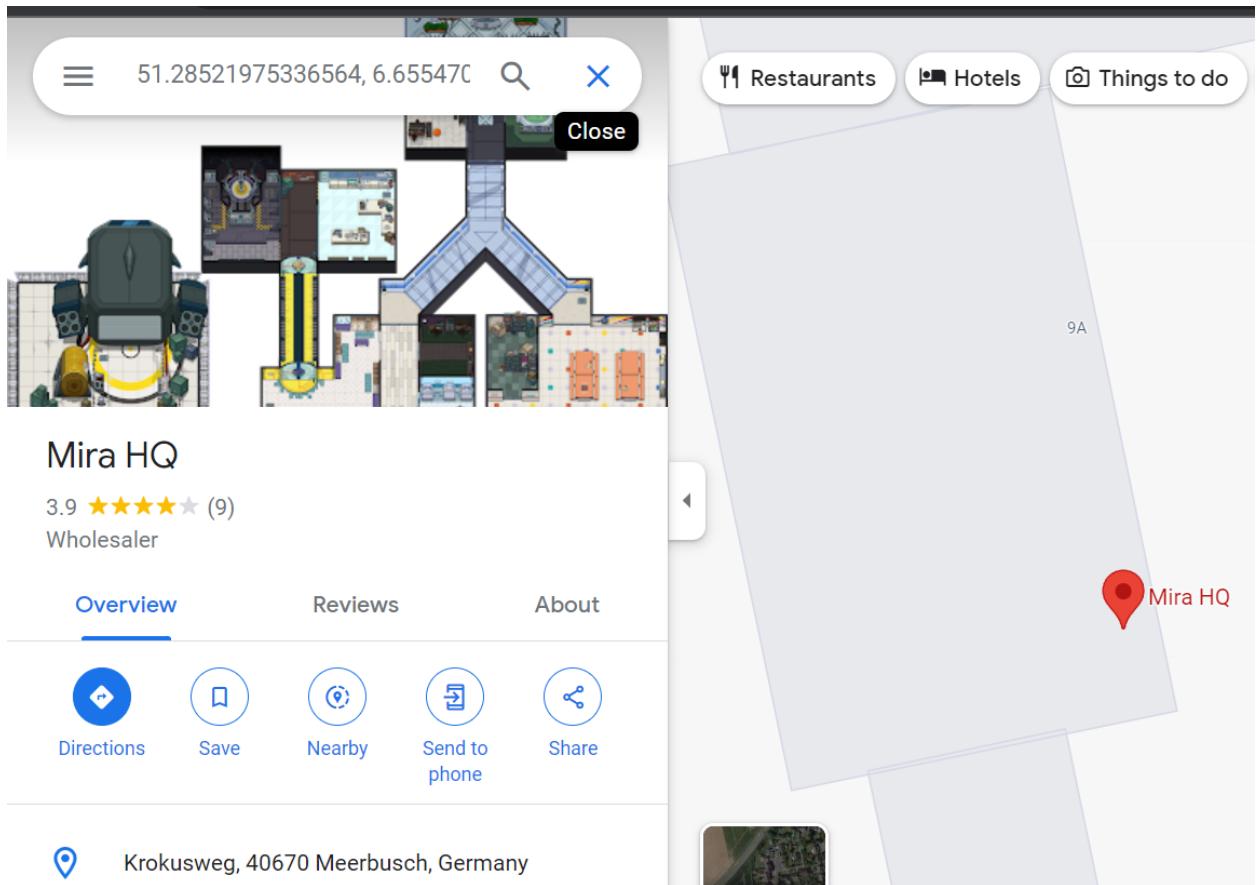


The flag found - **AmongUs{B14ck_c4t_k1ll3d}**

Content in launchpad.txt

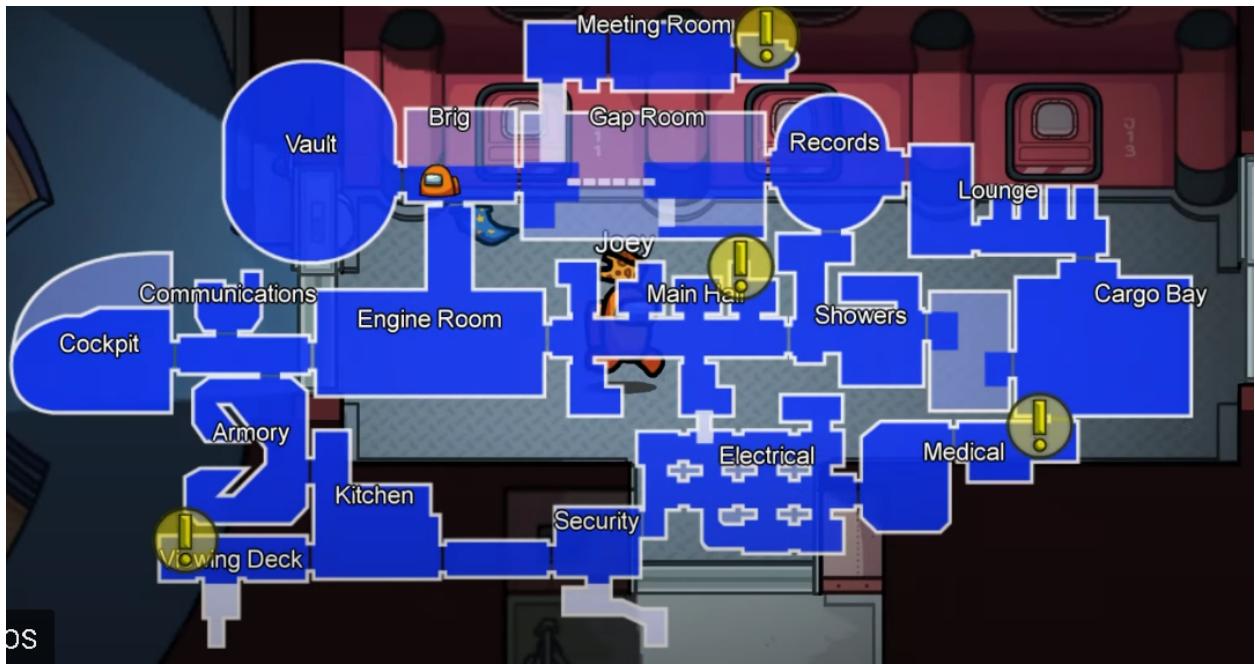
```
GNU nano 4.8
Launching in 3.. 2.. 1..
What country did I land in?
51.28521975336564, 6.655470674012466
AmongUs{country_name}
```

Country name is **Germany**



The flag is : **AmongUs{Germany}**

Room 3 : Airship



Entering **airship** map, the following files and directories can be found.

```
root@amongus:~# cd airship
root@amongus:~/airship# ls
1Readme.txt  electrical  kitchen  medical  meeting_room  security
root@amongus:~/airship# nano 1Readme.txt
```

Command : **nano 1Readme.txt**

```
GNU nano 4.8
1Readme.txt
You are 30 mins late to the meeting. Hurry up! Get to the meeting room soon!
```

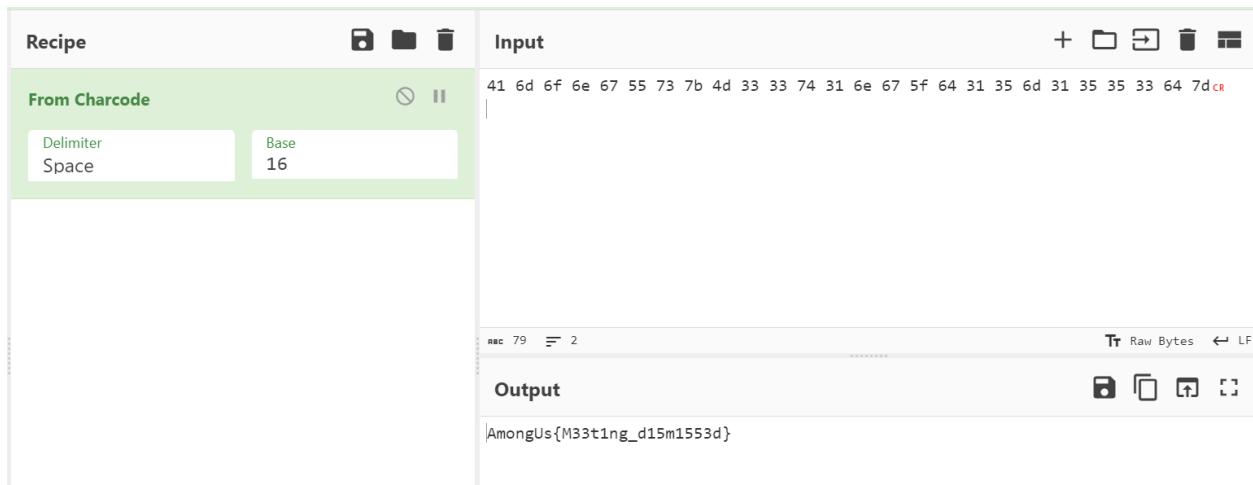
Meeting_room

The directory contains the file Important.c which has the flag

```
root@amongus:~/airship# ls
1Readme.txt electrical kitchen medical meeting_room security
root@amongus:~/airship# cd meeting_room
root@amongus:~/airship/meeting_room# ls
Important.c
root@amongus:~/airship/meeting_room# cat Important.c
#include <stdio.h>
int main() {
    int flag;
    while(1){
        printf("\nEnter a number : ");
        scanf("%d",&flag);
        if (flag == 30)
        {
            printf("Good Luck.");
            printf("\nThe flag -> 41 6d 6f 6e 67 55 73 7b 4d 33 33 74 31 6e 67 5f 64 31 35 6d 31 35 35
33 64 7d");
            break;
        }
        else{
            printf("\nTry again.");
        }
    }
    return 0;
}
root@amongus:~/airship/meeting_room#
```

Encrypted flag - 41 6d 6f 6e 67 55 73 7b 4d 33 33 74 31 6e 67 5f 64 31 35 6d 31 35 35
33 64 7d

It's encoded in charcode.



Decrypted Flag - **AmongUs{M33t1ng_d15m1553d}**

medical

Files inside medical

```
root@amongus:~/airship# cd medical
root@amongus:~/airship/medical# ls
medical.room.py  usethis.enc
root@amongus:~/airship/medical#
```

Command : nano usethix.enc

```
GNU nano 4.8
òöËËýÎÑè÷^wÑ¬ÄïóïçíÅ¶Ê ÄÓ^tèþ^wî
```

Content inside medical.room.py

```
GNU nano 4.8 medica
import sys
a = "!\\"#$%&'^)*+, -./0123456789:;<=>?@ABCDEFGHIJKLMNPQRSTUVWXYZ"+ \
    "[\\]^_`abcdefghijklmnopqrstuvwxyz{|}~"
def arg133(arg432):
    if arg432 == a[71]+a[64]+a[79]+a[79]+a[88]+a[66]+a[71]+a[64]+a[77]+a[66]+a[68] :
        return True
    else:
        print(a[51]+a[71]+a[64]+a[83]+a[94]+a[79]+a[64]+a[82]+a[82]+a[86]+a[78]+\
a[81]+a[67]+a[94]+a[72]+a[82]+a[94]+a[72]+a[77]+a[66]+a[78]+a[81]+\
a[81]+a[68]+a[66]+a[83])
        sys.exit(0)
    return False
def arg111(arg444):
    return arg122(arg444.decode(),a[64]+a[72]+a[81]+a[82]+a[71]+a[72]+a[79])
def arg232():
    return input(a[47]+a[75]+a[68]+a[64]+a[82]+a[68]+a[94]+a[68]+a[77]+a[83]+\
a[68]+a[81]+a[94]+a[66]+a[78]+a[81]+a[81]+a[68]+a[66]+a[83]+\
a[94]+a[79]+a[64]+a[82]+a[82]+a[86]+a[78]+a[81]+a[67]+a[94]+\
a[69]+a[78]+a[81]+a[94]+a[69]+a[75]+a[64]+a[70]+a[25]+a[94])
def arg132():
    return open('usethis.enc', 'rb').read()
def arg112():
    print(a[54]+a[68]+a[75]+a[66]+a[78]+a[76]+a[68]+a[94]+a[65]+a[64]+a[66]+\
a[74]+a[13]+a[13]+a[13]+a[94]+a[88]+a[78]+a[84]+a[81]+a[94]+a[69]+\
a[75]+a[64]+a[70]+a[11]+a[94]+a[84]+a[82]+a[68]+a[81]+a[25])
def arg122(arg432, arg423):
    arg433 = arg423
    i = 0
    while len(arg433) < len(arg432):
        arg433 = arg433 + arg423[i]
        i = (i + 1) % len(arg423)
    return "".join([chr(ord(arg422) ^ ord(arg422)+50) for (arg422,arg442) in zip(arg432,arg433)])
arg444 = arg132()
arg432 = arg232()
arg133(arg432)
arg112()
arg423 = arg111(arg444)
print(arg423)

sys.exit(0)
```

Decode and find the password - happychance

```
root@amongus:~/airship/medical#
root@amongus:~/airship/medical# python3 medical.room.py
Please enter correct password for flag: happychance
Welcome back... your flag, user:
AmongUs{l3t5_m@tch_th³_w1re5}
root@amongus:~/airship/medical#
```

Flag - **AmongUs{l3t5_m@tch_th3_w1re5}**

Electrical

Files inside electrical

```
root@amongus:~/airship# cd electrical
root@amongus:~/airship/electrical# ls
readit.txt  solve.S
root@amongus:~/airship/electrical# nano solve.S
```

Content inside readit.txt

```
GNU nano 4.8
readit.txt
What integer does this program print with arguments 3277441615 and 216522756? The obtained flag should be replaced with the hex of the output and the whole flag should be submitted.
(hex, no 0x. ex- 5614267 would be AmouguS{0055aabb})
```

Content inside solve.S

```
.arch armv8-a
.file "solve.S"
.text
.align 2
.global func1
.type func1, %function
func1:
    sub sp, sp, #16
    str w0, [sp, 12]
    str w1, [sp, 8]
    ldr w1, [sp, 12]
    ldr w0, [sp, 8]
    cmp w1, w0
    bne .L2
    ldr w0, [sp, 12]
    b .L3
.L2:
    ldr w0, [sp, 8]
.L3:
    add sp, sp, 16
    ret
.size func1, .-func1
.section .rodata
.align 3
.LC0:
    .string "Result: %lx\n"
.text
.align 2
.global main
.type main, %function
main:
    stp x29, x30, [sp, -48]!
    add x29, sp, 0
    str x19, [sp, 16]
    str w0, [x29, 44]
    str x1, [x29, 32]

    ldr w0, [x29, 44]    ; Load the first argument into w0
    ldr w1, [x29, 44]    ; Load the first argument again into w1

    lsl w0, w0, #3        ; Shift left by 3 bits (1 digit to the left)
    lsr w1, w1, #29       ; Shift right by 29 bits (1 digit to the right)
    orr w0, w0, w1        ; Combine the results

    bl func1
    adrp x0, .LC0
    add x0, x0, :lo12:.LC0
    bl printf
```

Assembly code that takes the 1st input and does a circular left shift and converts the value to hex.

The output is : a55e3b19

Flag - **AmongUs{a55e3b19}**

Security

Content inside security

```
root@amongus:~/airship# cd security
root@amongus:~/airship/security# ls
open.py
root@amongus:~/airship/security# nano open.py
```

Content inside open.py

```
def dfg3(ftg)
    ji, ij = ftg.ord / 10, (ftg.ord % 10) / 2
    (ftg.ord + "#{ij.to_i}#{(ij*ij).to_i}" .to_i).chr
end

def d45d(sde)
    dd, fr = sde.ord / 10, sde.ord % 10
    (dd**fr-2) + fr**fr-1 .to_i).chr
end

def pi8u
    an = ''
    (1..3).each { |i| an += i.to_s }
    an.to_i.chr
end

def xc44(rty)
    (rty.ord * 2 - 1).chr
end

bdy = ''

def ooki(wde)
    cv, vc = wde.ord / 10, wde.ord % 10
    ("#{(cv*vc).to_l}#{(cv+vc).to_i}" .to_i - 32).chr
end

def e439
    'n'
end

def llj
    'g'
end

def asew(cgh)
    (cgh.ord + cgh.ord / 10 - 32).chr
end

def ytrs(oki)
    dd, fr = oki.ord / 10, oki.ord % 10
    (dd**fr-2) + fr**fr-1 .to_i).chr
end

def gftr(mjk)
    vf = mjk.ord / 10
    num = vf * vf
    (num * 10 + num - 32).chr
end
```

```

def ytrs(ook)
  dd, fr = ook.ord / 10, ook.ord % 10
  (dd*(fr-2) + fr**((dd-1)) - (dd*fr)).chr
end

def gfr(mjk)
  vf = mjk.ord / 10
  num = vf * vf
  (num * 10 + num - 32).chr
end

def apt(vpr)
  vpr = vpr + pi8u + asew('d') + ytrs('$') + dfg3('`') + i9o8('?', 'z') + gftr('!') + dfg3('4')
  vpr = vpr + gftr('#') + ytrs('$') + dfg3('8') + d45d('$') + ookl('!') + xc44('!') + i9o8('!') + xc44('!') + fgtr4('`')
end

def i9o8(ft, ck = nil)
  ck = ft if ck.nil?
  (ft.ord + ck.ord - 32).chr
end

def fgtr4(xxc)
  nn, nn = xxc.ord / 10, xxc.ord % 10
  (xxc.ord + "#{(nn*10 + nn).to_i}").to_i + nn.ord).chr
end

def kij7(fgt)
  (fgt.ord + 2).chr
end

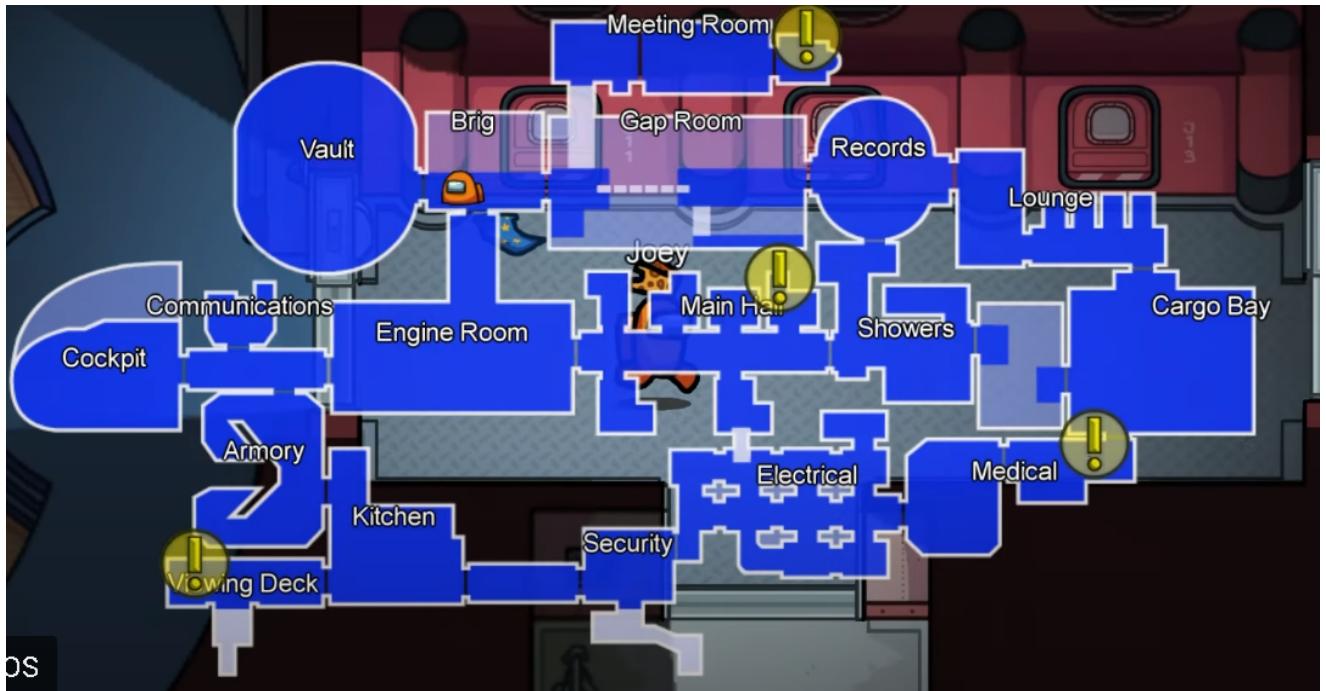
def zxis(ftr)
  ef, gh = ftr.ord / 10, ftr.ord % 10
  (-32 + ftr.ord + "#{((ef+1)*10 + gh+1)}".to_i).chr
end

akh = zxis('e') + fgtr4('`') + dfg3('o') + e439() + llj() + xc44('!') + xc44(';')
bby = apt(akh)
for i in 1..3
  bdy += llj
end
bby+== kij7(pi8u())
bdy += e439

```

Execute the ruby code to get the flag.
Flag - AmongUs{Nc:IC?Cc_HK?0Ym}

Room 4 - Polus



Entering **polus**, the following files can be found

```
root@amongus:~# cd polus
root@amongus:~/polus# ls
reactor.room.txt
root@amongus:~/polus#
```

reactor.room.txt

it is encrypted using Railfence cipher.

Encrypted flag - AogsKl3_nt3_rw43mnU{1ld@0hrc3mt}

Decrypted flag - **AmongUs{K1ll3d_@n0th3r_cr3wm4t3}**

If we try **ls -al**, we find another hidden file **.storage.txt**

```
root@amongus:~/polus# ls -al
total 16
drwxr-xr-x 2 amongus amongus 4096 Oct 11 23:57 .
drwx----- 9 amongus amongus 4096 Oct 11 17:18 ..
-rwxr--r-- 1 amongus amongus    73 Oct 11 17:18 reactor.room.txt
----- 1 root      root      26 Oct 11 23:55 .storage.txt
```

Flag- **AmongUs{jUmp_1n_th3_v3nt}**