WriteUp ApoorvCTF 2025

Cyber Alliance

hush1a & mshazaw

Reverse Engineering - Holy Rice (Unfinished)

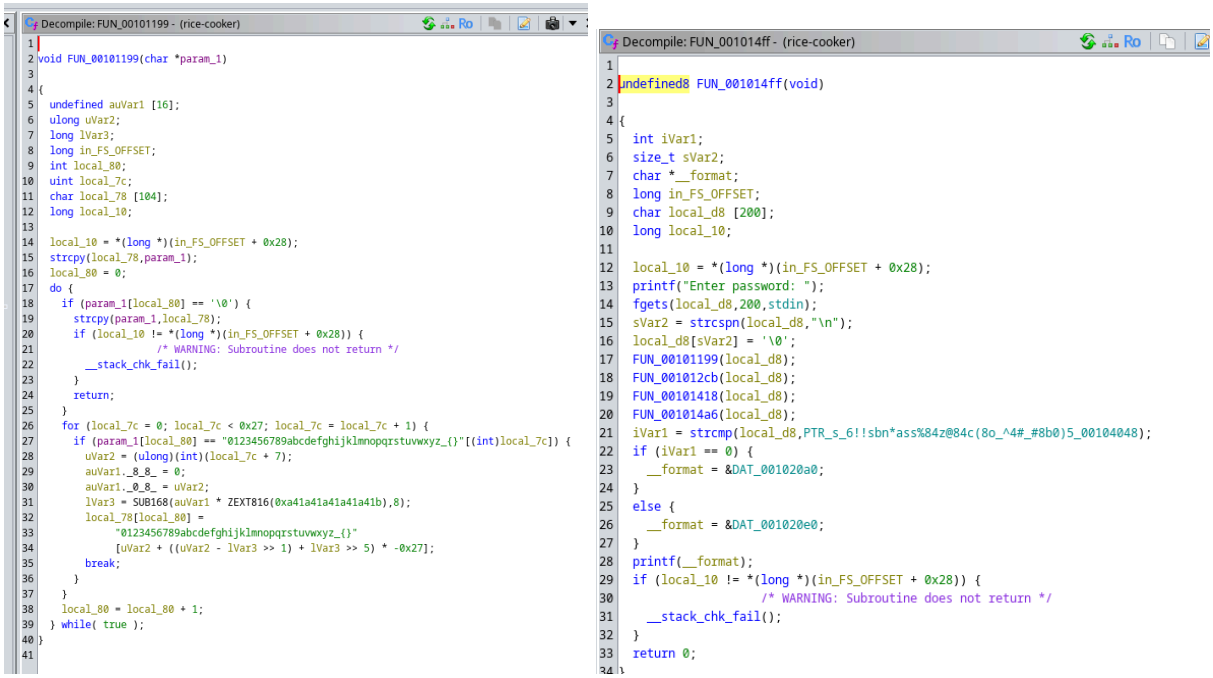
A wise monk made a "totally unhackable" locker to guard his holy rice. Spoiler: it wasn't. The pigeon mafia is already on it — crack it first and claim the rice for yourself. 🐦🍚

Author: *hampter*

```
┌──(~/Downloads/CTF/ApoorvCTF/rev)
└─❯ chmod +x rice-cooker

┌──(~/Downloads/CTF/ApoorvCTF/rev)
└─❯ ./rice-cooker
Enter password: test
no rice for you bro 🍚❌
```

I added the option to execute the binary file, and it prompted me for a password.



Decompile: FUN_00101199 - (rice-cooker)

```
void FUN_00101199(char *param_1)

{
  undefined auVar1 [16];
  ulong uVar2;
  long lVar3;
  long in_FS_OFFSET;
  int local_80;
  uint local_7c;
  char local_78 [104];
  long local_10;

  local_10 = *(long *)(in_FS_OFFSET + 0x28);
  strcpy(local_78,param_1);
  local_80 = 0;
  do {
    if (param_1[local_80] == '\0') {
      strcpy(param_1,local_78);
      if (local_10 != *(long *)(in_FS_OFFSET + 0x28)) {
                    /* WARNING: Subroutine does not return */
        __stack_chk_fail();
      }
      return;
    }
    for (local_7c = 0; local_7c < 0x27; local_7c = local_7c + 1) {
      if (param_1[local_80] == "0123456789abcdefghijklmnopqrstuvwxyz_{}"[(int)local_7c]) {
        uVar2 = (ulong)(int)(local_7c + 7);
        auVar1._8_8_ = 0;
        auVar1._0_8_ = uVar2;
        lVar3 = SUB168(auVar1 * ZEXT816(0xa41a41a41a41a41b),8);
        local_78[local_80] =
             "0123456789abcdefghijklmnopqrstuvwxyz_{}"
             [uVar2 + ((uVar2 - lVar3 >> 1) + lVar3 >> 5) * -0x27];
        break;
      }
    }
    local_80 = local_80 + 1;
  } while( true );
}
```

Decompile: FUN_001014ff - (rice-cooker)

```
undefined8 FUN_001014ff(void)

{
  int iVar1;
  size_t sVar2;
  char *__format;
  long in_FS_OFFSET;
  char local_d8 [200];
  long local_10;

  local_10 = *(long *)(in_FS_OFFSET + 0x28);
  printf("Enter password: ");
  fgets(local_d8,200,stdin);
  sVar2 = strcspn(local_d8,"\n");
  local_d8[sVar2] = '\0';
  FUN_00101199(local_d8);
  FUN_001012cb(local_d8);
  FUN_00101418(local_d8);
  FUN_001014a6(local_d8);
  iVar1 = strcmp(local_d8,PTR_s_6!!sbn*ass%84z@84c(8o_^4#_#8b0)5_00104048);
  if (iVar1 == 0) {
    __format = &DAT_001020a0;
  }
  else {
    __format = &DAT_001020e0;
  }
  printf(__format);
  if (local_10 != *(long *)(in_FS_OFFSET + 0x28)) {
                    /* WARNING: Subroutine does not return */
    __stack_chk_fail();
  }
  return 0;
}
```

When analyzing the code in Ghidra, I found this function in the main that seemingly does an encoding operation and also character shifting by 7. But there are other functions that I did not understand what they do.
There is also a charset for the password.

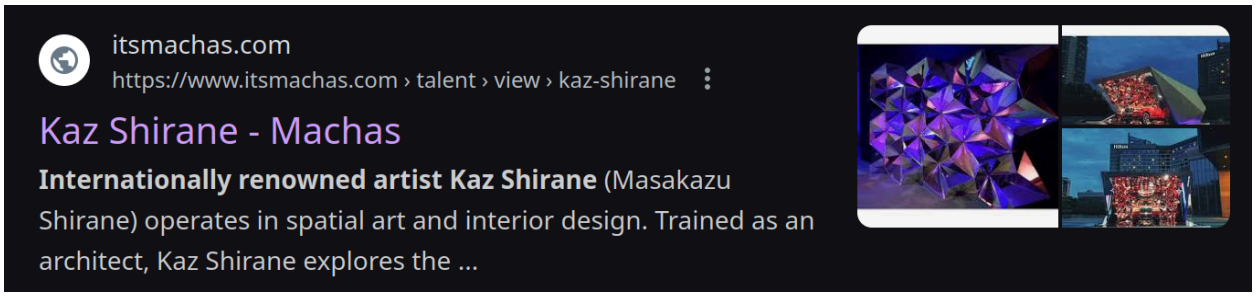
I tried to write a script to reverse the process but I couldn't find a string that the code compares with an inputted password, so :(

OSINT - I Love Japan: Identity Game - 100 pts

Oh! My friend send me this! So Pretty isn't it ^_^ I wonder who the designer could be?
Flag Format: apoorvctf{firstname_lastname} Note: The Flag is not a username/ nickname

Author: *Stargazer*

This picture is the Tokyo Tower, specifically the Top Deck.
When I searched for designers, a name came up : Kaz Shirane.



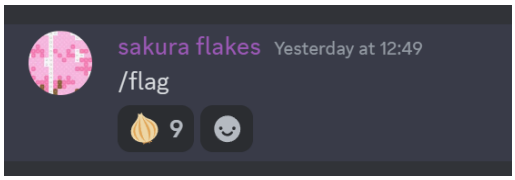
After searching his name, I found that it was a professional name and his real name is actually Masakazu Shirane.

Flag: **apoorvctf{Masakazu_shirane}**

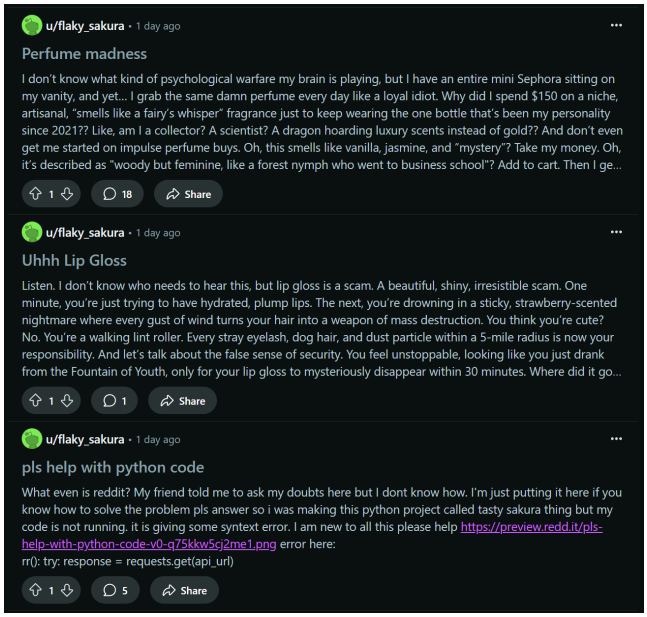
OSINT - Sakura beads (Unfinished)

Hey everyone, I have a friend Sakura. She wanted to learn how programming works so I gave her the best advice anyone could have given her- participate in ApoorvCTF.
She's trying her best to find flags and could only find the welcome flag, sending /flag XD.
If you consider yourself real OSINTers — stalk her.

Author: cooker



Found the discord account, which has a link to her reddit.



Bunch of rants,

This post was an odd one, but I couldn't find anything :/

OSINT - 404 Location Not Found - 385 pts

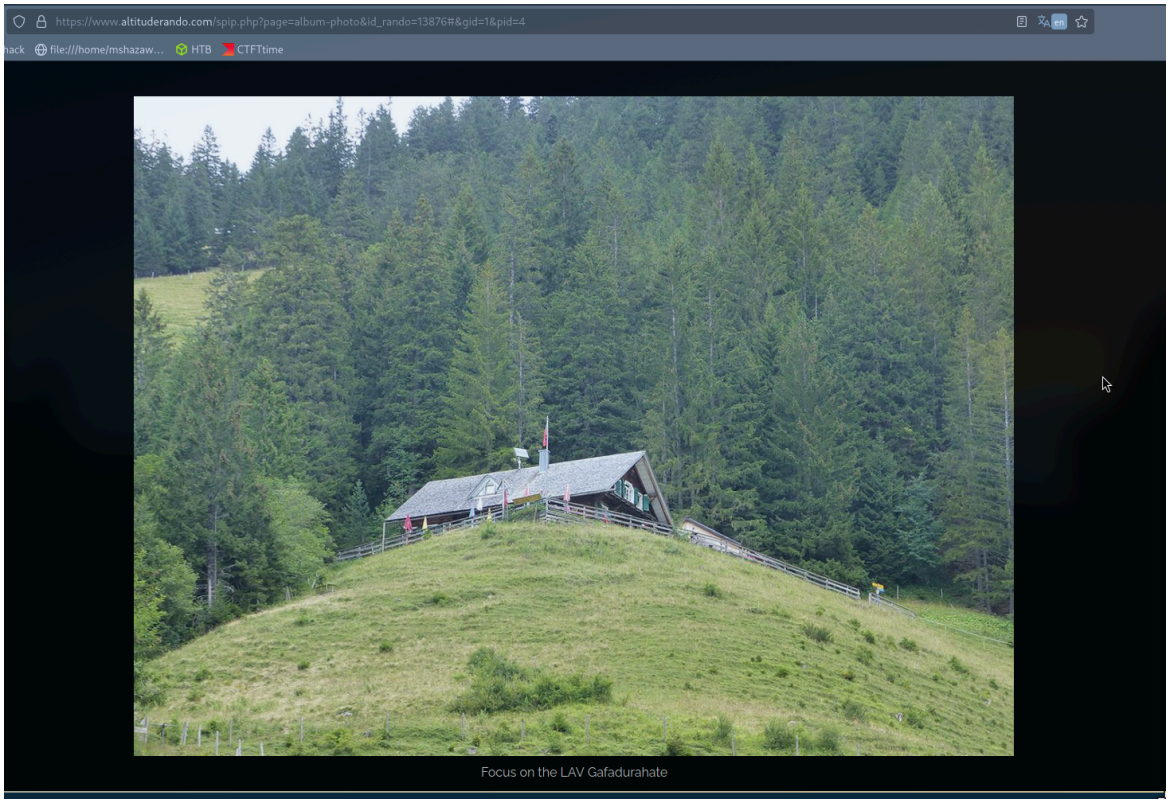
My friend Daniel S loved traveling. He used to send me pictures of his trips across Europe. Here's one from his favourite trail. I want to visit the place in his memory, help me find it.
flag_format: `apoorvctf{<coordinates>}`
coordinates should be truncated to the 4th decimal position

Example:
Eiffel Tower
Coordinates: 48.8583951,2.2949823
Flag: apoorvctf{48.8583,2.2949}

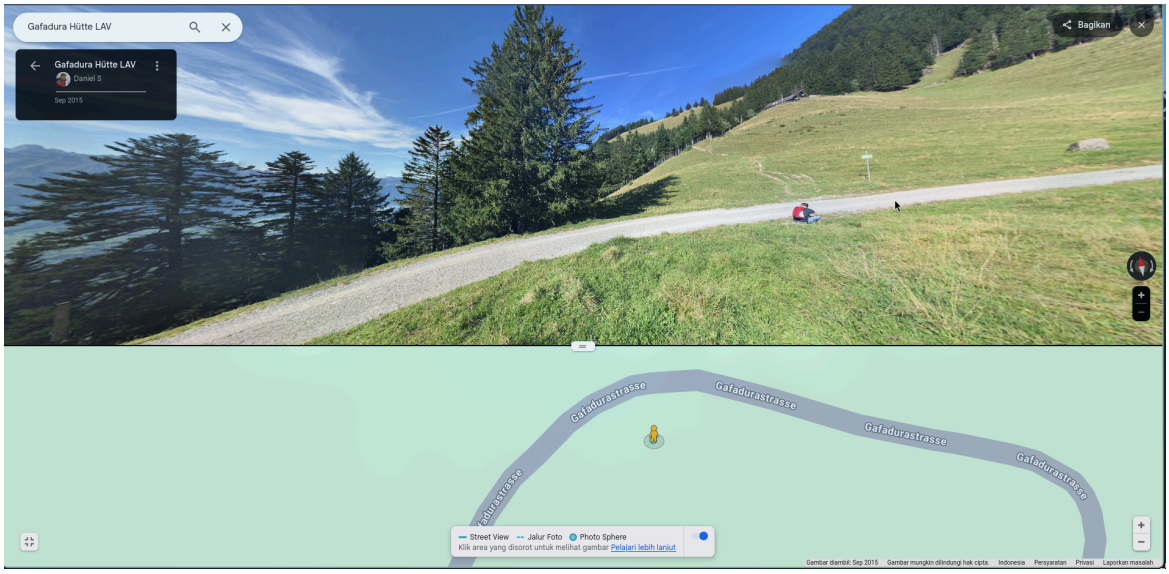
Author: *cooker*

So we were initially given this image and at first aside from it looking like it's in the Alps we have nothing to go on.

Then I noticed this hut in the corner of the image and tried reverse image searching it using google lens.



And that led me to this photo album which mentions the name LAV Gafadurahate and that was it.



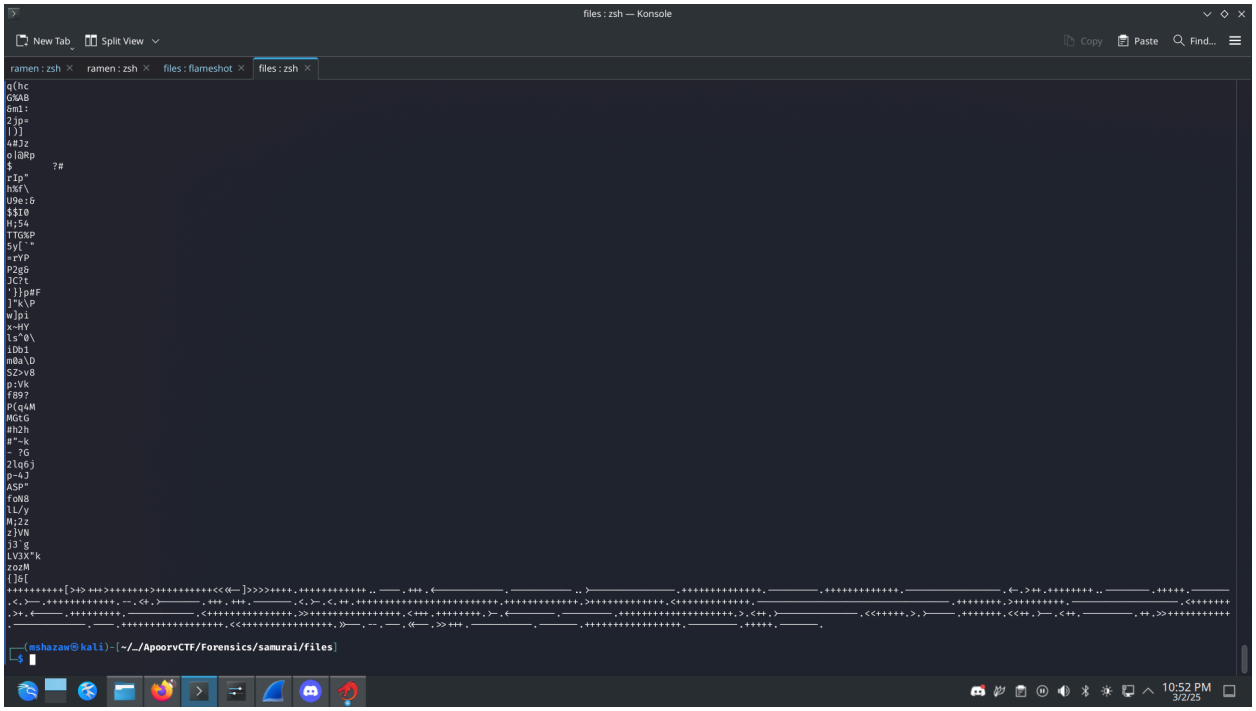
All I needed to do from then on was just search it up on google maps and try the placeable spots on google maps one by one.

Flag: **apoorvctf{47.1851,9.5663}**

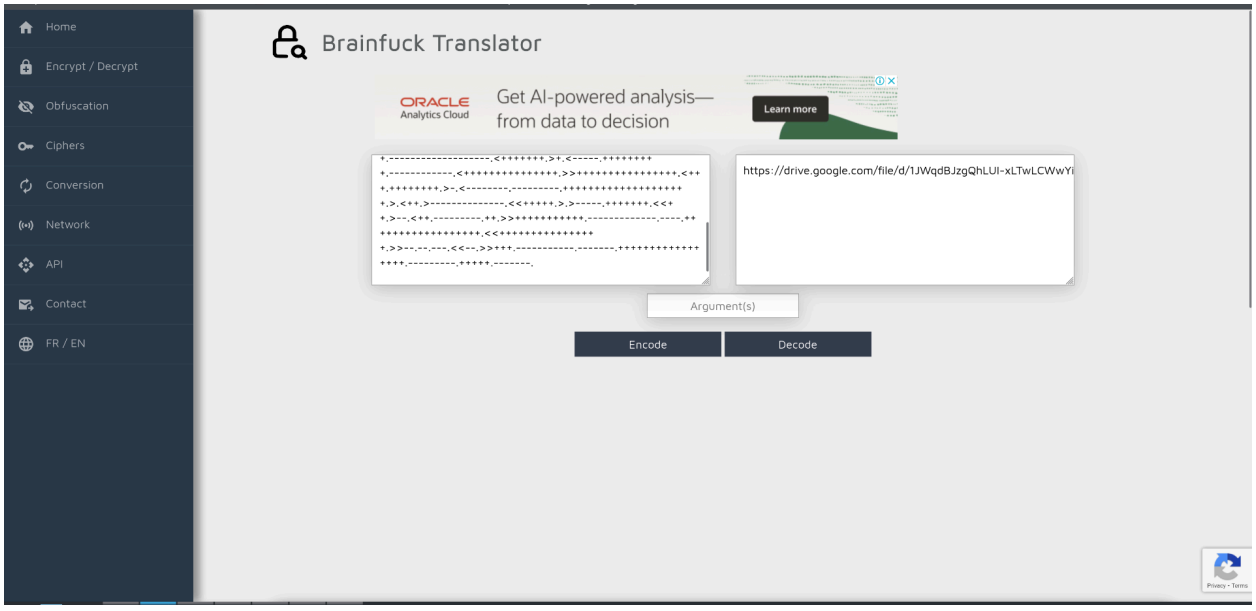
Forensics - Samurai Code (Unfinished)

Unveil the lost code of the Samurai and unlock the mystery hidden within.

Ok for this challenge, the first thing we do is just try and perform the command strings on the file sam.jpg and we receive brainfuck code on the bottom.



When we try and decode the code in a brainfuck translator we see that it leads to a google drive which gives us a file named samurai.



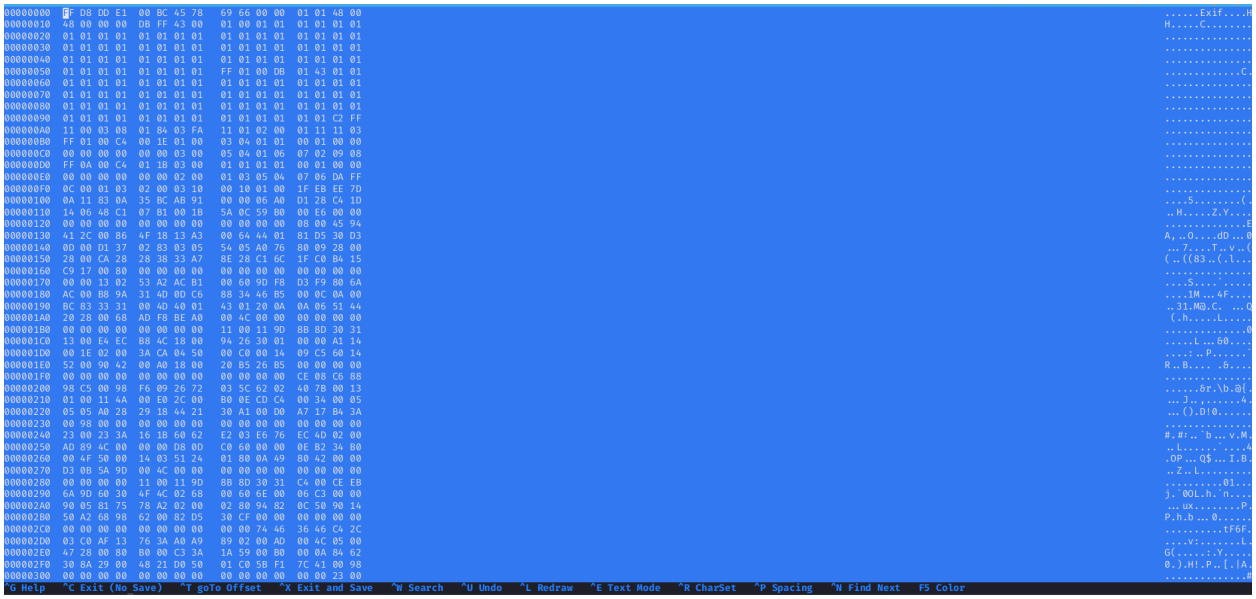
Upon performing basic analysis of this file, we see that it's just data and we don't even know anything about it other that the size.

```
┌──(mshazaw㉿kali)-[~/…/ApoorvCTF/Forensics/samurai/files]
└─$ file samurai
samurai: data

┌──(mshazaw㉿kali)-[~/…/ApoorvCTF/Forensics/samurai/files]
└─$ exiftool samurai
ExifTool Version Number         : 13.10
File Name                       : samurai
Directory                       : .
File Size                       : 217 kB
File Modification Date/Time     : 2025:03:02 21:26:06+07:00
File Access Date/Time           : 2025:03:02 21:26:07+07:00
File Inode Change Date/Time     : 2025:03:02 21:26:06+07:00
File Permissions                : -rw-rw-r--
Error                           : Unknown file type

┌──(mshazaw㉿kali)-[~/…/ApoorvCTF/Forensics/samurai/files]
└─$
```

The next thing I tried to check was the hexdata, I noticed that this looked like a badly made jpg file so I tried to go down that route.

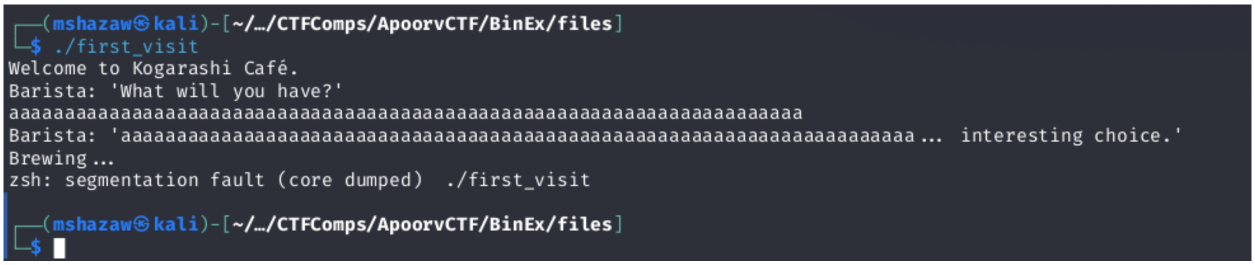


But after multiple tries, I still couldn't fix the file and that's where I threw in the towel.

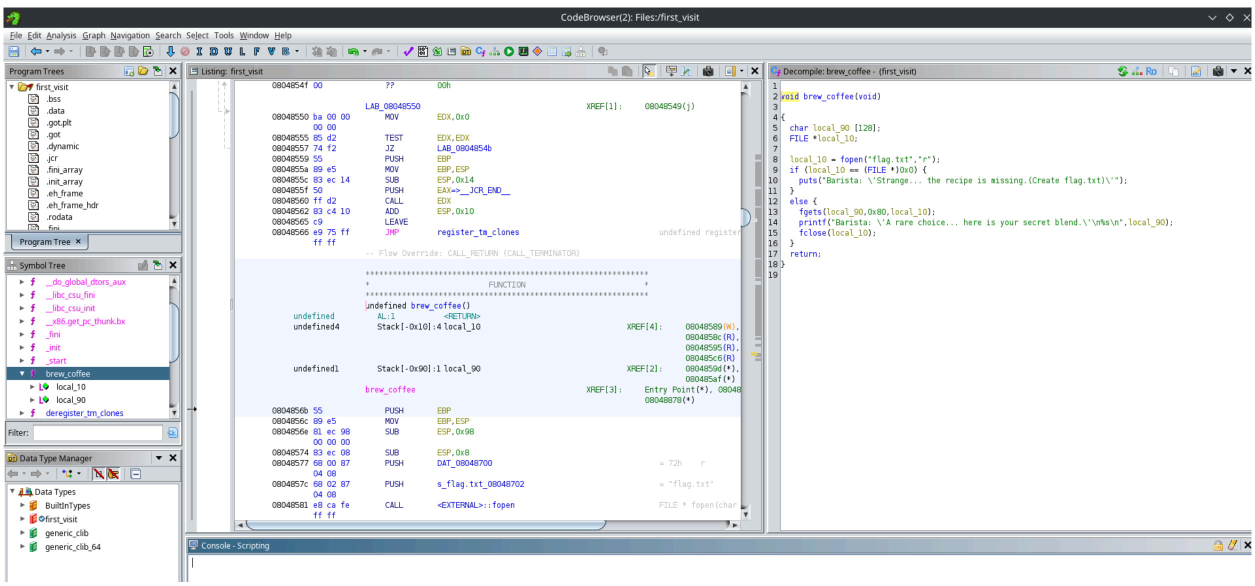
Binary Exploitation - “Kogarashi Cafe - The First Visit” (Unfinished)

A quiet café, a simple question. The barista waits, but your order may mean more than it seems.

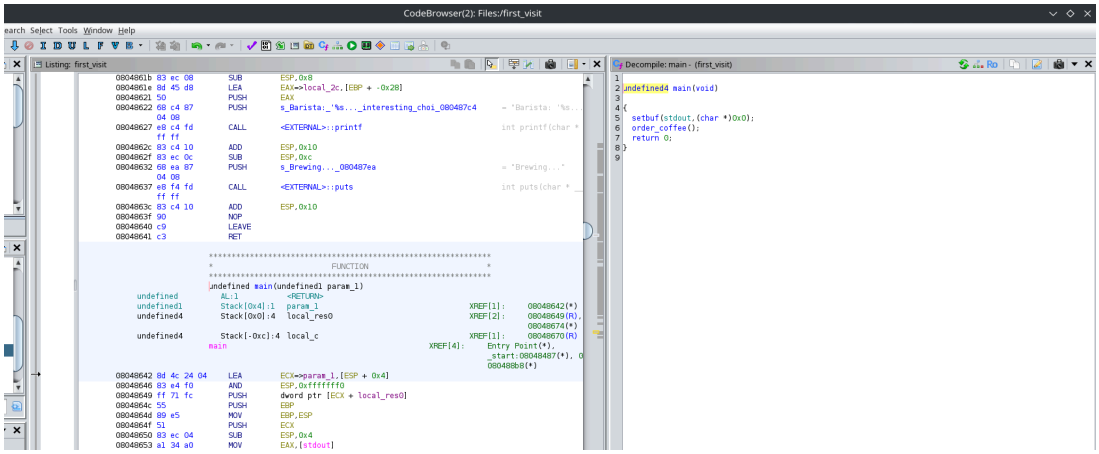
```
nc chals1.apoorvctf.xyz 3001
```



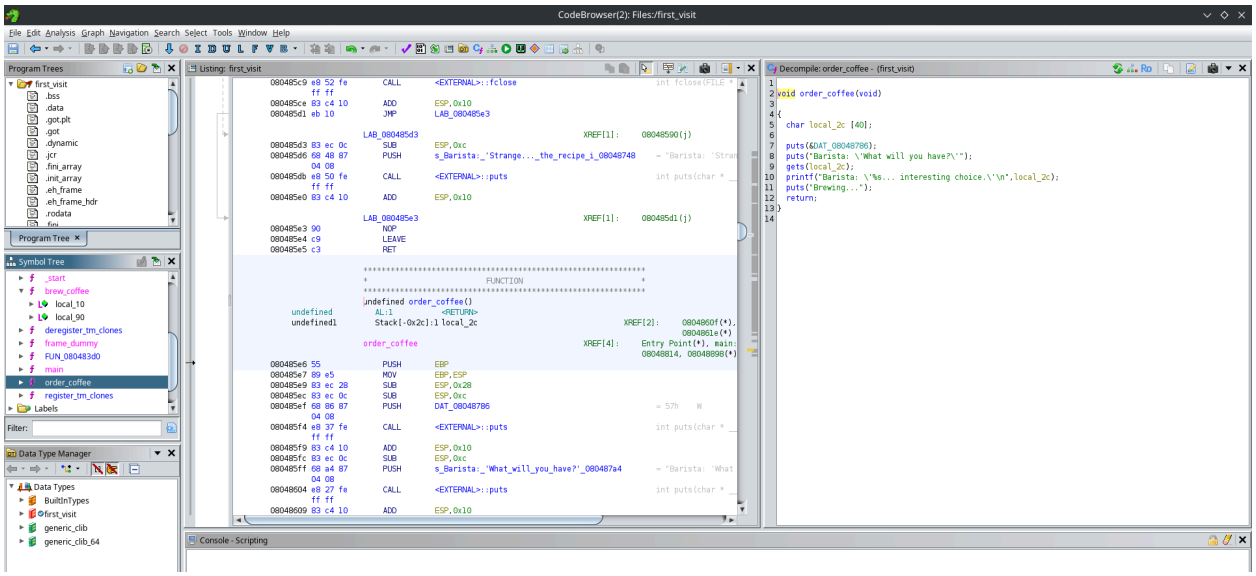
This one is supposed to be a pretty standard binary exploitation challenge, where you overload the buffer and get the flag.



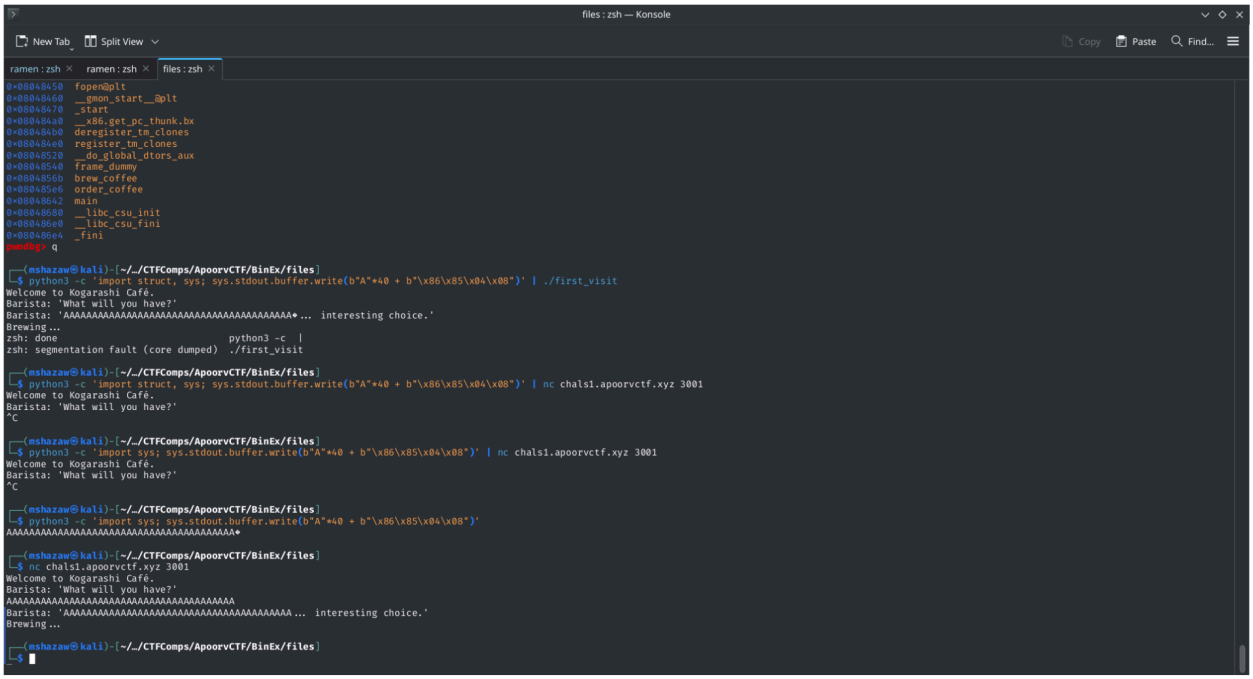
When looking at the code in ghidra, we can see the brew function which in this challenge is the win function. We now have to find a way to get here.



This one is the main function, and as we can see there's no path from it to the brew function and it only calls the order_coffee function.



And this is the order_coffee function, again with no calling of the brew function.



Because of the get() vulnerability we can write as many bytes as possible into the answer and they won't check it. So I tried to overflow the buffer and then afterwards redirect it to the brew_coffee function, but all my tries ended in failure, so I threw in the towel for this one.