Challenge: FlagCasino

# Challenge Description :

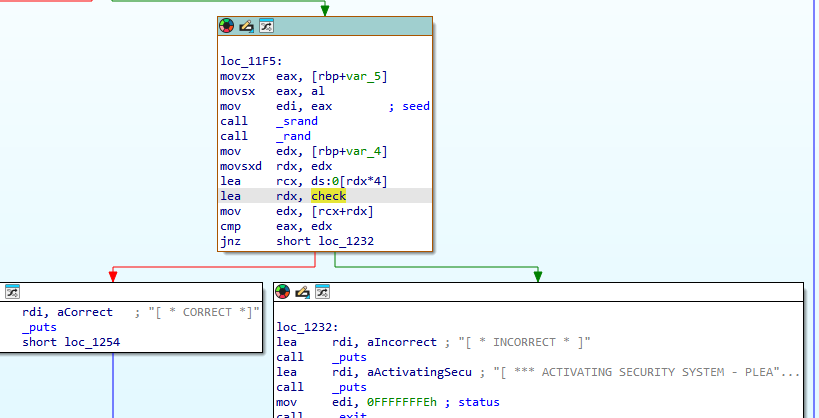
**The team stumbles into a long-abandoned casino. As you enter, the lights and music whir to life, and a staff of robots begin moving around and offering games, while skeletons of prewar patrons are slumped at slot machines. A robotic dealer waves you over and promises great wealth if you can win - can you beat the house and gather funds for the mission?**

# Context :

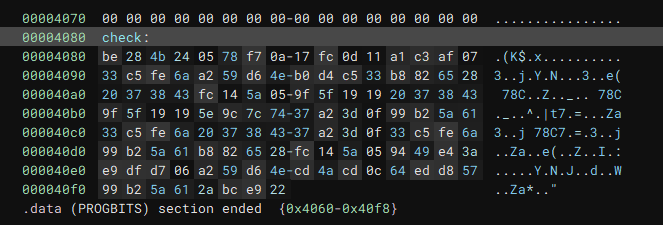
## You are given a compiled binary file, when looking through the file we can find a selection of 120 bytes of data that we need to format correctly, after formatting correctly we need to structure it and send the data to the IP:PORT that's hosting the binary file to gain the flag.

# Flag :

* **First I downloaded the file and opened it up with Hex-Ida, with much luck i didn't really find anything too interesting, i did find a bunch of data located in the variable [ Check ].**

****

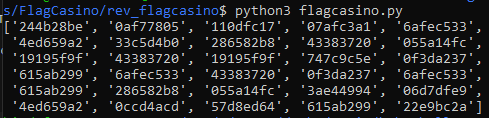
* **To get a better look and understanding of the file I opened it up with Binary ninja. With it I went to the main function and double checked the data in the [ Check ] variable .**
* **The data within the [ Check ] variable could be the data I need to input within the Binary file.**

****

* **Trying to find more information I went to [ Dogbolt.org ] to decompile it further. I finally found what I needed, I just needed a different decompiler to see it.**

****

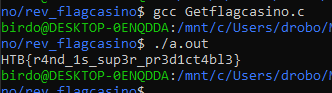
* **The Max amount of characters must be 30 because the loop variable i stops at 30. Given that there are 120 bytes of data, this means that each check corresponds to 4 bytes. To process the data effectively, we can divide the string into segments of 4 bytes each.**

****

* **Instead of converting it manually, let's create a simple python script to achieve this, which will also convert the bytes into a little-endian format.**

**[**[**Link**](https://github.com/Birdo1221/HTB-writeup/blob/main/Flagcasino/flagcasino.py)**]**

* **It is not read-able but it works. We will need to test every number / letter between 33 and 126, to brute-force the integers to be converted to ASCII characters, which will hopefully give us a better result.**

****

* **Below is the link for the script to gain the flag by converting the little-endian format to ascii characters : HTB{r4nd\_1s\_sup3r\_pr3d1ct4bl3}**

**[**[**Link**](https://github.com/Birdo1221/HTB-writeup/blob/main/Flagcasino/Getflagcasino.c)**]**