DESCRIPTION

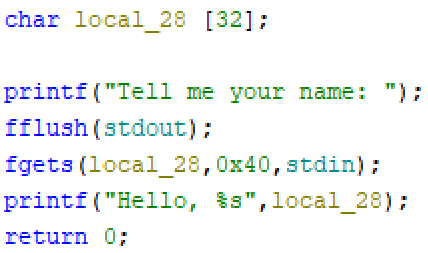
It's simple: get the flag from 141.85.224.106:31339.

RESOURCES

As part of the challenge I received an executable file called **piece_of_pie** as an attachment for analysis.

APPROACHES

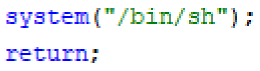
1. The first approach here was to run the program and after running it I could see that it is requiring a input string which after it is provided it is returned back;
2. Thus, I had no other idea than to decompile the program using Ghidra. The decompilation can be seen below:



```
char local_28 [32];

printf("Tell me your name: ");
fflush(stdout);
fgets(local_28,0x40,stdin);
printf("Hello, %s",local_28);
return 0;
```

3. Here we can directly see that we have a buffer overflow as we have a buffer of only 0x20 characters and we are reading 0x40 from the keyboard.
4. Additionally there was present in the binary another function called **make_it_easy**, which if we call we are going to get a shell:



```
system("/bin/sh");
return;
```

5. Thus my idea here was to direcly make a script that will overflow the input buffer and it will override the return address of the main function to **make_it_easy** function.
6. Then, if you run the script (**python3 script.py**) we will obtain a shell on the remote server.
7. If we then run **cat /home/ctf/flag** on the opened shell we will get the flag which is:

CNS_CTF{653a93fbd3d8574eea7acd4e23918989}