DESCRIPTION

Link me to print out the flag

RESOURCES

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

APPROACHES

1. My first approach was to directly run the executable file and got the following error:

```
mihnea@HOME-PC:/mnt/c/Users/mblot/Desktop/CNS$ ./link_me
./link_me: error while loading shared libraries: libmumu.so: cannot open shared object file: No such file or directory
```

2. The error made me think that a **libmumu.so** file is needed to be able to run the executable file. Thus, I created an empty **libmumu.c** file and generated a **libmumu shared object** from it. Additionally, I exported **LD_LIBRARY_PATH** to also search for libraries in the current directory. These actions were done using the following two commands:

```
mihnea@HOME-PC:/mnt/c/Users/mblot/Desktop/CNS$ gcc -fPIC -shared -o libmumu.so libmumu.c
mihnea@HOME-PC:/mnt/c/Users/mblot/Desktop/CNS$ export LD_LIBRARY_PATH=.
```

3. Then, tried to run again the **link_me** executable and end up with the following error:

```
mihnea@HOME-PC:/mnt/c/Users/mblot/Desktop/CNS$ ./link_me
./link_me: symbol lookup error: ./link_me: undefined symbol: string_xor_with_key
```

4. The error means that there should be a function called **string_xor_with_key** defined in the library so I searched inside **link_me** to see how it is called in order to find out how many parameters it receives:

```
undefined8 main(void)

{
  undefined4 local_28;
  undefined4 local_24;
  undefined4 local_20;
  undefined4 local_1c;
  undefined4 local_18;
  undefined4 local_14;
  uint local_c;

  local_28 = 0x138;
  local_24 = 0x1c5;
  local_20 = 0x240;
  local_1c = 0x7b;
  local_18 = 0x1bc;
  local_14 = 0x6d;
  string_xor_with_key(&DAT_00601090,6,0x22);
  local_c = array_sum(&local_28,6);
  sprintf(&DAT_00601096,"%d",(ulong)local_c);
  print_flag(&DAT_00601090,10);
  return 0;
}
```

5. From Ghidra I understood that the function receives **three parameters** and correlating it with the name of the function, it means that the first parameter is a string, the second one is the length of the string and the last one is the key with which we have to make **XOR** of every character in the string. Thus, the implementation follows:

```
void string_xor_with_key(char *str, int len, int key) {
    for (int i = 0; i < len; i++) {
        str[i] ^= key;
    }
}
```

6. Then, I compiled the **shared object** again and ran **link_me** to get the following error:

```
mihnea@HOME-PC:/mnt/c/Users/mblot/Desktop/CNS$ ./link_me
./link_me: symbol lookup error: ./link_me: undefined symbol: array_sum
```

7. From this error and from picture at point 4 above, it means that the library should contain another function called **array_sum** which receives two parameters and returns a value. Correlating with the name it means that it is a function that receives an array and the length of the array and computes the sum of that array which is then returned. Thus, the implementation follows:

```
int array_sum(int *arr, int len) {
    int sum = 0;
    for (int i = 0; i < len; i++) {
        sum += arr[i];
    }
    return sum;
}
```

8. Then, I compiled the **shared object** again and ran **link_me** to obtain the flag as:

CNS_CTF{Come_on_Yolanda_whats_Fonzie_like?}

9. I attached in the folder the **libmumu.c** which has to be compiled with the command (**gcc -fPIC -shared -o libmumu.so libmumu.c**), then use (export **LD_LIBRARY_PATH=.)** then **link_me** should be executed (**./link_me**) in order to reproduce the experiment and obtain the flag.