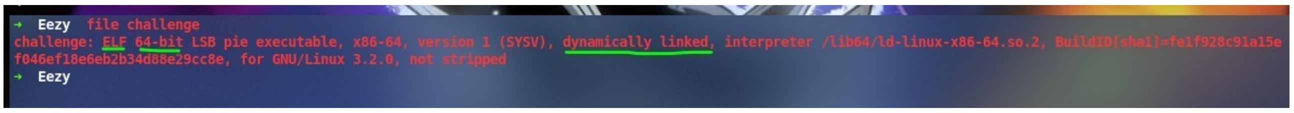
EEZY

DESCRIPTION : Don’t try to debug me. :(You can’t bypass me. Lets see if you can..
AUTHOUR : 0xRakesh Kumar

There are only one file : challenge

Do some static analysis:

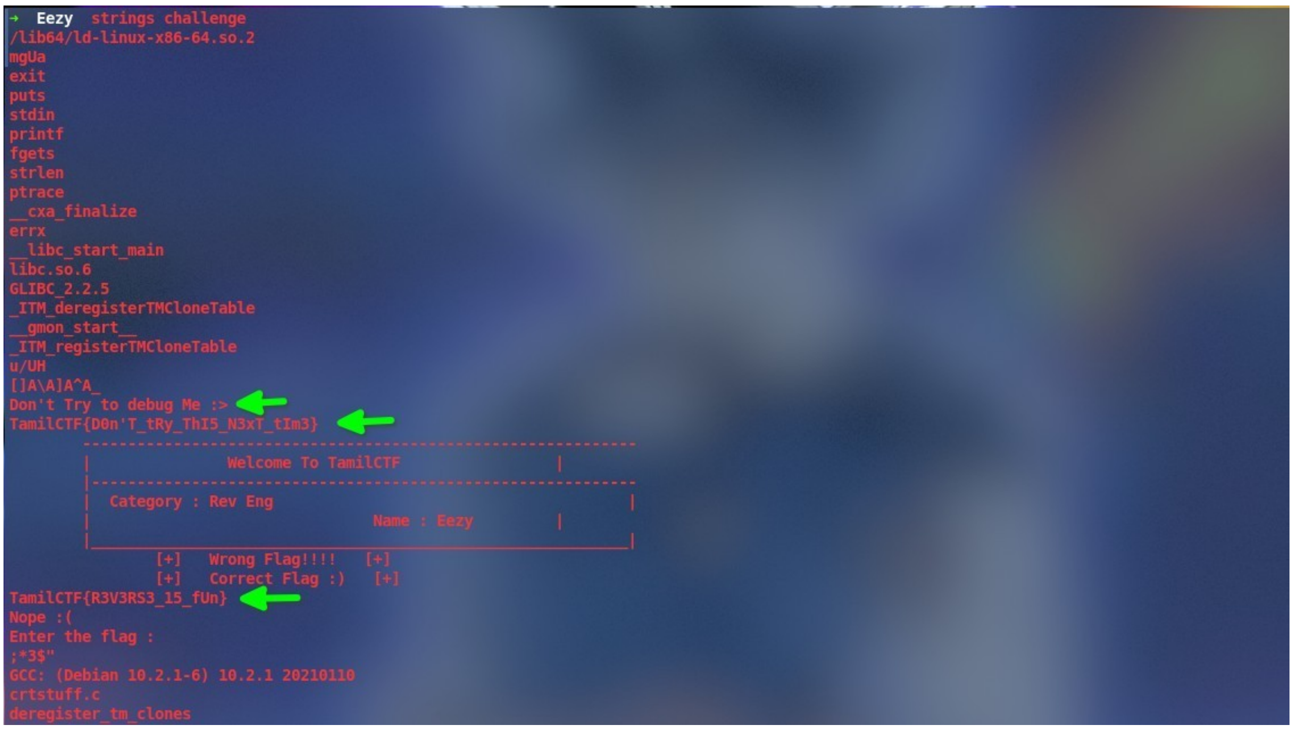
File command :



```
→ Eezy  file challenge
challenge: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=fe1f928c91a15e
f046ef18e6eb2b34d88e29cc8e, for GNU/Linux 3.2.0, not stripped
→ Eezy
```

It is ELF 64 bit binary and dynamically linked .

Strings command:



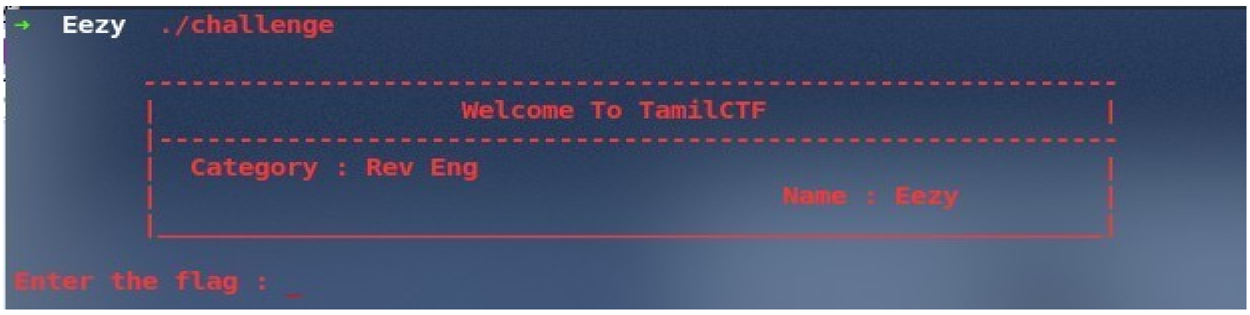
```
→ Eezy  strings challenge
/lib64/ld-linux-x86-64.so.2
mgUa
exit
puts
stdin
printf
fgets
strlen
ptrace
__cxa_finalize
errx
__libc_start_main
libc.so.6
GLIBC_2.2.5
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
u/UH
[]A\A]A^A_
Don't Try to debug Me :>
TamilCTF{D0n'T_tRy_ThI5_N3xT_tIm3}
        ------------------------------------------------------------
        |                Welcome To TamilCTF                      |
        |-----------------------------------------------------------
        |   Category : Rev Eng                                          |
        |                              Name : Eezy                |
        |___________________________________________________________|
                [+]   Wrong Flag!!!!   [+]
                [+]   Correct Flag :)   [+]
TamilCTF{R3V3RS3_15_fUn}
Nope :(
Enter the flag :
;*3$"
GCC: (Debian 10.2.1-6) 10.2.1 20210110
crtstuff.c
deregister_tm_clones
```

There are some **interest strings**:

```
TamilCTF{D0n'T_tRy_ThI5_N3xT_tIm3}
TamilCTF{R3V3RS3_15_fUn}
Don't Try to debug Me :>
```

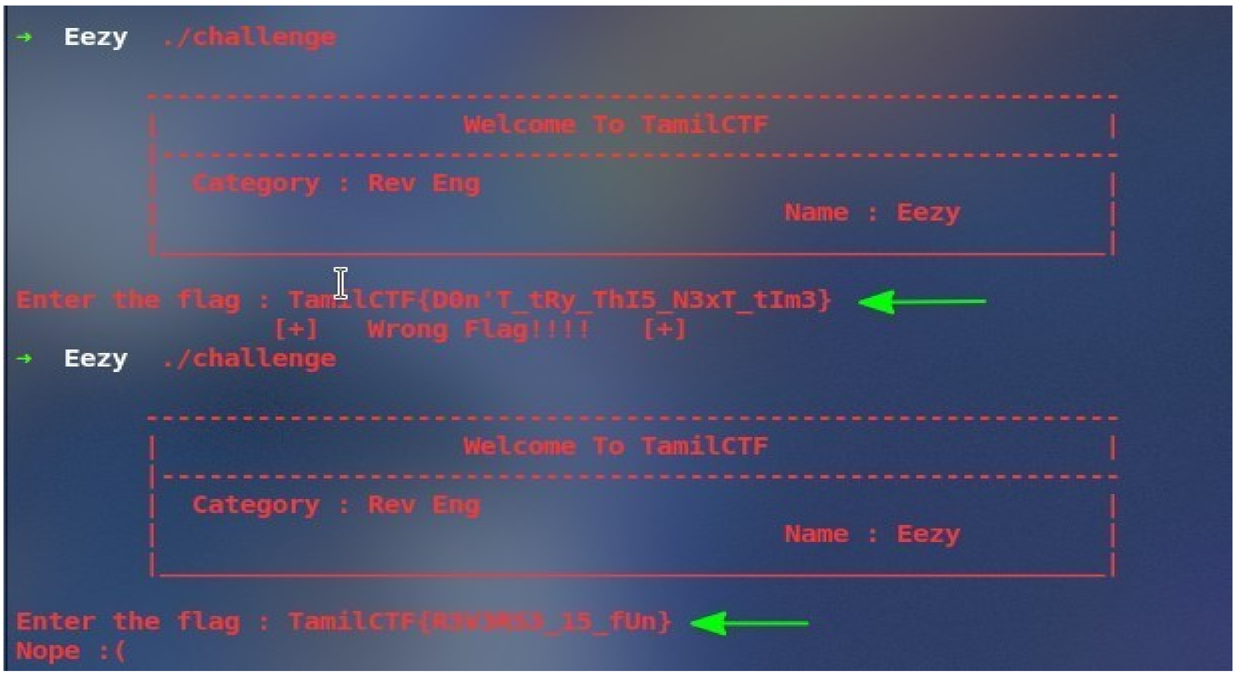
Run the binary:

```
➜ Eezy ./challenge
```



```
➜  Eezy  ./challenge

          ---------------------------------------------------------------
          |                  Welcome To TamilCTF                        |
          |--------------------------------------------------------------
          |  Category : Rev Eng                                         |
          |                                      Name : Eezy            |
          |_____________________________________________________________|

Enter the flag : _
```

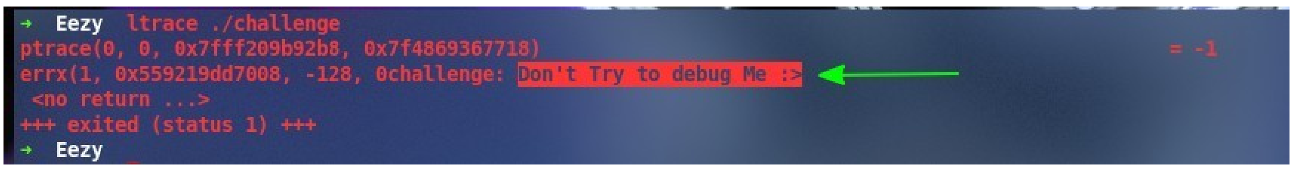
It asking the flag, so we can give this strings as input (TamilCTF{D0n'T_tRy_ThI5_N3xT_tIm3} , TamilCTF{R3V3RS3_15_fUn}). But it was fake flag. :(



```
➜  Eezy  ./challenge

          ---------------------------------------------------------------
          |                  Welcome To TamilCTF                        |
          |--------------------------------------------------------------
          |  Category : Rev Eng                                         |
          |                                      Name : Eezy            |
          |_____________________________________________________________|

Enter the flag : TamilCTF{D0n'T_tRy_ThI5_N3xT_tIm3}
                 [+]   Wrong Flag!!!!   [+]
➜  Eezy  ./challenge

          ---------------------------------------------------------------
          |                  Welcome To TamilCTF                        |
          |--------------------------------------------------------------
          |  Category : Rev Eng                                         |
          |                                      Name : Eezy            |
          |_____________________________________________________________|

Enter the flag : TamilCTF{R3V3RS3_15_fUn}
Nope :(
```

Ltrace command:

```
➜ Eezy  ltrace ./challenge
```



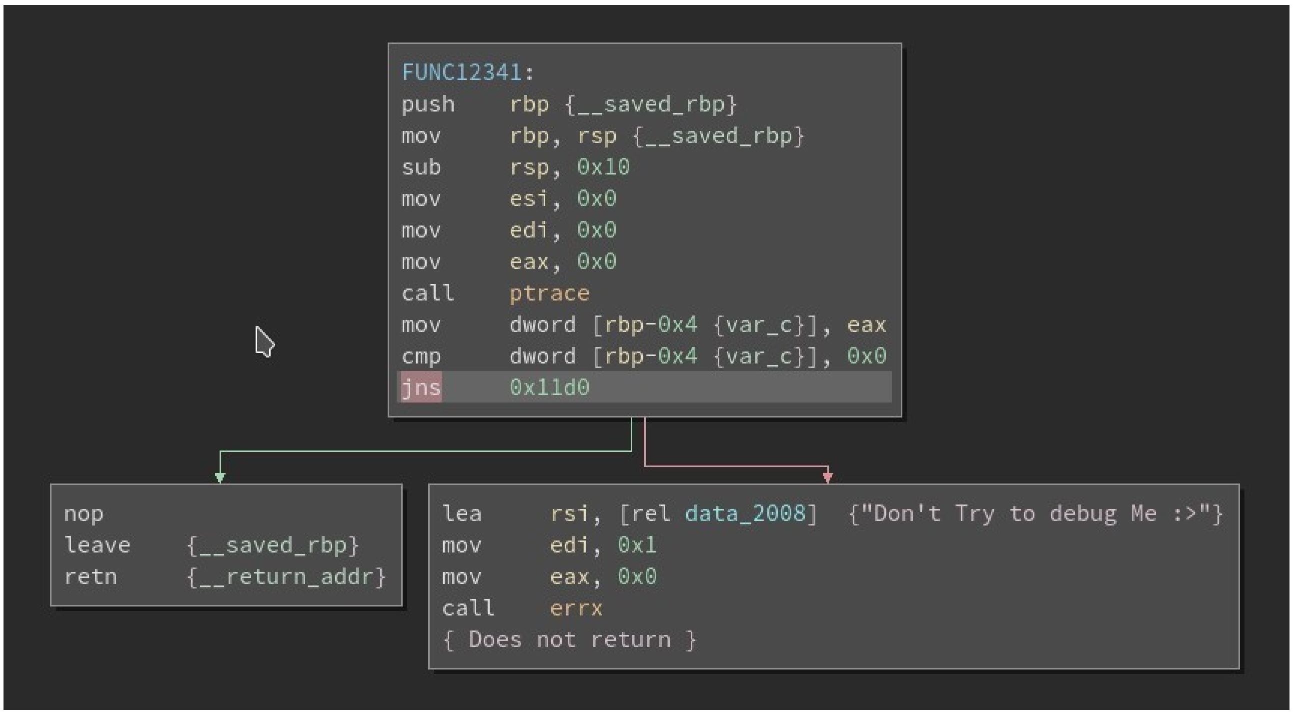
```
➜ Eezy  ltrace ./challenge
ptrace(0, 0, 0x7fff209b92b8, 0x7f4869367718)                                  = -1
errx(1, 0x559219dd7008, -128, 0challenge: Don't Try to debug Me :>
 <no return ...>
+++ exited (status 1) +++
➜ Eezy
```

They use some Anti-Reversing Techniques.Some we need patch the binary & debug the patch binary or analysis the binary statically.

[+]   Addition Information   [+]

Patch the Binary :

Open the binary in binaryninja. Look at the main function ,it call the function named FUNC12341 . The FUNC12341 function actually check the current program is debug or not.
If the current program is debugging ,then immediately the current program exit.

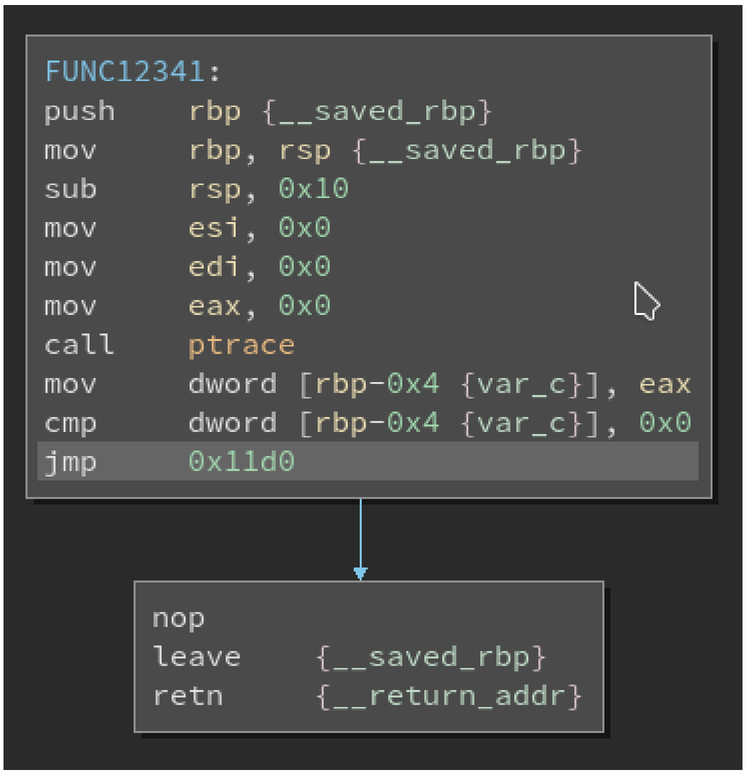


```
FUNC12341:
push    rbp {__saved_rbp}
mov     rbp, rsp {__saved_rbp}
sub     rsp, 0x10
mov     esi, 0x0
mov     edi, 0x0
mov     eax, 0x0
call    ptrace
mov     dword [rbp-0x4 {var_c}], eax
cmp     dword [rbp-0x4 {var_c}], 0x0
jns     0x11d0
```

```
nop
leave     {__saved_rbp}
retn      {__return_addr}
```

```
lea     rsi, [rel data_2008]  {"Don't Try to debug Me :>"}
mov     edi, 0x1
mov     eax, 0x0
call    errx
{ Does not return }
```

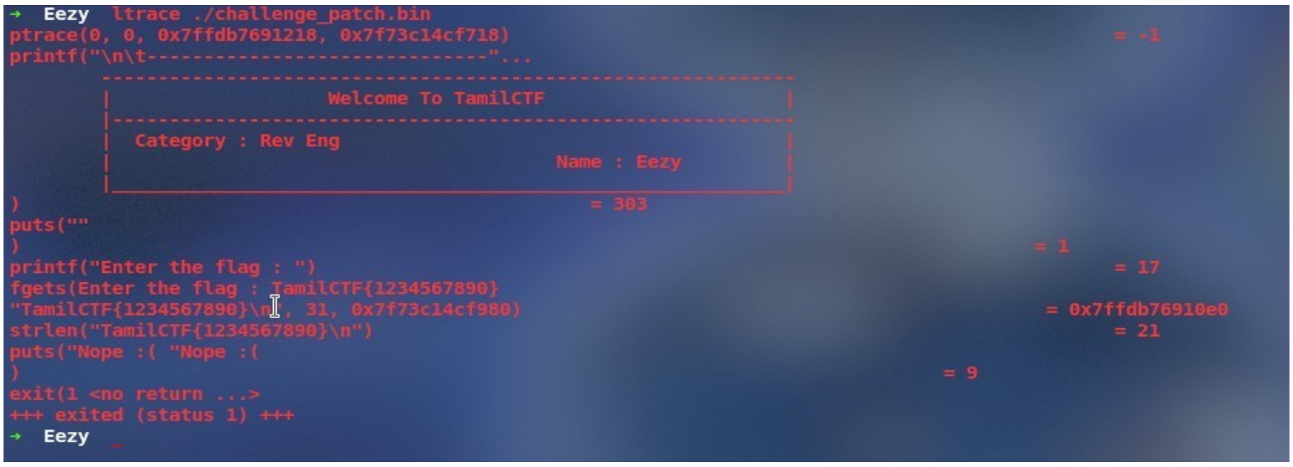
Now we need the change the branch condition. So right click the jns instruction ,select patch option and select always branch.
(**Right Click --> Patch ---> Always**)



```
FUNC12341:
push    rbp {__saved_rbp}
mov     rbp, rsp {__saved_rbp}
sub     rsp, 0x10
mov     esi, 0x0
mov     edi, 0x0
mov     eax, 0x0
call    ptrace
mov     dword [rbp-0x4 {var_c}], eax
cmp     dword [rbp-0x4 {var_c}], 0x0
jmp     0x11d0
```

```
nop
leave     {__saved_rbp}
retn      {__return_addr}
```

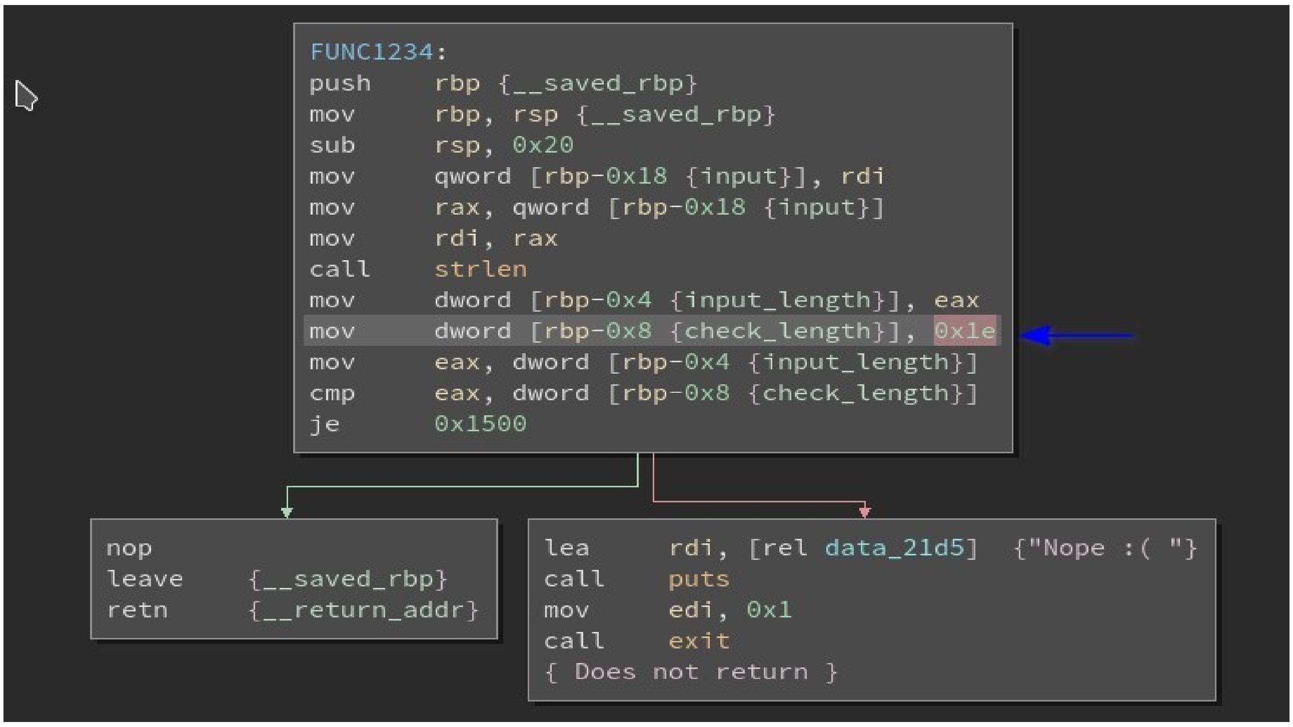
```
Now Try the ltrace command :
      ➜  Eezy  ltrace ./challenge
```



```
➜  Eezy  ltrace ./challenge_patch.bin
ptrace(0, 0, 0x7ffdb7691218, 0x7f73c14cf718)                                        = -1
printf("\n\t-----------------------------"...
        ------------------------------------------------------------
        |                 Welcome To TamilCTF                        |
        |------------------------------------------------------------
        |  Category : Rev Eng                                        |
        |                                    Name : Eezy             |
        |____________________________________________________________|
)                                                 = 303
puts(""
)                                                                              = 1
printf("Enter the flag : ")                                                        = 17
fgets(Enter the flag : TamilCTF{1234567890}
"TamilCTF{1234567890}\n", 31, 0x7f73c14cf980)                               = 0x7ffdb76910e0
strlen("TamilCTF{1234567890}\n")                                                   = 21
puts("Nope :( "Nope :(
)                                                                   = 9
exit(1 <no return ...>
+++ exited (status 1) +++
➜  Eezy  _
```

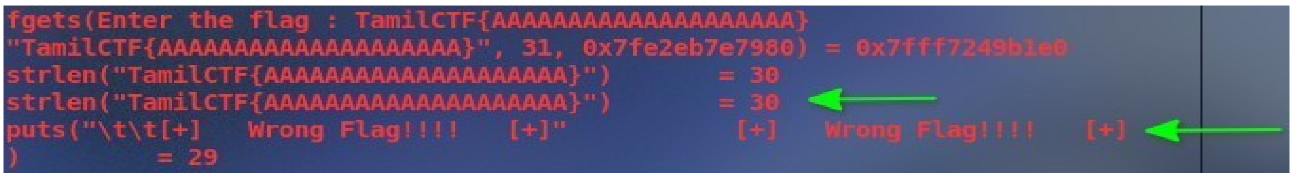
This is a wrong flag

After FUNC12341 function, it goes to FUNC1234 function,it calculate the length of strings and compare to 0x1e.If equal it continue ,otherwise it exit.



```
FUNC1234:
push    rbp {__saved_rbp}
mov     rbp, rsp {__saved_rbp}
sub     rsp, 0x20
mov     qword [rbp-0x18 {input}], rdi
mov     rax, qword [rbp-0x18 {input}]
mov     rdi, rax
call    strlen
mov     dword [rbp-0x4 {input_length}], eax
mov     dword [rbp-0x8 {check_length}], 0x1e
mov     eax, dword [rbp-0x4 {input_length}]
cmp     eax, dword [rbp-0x8 {check_length}]
je      0x1500

nop
leave   {__saved_rbp}
retn    {__return_addr}

lea     rdi, [rel data_21d5]  {"Nope :( "}
call    puts
mov     edi, 0x1
call    exit
{ Does not return }
```

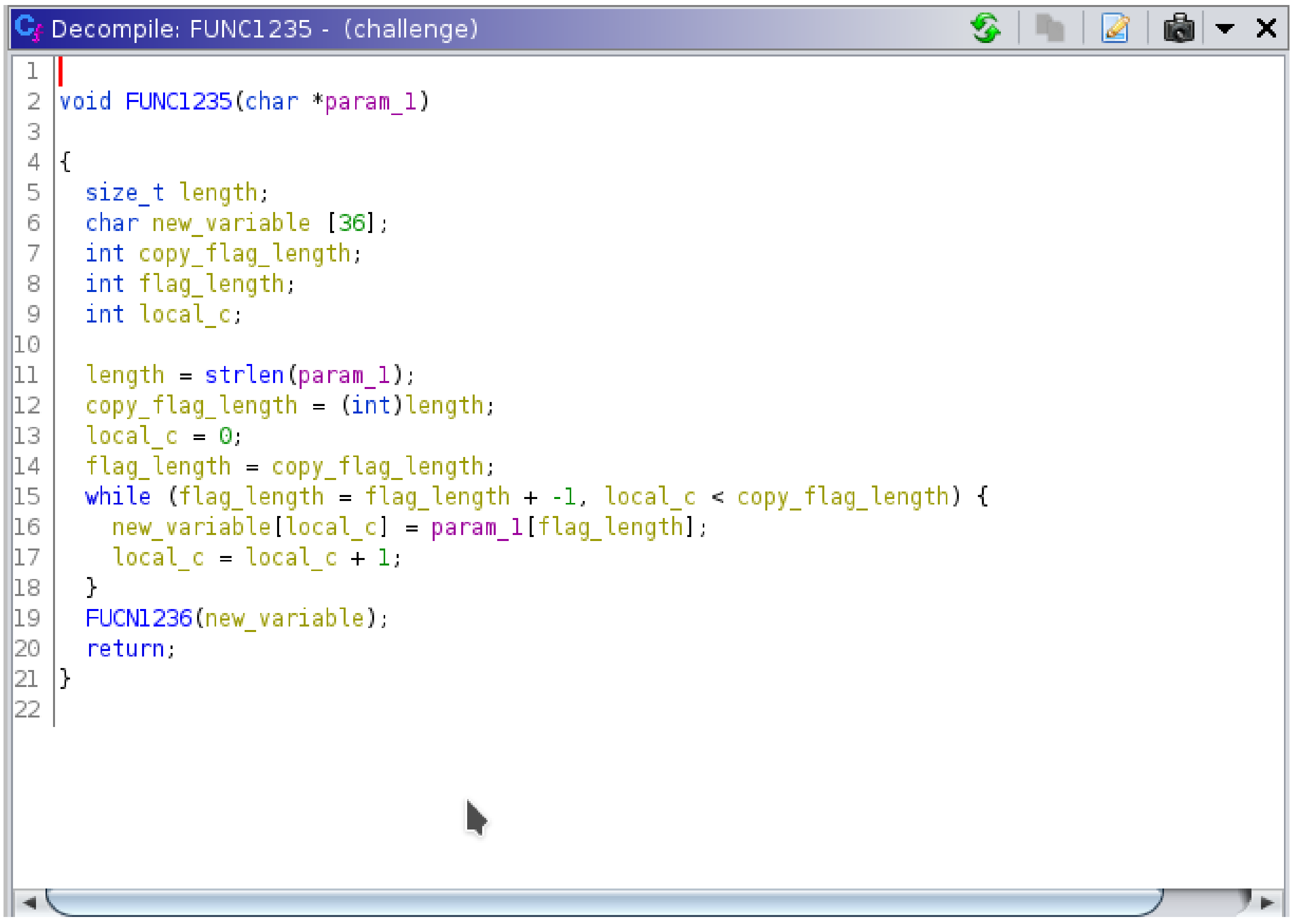
The decimal value of 0x1e is 30.So the length of the flag is 30.
So try some string with the length of 30.



```
fgets(Enter the flag : TamilCTF{AAAAAAAAAAAAAAAAAAAA}
"TamilCTF{AAAAAAAAAAAAAAAAAAAA}", 31, 0x7fe2eb7e7980) = 0x7fff7249b1e0
strlen("TamilCTF{AAAAAAAAAAAAAAAAAAAA}")            = 30
strlen("TamilCTF{AAAAAAAAAAAAAAAAAAAA}")            = 30
puts("\t\t[+]   Wrong Flag!!!!   [+]"               [+]   Wrong Flag!!!!   [+]
)         = 29
```

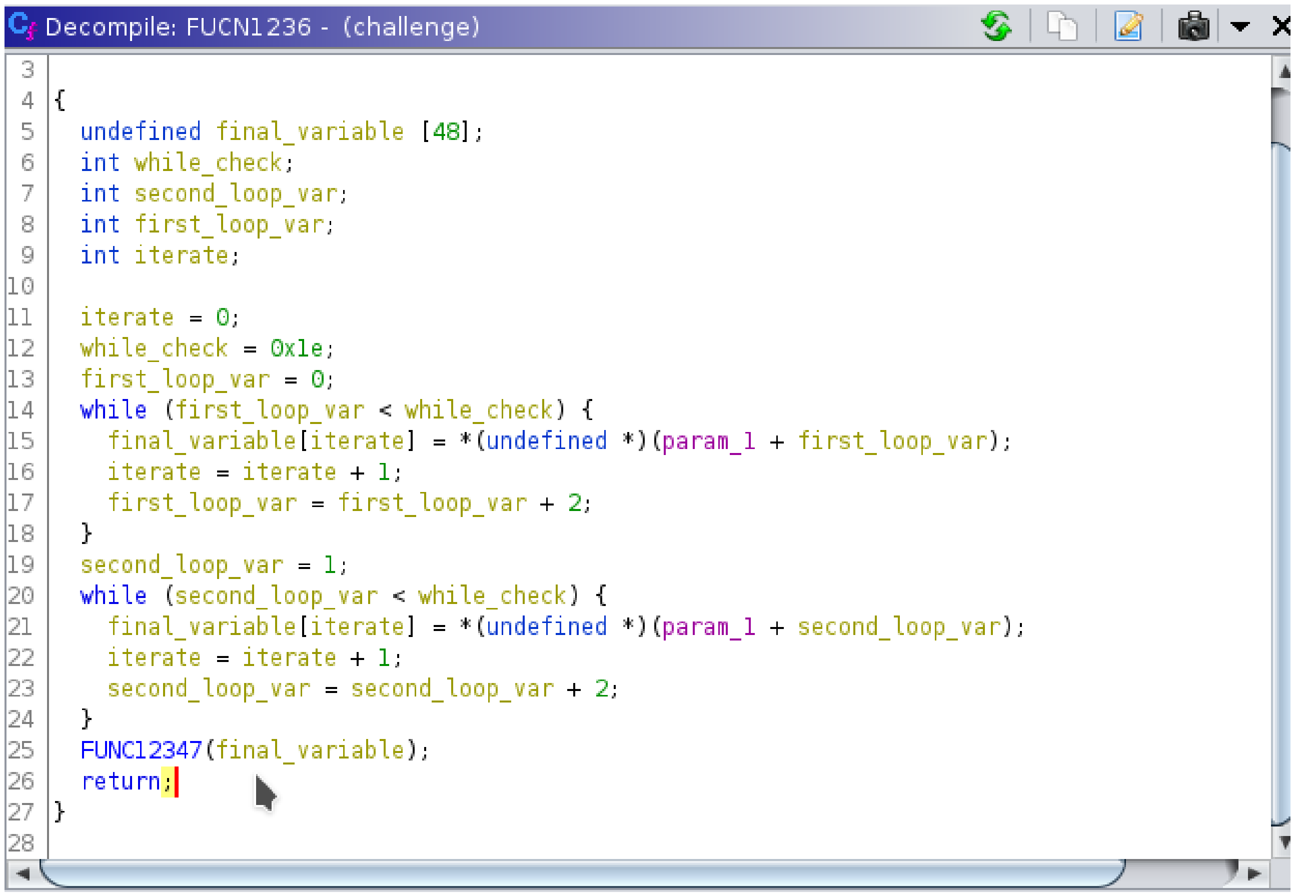
But this time the output is wrong flag.

After the FUNC1234 function ,it goes to FUNC1235 function.
Actually this do some stuff with our input. Analysis it in ghidra.



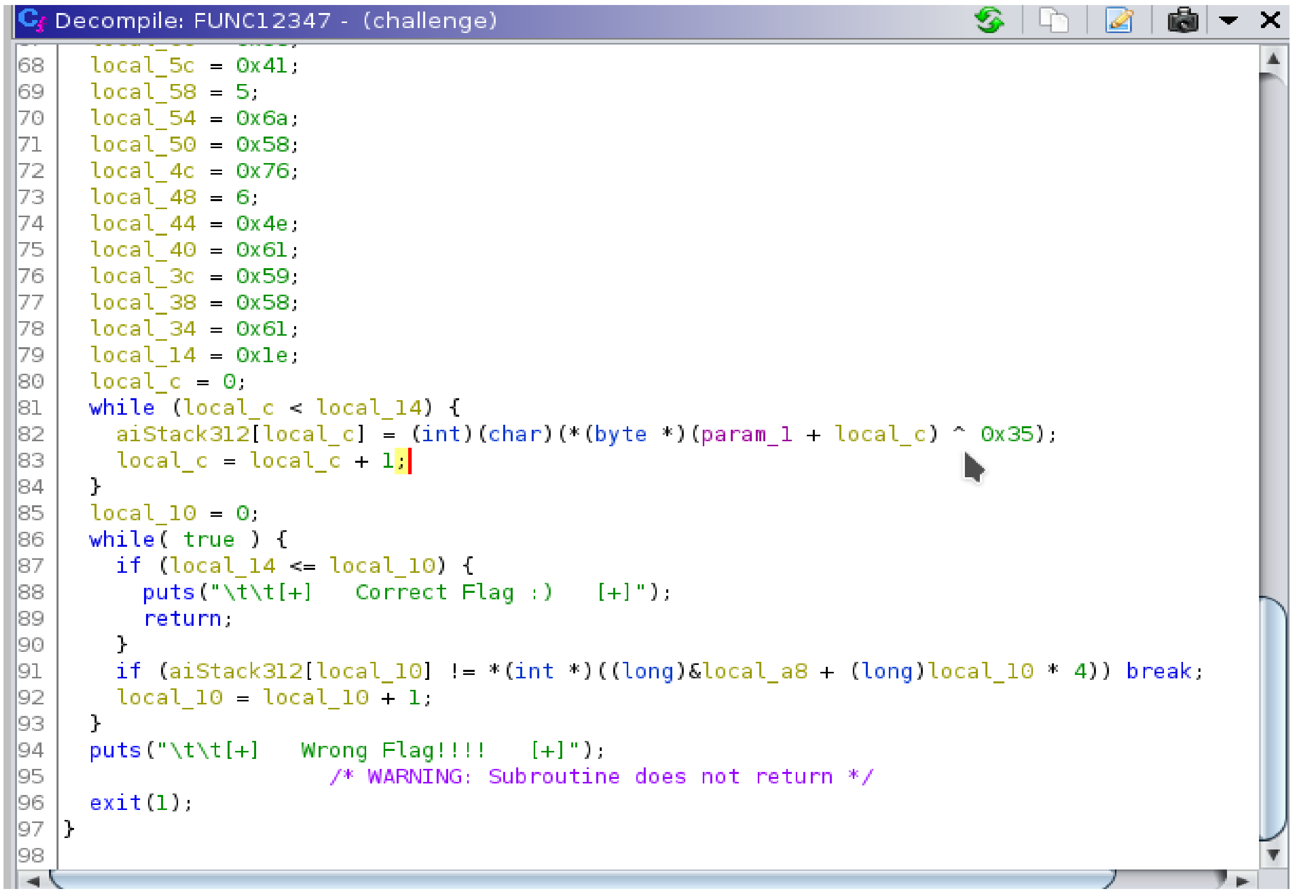
```c

void FUNC1235(char *param_1)

{
  size_t length;
  char new_variable [36];
  int copy_flag_length;
  int flag_length;
  int local_c;

  length = strlen(param_1);
  copy_flag_length = (int)length;
  local_c = 0;
  flag_length = copy_flag_length;
  while (flag_length = flag_length + -1, local_c < copy_flag_length) {
    new_variable[local_c] = param_1[flag_length];
    local_c = local_c + 1;
  }
  FUCN1236(new_variable);
  return;
}

```

Actually it inverse the strings and store in new_variable,then call the FUCN1236 function with argument of new_variable.



```c

{
  undefined final_variable [48];
  int while_check;
  int second_loop_var;
  int first_loop_var;
  int iterate;

  iterate = 0;
  while_check = 0x1e;
  first_loop_var = 0;
  while (first_loop_var < while_check) {
    final_variable[iterate] = *(undefined *)(param_1 + first_loop_var);
    iterate = iterate + 1;
    first_loop_var = first_loop_var + 2;
  }
  second_loop_var = 1;
  while (second_loop_var < while_check) {
    final_variable[iterate] = *(undefined *)(param_1 + second_loop_var);
    iterate = iterate + 1;
    second_loop_var = second_loop_var + 2;
  }
  FUNC12347(final_variable);
  return;
}

```

There are two loops, the first loop store even value of argument in final_varibale and the second loop store odd value of argument in final_variable.Finally the final_variable gives as argument for FUNC12347.The FUNC12347 function, xor the each value of argument with 0x35, and check with some values.If it equal then it print Correct Flag ,otherwise it print Wrong Flag.



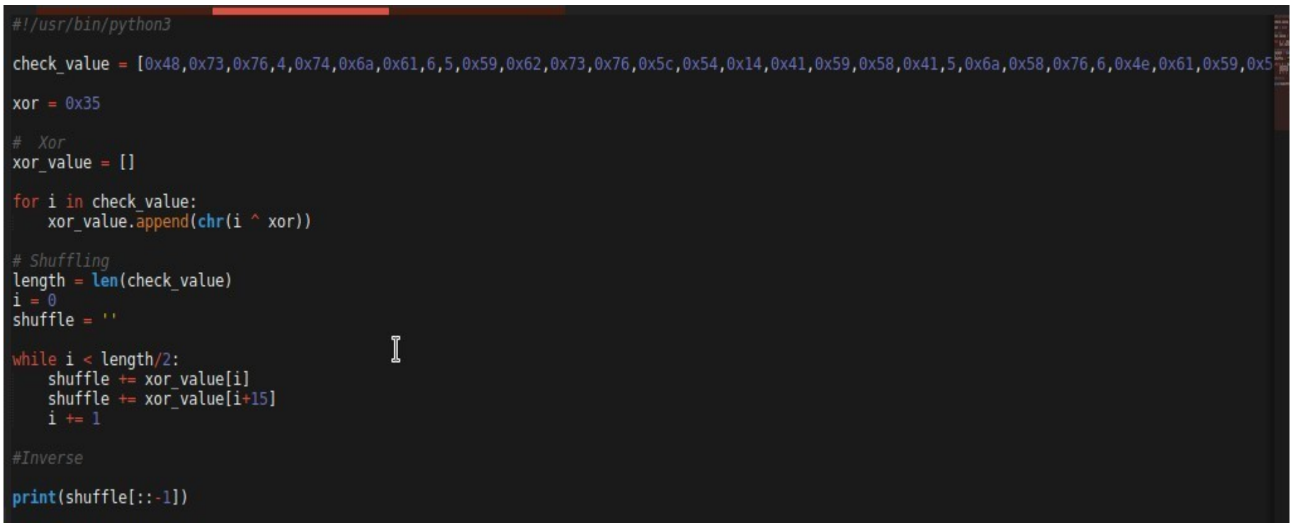
```
  local_5c = 0x41;
  local_58 = 5;
  local_54 = 0x6a;
  local_50 = 0x58;
  local_4c = 0x76;
  local_48 = 6;
  local_44 = 0x4e;
  local_40 = 0x61;
  local_3c = 0x59;
  local_38 = 0x58;
  local_34 = 0x61;
  local_14 = 0x1e;
  local_c = 0;
  while (local_c < local_14) {
    aiStack312[local_c] = (int)(char)(*(byte *)(param_1 + local_c) ^ 0x35);
    local_c = local_c + 1;
  }
  local_10 = 0;
  while( true ) {
    if (local_14 <= local_10) {
      puts("\t\t[+]    Correct Flag :)    [+]");
      return;
    }
    if (aiStack312[local_10] != *(int *)((long)&local_a8 + (long)local_10 * 4)) break;
    local_10 = local_10 + 1;
  }
  puts("\t\t[+]    Wrong Flag!!!!    [+]");
                    /* WARNING: Subroutine does not return */
  exit(1);
}
```

GOAL:

1. Xor the each value with 0x35.
2. Shuffle the xored value.
3. Inverse the shuffled value.

Python Script:

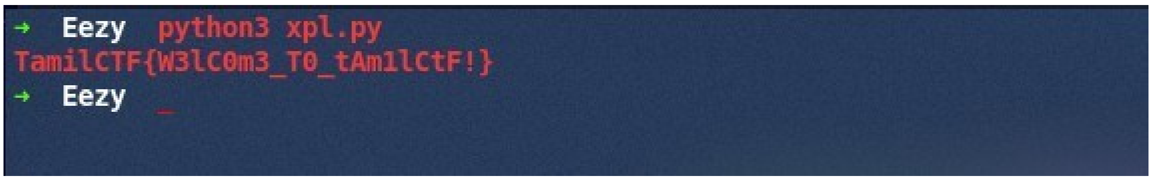
```
#!/usr/bin/python3
check_value =
[0x48,0x73,0x76,4,0x74,0x6a,0x61,6,5,0x59,0x62,0x73,0x76,0x5c,0x54,0x14,0x41,0x59,0x58,0x41,5,0x6a,0
x58,0x76,6,0x4e,0x61,0x59,0x58,0x61]
xor = 0x35
xor_value = []
for i in check_value:
    xor_value.append(chr(i ^ xor))
length = len(check_value)
i = 0
shuffle = ''
while i < length/2:
    shuffle += xor_value[i]
    shuffle += xor_value[i+15]
    i += 1
print(shuffle[::-1])
```



```python
#!/usr/bin/python3

check_value = [0x48,0x73,0x76,4,0x74,0x6a,0x61,6,5,0x59,0x62,0x73,0x76,0x5c,0x54,0x14,0x41,0x59,0x58,0x41,5,0x6a,0x58,0x76,6,0x4e,0x61,0x59,0x5

xor = 0x35

#  Xor
xor_value = []

for i in check_value:
    xor_value.append(chr(i ^ xor))

# Shuffling
length = len(check_value)
i = 0
shuffle = ''

while i < length/2:
    shuffle += xor_value[i]
    shuffle += xor_value[i+15]
    i += 1

#Inverse

print(shuffle[::-1])
```

Run the script:

```
➜  Eezy  python3 xpl.py
TamilCTF{W3lC0m3_T0_tAm1lCtF!}
```



```
➜  Eezy  python3 xpl.py
TamilCTF{W3lC0m3_T0_tAm1lCtF!}
➜  Eezy  _
```

Yeah,we finally find the flag :)