

Bite-code

Bite-code

I dunno what bytecode is.

Could you tell me what input of 'checkNum' will return true?

The flag is just a 32-bit signed integer as a decimal (nothing else.)

https://mega.nz/#!qfATFaKR!zaTNEqx3Bm1MjjnePjTGQyvnvLX_xZxhbGaMv_ypaxo

Flag: CTFlearn{-1352854872}

Writeup:

```
public static boolean checkNum(int);
  descriptor: (I)Z //Receive a boolean value after receiving an int parameter
  flags: ACC_PUBLIC, ACC_STATIC
  Code:
    stack=2, locals=3, args_size=1 //stack: the maximum operand stack, here the operation stack depth is 2;
    locals: the storage space required by local variables; args_size: the number of method parameters
    0: iload_0 loads int value from local variable 0
    1: iconst_3 pushes the int type constant 3 onto the stack, stack top=3
    2: ishl performs an int type left shift operation
    3: istore_1 stores the long type value in local variable 1
    4: iload_0 loads the int value from the local variable 0
    5: ldc #2 // int 525024598 pushes the items in the constant pool onto the stack
    7: ixor performs "logical XOR" operation on int type values
    8: istore_2 stores int type value in local variable 2
    9: iload_1 loads int value from local variable 1
    10: iload_2 loads int value from local variable 2
    11: ixor performs "logical XOR" operation on int type values
    12: ldc #3 // int -889275714 pushes the items in the constant pool onto the stack
    14: if_icmpne 21 If the two int types are not equal, jump
    17: iconst_1 push the int type constant 1 onto the stack
    18: goto 22
    21: iconst_0 pushes the int type constant 0 onto the stack
    22: ireturn returns int type data from the method
  LineNumberTable:
    line 3: 0
    line 4: 4
    line 5: 9
  StackMapTable: number_of_entries = 2
    frame_type = 253 /* append */
      offset_delta = 21
      locals = [int, int]
    frame_type = 64 /* same_locals_1_stack_item */
      stack = [int]
```

can be seen that this function moves the integer parameter x to the left by three bits and stores it in the first integer variable n1. Then store the value obtained by XORing x and the constant n2 into n2. Determine whether the value of n1 XOR or n2 is equal to the constant n3. If they are equal, return true, otherwise return false. This function written in C language will be this

```
bool checkNum(int x) {
    int n1 = x << 3;
    int n2 = 525024598;
    int n3 = -889275714;
    n2 = x ^ n2;
    if (n1 ^ n2 == n3) return true;
    else return false;
```

We can write a program to brute force the answer, after calculation, the correct value is -1352854872

files

