

## 逆向刷题

### 综合题库

#### test1.exe(xor)

```
1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     char v4[20]; // [esp+50h] [ebp-2Ch]
4     char Str[20]; // [esp+64h] [ebp-18h] BYREF
5     int i; // [esp+78h] [ebp-4h]
6
7     printf("Please give me your input:\n");
8     sub_401005((int)Str, 15);
9     if ( strlen(Str) == 10 )
10    {
11        for ( i = 0; i < 10; ++i )
12            v4[i] = byte_425A30[9 - i] ^ Str[i];
13        for ( i = 0; i < 10 && v4[i] == byte_425A3C[i]; ++i )
14            ;
15        if ( i == 10 )
16            printf("Congratulations! You are right!\n");
17        else
18            printf("2 Sorry, you are wrong!\n");
19        system("pause");
20        return 0;
21    }
22    else
23    {
24        printf("1 Sorry,you are wrong!\n");
25        system("pause");
26        return 0;
27    }
28 }
```

```
str1=[0x11, 0x02, 0x13, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x10]
str2=[0x57, 0x66, 0x67, 0x63, 0x59, 0x49, 0x71, 0x70, 0x69, 0x30]
str=[0]*11
for i in range(10):
    str[i]=str1[9-i]^str2[i]

str_output=''.join(chr(x) for x in str)
print("字符串:"+str_output)
```

Good\_Luck!

#### test2.exe(位运算)

```

1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     char Str[28]; // [esp+50h] [ebp-20h] BYREF
4     int i; // [esp+6Ch] [ebp-4h]
5
6     printf("Plase give me your answer:\n");
7     scanf("%s", Str);
8     if ( strlen(Str) == 22 )
9     {
10         for ( i = 0; i < 22; ++i )
11         {
12             printf("%c\n", Str[i]);
13             Str[i] = (4 * (Str[i] & 3)) | ((Str[i] & 0xC) >> 2) | Str[i] & 0xF0;
14             printf("Cipher[%d]=%c\n", i, Str[i]);
15         }
16         for ( i = 0; i < 22 && Str[i] == byte_429A30[i]; ++i )
17             ;
18         if ( i == 22 )
19             printf("Congratulations! You are right!\n");
20         else
21             printf("2 Sorry, you are wrong!\n");
22         system("pause");
23         return 0;
24     }
25     else
26     {
27         printf("1 Sorry,you are wrong!\n");
28         system("pause");
29         return 0;
30     }
31 }

```

输入22位

```
Str[i] = (4 * (Str[i] & 3)) | ((Str[i] & 0xC) >> 2) | Str[i] & 0xF0
```

1. 取低2位左移2位
2. 取低4位的高2位右移2位
3. 取高4位

逆向

```

str1 = [0x4E, 0x45, 0x45, 0x50, 0x5F, 0x56, 0x4F, 0x55, 0x58, 0x5F,
        0x41, 0x58, 0x45, 0x44, 0x47, 0x5F, 0x44, 0x43, 0x46, 0x59,
        0x45, 0x5F]
result = []
for x in str1:
    trans_x = (4 * (x & 0x3)) | ((x & 0xC) >> 2) | (x & 0xF0)
    result.append(chr(trans_x))
print("".join(result))

```

爆破法

```

str1 = [0x4E, 0x45, 0x45, 0x50, 0x5F, 0x56, 0x4F, 0x55, 0x58, 0x5F,
        0x41, 0x58, 0x45, 0x44, 0x47, 0x5F, 0x44, 0x43, 0x46, 0x59,
        0x45, 0x5F]
result = []
for target in str1:
    for x in range(256):
        trans_x = (4 * (x & 3)) | ((x & 0xC) >> 2) | (x & 0xF0)
        if trans_x == target:
            result.append(chr(x))
            break
print("".join(result))

```

## test3.exe(RC4改)

```
1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     signed int v4; // [esp+4Ch] [ebp-D0h]
4     signed int i; // [esp+50h] [ebp-CCh]
5     char v6[97]; // [esp+54h] [ebp-C8h] BYREF
6     __int16 v7; // [esp+B5h] [ebp-67h]
7     char v8; // [esp+B7h] [ebp-65h]
8     char Str[100]; // [esp+B8h] [ebp-64h] BYREF
9
10    memset(v6, 0, sizeof(v6));
11    v7 = 0;
12    v8 = 0;
13    puts("please input a correct string to encrypt:");
14    scanf("%s", Str);
15    v4 = strlen(Str);
16    sub_40100A(Str, (int)v6);
17    for ( i = 0; i < v4; ++i )
18    {
19        if ( v6[i] != byte_429B30[i] )
20        {
21            puts("Sorry you are wrong!");
22            system("pause");
23            return -1;
24        }
25    }
26    puts("WoW!!!Great!!!You are a genius!!!");
27    system("pause");
28    return 0;
29 }
```

sub\_40100A

```
1 int __cdecl sub_401410(char *Str, int a2)
2 {
3     char v2; // bl
4     int result; // eax
5     signed int i; // [esp+4Ch] [ebp-8h]
6     signed int v5; // [esp+50h] [ebp-4h]
7
8     dword_42D23C = 0;
9     dword_42D240 = 0;
10    v5 = strlen(Str);
11    sub_401014();
12    for ( i = 0; i < v5; ++i )
13    {
14        v2 = Str[i];
15        *(_BYTE *)(i + a2) = sub_401005() ^ v2;
16    }
17    result = i + a2;
18    *(_BYTE *)(i + a2) = 0;
19    return result;
20 }
```

sub\_401014, S盒被初始化为0-255的升序数组

```

1 size_t sub_401190()
2 {
3     size_t result; // eax
4     unsigned __int8 v1; // [esp+4Ch] [ebp-10h]
5     signed int v2; // [esp+50h] [ebp-Ch]
6     int v3; // [esp+54h] [ebp-8h]
7     signed int i; // [esp+58h] [ebp-4h]
8     int j; // [esp+58h] [ebp-4h]
9     int k; // [esp+58h] [ebp-4h]
10
11     result = strlen(Str);
12     v2 = result;
13     for ( i = 0; i < v2; ++i )
14     {
15         Str[i] += 2;
16         result = i + 1;
17     }
18     for ( j = 0; j < 256; ++j )
19     {
20         result = j;
21         dword_42CE3C[j] = j;
22     }
23     v3 = 0;
24     for ( k = 0; k < 256; ++k )
25     {
26         v3 = ((unsigned __int8)Str[k % v2] + dword_42CE3C[k] + v3) % 256;
27         v1 = dword_42CE3C[k];
28         dword_42CE3C[k] = dword_42CE3C[v3];
29         result = v1;
30         dword_42CE3C[v3] = v1;
31     }
32     return result;
33 }

```

sub\_401005, 密钥流生成

```

1 char sub_401300()
2 {
3     unsigned __int8 v1; // [esp+50h] [ebp-4h]
4
5     dword_42D23C = (dword_42D23C + 1) % 256;
6     dword_42D240 = (dword_42CE3C[dword_42D23C] + dword_42D240) % 256;
7     v1 = dword_42CE3C[dword_42D23C];
8     dword_42CE3C[dword_42D23C] = dword_42CE3C[dword_42D240];
9     dword_42CE3C[dword_42D240] = v1;
10    return dword_42CE3C[(dword_42CE3C[dword_42D240] + dword_42CE3C[dword_42D23C]) % 256];
11 }

```

看到 %256、S盒初始化、密钥流生成的代码可以判断为RC4

找到初始密钥

```
Str db 'QB3jdx',0
```

每个字符+2才是初始密钥

```
10
11 result = strlen(Str);
12 v2 = result;
13 for ( i = 0; i < v2; ++i )
14 {
15     Str[i] += 2;
16     result = i + 1;
17 }
18 for ( j = 0; j < 256; ++j )
19 {
20     result = j;
21     str2[j] = j;
22 }
23 v3 = 0;
24 for ( k = 0; k < 256; ++k )
25 {
26     v3 = ((unsigned __int8)Str[k % v2] + str2[k] + v3) % 256;
27     v1 = str2[k];
28     str2[k] = str2[v3];
29     result = v1;
30     str2[v3] = v1;
31 }
32 return result;
```

多出来的一部分

给S赋初值

S的初始置换

```
def rc4_decrypt(ciphertext, key):
    s = list(range(256))
    j = 0
    for i in range(256):
        j = (j + s[i] + key[i % len(key)]) % 256
        s[i], s[j] = s[j], s[i]
    i, j = 0, 0
    plaintext = []
    for byte in ciphertext:
        i = (i + 1) % 256
        j = (j + s[i]) % 256
        s[i], s[j] = s[j], s[i]
        plaintext.append(byte ^ s[(s[i] + s[j]) % 256])
    return bytes(plaintext)

ciphertext = [0x1C, 0xF3, 0x73, 0x68, 0x13, 0xC8, 0x96, 0xE3, 0x5C, 0xB6,
              0x83, 0xDE, 0x28, 0x97, 0xE7, 0x84, 0x57, 0x45, 0xF0]
key = [0x51, 0x42, 0x33, 0x6a, 0x64, 0x78]
for i in range(6):
    key[i] = key[i] + 2
print(rc4_decrypt(ciphertext, key).decode())
```

A\_GOOD\_REST\_OF\_LIFE

test4.exe(DES)

```

1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     int i; // [esp+4Ch] [ebp-2Ch]
4     char v5[8]; // [esp+50h] [ebp-28h] BYREF
5     char Str[20]; // [esp+58h] [ebp-20h] BYREF
6     char v7[12]; // [esp+6Ch] [ebp-Ch] BYREF
7
8     strcpy(v7, "Reserse_");
9     puts("give me a string to encrypt:");
10    scanf("%s", Str);
11    if ( strlen(Str) == 8 )
12    {
13        sub_40100F(v7);
14        sub_401032(Str, v5);
15        for ( i = 0; i < 8; ++i )
16        {
17            if ( v5[i] != byte_42AA30[i] )
18            {
19                puts("Wrong!!");
20                system("pause");
21                return -1;
22            }
23        }
24        puts("Proud of you!!");
25        system("pause");
26        return 0;
27    }
28    else
29    {
30        puts("Wrong!!");
31        system("pause");
32        return -1;
33    }
34 }

```

看到DES的IP置换表了

byte\_42801C db 3Ah

db 32h ; 2

db 2Ah ; \*

db 22h ; "

db 1Ah

db 12h

db 0Ah

db 2

db 3Ch ; <

db 34h ; 4

db 2Ch ; ,

db 24h ; \$

db 1Ch

IP\_hex = [

0x3A, 0x32, 0x2A, 0x22, 0x1A, 0x

0x3C, 0x34, 0x2C, 0x24, 0x1C, 0x

0x3E, 0x36, 0x2E, 0x26, 0x1E, 0x

0x40, 0x38, 0x30, 0x28, 0x20, 0x

0x39, 0x31, 0x29, 0x21, 0x19, 0x

0x3B, 0x33, 0x2B, 0x23, 0x1B, 0x

0x3D, 0x35, 0x2D, 0x25, 0x1D, 0x

0x3F, 0x37, 0x2F, 0x27, 0x1F, 0x

]

```

from Crypto.Cipher import DES
encrypted=bytes([0xDA, 0x68, 0x84, 0xC1, 0xC9, 0x07, 0x1E, 0x48])
key=bytes([0x52,0x65,0x73,0x65,0x72,0x73,0x65,0x5f])
print(DES.new(key, DES.MODE_ECB).decrypt(encrypted))

```

## test5.exe(花指令)

花指令

```

loc_401010:                                     ; COI
push      ebp
mov       ebp, esp
sub       esp, 40h
push      ebx
push      esi
push      edi
lea       edi, [ebp-40h]
mov       ecx, 10h
mov       eax, 0CCCCCCCCh
rep stosd
push      offset aCanYouHelpMe                 ; "Ca
call      _printf

add       esp, 4
xor       eax, eax
jz        short near ptr loc_401039+1

```

```

loc_401039:                                     ; COI
jmp       near ptr 428044A6h

```

```

; -----
dw 0E800h

```

```

C0 74+dd 14Ch, 3304C483h, 0E90174C0h

```

```

; -----
push      offset aPleaseFixItInT                 ; "P]
call      _printf

```

```

add       esp, 4
push      offset aPause                         ; "n

```

NOP掉



```

xor     eax, eax
jz      short loc_40103A

```

```

; -----
db      90h
; -----

```

```

loc_40103A:                                ; CO
00      push     offset aThisProgramIsA    ; "T
00      call     _printf

```

这个也nop掉  
主函数上按p

```

; int __cdecl main(int argc, const char
_main:
jmp      loc_401010

```

```

; -----
align 10h

```

```

loc_401010:
push     ebp
mov      ebp, esp
sub      esp, 40h
push     ebx
push     esi
push     edi
lea      edi, [ebp-40h]
mov      ecx, 10h
mov      eax, 0CCCCCCCCh
rep stosd

```

修复完了

```
1 int sub_401010()
2 {
3     printf("Can you help me?\n");
4     printf("This program is accursed by a wizard!\n");
5     printf("Please fix it !!!(in two places)!!!\n");
6     system("pause");
7     return 0;
8 }
```

本题没有flag

### test6.exe(xor)

```
1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     char v4[40]; // [esp+50h] [ebp-40h]
4     char Str[20]; // [esp+78h] [ebp-18h] BYREF
5     int i; // [esp+8Ch] [ebp-4h]
6
7     printf("Please give me your input:\n");
8     sub_401005(Str, 15);
9     if ( strlen(Str) == 10 )
10    {
11        for ( i = 0; i < 10; ++i )
12            v4[i + 20] = byte_427A30[9 - i] ^ Str[i];
13        for ( i = 0; i < 10; ++i )
14            v4[i] = byte_427A3C[i] ^ v4[i + 20];
15        for ( i = 0; i < 10 && v4[i] == byte_427A48[i]; ++i )
16            ;
17        if ( i == 10 )
18            printf("Congratulations! You are right!\n");
19        else
20            printf("2 Sorry, you are wrong!\n");
21        system("pause");
22        return 0;
23    }
24    else
25    {
26        printf("1 Sorry,you are wrong!\n");
27        system("pause");
28        return 0;
29    }
30 }
```

```
str1=[0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A]
str2=[0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17, 0x18, 0x19, 0x1A]
str3=[0x5C, 0x54, 0x54, 0x57, 0x4C, 0x5F, 0x46, 0x58, 0x50, 0x3A]
v1=[]
str=[]
for i in range(10):
    v1.append(str2[i]^str3[i])
for i in range(10):
    str.append(str1[9-i]^v1[i])
print(''.join(chr(x) for x in str))
```

## test7.exe(位运算)

```

1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     char Str[16]; // [esp+50h] [ebp-14h] BYREF
4     int i; // [esp+60h] [ebp-4h]
5
6     printf("Plase give me your answer:\n");
7     scanf("%s", Str);
8     if ( strlen(Str) == 13 )
9     {
10         for ( i = 0; i < 13; ++i )
11         {
12             Str[i] = (8 * (Str[i] & 7)) | ((Str[i] & 0x38) >> 3) | Str[i] & 0xC0;
13             Str[i] ^= byte_429A40[i];
14         }
15         for ( i = 0; i < 13 && Str[i] == byte_429A30[i]; ++i )
16             ;
17         if ( i == 13 )
18             printf("Congratulations! You are right!\n");
19         else
20             printf("2 Sorry, you are wrong!\n");
21         system("pause");
22         return 0;
23     }
24     else
25     {
26         printf("1 Sorry,you are wrong!\n");
27         system("pause");
28         return 0;
29     }
30 }

```

```

target=[0x5B, 0x60, 0x69, 0x64, 0x4E, 0x7D,
        0x46, 0x40, 0x5B, 0x6A, 0x63, 0x5E, 0x01]
xorstr=[0x01, 0x02, 0x03, 0x04, 0x05, 0x06,
        0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D]
result=[]
str1=[(target[i]^xorstr[i]) for i in range(13)]
for a in str1:
    for x in range(256):
        trans_x = (8 * (x & 7)) | ((x & 0x38) >> 3) | (x & 0xC0)
        if trans_x == a:
            result.append(chr(x))
            break
print("".join(result))

```

## test8.exe(RC4改)

```

1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     signed int v4; // [esp+4Ch] [ebp-D0h]
4     signed int i; // [esp+50h] [ebp-CCh]
5     char v6[97]; // [esp+54h] [ebp-C8h] BYREF
6     __int16 v7; // [esp+B5h] [ebp-67h]
7     char v8; // [esp+B7h] [ebp-65h]
8     char Str[100]; // [esp+B8h] [ebp-64h] BYREF
9
10    memset(v6, 0, sizeof(v6));
11    v7 = 0;
12    v8 = 0;
13    puts("please input a correct string to encrypt:");
14    scanf("%s", Str);
15    v4 = strlen(Str);
16    sub_40100A(Str, (int)v6);
17    for ( i = 0; i < v4; ++i )
18    {
19        if ( v6[i] != byte_429B30[i] )
20        {
21            puts("Sorry you are wrong!");
22            system("pause");
23            return -1;
24        }
25    }
26    puts("Wow!!!Great!!!You are a genius!!!");
27    system("pause");
28    return 0;
29 }

```

跟test3一样的

```

1 size_t sub_401190()
2 {
3     size_t result; // eax
4     unsigned __int8 v1; // [esp+4Ch] [ebp-10h]
5     signed int v2; // [esp+50h] [ebp-Ch]
6     int v3; // [esp+54h] [ebp-8h]
7     signed int i; // [esp+58h] [ebp-4h]
8     int j; // [esp+58h] [ebp-4h]
9     int k; // [esp+58h] [ebp-4h]
10
11     result = strlen(Str);
12     v2 = result;
13     for ( i = 0; i < v2; ++i )
14     {
15         Str[i] += 2;
16         result = i + 1;
17     }
18     for ( j = 0; j < 256; ++j )
19     {
20         result = j;
21         dword_42CE3C[j] = j;
22     }
23     v3 = 0;
24     for ( k = 0; k < 256; ++k )
25     {
26         v3 = ((unsigned __int8)Str[k % v2] + dword_42CE3C[k] + v3) % 256;
27         v1 = dword_42CE3C[k];
28         dword_42CE3C[k] = dword_42CE3C[v3];
29         result = v1;
30         dword_42CE3C[v3] = v1;
31     }
32     return result;
33 }

```

```

def rc4_decrypt(ciphertext, key):
    s = list(range(256))
    j = 0
    for i in range(256):
        j = (j + s[i] + key[i % len(key)]) % 256
        s[i], s[j] = s[j], s[i]
    i, j = 0, 0
    plaintext = []
    for byte in ciphertext:
        i = (i + 1) % 256
        j = (j + s[i]) % 256
        s[i], s[j] = s[j], s[i]
        plaintext.append(byte ^ s[(s[i] + s[j]) % 256])
    return bytes(plaintext)

ciphertext = [0xD5, 0x23, 0xA5, 0x22, 0x75, 0xD8, 0xB7, 0x80]
key = [0x57, 0x4F, 0x52, 0x4B, 0x45, 0x52]
for i in range(6):
    key[i] = key[i] + 2
print(rc4_decrypt(ciphertext, key).decode())

```

```

1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     int i; // [esp+4Ch] [ebp-2Ch]
4     char v5[8]; // [esp+50h] [ebp-28h] BYREF
5     char Str[20]; // [esp+58h] [ebp-20h] BYREF
6     char v7[12]; // [esp+6Ch] [ebp-Ch] BYREF
7
8     strcpy(v7, "REVERSE!");
9     puts("give me a string to encrypt:");
10    scanf("%s", Str);
11    if ( strlen(Str) == 8 )
12    {
13        sub_40100F(v7);
14        sub_401032(Str, v5);
15        for ( i = 0; i < 8; ++i )
16        {
17            if ( v5[i] != byte_42AA30[i] )
18            {
19                puts("Wrong!!");
20                system("pause");
21                return -1;
22            }
23        }
24        puts("Proud of you!!");
25        system("pause");
26        return 0;
27    }
28    else
29    {
30        puts("Wrong!!");
31        system("pause");
32        return -1;
33    }
34 }

```

PC2置换表→DES

```
, char byte_428104[000]
```

```
byte_428164 db 0Eh
```

```
db 4
```

```
db 0Dh
```

```
db 1
```

```
db 2
```

```
db 0Fh
```

```
db 0Bh
```

```
db 8
```

```
db 3
```

```
db 0Ah
```

```
db 6
```

```
db 0Ch
```

```
db 5
```

```
db 9
```

```
db 0
```

```
db 7
```

```
db 0
```

```
db 0Fh
```

```
db 7
```

```
db 4
```

```
db 0Eh
```

```
db 2
```

```
db 0Dh
```

```
..
```

```
from Crypto.Cipher import DES
encrypted=bytes([0x66, 0xD3, 0xD5, 0xF4, 0x1A, 0xBF, 0x81, 0x28])
key=bytes([0x52,0x45,0x56,0x45,0x52,0x53,0x45,0x21])

print(DES.new(key, DES.MODE_ECB).decrypt(encrypted))
```

NEWYEAR!

test10.exe(花指令+仿射)

```

loc_4010C7:                                     ; COD
xor      eax, eax
jz       short near ptr loc_4010CB+1

```

```

loc_4010CB:                                     ; COD
jmp      near ptr 0C85697h

```

```

; -----
8B 55 88 83 C2+dd 0EB000000h, 88558B09h, 8901C283h, 458B8855
45 88 3B 45 8C+dd 0F90458Bh, 3FC45AFh, 0B999F845h, 1Ah, 5589
; -----
push     offset aQvldxt                        ; "qv
lea      ecx, [ebp-6Ch]
push     ecx
call     _strcmp

```

去花



```

11
12 v8 = 5;
13 v7 = 7;
14 memset(Str, 0, sizeof(Str));
15 v5 = 0;
16 v6 = 0;
17 puts("please input a string:");
18 scanf("%s", Str);
19 v3 = strlen(Str);
20 for ( i = 0; i < v3; ++i )
21 {
22     if ( Str[i] < 97 || Str[i] > 122 )
23     {
24         printf("Sorry! Hang on!");
25         return -1;
26     }
27 }
28 for ( j = 0; j < v3; ++j )
29     Str[j] = (v7 + v8 * (Str[j] - 97)) % 26 + 97;
30 if ( !strcmp(Str, Str2) )
31     puts("Ok, you know it. Just hang on.");
32 else
33     puts("Sorry! Hang on!");
34 system("pause");
35 return 0;
36 }

```

000010AE: sub 401010+22, (4010AE)

```

target = 'qvlxdt'
a=5
k=7
m=26
a_inv=pow(a, -1, m)
result = []
for i in range(6):
    result.append((((ord(target[i])-97-7)*a_inv)%26)+97)
print("".join(chr(x) for x in result))

```

higuys

test12.exe(位运算)

```

1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     char Str[28]; // [esp+50h] [ebp-20h] BYREF
4     int i; // [esp+6Ch] [ebp-4h]
5
6     printf("Plase give me your answer:\n");
7     scanf("%s", Str);
8     if ( strlen(Str) == 16 )
9     {
10         for ( i = 0; i < 16; ++i )
11             Str[i] = (4 * (Str[i] & 3)) | ((Str[i] & 0xC) >> 2) | Str[i] & 0xF0;
12         for ( i = 0; i < 16 && Str[i] == byte_429A30[i]; ++i )
13             ;
14         if ( i == 16 )
15             printf("Congratulations! You are right!\n");
16         else
17             printf("2 Sorry, you are wrong!\n");
18         system("pause");
19         return 0;
20     }
21     else
22     {
23         printf("1 Sorry,you are wrong!\n");
24         system("pause");
25         return 0;
26     }
27 }

```

```

str1=[0x5C, 0x65, 0x6C, 0x75, 0x78, 0x66, 0x71, 0x76,
      0x46, 0x7C, 0x50, 0x75, 0x7A, 0x7A, 0x63, 0x65]
result = []
for target in str1:
    for x in range(256):
        trans_x = (4 * (x & 3)) | ((x & 0xC) >> 2) | x & 0xF0
        if trans_x == target:
            result.append(chr(x))
            break

print("".join(result))

```

SecurityIsPuzzle

test13.exe(MD5)

```

13  memset(Source, 0, sizeof(Source));
14  v9 = 0;
15  v10 = 0;
16  Destination = 0;
17  v7 = 0;
18  printf("Please input your flag:\n");
19  scanf("%s", Source);
20  v4 = strlen(Source);
21  for ( i = 0; i < v4; ++i )
22  {
23      if ( Source[i] < 97 || Source[i] > 122 )
24      {
25          printf("Wrong,try again!\n");
26          return -1;
27      }
28  }
29  if ( v4 >= 5 )
30  {
31      strncpy(&Destination, Source, 4u);
32      sub_401014(Str1, &Destination, 4u);
33      if ( !strcmp(Str1, "b5c0b187fe309af0f4d35982fd961d7e", 0x20u) )
34          printf("Correct!\n");
35      else
36          printf("Wrong,try again!\n");
37  }
38  else
39  {
40      printf("Wrong,try again!\n");
41  }
42  system("pause");
43  return 0;
44 }

```

发现轮函数

很明显的MD5

转换几个数字为16进制也会发现常量与MD5的匹配

```

13  sub_401028(&v9, v8, v7, v6, *a5, 0xD76AA478, 7);
14  sub_401028(&v6, v9, v8, v7, a5[1], 0xE8C7B756, 12);
15  sub_401028(&v7, v6, v9, v8, a5[2], 0x242070DB, 17);
16  sub_401028(&v8, v7, v6, v9, a5[3], 0xC1BDCEEE, 22);
17  sub_401028(&v9, v8, v7, v6, a5[4], 0xF57C0FAF, 7);
18  sub_401028(&v6, v9, v8, v7, a5[5], 0x4787C62A, 12);
19  sub_401028(&v7, v6, v9, v8, a5[6], -1473231341, 17);
20  sub_401028(&v8, v7, v6, v9, a5[7], -45705983, 22);
21  sub_401028(&v9, v8, v7, v6, a5[8], 1770035416, 7);
22  sub_401028(&v6, v9, v8, v7, a5[9], -1958414417, 12);
23  sub_401028(&v7, v6, v9, v8, a5[10], -42063, 17);
24  sub_401028(&v8, v7, v6, v9, a5[11], -1990404162, 22);
25  sub_401028(&v9, v8, v7, v6, a5[12], 1804603682, 7);
26  sub_401028(&v6, v9, v8, v7, a5[13], -40341101, 12);
27  sub_401028(&v7, v6, v9, v8, a5[14], -1502002290, 17);
28  sub_401028(&v8, v7, v6, v9, a5[15], 1236535329, 22);
29  sub_401005(&v9, v8, v7, v6, a5[1], -165796510, 5);
30  sub_401005(&v6, v9, v8, v7, a5[6], -1069501632, 9);
31  sub_401005(&v7, v6, v9, v8, a5[11], 643717713, 14);

```

```
from hashlib import md5
```

```

from itertools import product
from string import ascii_letters, digits
import time

def brute_md5(md5_value, max_len=4):
    md5_value = md5_value.lower()
    if len(md5_value) != 32:
        print("Invalid MD5 hash!")
        return

    chars = ascii_letters + digits # 可扩展为更多字符
    start = time.time()

    for length in range(1, max_len + 1):
        for attempt in product(chars, repeat=length):
            s = ''.join(attempt)
            if md5(s.encode()).hexdigest() == md5_value:
                print(f"Found: {s} (Time: {time.time() - start:.2f}s)")
                return

    print(f"Not found in {max_len} chars.")

brute_md5(input("MD5 hash: "), max_len=10)

```

love + 任意小写字母

长度大于等于5

### test14.exe(仿射)

```

11  int v12; // [esp+C4h] [ebp-4h]
12
13  v12 = 9;
14  v11 = 7;
15  v10 = 3;
16  memset(Str1, 0, sizeof(Str1));
17  v8 = 0;
18  v9 = 0;
19  puts("please input a string:");
20  scanf("%s", Str1);
21  v6 = strlen(Str1);
22  for ( i = 0; i < v6; ++i )
23  {
24      if ( Str1[i] < 65 || Str1[i] > 90 )
25      {
26          printf("Sorry! Hang on!");
27          return -1;
28      }
29  }
30  for ( j = 0; j < v6; ++j )
31      Str1[j] = (v11 + v12 * (Str1[j] - 65)) % 26 + 65;
32  if ( !strcmp(Str1, Str2) )
33      puts("Ok, you know it. Just hang on.");
34  else
35      puts("Sorry! Hang on!");
36  return system("pause");
37 }

```

仿射加密

```

ciphertext_string = 'SHUJDU'
a = 9
k = 7
m = 26
base_ord = 65

a_inv = pow(a, -1, m)
result_codes = []
for char in ciphertext_string:
    char_code = ord(char)
    result_codes.append(((char_code - base_ord - k) * a_inv) % m) + base_ord)
plaintext_string = "".join(chr(code) for code in result_codes)
print(f"字符串格式: {plaintext_string}")

```

HANGON

## test15.exe(DES)

```

1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     int i; // [esp+4Ch] [ebp-2Ch]
4     char v5[8]; // [esp+50h] [ebp-28h] BYREF
5     char Str[20]; // [esp+58h] [ebp-20h] BYREF
6     char v7[12]; // [esp+6Ch] [ebp-Ch] BYREF
7
8     strcpy(v7, "TakeEasy");
9     puts("give me a string to encrypt:");
10    scanf("%s", Str);
11    if ( strlen(Str) == 8 )
12    {
13        sub_40100F(v7);
14        sub_401032(Str, v5);
15        for ( i = 0; i < 8; ++i )
16        {
17            if ( v5[i] != byte_42AA30[i] )
18            {
19                puts("Wrong!!");
20                system("pause");
21                return -1;
22            }
23        }
24        puts("G00d Job!!");
25        system("pause");
26        return 0;
27    }
28    else
29    {
30        system("pause");
31        return -1;
32    }
33 }

```

找到IP置换表

```
byte_42801C db 3Ah
```

```
db 32h ; 2
```

```
db 2Ah ; *
```

```
db 22h ; "
```

```
db 1Ah
```

```
db 12h
```

```
db 0Ah
```

```
db 2
```

```
db 3Ch ; <
```

```
db 34h ; 4
```

```
db 2Ch ; ,
```

```
db 24h ; $
```

```
db 1Ch
```

```
db 14h
```

```
db 0Ch
```

```
db 1
```

```
from Crypto.Cipher import DES
encrypted=bytes([0x28, 0x70, 0x77, 0x48, 0x7B, 0x4F, 0xFF, 0x3D])
key=bytes([0x54,0x61,0x6b,0x65,0x45,0x61,0x73,0x79])
print(DES.new(key, DES.MODE_ECB).decrypt(encrypted))
```

itiseasy

test16.exe(RC4+xor)

```
21  v13 = 0;
22  memset(v14, 0, sizeof(v14));
23  v15 = 0;
24  v16 = 0;
25  memset(v10, 0, sizeof(v10));
26  v11 = 0;
27  v12 = 0;
28  memset(v7, 0, sizeof(v7));
29  v8 = 0;
30  v9 = 0;
31  puts("please input a correct string to encrypt:");
32  scanf("%s", Str);
33  v4 = strlen(Str);
34  strncpy(Destination, Str, 0xCu);
35  sub_40100A(Destination, (int)v10);
36  strncpy(v17, Source, 7u);
37  for ( i = 0; i < 7; ++i )
38      v7[i] = byte_42C600[i] ^ v17[i];
39  strncpy(&v10[12], v7, 7u);
40  for ( j = 0; j < v4; ++j )
41  {
42      if ( v10[j] != byte_42C700[j] )
43      {
44          puts("Wrong!!!");
45          system("pause");
46          return -1;
47      }
48  }
49  puts("Great!!!");
50  system("pause");
51  return 0;
52 }
```

```

1 int __cdecl sub_4013D0(char *Str, int a2)
2 {
3     char v2; // b1
4     int result; // eax
5     signed int i; // [esp+4Ch] [ebp-8h]
6     signed int v5; // [esp+50h] [ebp-4h]
7
8     dword_42D23C = 0;
9     dword_42D240 = 0;
10    v5 = strlen(Str);
11    sub_401014();
12    for ( i = 0; i < v5; ++i )
13    {
14        v2 = Str[i];
15        *(_BYTE *)(i + a2) = sub_401005() ^ v2;
16    }
17    result = i + a2;
18    *(_BYTE *)(i + a2) = 0;
19    return result;
20 }

```

```

1 char sub_4012C0()
2 {
3     unsigned __int8 v1; // [esp+50h] [ebp-4h]
4
5     dword_42D23C = (dword_42D23C + 1) % 256;
6     dword_42D240 = (dword_42CE3C[dword_42D23C] + dword_42D240) % 256;
7     v1 = dword_42CE3C[dword_42D23C];
8     dword_42CE3C[dword_42D23C] = dword_42CE3C[dword_42D240];
9     dword_42CE3C[dword_42D240] = v1;
10    return dword_42CE3C[(dword_42CE3C[dword_42D240] + dword_42CE3C[dword_42D23C]) % 256];
11 }

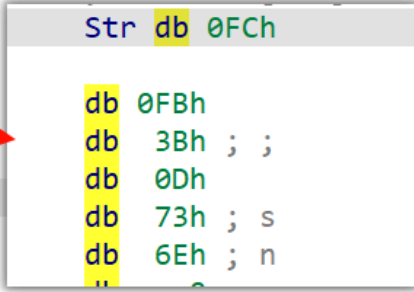
```



```

1 size_t sub_401190()
2 {
3     size_t result; // eax
4     unsigned __int8 v1; // [esp+4Ch] [ebp-10h]
5     signed int v2; // [esp+50h] [ebp-Ch]
6     int v3; // [esp+54h] [ebp-8h]
7     int i; // [esp+58h] [ebp-4h]
8     int j; // [esp+58h] [ebp-4h]
9
10    result = strlen(Str);
11    v2 = result;
12    for ( i = 0; i < 256; ++i )
13    {
14        dword_42CE3C[i] = i;
15        result = i + 1;
16    }
17    v3 = 0;
18    for ( j = 0; j < 256; ++j )
19    {
20        v3 = ((unsigned __int8)Str[j % v2] + dword_42CE3C[j] + v3) % 256;
21        v1 = dword_42CE3C[j];
22        dword_42CE3C[j] = dword_42CE3C[v3];
23        dword_42CE3C[v3] = v1;
24        result = j + 1;
25    }
26    return result;
27 }

```



Str db 0FCh

db 0FBh

db 3Bh ; ;

db 0Dh

db 73h ; s

db 6Eh ; n

前12位是RC4加密，后6位是异或加密

目标字符串只有12位，剩下的6位对应0，任何数异或0还是其本身

所以用RC4解密出的字符串直接连接异或密钥对应字符串

```

def rc4_decrypt(ciphertext, key):
    s = list(range(256))
    j = 0
    for i in range(256):
        j = (j + s[i] + key[i % len(key)]) % 256
        s[i], s[j] = s[j], s[i]

    i, j = 0, 0
    plaintext = []
    for byte in ciphertext:
        i = (i + 1) % 256
        j = (j + s[i]) % 256
        s[i], s[j] = s[j], s[i]
        plaintext.append(byte ^ s[(s[i] + s[j]) % 256])
    return bytes(plaintext)

key = [0xFC, 0xFB, 0x3B, 0x0D, 0x73, 0x6E]
target=[0x6E, 0x65, 0x11, 0xCF, 0x1D, 0x80, 0x3B, 0x4E,
        0x20, 0x2A, 0xE0, 0xB6]
xorstr=[0x4E, 0x6F, 0x74, 0x47, 0x6F, 0x6F, 0x64]
ciphertext=[0x6E, 0x65, 0x11, 0xCF, 0x1D, 0x80, 0x3B, 0x4E, 0x20, 0x2A, 0xE0, 0xB6]
xor=""
for x in xorstr:
    xor=xor+chr(x)
print(rc4_decrypt(ciphertext, key).decode()+xor)

```

NetworkClassNotGood

这题有个陷阱，如果输入为 NetworkClass 也会回应成功，但其实后面还有算法

test17.exe(SHA改)

```

18 Destination = 0;
19 v9 = 0;
20 printf("Please input your flag:\n");
21 scanf("%s", Source);
22 v5 = strlen(Source);
23 if ( v5 >= 5 )
24 {
25     for ( i = 0; i < v5; ++i )
26     {
27         if ( Source[i] < 65 || Source[i] > 122 )
28         {
29             printf("The inputs are out of the scope!");
30             system("pause");
31         }
32     }
33     strncpy(&Destination, Source, 4u);
34     v4 = strlen(&Destination);
35     sub_401005(Str1, (int)&Destination, v4);
36     for ( j = 0; j < 64; ++j )
37         ++Str1[j];
38     if ( !strcmp(Str1, "1f2e28649c4g:25:8bb:24c3D3EGF6GFg22dff:1dbd916df13239513g21e4663", 0x40u) )
39         printf("Correct!\n");
40     else
41         printf("Wrong,try again!\n");
42     system("pause");
43     return 0;
44 }
45 else
46 {
47     printf("Wrong,try again!\n");
48     system("pause");
49     return 0;
50 }
51 }

1 int __cdecl sub_401B20(char *Buffer, void *Src, size_t Size)
2 {
3     void *v4; // [esp+4Ch] [ebp-30h]
4     int v5; // [esp+50h] [ebp-2Ch] BYREF
5     unsigned int v6; // [esp+54h] [ebp-28h]
6     int v7; // [esp+58h] [ebp-24h] BYREF
7     int v8; // [esp+5Ch] [ebp-20h] BYREF
8     int v9; // [esp+60h] [ebp-1Ch] BYREF
9     int v10; // [esp+64h] [ebp-18h] BYREF
10    int v11; // [esp+68h] [ebp-14h] BYREF
11    int v12; // [esp+6Ch] [ebp-10h] BYREF
12    int v13; // [esp+70h] [ebp-Ch] BYREF
13    int v14; // [esp+74h] [ebp-8h] BYREF
14    unsigned int i; // [esp+78h] [ebp-4h]
15
16    v4 = malloc(0x40u);
17    sub_40100A(&v14, &v13, &v12, &v11, &v10, &v9, &v8, &v7);
18    v6 = sub_40100F((int)&v5, Src, Size);
19    for ( i = 0; i < v6 >> 6; ++i )
20    {
21        sub_401023(v4, (i << 6) + v5);
22        sub_40103C(&v14, &v13, &v12, &v11, &v10, &v9, &v8, &v7, v4);
23    }
24    sprintf(Buffer, "%08x%08x%08x%08x%08x%08x%08x%08x", v14, v13, v12, v11, v10, v9, v8, v7);
25    sub_402D70(v4);
26    return sub_402D70(v5);
27 }

```

找到了SHA的kt常量

```

35     v27[i] = *(_DWORD*)(a9 + 4 * i);
36     for ( i = 16; i < 64; ++i )
37     {
38         v9 = sub_401019(&v25 + i);
39         v10 = *(&v20 + i) + v9;
40         v11 = sub_40101E(v17[i + 1]);
41         v27[i] = v17[i] + v11 + v10;
42     }
43     for ( i = 0; i < 64; ++i )
44     {
45         v12 = sub_401032(v22);
46         v13 = v12 + v19;
47         v14 = sub_401028(v22, v21, v20);
48         v29 = v27[i] + dword_42801C[i] + v14 + v13;
49         v15 = sub_40102D(v26);
50         v28 = sub_401014(v26, v25, v24) + v15;
51         v19 = v20;
52         v20 = v21;
53         v21 = v22;
54         v22 = v29 + v23;
55         v23 = v24;
56         v24 = v25;
57         v25 = v26;

```

```

; int dword_42801C[64]
dword_42801C dd 428A2F98h
db 91h
db 44h ; D
db 37h ; 7
db 71h ; q
db 0CFh
db 0FBh
db 0C0h
db 0B5h
db 0A5h
db 0DBh
db 0B5h
db 0E9h
db 5Bh ; [
db 0C2h
db 56h ; V
db 39h ; 9
db 0F1h
db 11h

```

比较的字符串要整体-1

```

for ( j = 0; j < 64; ++j )
    ++Str1[j];
if ( !strncmp(Str1, "1f2e28649c4g:25:8bb:24c3D3EGF6GFg22df

```

```

import hashlib
import itertools
import string

def find_hash_collision(target_hash_hex):
    # 将16进制的哈希字符串转换为字节
    target_hash = bytes.fromhex(target_hash_hex)
    # 定义字符集（可以添加更多字符）
    # charset = string.ascii_letters + string.digits + string.punctuation
    charset = string.ascii_letters + string.digits
    # 定义要尝试的密码长度范围
    for length in range(3, 10): # 尝试长度
        # 枚举所有可能的字符组合
        for candidate in itertools.product(charset, repeat=length):
            candidate_str = ''.join(candidate)
            # 计算当前候选字符串的哈希值
            hashed = hashlib.sha256(candidate_str.encode()).digest()
            if hashed == target_hash:
                return candidate_str
    return None

if __name__ == "__main__":
    original_string = '1f2e28649c4g:25:8bb:24c3D3EGF6GFg22dff:1dbd916df13239513g21e4663'
    target_hash_hex = ''.join([chr(ord(char) - 1) for char in original_string])
    if len(target_hash_hex) != 64:
        print("错误：哈希值应该为64个字符的16进制字符串")
    else:
        result = find_hash_collision(target_hash_hex)

```

Seey +任意字母  
长度大于等于5

test18.exe(位运算+xor)

```

1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     char Str[16]; // [esp+50h] [ebp-14h] BYREF
4     int i; // [esp+60h] [ebp-4h]
5
6     printf("Plase give me your answer:\n");
7     scanf("%s", Str);
8     if ( strlen(Str) == 8 )
9     {
10         for ( i = 0; i < 8; ++i )
11         {
12             Str[i] = ((Str[i] & 1) << 7) | ((Str[i] & 0x80) >> 7) | Str[i] & 0x7E;
13             Str[i] ^= byte_429A38[i];
14         }
15         for ( i = 0; i < 8 && Str[i] == byte_429A30[i]; ++i )
16             ;
17         if ( i == 8 )
18             printf("Congratulations! You are right!\n");
19         else
20             printf("Sorry, you are wrong!\n");
21         system("pause");
22         return 0;
23     }
24     else
25     {
26         printf("Sorry,you are wrong!\n");
27         system("pause");
28         return 0;
29     }
30 }

```

```

xorstr=[0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D]
target=[0x52, 0xC7, 0xC2, 0xCD, 0xEE, 0xEB, 0xFE, 0xF5]
for i in range(8):
    xorstr[i]=xorstr[i]^target[i]
for i in range(8):
    xorstr[i]=((xorstr[i] & 1) << 7) | ((xorstr[i] & 0x80) >> 7) | xorstr[i] & 0x7E
print(''.join([chr(byte) for byte in xorstr]))

```

TAKEasy

## test19.exe(取反)

```

1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     int i; // [esp+4Ch] [ebp-8h]
4     int v5; // [esp+50h] [ebp-4h]
5
6     puts("----- This one is pretty easy! JUST ENTER YOUR FLAG: ");
7     sub_40100A();
8     v5 = 0;
9     for ( i = 0; i < 14; ++i )
10    {
11        byte_42CCBC[i] = ~byte_42CCBC[i];
12        v5 += (unsigned __int8)byte_42CCBC[i] - (unsigned __int8)byte_429A30[i];
13    }
14    if ( v5 )
15        puts("\nSorry! Give it another try!\n");
16    else
17        puts("\nCongratulations! You get the right flag!\n");
18    system("pause");
19    return 0;
20 }

```

```

str=[0xBD, 0x9A, 0x9E, 0x8B, 0xD5, 0xCF, 0x92, 0x96,
    0x9C, 0x8D, 0x90, 0x91, 0xD5, 0xDE]
for i in range(len(str)):
    str[i]=~(str[i])&0xFF
print(''.join([chr(byte) for byte in str]))

```

Beat\*0micron\*!

## test20.exe(异常+仿射)

```

1 int sub_401190()
2 {
3     puts("Please input your flag: ");
4     scanf("%s", Str);
5     if ( strlen(Str) > 24 )
6         ++str2;
7     byte_42CD91 = 0;
8     return printf("\n--->The flag you just enter is \"%s\"\n\n", Str);
9 }

```

异常处理

```

9     ms_exc.registration.TryLevel = -1;
10    ms_exc.registration.ScopeTable = &stru_427078;
11    ms_exc.registration.ExceptionHandler = _except_handler3;
12    ms_exc.registration.Next = (struct _EH3_EXCEPTION_REGISTRATION *)NtCurrentTeb()->NtTib.ExceptionList;
13    v5 = a1;
14    v4 = a3;
15    v3 = a2;
16    ms_exc.old_esp = (DWORD)&v3;
17    v6 = 0;
18    while ( v6 < 5 )
19    {
20        if ( Str[str1[v6]] == '-' )
21        {
22            ++v6;
23        }
24        else
25        {
26            ++str2;
27            sub_401019(v3, v4, v5);
28            system("pause");
29        }
30    }
31    ms_exc.registration.TryLevel = 0;
32    if ( 1 / (v6 - 5) ) 除0异常
33        ++str2;
34    sub_401019(v3, v4, v5);
35    exit(0);
36 }

```

```

while ( v6 < 5 )
{
    if ( Str[str1[v6]] == 0x2D )
    {
        ++v6;
    }
    else
    {
        ++str2;
        sub_401019(v3, v4, v5);
        system("pause");
    }
}

```

长度为5的数组

```

str1 dd 9
db 0Ch
db 0
db 0
db 0
db 10h
db 0
db 0
db 0
db 13h
db 0
db 0
db 0
db 15h
db 0
db 0
db 0

```

得知input的第9、12、16、19、21为

进入sub\_401019查看

```

1 int sub_401120()
2 {
3     const char *v0; // eax
4
5     v0 = (const char *)sub_401005(&unk_429AC4, str2);
6     puts(v0);
7     return puts("\n");
8 }

```

```

1 int __cdecl sub_401060(int a1, int a2)
2 {
3     int i; // [esp+4Ch] [ebp-4h]
4     int j; // [esp+4Ch] [ebp-4h]
5
6     for ( i = 0; i < 19; ++i )
7         *(_BYTE *)(i + a1) ^= byte_429AF0[i];
8     for ( j = 0; j < 19; ++j )
9         *(_BYTE *)(j + a1 + 21) ^= byte_429AF0[j];
10    if ( a2 )
11        return a1 + 21;
12    else
13        return a1;
14 }

```

这里判断a2是否为0，若为0返回前半部分，不为0返回后半部分  
写脚本对sub\_401060函数进行破解

```
unk429AC4=[0x08, 0x04, 0x00, 0x13, 0x12, 0x40, 0x4F, 0x1E, 0x54, 0x52,
0x08, 0x1D, 0x44, 0x1C, 0x49, 0x02, 0x1A, 0x06, 0x4E, 0x00,
0x00, 0x1C, 0x19, 0x17, 0x00, 0x1F, 0x40, 0x4F, 0x1E, 0x54,
0x52, 0x08, 0x1D, 0x44, 0x19, 0x52, 0x0A, 0x1C, 0x15, 0x4E]
xor=[0x4F, 0x76, 0x65, 0x72, 0x66, 0x6C, 0x6F, 0x77, 0x20, 0x72,
0x61, 0x6E, 0x64, 0x6E, 0x20, 0x65, 0x72, 0x72, 0x6F, 0x72]
for i in range(19):
    unk429AC4[i]=xor[i]^unk429AC4[i]
for i in range(19):
    unk429AC4[i+21]=xor[i]^unk429AC4[i+21]
print(''.join([chr(byte) for byte in unk429AC4]))
```

Great, it is right! Sorry, it is wrong!

得到的提示是，a2应该为0，这样才能返回Great

sub\_401019的功能就是结束程序，并输出Great或者Sorry

从外层函数看到a2是str2，再到外层看到str2++的情况

```
16 ms_exc.014_esp = (DWORD)&v5,
17 v6 = 0;
18 while ( v6 < 5 )
19 {
20     if ( Str[str1[v6]] == '-' )
21     {
22         ++v6;
23     }
24     else
25     {
26         ++str2;
27         sub_401019(v3, v4, v5);
28         system("pause");
29     }
30 }
31 ms_exc.registration.TryLevel = 0;
32 if ( 1 / (v6 - 5) )
33     ++str2;
34 sub_401019(v3, v4, v5);
35 exit(0);
36 }
```

判断Str[str1[v6]]=='-' 对v6=0~5成立，如果不成立就会调用401019结束

抛出异常后执行的函数在ScopeTable里会有

(ScopeTable是一张标注各种异常处理调用的函数位置的"地图")

```
9 ms_exc.registration.TryLevel = -1;
10 ms_exc.registration.ScopeTable = &stru_427078;
11 ms_exc.registration.ExceptionHandler = _except_handler3;
12 ms_exc.registration.Next = (struct _EH3_EXCEPTION_REGISTRATION *)NtCurrentTeb()->NtTib.ExceptionList;
13 v5 = a1;
14 v4 = a3;
```

双击查看，三个参数，第二个是筛选函数(筛选异常类型)，第三个是处理函数

```
db 0
30 15+stru_427078 _SCOPETABLE_ENTRY {0FFFFFFFh, offset loc_40152A, offset loc_401530>
; DATA XREF: sub_401470+510
align 0
```

直接进入loc\_401530查看，调用了401014

```
loc_401530:                                ; DATA XREI
;   __except(loc_40152A) // owned by 4014F6
mov     esp, [ebp+ms_exc.old_esp]
call    sub_401014
;   } // starts at 4014F6
```

```
sub_401014 proc near
jmp     sub_4012E0
```

终于找到了

```
1 int sub_4012E0()
2 {
3     signed int i; // [esp+54h] [ebp-14h]
4     signed int v2; // [esp+58h] [ebp-10h]
5
6     v2 = strlen(Str);
7     for ( i = 0; i < v2; ++i )
8     {
9         if ( Str[i] < 97 || Str[i] > 122 )
10        {
11            if ( Str[i] >= 65 && Str[i] <= 90 )
12                Str[i] = (3 * (Str[i] - 65) + 7) % 26 + 65;
13        }
14        else
15        {
16            Str[i] = (3 * (Str[i] - 97) + 7) % 26 + 97;
17        }
18    }
19    return sub_40100A();
20 }
```

是一个仿射加密



```
1 size_t sub_401230()  
2 {  
3     size_t result; // eax  
4     int i; // [esp+4Ch] [ebp-8h]  
5  
6     result = strlen(Str);  
7     if ( result == 24 )  
8     {  
9         for ( i = 0; i < 24; ++i )  
10        {  
11            result = Str[i];  
12            if ( result != dword_429A44[i] )  
13                ++str2;  
14        }  
15    }  
16    else  
17    {  
18        return ++str2;  
19    }  
20    return result;  
21 }
```

target字符串给的是数组格式，选C变量导出自动删除了多余的空格

- ☐ C unsigned char array (hex)
- ☐ C unsigned char array (decimal)
- ☒ initialized C variable
- ☐ raw bytes

☐ Save data to clipboard

Preview

```
int dword_429A44[32] =  
{  
    119,  
    111,  
    104,  
    122,  
    123,  
    77,  
    49,  
    104,  
    117,  
    45,  
    55,  
    49,  
    45,  
    52,
```

Line:1 Column:1

Output file export\_results.txt

Export

```
def affine_decrypt(ciphertext, a, k, m=26):
    plaintext = []
    try:
        a_inv = pow(a, -1, m)
    except ValueError:
        raise ValueError("参数 'a' 没有模逆元，无法解密。")
    for char in ciphertext:
        if 'a' <= char <= 'z':
            base_ord = 97
            y = ord(char) - base_ord
            decrypted_code = (a_inv * (y - k)) % m + base_ord
            plaintext.append(chr(decrypted_code))
        elif 'A' <= char <= 'Z':
            base_ord = 65
            y = ord(char) - base_ord
            decrypted_code = (a_inv * (y - k)) % m + base_ord
            plaintext.append(chr(decrypted_code))
        else:
            plaintext.append(char)
    return "".join(plaintext)

if __name__ == '__main__':
    a = 3
    k = 7
    ciphertext_hex = [119,111,104,122,123,77,49,104,
```

```

117,45,55,49,45,52,104,48,
45,57,116,45,104,45,33,125]
ciphertext_string = "".join(chr(code) for code in ciphertext_hex)
decrypted_text = affine_decrypt(ciphertext_string, a, k)
hex_array_output = [hex(ord(char)) for char in decrypted_text]
print(f"密文: {ciphertext_string}")
print(f"明文: {decrypted_text}")
print(f'Hex数组: [{"", ".join(hex_array_output)}]')

```

flag{T1an-71-4a0-9e-a-!}

## test21.exe(hook+xor)

hook

```

18 v10[0] = -25;
19 v10[1] = 0;
20 *(_DWORD *)&v10[2] = 0;
21 strcpy((char *)Buffer, "realpwd");
22 hModule = GetModuleHandleA("kernel32.dll");
23 WriteFile = (BOOL (__stdcall *)(HANDLE, LPCVOID, DWORD, LPDWORD, LPOVERLAPPED))GetProcAddress(hModule, "WriteFile");
24 lpAddress = WriteFile;
25 if ( VirtualProtect(WriteFile, 5u, 0x40u, &f10ldProtect) )
26 {
27     memcpy(&unk_42DC8C, lpAddress, 5u);
28     Src = (char *)sub_40100A - (char *)WriteFile - 5;
29     memcpy(&v10[1], &Src, 4u);
30     memcpy(WriteFile, v10, 5u);
31     VirtualProtect(WriteFile, 5u, f10ldProtect, &f10ldProtect);
32 }
33 hFile = CreateFileA("pwd.txt", 0x10000000u, 0, 0, 2u, 0x80u, 0);
34 v3 = strlen((const char *)Buffer);
35 ::WriteFile(hFile, Buffer, v3, (LPDWORD)&Buffer[2], 0);
36 CloseHandle(hFile);
37 Stream = fopen("pwd.txt", "r");
38 if ( Stream )
39     printf("File open success\n");
40 else
41     printf("File open fail\n");
42 if ( Stream )
43     fscanf(Stream, "%s", String2);
44 else
45     printf("scan fail\n");
46 if ( Stream )
47     fclose(Stream);
48 if ( !strcmpA((LPCSTR)Buffer, String2) )
49     printf("try again!\n");
50 else
51     printf("congratulations!\n");
52     system("Pause");
53     return 0;
54 }

```

sub\_40100A为回调函数

```

10 Str[0] = 26;
11 Str[1] = 10;
12 Str[2] = 14;
13 Str[3] = 7;
14 Str[4] = 17;
15 Str[5] = 7;
16 Str[6] = 13;
17 Str[7] = 0;
18 sub_401005(this);
19 printf("Please input your flag \n");
20 scanf("%s", v11);
21 for ( i = 0; i < 7; ++i )
22     Str[i] ^= v11[i];
23 ModuleHandleA = GetModuleHandleA("kernel32.dll");
24 WriteFile = (BOOL ( __stdcall *) (HANDLE, LPCVOID, DWORD, LPDWORD, LPOVERLAPPED))GetProcAddress(
25     ModuleHandleA,
26     "WriteFile");
27 v7 = strlen(Str);
28 WriteFile(a2, Str, v7, a5, a6);
29 return 1;
30 }

```

```

WriteFile = (BOOL ( __stdcall *) (HANDLE, LPCVOID, DWORD, LPDWORD, LPOVERLAPPED))GetProcAddress(
    ModuleHandleA,
    "WriteFile");
if ( VirtualProtect(WriteFile, 5u, 0x40u, &f10ldProtect) )
{
    memcpy(&unk_42DC8C, lpAddress, 5u);
    Src = (char *)sub_40100A - (char *)WriteFile - 5;
    memcpy(&v10[1], &Src, 4u);
    memcpy(WriteFile, v10, 5u);
    VirtualProtect(WriteFile, 5u, f10ldProtect, &f10ldProtect);
}
hFile = CreateFileA("pwd.txt", 0x10000000u, 0, 0, 2u, 0x80u, 0);

```

```

21 strcpy((char *)Buffer, "realpwd");
22 hModule = GetModuleHandleA("kernel32.dll");
23 WriteFile = (BOOL ( __stdcall *) (HANDLE, LP
48 if ( lstrcmpA((LPCSTR)Buffer, String2) )
49     printf("try again!\n");
50 else

```

原理解释

```

21 strcpy((char *)Buffer, "realpwd");
22 hModule = GetModuleHandleA("kernel32.dll");
23 WriteFile = (BOOL ( __stdcall *) (HANDLE, LPCVOID, DWORD, LPDWORD, LPOVERLAPPED))GetProcAddress(
24     hModule,
25     "WriteFile");
26 if ( VirtualProtect(WriteFile, 5u, 0x40u, &f10ldProtect) )
27 {
28     memcpy(&unk_42DC8C, lpAddress, 5u);
29     Src = (char *)sub_40100A - (char *)WriteFile - 5;
30     memcpy(&v10[1], &Src, 4u);
31     memcpy(WriteFile, v10, 5u);
32     VirtualProtect(WriteFile, 5u, f10ldProtect, &f10ldProtect);
33 }
34 hFile = CreateFileA("pwd.txt", 0x10000000u, 0, 0, 2u, 0x80u, 0);
35 v3 = strlen((const char *)Buffer);
36 ::WriteFile(hFile, Buffer, v3, (LPDWORD)&Buffer[2], 0);
37 CloseHandle(hFile);
38 Stream = fopen("pwd.txt", "r");
39 if ( Stream )
    printf("File open success\n");

```

**找到WriteFile地址**

**修改权限为 可读可写可执行**

**构建跳转指令**

**修改WriteFile前5个字节 替换成自定义函数地址**

```

9
10 Str[0] = 26;
11 Str[1] = 10;
12 Str[2] = 14;
13 Str[3] = 7;
14 Str[4] = 17;
15 Str[5] = 7;
16 Str[6] = 13;
17 Str[7] = 0;
18 sub_401005(this);
19 printf("Please input your flag \n");
20 scanf("%s", v11);
21 for ( i = 0; i < 7; ++i )
22     Str[i] ^= v11[i];
23 ModuleHandleA = GetModuleHandleA("kernel32.dll");
24 WriteFile = (BOOL (__stdcall *)(HANDLE, L
25
26
27 v7 = strlen(Str);
28 WriteFile(a2, Str, v7, a5, a6);
29 return 1;
30 }

```

脱钩

```

1 BOOL sub_401030()
2 {
3     HMODULE ModuleHandleA; // eax
4     FARPROC lpAddress; // [esp+50h] [ebp-8h]
5     DWORD f10ldProtect; // [esp+54h] [ebp-4h] BYREF
6
7     ModuleHandleA = GetModuleHandleA("kernel32.dll");
8     lpAddress = GetProcAddress(ModuleHandleA, "WriteFile");
9     VirtualProtect(lpAddress, 5u, 0x40u, &f10ldProtect);
10    memcpy(lpAddress, &unk_42DC8C, 5u);
11    return VirtualProtect(lpAddress, 5u, f10ldProtect, &f10ldProtect);
12 }

```

```

str=[26,10,14,7,17,7,13]
str2 = 'realpwd'
H = [ord(str2[i]) ^ str[i] for i in range(7)]
print(''.join(chr(x) for x in H))

```

hookapi

test22.exe(xor)

```

8  strcpy(v6, "@URU@KE");
9  v5[0] = 2;
10 v5[1] = 0;
11 v5[2] = 2;
12 v5[3] = 1;
13 v5[4] = 6;
14 v5[5] = 1;
15 v5[6] = 22;
16 v5[7] = 0;
17 printf("Plase give me your answer:\n");
18 scanf("%s", Str);
19 if ( strlen(Str) == 7 )
20 {
21     for ( i = 0; i < 7; ++i )
22         Str[i] ^= v5[i];
23     for ( i = 0; i < 7 && Str[i] == v6[i]; ++i )
24         ;
25     if ( i == 7 )
26         printf("Congratulations! You are right!\n");
27     else
28         printf("Sorry, you are wrong!\n");
29     system("pause");
30     return 0;
31 }
32 else
33 {
34     printf("Sorry,you are wrong!\n");
35     system("pause");
36     return 0;
37 }
38 }

```

```

v5=[2,0,2,1,6,1,22]
v6='@URU@KE'
H = [ord(v6[i]) ^ v5[i] for i in range(7)]
print(''.join(chr(x) for x in H))

```

BUPTFJS

test23.exe(花指令+仿射)

```
loc_4010B8: ; CODE
jmp     short loc_401085
```

```
; -----
```

```
loc_4010BA: ; CODE
xor     eax, eax
jz      short near ptr loc_4010BE+1
```

```
loc_4010BE: ; CODE
jmp     near ptr 0C8568Ah
```

```
; -----
```

```
align 4
```

```
5 88 83 C2 01+dd 9EB0000h, 8388558Bh, 558901C2h, 88458B88h, 7
3 3B 45 8C 7D+dd 0AF0F9045h, 4503FC45h, 1AB999F8h, 0F7000000h
dd offset aKbcwsxxsz ; "kbcw
```

```
; -----
```

```
12 v8 = 5;
13 v7 = 5;
14 memset(Str, 0, sizeof(Str));
15 v5 = 0;
16 v6 = 0;
17 puts("please input a string:");
18 scanf("%s", Str);
19 v3 = strlen(Str);
20 for ( i = 0; i < v3; ++i )
21 {
22     if ( Str[i] < 97 || Str[i] > 122 )
23         return -1;
24 }
25 for ( j = 0; j < v3; ++j )
26     Str[j] = (v7 + v8 * (Str[j] - 97)) % 26 + 97;
27 if ( !strcmp(Str, Str2) )
28 {
29     puts("ok, you really know");
30     puts("By the way, can you list all the Sections of this binary file???");
31 }
32 else
33 {
34     puts("sorry");
35 }
36 return system("pause");
37 }
```

```
ciphertext_string = 'kbcwsxxsz'
a = 5
k = 5
m = 26
base_ord = 97
a_inv = pow(a, -1, m)
result_codes = []
for char in ciphertext_string:
    char_code = ord(char)
    result_codes.append((((char_code - base_ord - k) * a_inv) % m) + base_ord)
```

```
plaintext_string = "".join(chr(code) for code in result_codes)
hex_array_output = [hex(code) for code in result_codes]

print(f"字符串格式: {plaintext_string}")
print(f"Hex数组格式: {hex_array_output}")
```

buptnoone

## test24.exe(MD5)

```
11
12  memset(Str1, 0, 33);
13  memset(Source, 0, sizeof(Source));
14  v9 = 0;
15  v10 = 0;
16  Destination = 0;
17  v7 = 0;
18  printf("Please input your flag:\n");
19  scanf("%s", Source);
20  v5 = strlen(Source);
21  for ( i = 0; i < v5; ++i )
22  {
23      if ( Source[i] < 48 || Source[i] > 57 )
24          return -1;
25  }
26  if ( v5 >= 5 )
27  {
28      strncpy(&Destination, Source, 4u);
29      sub_401014(Str1, (int)&Destination, 4);
30      if ( !strcmp(Str1, "eb62f6b9306db575c2d596b1279627a4", 0x20u) )
31          printf("Correct!\n");
32      else
33          printf("Wrong,try again!\n");
34  }
35  else
36  {
37      printf("Wrong,try again!\n");
38  }
39  system("pause");
40  return 0;
41 }
```

看着像Hash  
输入为数字

```
{
    if ( Source[i] < '0' || Source[i] > '9' )
        return -1;
}
```



```

13 unsigned int v14; // [esp+b4n] [ebp-8n]
14 unsigned int i; // [esp+68h] [ebp-4h]
15
16 v8 = malloc(0x40u);
17 v14 = sub_401046((int)&v13, Src, Size);
18 sub_401050(&v12, &v11, &v10, &v9);
19 for ( i = 0; i < v14 >> 6; ++i )
20 {
21     sub_40100F(v8, (void *)((i << 6) + v13));
22     sub_401037(&v12, &v11, &v10, &v9, v8);
23 }
24 v7 = sub_401032(v9);
25 v6 = sub_401032(v10);
26 v5 = sub_401032(v11);
27 v3 = sub_401032(v12);
28 sprintf(Buffer, "%08x%08x%08x%08x", v3, v5, v6, v7);
29 sub_403750(v8);
30 return sub_403750(v13);
31 }

```

找到MD5的轮常量表

```

12 v6 = *a4;
13 sub_401028(&v9, v8, v7, v6, *a5, 0xD76AA478, 7);
14 sub_401028(&v6, v9, v8, v7, a5[1], 0xE8C7B756, 12);
15 sub_401028(&v7, v6, v9, v8, a5[2], 0x242070DB, 17);
16 sub_401028(&v8, v7, v6, v9, a5[3], 0xC1BDCEEE, 22);
17 sub_401028(&v9, v8, v7, v6, a5[4], 0xF57C0FAF, 7);
18 sub_401028(&v6, v9, v8, v7, a5[5], 0x4787C62A, 12);
19 sub_401028(&v7, v6, v9, v8, a5[6], 0xA8304613, 17);
20 sub_401028(&v8, v7, v6, v9, a5[7], 0xFD469501, 22);
21 sub_401028(&v9, v8, v7, v6, a5[8], 0x698098D8, 7);
22 sub_401028(&v6, v9, v8, v7, a5[9], -1958414417, 12);
23 sub_401028(&v7, v6, v9, v8, a5[10], -42063, 17);
24 sub_401028(&v8, v7, v6, v9, a5[11], -1990404162, 22);
25 sub_401028(&v9, v8, v7, v6, a5[12], 1804603682, 7);
26 sub_401028(&v6, v9, v8, v7, a5[13], -40341101, 12);
27 sub_401028(&v7, v6, v9, v8, a5[14], -1502002290, 17);
28 sub_401028(&v8, v7, v6, v9, a5[15], 1236535329, 22);
29 sub_401005(&v9, v8, v7, v6, a5[1], -165796510, 5);

```

只需要穷举数字

```

from hashlib import md5
from string import ascii_letters, digits, punctuation
from itertools import permutations
import time
all=digits
def brute_md5(md5_value):
    md5_value=md5_value.lower()

```

```

if len(md5_value)==32:
    count=4
    start=time.time()
    while(1):
        for item in permutations(all,count):
            item="".join(item)
            if md5(item.encode()).hexdigest()==md5_value:
                end=time.time()
                print(end-start)
                print(item)
                return
            count+=1

md5_value = input()
brute_md5(md5_value)

```

0123 +任意数字  
长度大于等于5

## test25.exe(RC4改)

```

1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     signed int v4; // [esp+4Ch] [ebp-D0h]
4     signed int i; // [esp+50h] [ebp-CCh]
5     char v6[97]; // [esp+54h] [ebp-C8h] BYREF
6     __int16 v7; // [esp+B5h] [ebp-67h]
7     char v8; // [esp+B7h] [ebp-65h]
8     char Str[100]; // [esp+B8h] [ebp-64h] BYREF
9
10    memset(v6, 0, sizeof(v6));
11    v7 = 0;
12    v8 = 0;
13    puts("please input a correct string to encrypt:");
14    scanf("%s", Str);
15    v4 = strlen(Str);
16    sub_40100A(Str, (int)v6);
17    for ( i = 0; i < v4; ++i )
18    {
19        if ( v6[i] != byte_429B30[i] )
20        {
21            puts("Wrong!!!");
22            system("pause");
23            return -1;
24        }
25    }
26    puts("Great!!!");
27    system("pause");
28    return 0;
29 }

```

RC4

不太对劲，密钥流生成完调用的时候取反了

```

1 int __cdecl sub_4013D0(char *Str, int a2)
2 {
3     char v2; // b1
4     int result; // eax
5     signed int i; // [esp+4Ch] [ebp-8h]
6     signed int v5; // [esp+50h] [ebp-4h]
7
8     dword_42D23C = 0;
9     dword_42D240 = 0;
10    v5 = strlen(Str);
11    sub_401014();
12    for ( i = 0; i < v5; ++i )
13    {
14        v2 = Str[i];
15        *(_BYTE *)(i + a2) = ~(unsigned __int8)sub_401005() ^ v2;
16    }
17    result = i + a2;
18    *(_BYTE *)(i + a2) = 0;
19    return result;
20 }

```

```

1 size_t sub_401190()
2 {
3     size_t result; // eax
4     unsigned __int8 v1; // [esp+4Ch] [ebp-10h]
5     signed int v2; // [esp+50h] [ebp-Ch]
6     int v3; // [esp+54h] [ebp-8h]
7     int i; // [esp+58h] [ebp-4h]
8     int j; // [esp+58h] [ebp-4h]
9
10    result = strlen(Str);
11    v2 = result;
12    for ( i = 0; i < 256; ++i )
13    {
14        dword_42CE3C[i] = i;
15        result = i + 1;
16    }
17    v3 = 0;
18    for ( j = 0; j < 256; ++j )
19    {
20        v3 = ((unsigned __int8)Str[j % v2] + dword_42CE3C[j] + v3) % 256;
21        v1 = dword_42CE3C[j];
22        dword_42CE3C[j] = dword_42CE3C[v3];
23        dword_42CE3C[v3] = v1;
24        result = j + 1;
25    }
26    return result;
27 }

```

```

1 char sub_4012C0()
2 {
3     unsigned __int8 v1; // [esp+50h] [ebp-4h]
4
5     dword_42D23C = (dword_42D23C + 1) % 256;
6     dword_42D240 = (dword_42CE3C[dword_42D23C] + dword_42D240) % 256;
7     v1 = dword_42CE3C[dword_42D23C];
8     dword_42CE3C[dword_42D23C] = dword_42CE3C[dword_42D240];
9     dword_42CE3C[dword_42D240] = v1;
10    return dword_42CE3C[(dword_42CE3C[dword_42D240] + dword_42CE3C[dword_42D23C]) % 256];
11 }

```

核心是: `plaintext.append(byte ^ ~(s[(s[i] + s[j]) % 256])&0xFF)`

```

def rc4_decrypt(ciphertext, key):
    s = list(range(256))
    j = 0
    for i in range(256):
        j = (j + s[i] + key[i % len(key)]) % 256
        s[i], s[j] = s[j], s[i]

    i, j = 0, 0
    plaintext = []
    for byte in ciphertext:
        i = (i + 1) % 256
        j = (j + s[i]) % 256
        s[i], s[j] = s[j], s[i]
        plaintext.append(byte ^ ~(s[(s[i] + s[j]) % 256])&0xFF)
    return bytes(plaintext)

ciphertext = [0x10, 0x2C, 0x02, 0xFC, 0xFB, 0x3B, 0x0D, 0x73, 0x6E, 0xBC,
               0xB9, 0xA7, 0x6F, 0x2F]
key = b'secrets'
plaintext = rc4_decrypt(ciphertext, key)
print(plaintext.decode())

```

seeyounextyear

## test26.exe(异常+位运算 23)

#23年第一题

```

10 v9 = 1;
11 v8 = 0;
12 i = 0;
13 printf("Please input the password : ");
14 scanf("%s", Str);
15 v4 = strlen(Str);
16 if ( v4 != 18 )
17 {
18     printf("Sorry,you are wrong!\n");
19     system("pause");
20 }
21 strncpy(Destination, Str, 0x10u);
22 v6 = v9 / v8;
23 for ( i = 0; i < v4; ++i )
24     Destination[i] = Destination[i] & 3 | (16 * (Destination[i] & 0xC)) | ((Destination[i] & 0xF0) >> 2);
25 for ( i = 0; i < v4 && Destination[i] == byte_429A30[i]; ++i )
26 ;
27 if ( i == 16 )
28     printf("Congratulations! You are right!\n");
29 else
30     printf("Sorry, you are wrong!\n");
31 system("pause");
32 printf("What a pity, you found a wrong way.\n");
33 system("pause");
34 return -1;
35 }

```

除0异常

查看除0异常所在位置

```

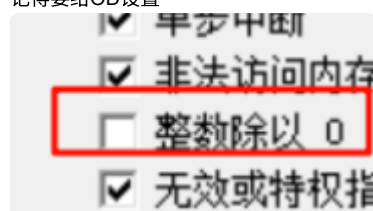
ext:0040121D C7 45 D4 05 10 40 00 mov [ebp+var_2C], 0115E1 sub_401005
ext:00401224 FF 75 D4 push [ebp+var_2C]
ext:00401227 64 FF 35 00 00 00 00 push large dword ptr fs:0
ext:0040122E 64 89 25 00 00 00 00 mov large fs:0, esp
ext:00401235 8B 45 FC mov eax, [ebp+var_4]
ext:00401238 99 cdq
ext:00401239 F7 7D F8 idiv [ebp+var_8]
ext:0040123C 89 45 F0 mov [ebp+var_10], eax
ext:0040123F C7 45 F4 00 00 00 00 mov [ebp+var_C], 0
ext:00401246 EB 09 jmp short loc_401251
ext:00401246
ext:00401248 ; -----
ext:00401248

```

直接用od打开定位，下断点

00401227	. 64:8925 000000	push	dword ptr fs:[0]
0040122E	. 64:8925 000000	mov	fs:[0], esp
00401235	. 8B45 FC	mov	eax, [local.1]
00401238	. 99	cdq	
00401239	. F77D F8	idiv	[local.2]
0040123C	. 8945 F0	mov	[local.4], eax
0040123F	. C745 F4 000000	mov	[local.3], 0
00401246	. EB 09	jmp	short 00401251
00401248	> 8B45 F4	mov	eax, [local.3]
0040124B	. 83C0 01	add	eax, 1

记得要给OD设置



重载F9运行至断点

要输入长度为18的字符串，否则在执行到除0异常前就被拦下了

00401239	. F77D F8	idiv	[local.2]
0040123C	. 8945 F0	mov	[local.4], eax
0040123F	. C745 F4 000000	mov	[local.3], 0
00401246	. EB 09	jmp	short 00401251
00401248	> 8B45 F4	mov	eax, [local.3]

当然也可以不需要在IDA里找到位置，直接在OD中运行自然会断在idiv处，此时在下断点即可

00401239	. F77D F8	idiv	[local.2]
0040123C	. 8945 F0	mov	[local.4], eax
0040123F	. C745 F4 000000	mov	[local.3], 0
00401246	. EB 09	jmp	short 00401251
00401248	> 8B45 F4	mov	eax, [local.3]

断下后查看→SEH链，双击第一个

地址	SE处理程序
0019FEB4	test26.00401005
0019FF64	test26.00406380
0019FFCC	ntd11.778F22E0
0019FFE4	ntd11.7792212C

按enter跳转

00401004	. CC	int3
00401005	. E9 16000000	jmp 00401020
0040100A	. E9 81010000	jmp 00401190
0040100F	. CC	int3
00401010	. CC	int3
00401020	> 55 push ebp	
00401021	. 8BEC mov ebp, esp	
00401023	. 83EC 48 sub esp, 48	
00401026	. 53 push ebx	
00401027	. 56 push esi	
00401028	. 57 push edi	
00401029	. 8D7D B8 lea edi, [local.18]	
0040102C	. B9 12000000 mov ecx, 12	
00401031	. B8 CCCCCCCC mov eax, CCCCCCCC	
00401036	. F3:AB rep stos dword ptr es:[edi]	
00401038	. C745 FC 0000 mov [local.1], 0	
0040103F	. 68 ACCC4200 push 0042CCAC	
00401044	. E8 E7040000 call 00401530	
00401049	. 83C4 04 add esp, 4	
0040104C	. 8945 F8 mov [local.2], eax	
0040104F	. C745 FC 0000 mov [local.1], 0	
00401056	. EB 09 jmp short 00401061	
00401058	> 8B45 FC mov eax, [local.1]	
0040105B	. 83C0 01 add eax, 1	
0040105E	. 8945 FC mov [local.1], eax	
00401061	> 8B4D FC mov ecx, [local.1]	
00401064	. 3B4D F8 cmp ecx, [local.2]	
00401067	. 7D 57 jge short 004010C0	
00401069	. 8B55 FC mov edx, [local.1]	
0040106C	. 0FB82 ACCC4 movsx eax, byte ptr [edx+42CCAC]	ASCII "fndsmkejrnfghdja"

找到异常处理函数地址，回到ida里查看

```

1 int sub_401020()
2 {
3     signed int v1; // [esp+4Ch] [ebp-8h]
4     signed int i; // [esp+50h] [ebp-4h]
5     signed int j; // [esp+50h] [ebp-4h]
6
7     v1 = strlen(Destination);
8     for ( i = 0; i < v1; ++i )
9     {
10         Destination[i] += i;
11         Destination[i] = ((Destination[i] & 3) << 6) | (4 * (Destination[i] & 0xC)) | ((Destination[i] & 0xF0) >> 4);
12     }
13     for ( j = 0; j < v1 && Destination[j] == byte_429A30[j]; ++j )
14     ;
15     if ( j == 16 )
16         printf("Congratulations! You are right!\n");
17     else
18         printf("Sorry, you are wrong!\n");
19     system("pause");
20     return 0;
21 }

```

- 每位加上 i
- 最低2位移动到最高2位
- 低4位的高2位左移2位
- 高4位移至低4位

```

target=[0xD5, 0x96, 0xC4, 0xF6, 0x07, 0x45, 0x57, 0x77, 0x76, 0xE5,
        0xF6, 0x48, 0x47, 0xF7, 0x48, 0x17]
result = []
for i in range(len(target)):
    trans_x = ((target[i] & 0x0F)<<4) | ((target[i] & 0xC0) >> 6) | ((target[i] & 0x30)>>2)
    result.append(chr(trans_x-i))

```



```
print("".join(result))
```

WeAllLoveReverse +任意字符

长度为18位

## test27.exe(仿射改 23)

#23年第二题

```
11
12  v11 = 2;
13  v10 = 7;
14  memset(Str1, 0, sizeof(Str1));
15  v8 = 0;
16  v9 = 0;
17  puts("Please input a string : ");
18  scanf("%s", Str1);
19  v6 = strlen(Str1);
20  for ( i = 0; i < v6; ++i )
21  {
22      if ( Str1[i] < 65 || Str1[i] > 89 )
23      {
24          printf("Sorry! Hang on!");
25          return -1;
26      }
27  }
28  for ( j = 0; j < v6; ++j )
29      Str1[j] = (v10 + v11 * (Str1[j] - 65)) % 25 + 65;
30  if ( !strcmp(Str1, Str2) )
31      puts("Ok, you know it. Just hang on.");
32  else
33      puts("Sorry! Hang on!");
34  system("pause");
35  return 0;
36 }
```

注意模数变成25了

```
ciphertext_string = 'HIPHSFUPSU'
a = 2
k = 7
m = 25
base_ord = 65
a_inv = pow(a, -1, m)

result_codes = []
for char in ciphertext_string:
    char_code = ord(char)
    result_codes.append((((char_code - base_ord - k) * a_inv) % m) + base_ord)

plaintext_string = "".join(chr(code) for code in result_codes)
hex_array_output = [hex(code) for code in result_codes]

print(f"字符串格式: {plaintext_string}")
print(f"Hex数组格式: {hex_array_output}")
```

## test28.exe(xor 23)

#23年第三题

```

1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     char v4[40]; // [esp+50h] [ebp-40h]
4     char Str[20]; // [esp+78h] [ebp-18h] BYREF
5     int i; // [esp+8Ch] [ebp-4h]
6
7     printf("Please give me your input:\n");
8     sub_401005(Str, 15);
9     if ( strlen(Str) == 10 )
10    {
11        for ( i = 0; i < 10; ++i )
12            v4[i + 20] = byte_427A30[9 - i] ^ Str[i];
13        for ( i = 0; i < 10; ++i )
14            v4[i] = byte_427A3C[i] ^ v4[i + 20];
15        for ( i = 0; i < 10 && v4[i] == byte_427A48[i]; ++i )
16            ;
17        if ( i == 10 )
18            printf("Congratulations! You are right!\n");
19        else
20            printf("2 Sorry, you are wrong!\n");
21        system("pause");
22        return 0;
23    }
24    else
25    {
26        printf("1 Sorry,you are wrong!\n");
27        system("pause");
28        return 0;
29    }
30 }

```

```

target=[0x52, 0x68, 0x53, 0x72, 0x70, 0x70, 0x59, 0x7A, 0x77, 0x2C]
xor=[0x11, 0x12, 0x13, 0x14, 0x15, 0x01, 0x02, 0x03, 0x04, 0x05]
xor2=[0x16, 0x17, 0x18, 0x19, 0x1A, 0x06, 0x07, 0x08, 0x09, 0x0A]
result=[]
for i in range(10):
    result.append(target[i]^xor[i])
for i in range(10):
    result[i]=xor2[9-i]^result[i]
print(''.join([chr(byte) for byte in result]))

```

IsHackBad?

## test29.exe(DES 23)



```
1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     int i; // [esp+4Ch] [ebp-2Ch]
4     char v5[8]; // [esp+50h] [ebp-28h] BYREF
5     char Str[20]; // [esp+58h] [ebp-20h] BYREF
6     char v7[12]; // [esp+6Ch] [ebp-Ch] BYREF
7
8     strcpy(v7, "$12*&^");
9     puts("give me a string to encrypt:");
10    scanf("%s", Str);
11    if ( strlen(Str) == 8 )
12    {
13        sub_40100F(v7);
14        sub_401032(Str, v5);
15        for ( i = 0; i < 8; ++i )
16        {
17            if ( v5[i] != byte_42AA30[i] )
18            {
19                puts("Wrong!!");
20                system("pause");
21                return -1;
22            }
23        }
24        puts("Proud of you!!");
25        system("pause");
26        return 0;
27    }
28    else
29    {
30        puts("Wrong!!");
31        system("pause");
32        return -1;
33    }
34 }
```

找到DES的循环移位表

```
byte_428154 db 1
```

```
db 1
db 2
db 2
db 2
db 2
db 2
db 2
db 1
db 2
db 2
db 2
db 2
db 2
db 2
db 1
```

```
from Crypto.Cipher import DES
encrypted = bytes([0x42, 0xAC, 0x43, 0xD3, 0xF1, 0x44, 0xB1, 0x36])
key = b'#$12*&^'
print(DES.new(key, DES.MODE_ECB).decrypt(encrypted))
```

1\_L0V5\_@

## test30.exe(SHA改 23)

#23年第五题

```
1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     char Str; // [esp+4Ch] [ebp-44Ch] BYREF
4     int v5; // [esp+4Dh] [ebp-44Bh]
5     char v6; // [esp+51h] [ebp-447h]
6     char Source[1021]; // [esp+54h] [ebp-444h] BYREF
7     __int16 v8; // [esp+451h] [ebp-47h]
8     char v9; // [esp+453h] [ebp-45h]
9     char Buffer[68]; // [esp+454h] [ebp-44h] BYREF
10
11     memset(Buffer, 0, 65);
12     memset(Source, 0, sizeof(Source));
13     v8 = 0;
14     v9 = 0;
15     Str = 0;
16     v5 = 0;
17     v6 = 0;
18     printf("Note: all the inputs should be located in the scope of a to z. Good Luck!\n");
19     sub_401028(Source);
20     sub_40100F(&Str, Source, Buffer);
21     sub_401014(Buffer);
22     system("pause");
23     return 0;
24 }
```

有三个函数，一个一个看

sub\_401028, 对输入进行翻转, 不改变输入是小写字母的情况

```

1 int __cdecl sub_401D70(char *Str)
2 {
3     puts("Please input your flag: ");
4     scanf("%s", Str);
5     if ( strlen(Str) != 6 )
6     {
7         printf("\nSorry, but try it again!\n\n");
8         system("pause");
9         exit(-1);
10    }
11    return sub_40100A(Str);
12 }

```

```

1 size_t __cdecl sub_401CC0(char *Str)
2 {
3     size_t result; // eax
4     char v2; // [esp+4Ch] [ebp-Ch]
5     signed int i; // [esp+50h] [ebp-8h]
6     signed int v4; // [esp+54h] [ebp-4h]
7
8     result = strlen(Str) - 1;
9     v4 = result;
10    for ( i = 0; i < v4; ++i )
11    {
12        v2 = Str[v4];
13        Str[v4] = Str[i];
14        Str[i] = v2;
15        result = i + 1;
16        --v4;
17    }
18    return result;
19 }

```

sub\_40100F

```

1 int __cdecl sub_401B40(char *Buffer, void *Src, size_t Size)
2 {
3     void *v4; // [esp+4Ch] [ebp-30h]
4     int v5; // [esp+50h] [ebp-2Ch] BYREF
5     unsigned int v6; // [esp+54h] [ebp-28h]
6     int v7; // [esp+58h] [ebp-24h] BYREF
7     int v8; // [esp+5Ch] [ebp-20h] BYREF
8     int v9; // [esp+60h] [ebp-1Ch] BYREF
9     int v10; // [esp+64h] [ebp-18h] BYREF
10    int v11; // [esp+68h] [ebp-14h] BYREF
11    int v12; // [esp+6Ch] [ebp-10h] BYREF
12    int v13; // [esp+70h] [ebp-Ch] BYREF
13    int v14; // [esp+74h] [ebp-8h] BYREF
14    unsigned int i; // [esp+78h] [ebp-4h]
15
16    v4 = malloc(0x40u);
17    sub_401019(&v14, &v13, &v12, &v11, &v10, &v9, &v8, &v7);
18    v6 = sub_40101E((int)&v5, Src, Size);
19    for ( i = 0; i < v6 >> 6; ++i )
20    {
21        sub_401037(v4, (i << 6) + v5);
22        sub_401050(&v14, &v13, &v12, &v11, &v10, &v9, &v8, &v7, v4);
23    }
24    sprintf(Buffer, "%08x%08x%08x%08x%08x%08x%08x%08x", v14, v13, v12, v11, v10, v9, v8, v7);
25    sub_402E70(v4);
26    return sub_402E70(v5);
27 }

```

像Hash

```
dword_42801C dd 428A2F98h
```

```
db 91h
```

```
db 44h ; D
```

```
db 37h ; 7
```

```
db 71h ; q
```

```
db 0CFh
```

```
db 0FBh
```

```
db 0C0h
```

```
db 0B5h
```

```
db 0A5h
```

```
db 0DBh
```

```
db 0B5h
```

```
db 0E9h
```

```
db 5Bh ; [
```

```
db 0C2h
```

```
db 56h ; V
```

```
db 39h ; 9
```

```
db 0F1h
```

```
db 11h
```

sub\_401014

```
1 int __cdecl sub_401E90(char *Str1)
2 {
3     if ( !strncmp(Str1, Str2, 0x40u) )
4         return printf("\n\n-----\nAwesome, wish you a nice day!\n-----\n\n");
5     else
6         return printf("\n\nSorry, but try it again!\n\n");
7 }
```

```
import hashlib
import itertools
import string

def find_hash_collision(target_hash_hex):
    target_hash = bytes.fromhex(target_hash_hex)
    charset = string.ascii_lowercase

    for length in range(4, 5):
        for candidate in itertools.product(charset, repeat=length):
            candidate_str = ''.join(candidate)
            hashed = hashlib.sha256(candidate_str.encode()).digest()
            if hashed == target_hash:
                return candidate_str

    return None

if __name__ == "__main__":

    target_hash_hex = "686f7446a95b6f836d7d70567c302c3f9ebb5ee0def3d1220ee9d4e9f34f5e131"

    if len(target_hash_hex) != 64:
        print("错误：哈希值应该为64个字符的16进制字符串")
    else:
        result = find_hash_collision(target_hash_hex)
        if result:
            print(f"找到匹配的字符串: {result}")
        else:
            print("未找到匹配的字符串")
```

得到 love

只有4位，而输入有6位，6位经翻转后前4位为love

由此得到

任意2位小写字母+evol

## test31.exe(xor)

```
1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     char v4[40]; // [esp+50h] [ebp-40h]
4     char Str[20]; // [esp+78h] [ebp-18h] BYREF
5     int i; // [esp+8Ch] [ebp-4h]
6
7     printf("Please give me your input:\n");
8     sub_401005(Str, 15);
9     if ( strlen(Str) == 10 )
10    {
11        for ( i = 0; i < 10; ++i )
12            v4[i + 20] = byte_427A30[9 - i] ^ Str[i];
13        for ( i = 0; i < 10; ++i )
14            v4[i] = byte_427A3C[i] ^ v4[i + 20];
15        for ( i = 0; i < 10 && v4[i] == byte_427A48[i]; ++i )
16            ;
17        if ( i == 10 )
18            printf("Congratulations! You are right!\n");
19        else
20            printf("Sorry, you are wrong!\n");
21        system("pause");
22        return 0;
23    }
24    else
25    {
26        printf("Sorry, the length is wrong!\n");
27        system("pause");
28        return 0;
29    }
30 }
```

```
target=[0x4C, 0x7E, 0x50, 0x7D, 0x7C, 0x64, 0x5A, 0x6F, 0x54, 0x70]
xor=[0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17, 0x18, 0x19, 0x1A]
xor2=[0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A]
result=[]
for i in range(10):
    result.append(target[i]^xor[i])
for i in range(10):
    result[i]=xor2[9-i]^result[i]
print(''.join([chr(byte) for byte in result]))
```

WeKnowItOk

## test32.exe(位运算)

```

1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     char Str[12]; // [esp+50h] [ebp-10h] BYREF
4     int i; // [esp+5Ch] [ebp-4h]
5
6     printf("Plase give me your answer:\n");
7     scanf("%s", Str);
8     if ( strlen(Str) == 10 )
9     {
10         for ( i = 0; i < 10; ++i )
11             Str[i] = (4 * (Str[i] & 3)) | ((Str[i] & 0xC) >> 2) | Str[i] & 0xF0;
12         for ( i = 0; i < 10 && Str[i] == byte_429A30[i]; ++i )
13             ;
14         if ( i == 10 )
15             printf("Congratulations! You are right!\n");
16         else
17             printf("Sorry, you are wrong!\n");
18         system("pause");
19         return 0;
20     }
21     else
22     {
23         printf("Sorry,you are wrong!\n");
24         system("pause");
25         return 0;
26     }
27 }

```

```

target=[0x4E, 0x45, 0x45, 0x50, 0x5F, 0x41, 0x58, 0x45, 0x44, 0x47]
result = []
for target in target:
    for x in range(256):
        trans_x = (4 * (x & 3)) | ((x & 0xC) >> 2) | x & 0xF0
        if trans_x == target:
            result.append(chr(x))
            break

print("".join(result))

```

KEEP\_DREAM

test33.exe(仿射)

```

13  v12 = 9;
14  v11 = 7;
15  v10 = 3;
16  memset(Str1, 0, sizeof(Str1));
17  v8 = 0;
18  v9 = 0;
19  puts("please input a string:");
20  scanf("%s", Str1);
21  v6 = strlen(Str1);
22  for ( i = 0; i < v6; ++i )
23  {
24      if ( Str1[i] < 65 || Str1[i] > 90 )
25      {
26          printf("Sorry! Hang on!");
27          return -1;
28      }
29  }
30  for ( j = 0; j < v6; ++j )
31      Str1[j] = (v11 + v12 * (Str1[j] - 65)) % 26 + 65;
32  if ( !strcmp(Str1, Str2) )
33      puts("Ok, you know it. Just hang on.");
34  else
35      puts("Sorry! Hang on!");
36  return system("pause");
37 }

```

```

ciphertext_string = 'QFMWRE'
a = 9
k = 7
m = 26
base_ord = 65
a_inv = pow(a, -1, m)
result_codes = []
for char in ciphertext_string:
    char_code = ord(char)
    result_codes.append((((char_code - base_ord - k) * a_inv) % m) + base_ord)

plaintext_string = "".join(chr(code) for code in result_codes)
hex_array_output = [hex(code) for code in result_codes]

print(f"字符串格式: {plaintext_string}")
print(f"Hex数组格式: {hex_array_output}")

```

BUPTER

test34.exe(MD5)

```

12  memset(Str1, 0, 33);
13  memset(Source, 0, sizeof(Source));
14  v9 = 0;
15  v10 = 0;
16  Destination = 0;
17  v7 = 0;
18  printf("Please input your flag:\n");
19  scanf("%s", Source);
20  v4 = strlen(Source);
21  for ( i = 0; i < v4; ++i )
22  {
23      if ( Source[i] < 65 || Source[i] > 90 )
24      {
25          printf("Wrong,try again!\n");
26          return -1;
27      }
28  }
29  if ( v4 >= 5 )
30  {
31      strncpy(&Destination, Source, 4u);
32      sub_401014(Str1, (int)&Destination, 4);
33      if ( !strcmp(Str1, "a3abe5e290d0bcc3c82ad572837b5d8d", 0x20u) )
34          printf("Correct!\n");
35      else
36          printf("Wrong,try again!\n");
37  }
38  else
39  {
40      printf("Wrong,try again!\n");
41  }
42  system("pause");
43  return 0;
44 }

```

MD5的轮常量表



vu = a4,

```
sub_401028(&v9, v8, v7, v6, *a5, 0xD76AA478, 7);
sub_401028(&v6, v9, v8, v7, a5[1], 0xE8C7B756, 12);
sub_401028(&v7, v6, v9, v8, a5[2], 0x242070DB, 17);
sub_401028(&v8, v7, v6, v9, a5[3], 0xC1BDCEEE, 22);
sub_401028(&v9, v8, v7, v6, a5[4], -176418897, 7);
sub_401028(&v6, v9, v8, v7, a5[5], 1200080426, 12);
sub_401028(&v7, v6, v9, v8, a5[6], -1473231341, 17);
sub_401028(&v8, v7, v6, v9, a5[7], -45705983, 22);
sub_401028(&v9, v8, v7, v6, a5[8], 1770035416, 7);
sub_401028(&v6, v9, v8, v7, a5[9], -1958414417, 12);
sub_401028(&v7, v6, v9, v8, a5[10], -42063, 17);
sub_401028(&v8, v7, v6, v9, a5[11], -1990404162, 22);
sub_401028(&v9, v8, v7, v6, a5[12], 1804603682, 7);
sub_401028(&v6, v9, v8, v7, a5[13], -40341101, 12);
sub_401028(&v7, v6, v9, v8, a5[14], -1502002290, 17);
sub_401028(&v8, v7, v6, v9, a5[15], 1236535329, 22);
sub_401005(&v9, v8, v7, v6, a5[1], -165796510, 5);
sub_401005(&v6, v9, v8, v7, a5[2], -556546863, 10);
```

```
from hashlib import md5
from itertools import product
from string import ascii_letters, digits
import time

def brute_md5(md5_value, max_len=4):
    md5_value = md5_value.lower()
    if len(md5_value) != 32:
        print("Invalid MD5 hash!")
        return
    chars = ascii_letters + digits # 可扩展为更多字符
    start = time.time()
    for length in range(1, max_len + 1):
        for attempt in product(chars, repeat=length):
            s = ''.join(attempt)
            if md5(s.encode()).hexdigest() == md5_value:
                print(f"Found: {s} (Time: {time.time() - start:.2f}s)")
                return
    print(f"Not found in {max_len} chars.")
brute_md5(input("MD5 hash: "), max_len=10)
```

SEPT +任意大写字母

test35.exe(DES)

```

1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     int i; // [esp+4Ch] [ebp-2Ch]
4     char v5[8]; // [esp+50h] [ebp-28h] BYREF
5     char Str[20]; // [esp+58h] [ebp-20h] BYREF
6     char v7[12]; // [esp+6Ch] [ebp-Ch] BYREF
7
8     strcpy(v7, "Mbuilder");
9     puts("give me a string to encrypt:");
10    scanf("%s", Str);
11    if ( strlen(Str) == 8 )
12    {
13        sub_40100F(v7);
14        sub_401032(Str, v5);
15        for ( i = 0; i < 8; ++i )
16        {
17            if ( v5[i] != byte_42AA30[i] )
18            {
19                puts("Wrong!!");
20                system("pause");
21                return -1;
22            }
23        }
24        puts("Proud of you!!");
25        system("pause");
26        return 0;
27    }
28    else
29    {
30        puts("Wrong!!");
31        system("pause");
32        return -1;
33    }
34 }

```

DES的IP置换表

```

byte_42801C db 3Ah
db 32h ; 2
db 2Ah ; *
db 22h ; "
db 1Ah
db 12h
db 0Ah
db 2
db 3Ch ; <
db 34h ; 4
db 2Ch ; ,
db 24h ; $
db 1Ch

```

```

from Crypto.Cipher import DES
encrypted = bytes([0x34, 0x69, 0xB1, 0x78, 0x3F, 0x5C, 0x9E, 0x3F])
key = b'Mbuilder'
print(DES.new(key, DES.MODE_ECB).decrypt(encrypted))
from Crypto.Cipher import DES
encrypted = bytes([0x34, 0x69, 0xB1, 0x78, 0x3F, 0x5C, 0x9E, 0x3F])
key = b'Mbuilder'
print(DES.new(key, DES.MODE_ECB).decrypt(encrypted))

```

achanceo

## test36.exe(SHA改 24)

#24年第一题

```

12 | memset(Str1, 0, 65);
13 | memset(Source, 0, sizeof(Source));
14 | v9 = 0;
15 | v10 = 0;
16 | Destination = 0;
17 | v7 = 0;
18 | printf("Please input your flag:\n");
19 | scanf("%s", Source);
20 | if ( strlen(Source) >= 5 )
21 | {
22 |     strncpy(&Destination, Source, 4u);
23 |     v4 = strlen(&Destination);
24 |     sub_401005(Str1, (int)&Destination, v4);
25 |     for ( i = 0; i < 64; ++i )
26 |     {
27 |         if ( ++Str1[i] == 103 )
28 |             Str1[i] = 97;
29 |         if ( Str1[i] == 58 )
30 |             Str1[i] = 48;
31 |     }
32 |     if ( !strcmp(Str1, "493f877692ea8d507fa98355a054efede85e7c7bbc9ba9890ea99b7b33e281fc", 0x40u) )
33 |         printf("Well done!");
34 |     else
35 |         printf("Wrong!");
36 |     system("pause");
37 |     return 0;
38 | }
39 | else
40 | {
41 |     printf("Wrong,try again!\n");
42 |     system("pause");
43 |     return 0;
44 | }
45 | }

```

SHA的常数Ki

```
, int dword_42801C[17]
```

```
dword_42801C dd 428A2F98h
```

```
db 91h
```

```
db 44h ; D
```

```
db 37h ; 7
```

```
db 71h ; q
```

```
db 0CFh
```

```
db 0FBh
```

```
db 0C0h
```

```
db 0B5h
```

```
db 0A5h
```

```
db 0DBh
```

```
db 0B5h
```

加密完经过了变换

```

...
sub_401005(Str1, (int)&Destination, v4);
for ( i = 0; i < 64; ++i )
{
    if ( ++Str1[i] == 103 )
        Str1[i] = 97;
    if ( Str1[i] == 58 )
        Str1[i] = 48;
}
if ( !strncmp(Str1, "493f877692ea8d507fa98355a

```

注意 ++Str1[i] 会对每个字符+1

再把ASCII为103的变为97，ASCII为58的变为48

逆向就是先97→103，48→58，再整体-1

```

import hashlib
import itertools
import string

def find_hash_collision(target_hash_hex):

    target_hash = bytes.fromhex(target_hash_hex)
    charset = string.ascii_letters + string.digits

    for length in range(4, 5):
        for candidate in itertools.product(charset, repeat=length):
            candidate_str = ''.join(candidate)
            hashed = hashlib.sha256(candidate_str.encode()).digest()
            if hashed == target_hash:
                return candidate_str
    return None

if __name__ == "__main__":
    s='493f877692ea8d507fa98355a054efede85e7c7bbc9ba9890ea99b7b33e281fc'
    result = [ord(c) for c in s]
    for i in range(len(result)):
        if result[i]== 97:
            result[i]= 103
        if result[i]== 48:
            result[i]= 58
        result[i]-=1
    target_hash_hex=''.join([chr(byte) for byte in result])
    result = find_hash_collision(target_hash_hex)
    if result:
        print(f"找到匹配的字符串: {result}")
    else:
        print("未找到匹配的字符串")

```

Luck +任意字符

## test37.exe(动态调试+xor 24)

#24年第二题

```

1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     ((void (*)(void))sub_401005)();
4     sub_401014();
5     sub_40100A();
6     system("pause");
7     return 0;
8 }

```

有3个函数，一个一个看

sub\_401005

```

1 int sub_401090()
2 {
3     int result; // eax
4
5     fputs("Please input your flag: ", &Stream);
6     scanf("%s", &Str);
7     result = strlen(&Str) - byte_429A31;
8     if ( result )
9     {
10         printf("You are wrong in the initial phase!");
11         return system("pause");
12     }
13     return result;
14 }

```

输入的长度为byte\_429A31的值(0x0F)，即15

sub\_401014

```

1 void *sub_401120()
2 {
3     void *result; // eax
4     int i; // [esp+4Ch] [ebp-4h]
5
6     result = memcpy(byte_42CCAC, (const void *)0x40004E, 0x2Bu);
7     for ( i = 0; i < 43; ++i )
8     {
9         byte_429A30 ^= byte_42CCAC[4 * i];
10        result = (void *)(i + 1);
11    }
12    return result;
13 }

```

byte\_42CCAC被赋值为内存地址 [0x40004E] 的值，无法查看，只能动态调试获得其值

```

1 void *sub_401120()
2 {
3     void *result; // eax
4     int i; // [esp+4Ch] [ebp-4h]
5
6     result = memcpy(byte_42CCAC, &unk_40004E, 43u);
7     for ( i = 0; i < 43; ++i )
8     {
9         byte_429A30 ^= byte_42CCAC[4 * i];
10        result = (void *)(i + 1);
11    }
12    return result;
13 }

```

EAX

```
.data:0042CCAC ; char byte_42CCAC[172]
.data:0042CCAC byte_42CCAC db 54h
.data:0042CCAC
• .data:0042CCAD db 68h ; h
• .data:0042CCAE db 69h ; i
• .data:0042CCAF db 73h ; s
• .data:0042CCB0 db 20h
• .data:0042CCB1 db 70h ; p
• .data:0042CCB2 db 72h ; r
• .data:0042CCB3 db 6Fh ; o
• .data:0042CCB4 db 67h ; g
• .data:0042CCB5 db 72h ; r
• .data:0042CCB6 db 61h ; a
• .data:0042CCB7 db 6Dh ; m
• .data:0042CCB8 db 20h
• .data:0042CCB9 db 63h ; c
• .data:0042CCBA db 61h ; a
• .data:0042CCBB db 6Eh ; n
• .data:0042CCBC db 6Eh ; n
• .data:0042CCBD db 6Fh ; o
• .data:0042CCBE db 74h ; t
• .data:0042CCBF db 20h
```

数组长度为43

```
byte_429A30 db 66h
```

```
byte_429A31 db 0Fh
```

```
align 4
```

```
byte_429A34 db 33h
```

```
db 19h
```

```
db 11h
```

```
db 32h ; 2
```

```
db 0Dh
```

```
db 27h ; '
db 21h ; !
```

```
db 11h
```

```
db 22h ; "
db 10h
```

```
db 11h
```

```
db 27h ; '
db 28h ; (
```

```
db 3Dh ; =
```

```
db 36h ; 6
```

```
db 0
```

```
db 0
```

```
db 0
```

```
db 0
```

```
db 0
```

Export as

- ☐ hex string (unspaced)
- ☐ hex string (spaced)
- ☐ string literal
- ☒ C unsigned char array (hex)
- ☐ C unsigned char array (decimal)
- ☐ initialized C variable
- ☐ raw bytes

☐ Save data to clipboard

Preview

```
unsigned char ida_chars[] =
{
    0x66, 0x0F, 0x00, 0x00, 0x33, 0x19, 0x11, 0x32, 0x0D, 0x27,
    0x21, 0x11, 0x22, 0x10, 0x11, 0x27, 0x28, 0x3D, 0x36
};
```

下面这个异或有点越界了？不管了，算都懒得算

直接在异或之后下断点

```
1 void *sub_401120()
2 {
3     void *result; // eax
4     int i; // [esp+4Ch] [ebp-4h]
5
6     result = memcpy(byte_42CCAC, &unk_40004E, 43u);
7     for ( i = 0; i < 43; ++i )
8     {
9         byte_429A30 ^= byte_42CCAC[4 * i];
10        result = (void *)(i + 1);
11    }
12    return result;
13 }
```

得到所有要用的值

```
00429A30 byte_429A30 db 78h
00429A30
00429A30
00429A31 byte_429A31 db 0Fh
00429A31
00429A32 align 4
00429A34 ; char byte_429A34[28]
00429A34 byte_429A34 db 33h
00429A35 db 19h
00429A36 db 11h
00429A37 db 32h ; 2
00429A38 db 0Dh
00429A39 db 27h ; '
00429A3A db 21h ; !
00429A3B db 11h
00429A3C db 22h ; "
00429A3D db 10h
00429A3E db 11h
00429A3F db 27h ; '
00429A40 db 28h ; (
00429A41 db 3Dh ; =
00429A42 db 36h ; 6
00429A43 db 0
```

byte\_429A30=0x78

byte\_429A31=0x0F

byte\_429A34=[0x33, 0x19, 0x11, 0x32, 0x0D, 0x27, 0x21, 0x11,

```
0x22, 0x10, 0x11, 0x27, 0x28, 0x3D, 0x36]
```

也可以自己写脚本算，越界部分都是0，异或0值不变，因此只用考虑  $4 \leq i < 43$  时的运算

```
byte_429A30=0x66

byte_42CCAC=[0x54, 0x68, 0x69, 0x73, 0x20, 0x70, 0x72, 0x6f, 0x67, 0x72,
             0x61, 0x6d, 0x20, 0x63, 0x61, 0x6e, 0x6e, 0x6f, 0x74, 0x20,
             0x62, 0x65, 0x20, 0x72, 0x75, 0x6e, 0x20, 0x69, 0x6e, 0x20,
             0x44, 0x4f, 0x53, 0x20, 0x6d, 0x6f, 0x64, 0x65, 0x2e, 0x0D,
             0x0D, 0x0A, 0x24]

for i in range(11):
    byte_429A30 ^= byte_42CCAC[i*4]
print(hex(byte_429A30))
```

得到byte\_429A30=0x78，跟直接动态调试看到的是一致的

进入第三个函数sub\_40100A

```
1 int sub_4011E0()
2 {
3     int i; // [esp+4Ch] [ebp-4h]
4
5     for ( i = 0; i < byte_429A31; ++i )
6     {
7         Str[i] ^= byte_429A30;
8         if ( Str[i] != byte_429A34[i] )
9         {
10             fputs("\n--- Sorry, but try it again! ---\n\n", &Stream);
11             system("pause");
12             return 0;
13         }
14     }
15     fputs("\n*** Good work! ***\n\n", &Stream);
16     return 0;
17 }
```

这些数组的值都能在刚才动态调试中得到

```
str=[0x33, 0x19, 0x11, 0x32, 0x0D, 0x27, 0x21, 0x11,
     0x22, 0x10, 0x11, 0x27, 0x28, 0x3D, 0x36]
for i in range(15):
    str[i]^=0x78
print(''.join([chr(byte) for byte in str]))
```

KaiJu\_YiZhi\_PEN

test38.exe(RC4改 24)



```
1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     signed int v4; // [esp+4Ch] [ebp-D0h]
4     signed int i; // [esp+50h] [ebp-CCh]
5     char v6[97]; // [esp+54h] [ebp-C8h] BYREF
6     __int16 v7; // [esp+B5h] [ebp-67h]
7     char v8; // [esp+B7h] [ebp-65h]
8     char Str[100]; // [esp+B8h] [ebp-64h] BYREF
9
10    memset(v6, 0, sizeof(v6));
11    v7 = 0;
12    v8 = 0;
13    puts("please input a correct string to encrypt:");
14    scanf("%s", Str);
15    v4 = strlen(Str);
16    sub_40100A(Str, (int)v6);
17    for ( i = 0; i < v4; ++i )
18    {
19        if ( v6[i] != byte_429B30[i] )
20        {
21            puts("Sorry you are wrong!");
22            system("pause");
23            return -1;
24        }
25    }
26    puts("WoW!!!Great!!!You are a genius!!!");
27    system("pause");
28    return 0;
29 }
```

一看就是RC4

```

1 int __cdecl sub_401410(char *Str, int a2)
2 {
3     char v2; // b1
4     int result; // eax
5     signed int i; // [esp+4Ch] [ebp-8h]
6     signed int v5; // [esp+50h] [ebp-4h]
7
8     dword_42D23C = 0;
9     dword_42D240 = 0;
10    v5 = strlen(Str);
11    sub_401014();
12    for ( i = 0; i < v5; ++i )
13    {
14        v2 = Str[i];
15        *(_BYTE *)(i + a2) = sub_401005() ^ v2;
16    }
17    result = i + a2;
18    *(_BYTE *)(i + a2) = 0;
19    return result;
20 }

```

作了手脚

```

11    result = strlen(Str);
12    v2 = result;
13    for ( i = 0; i < v2; ++i )
14    {
15        Str[i] ^= 0x76u;
16        result = i + 1;
17    }
18    for ( j = 0; j < 256; ++j )
19    {
20        result = j;
21        dword_42CE3C[j] = j;
22    }
23    v3 = 0;
24    for ( k = 0; k < 256; ++k )
25    {
26        v3 = ((unsigned __int8)Str[k % v2] + dword_42CE3C[k] + v3) % 256;
27        v1 = dword_42CE3C[k];
28        dword_42CE3C[k] = dword_42CE3C[v3];
29        result = v1;
30        dword_42CE3C[v3] = v1;
31    }
32    return result;
33 }

```

```

def rc4_decrypt(ciphertext, key):
    s = list(range(256))
    j = 0

```

```

for i in range(256):
    j = (j + s[i] + key[i % len(key)]) % 256
    s[i], s[j] = s[j], s[i]
i, j = 0, 0
plaintext = []
for byte in ciphertext:
    i = (i + 1) % 256
    j = (j + s[i]) % 256
    s[i], s[j] = s[j], s[i]
    plaintext.append(byte ^ s[(s[i] + s[j]) % 256])
return bytes(plaintext)
ciphertext = [0x61, 0x6E, 0x92, 0x2C, 0xEF, 0x13, 0x8B, 0x13, 0x28]
key = [0x45, 0x42, 0x50, 0x46, 0x31, 0x53, 0x47, 0x4F, 0x4F, 0x44]
for i in range(10):
    key[i]^=0x76
print(rc4_decrypt(ciphertext, key).decode())

```

AREYOUOK?

## test39.exe(花指令+DES 24)

#24年第四题

```

1 // attributes: thunk
2 int __cdecl main(int argc, const char **argv, const char **envp)
3 {
4     JUMPOUT(0x4010A0);
5 }

```

双击来到这里

010A0		; int __cdecl main_0(int argc, const char **argv, const char **envp)
010A0		_main_0: ; CODE XREF: _main↑j
010A0 55	push	ebp
010A1 8B EC	mov	ebp, esp
010A3 83 EC 6C	sub	esp, 6Ch
010A6 53	push	ebx
010A7 56	push	esi
010A8 57	push	edi
010A9 8D 7D 94	lea	edi, [ebp-6Ch]
010AC B9 1B 00 00 00	mov	ecx, 1Bh
010B1 B8 CC CC CC CC	mov	eax, 0CCCCCCCCh
010B6 F3 AB	rep stosd	
010B8 C6 45 F4 EF	mov	byte ptr [ebp-0Ch], 0EFh
010BC C6 45 F5 34	mov	byte ptr [ebp-0Bh], 34h ; '4'
010C0 C6 45 F6 D4	mov	byte ptr [ebp-0Ah], 0D4h
010C4 C6 45 F7 A3	mov	byte ptr [ebp-9], 0A3h
010C8 C6 45 F8 C6	mov	byte ptr [ebp-8], 0C6h
010CC C6 45 F9 84	mov	byte ptr [ebp-7], 84h
010D0 C6 45 FA E4	mov	byte ptr [ebp-6], 0E4h
010D4 C6 45 FB 23	mov	byte ptr [ebp-5], 23h ; '#'
010D8 33 C0	xor	eax, eax
010DA 88 45 FC	mov	[ebp-4], al
010DD C7 45 D4 00 00 00 00	mov	dword ptr [ebp-2Ch], 0
010E4 68 B4 84 42 00	push	offset aGiveMeAStringT ; "give me a string to encryp
010E9 E8 82 0E 00 00	call	_puts
010E9		
010EE 83 C4 04	add	esp, 4
010F1 8D 4D E0	lea	ecx, [ebp-20h]
010F4 51	nush	ecx

花指令

```
add     esp, 8
xor     eax, eax
jz      short near ptr loc_401106+1
```

```
loc_401106:                                ; COI
jmp     short loc_401095
```

```
00000009 v7[0] = -17;
00000010 v7[1] = 52;
00000011 v7[2] = -44;
00000012 v7[3] = -93;
00000013 v7[4] = -58;
00000014 v7[5] = -124;
00000015 v7[6] = -28;
00000016 strcpy(v8, "#");
00000017 puts("give me a string to encrypt:");
00000018 scanf("%s", Str);
00000019 if ( strlen(Str) == 8 )
00000020 {
00000021     sub_40100F(v7);
00000022     sub_401032(Str, v5);
00000023     for ( i = 0; i < 8; ++i )
00000024     {
00000025         if ( v5[i] != byte_42AA30[i] )
00000026         {
00000027             puts("Wrong!!");
00000028             system("pause");
00000029             return -1;
00000030         }
00000031     }
00000032     puts("Good Job!!\n");
00000033     puts("Wait a moment!!!\n");
00000034     puts("By the way, can you tell me a way to get the number of the external functions in this program, please?");
00000035     system("pause");
00000036     return 0;
00000037 }
00000038 else
00000039 {
00000040     system("pause");
00000041     return -1;
00000042 }
00000043 }
```

找到DES的循环移位表

```
; char byte_428154[16]
```

```
byte_428154 db 1
```

```
db 1
```

```
db 2
```

```
db 2
```

```
db 2
```

```
db 2
```

```
db 2
```

```
db 2
```

```
db 1
```

```
db 2
```

```
db 2
```

```
db 2
```

密钥是v7和v8（一共8位）

```
9  v7[0] = -17;  
0  v7[1] = 52;  
1  v7[2] = -44;  
2  v7[3] = -93;  
3  v7[4] = -58;  
4  v7[5] = -124;  
5  v7[6] = -28;  
6  strcpy(v8, "#");  
7  puts("give me a string to e  
8  scanf("%s", Str);  
9  if ( strlen(Str) == 8 )  
0  {  
1      sub_40100F(v7);  
2      sub_401032(Str, v5);  
3      for ( i = 0; i < 8; ++i )
```

```
from Crypto.Cipher import DES  
encrypted=bytes([0xCD, 0x49, 0x33, 0x09, 0xF3, 0x23, 0x21, 0x8A])  
key=bytes([0xEF, 0x34, 0xD4, 0xA3, 0xC6, 0x84, 0xE4, 0x23])  
print(DES.new(key, DES.MODE_ECB).decrypt(encrypted))
```

partplan

test40.exe(花指令+位运算 24)

#24年第五题

花指令

align 10h

```
loc_401010: ;
push     ebp
mov      ebp, esp
sub      esp, 5Ch
push     ebx
push     esi
push     edi
lea      edi, [ebp-5Ch]
mov      ecx, 17h
mov      eax, 0CCCCCCCCh
rep stosd
xor      eax, eax
jz       short near ptr loc_40102C+1
```

```
loc_40102C: ;
jmp      near ptr 42B09C99h
```

```
; -----
db 0, 0E8h, 49h
```

还有

```
xor      eax, eax
jz       short near ptr loc_40103E+1
```

```
loc_40103E:
jmp      near ptr 512855D0h
```

```
; -----
db 68h ; h
dd offset aS
```

C0+dd 2D3E8h, 8C48300h, 174C033h, 0E84D8DE9

还有

```
xor     eax, eax
jz      short near ptr loc_401054+1
```

```
loc_401054:
jmp     near ptr 52285DE6h
```

还有

```
xor     eax, eax
jz      short near ptr loc_401068+1
```

```
loc_401068:
jmp     near ptr 12248DF0h
```

还有

```
loc_401090:
xor     eax, eax
jz      short near ptr loc_401094+1
```

```
loc_401094:
jmp     near ptr 13C5660h
```

还有

```
xor     eax, eax
jz      short near ptr loc_4010EA+1
```

```
loc_4010EA:
jmp     near ptr 13C56B6h
```

还有

```
xor     eax, eax
jz      short near ptr loc_401123+1
```

```
loc_401123:
jmp     near ptr 123C8EABh
```

还有

```
loc_401140:  
xor     eax, eax  
jz      short near ptr loc_40114A+1  
  
loc_40114A:  
jmp     near ptr 42B075B7h
```

有了

```
1 int __cdecl main_0(int argc, const char **argv, const char **envp)  
2 {  
3     char Str[20]; // [esp+50h] [ebp-18h] BYREF  
4     int i; // [esp+64h] [ebp-4h]  
5  
6     printf("Plase give me your answer:\n");  
7     scanf("%s", Str);  
8     if ( strlen(Str) == 17 )  
9     {  
10        for ( i = 0; i < 17; ++i )  
11            Str[i] = (4 * (Str[i] & 3)) | ((Str[i] & 0xC) >> 2) | Str[i] & 0xF0;  
12        for ( i = 0; i < 17 && Str[i] == byte_429A30[i]; ++i )  
13            ;  
14        if ( i == 17 )  
15            printf("Congratulations! You are right!\n");  
16        else  
17            printf("2 Sorry, you are wrong!\n");  
18        system("pause");  
19        return 0;  
20    }  
21    else  
22    {  
23        printf("1 Sorry,you are wrong!\n");  
24        system("pause");  
25        return 0;  
26    }  
27 }
```

```
str1 = [0x46, 0x44, 0x47, 0x5C, 0x4F, 0x58, 0x58, 0x56, 0x46, 0x44,  
        0x47, 0x4D, 0x55, 0x46, 0x43, 0x51, 0x56]  
result = []  
for x in str1:  
    trans_x = (4 * (x & 0x3)) | ((x & 0xC) >> 2) | (x & 0xF0)  
    result.append(chr(trans_x))  
print("".join(result))
```

IAMSORRYIANGUILTY

test41.exe(xor)



```

1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     char v4[20]; // [esp+50h] [ebp-2Ch]
4     char Str[20]; // [esp+64h] [ebp-18h] BYREF
5     int i; // [esp+78h] [ebp-4h]
6
7     printf("Please give me your input:\n");
8     sub_401005((int)Str, 15);
9     if ( strlen(Str) == 10 )
10    {
11        for ( i = 0; i < 10; ++i )
12            v4[i] = byte_427A30[9 - i] ^ Str[i];
13        for ( i = 0; i < 10 && v4[i] == byte_427A3C[i]; ++i )
14            ;
15        if ( i == 10 )
16            printf("Congratulations! You are right!\n");
17        else
18            printf("2 Sorry, you are wrong!\n");
19        system("pause");
20        return 0;
21    }
22    else
23    {
24        printf("1 Sorry,you are wrong!\n");
25        system("pause");
26        return 0;
27    }
28 }

```

```

xor1=[0x57, 0x66, 0x67, 0x63, 0x59, 0x47, 0x7D, 0x76, 0x23, 0x30]
xor2=[0x11, 0x02, 0x13, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x10]
for i in range(10):
    xor1[i]=xor1[i]^xor2[9-i]
print(''.join([chr(byte) for byte in xor1]))

```

Good\_Bye!!

test42.exe(位运算)

```

1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     char Str[28]; // [esp+50h] [ebp-20h] BYREF
4     int i; // [esp+6Ch] [ebp-4h]
5
6     printf("Plase give me your answer:\n");
7     scanf("%s", Str);
8     if ( strlen(Str) == 16 )
9     {
10         for ( i = 0; i < 16; ++i )
11             Str[i] = (4 * (Str[i] & 3)) | ((Str[i] & 0xC) >> 2) | Str[i] & 0xF0;
12         for ( i = 0; i < 16 && Str[i] == byte_429A30[i]; ++i )
13             ;
14         if ( i == 16 )
15             printf("Congratulations! You are right!\n");
16         else
17             printf("2 Sorry, you are wrong!\n");
18         system("pause");
19         return 0;
20     }
21     else
22     {
23         printf("1 Sorry,you are wrong!\n");
24         system("pause");
25         return 0;
26     }
27 }

```

```

str1 = [0x5C, 0x65, 0x6C, 0x75, 0x78, 0x66, 0x71, 0x76, 0x46, 0x7C,
0x5C, 0x6F, 0x4C, 0x6F, 0x6F, 0x63]
result = []
for x in str1:
    trans_x = (4 * (x & 0x3)) | ((x & 0xC) >> 2) | (x & 0xF0)
    result.append(chr(trans_x))
print("".join(result))

```

SecurityIsSoCool

### test43.exe(花指令+仿射)

```

loc_4010BA:
xor     eax, eax
jz      short near ptr loc_4010BE+1

loc_4010BE:
jmp     near ptr 0C8568Ah

```

```

11
12 v11 = 5;
13 v10 = 5;
14 memset(Str, 0, sizeof(Str));
15 v8 = 0;
16 v9 = 0;
17 puts("please input a string:");
18 scanf("%s", Str);
19 v6 = strlen(Str);
20 for ( i = 0; i < v6; ++i )
21 {
22     if ( Str[i] < 97 || Str[i] > 122 )
23         return -1;
24 }
25 for ( j = 0; j < v6; ++j )
26     Str[j] = (v10 + v11 * (Str[j] - 97)) % 26 + 97;
27 if ( !strcmp(Str, Str2) )
28 {
29     puts("ok, you really know");
30     puts("By the way, can you list all the Sections of this binary file???");
31 }
32 else
33 {
34     puts("sorry");
35 }
36 return system("pause");
37 }

```

```

ciphertext_string = 'kbcwsxxsz'
a = 5
k = 5
m = 26
base_ord = 97
a_inv = pow(a, -1, m)
result_codes = []
for char in ciphertext_string:
    char_code = ord(char)
    result_codes.append((((char_code - base_ord - k) * a_inv) % m) + base_ord)

plaintext_string = "".join(chr(code) for code in result_codes)
hex_array_output = [hex(code) for code in result_codes]

print(f"字符串格式: {plaintext_string}")
print(f"Hex数组格式: {hex_array_output}")

```

buptnoone

test44.exe(MD5)

```

18 printf("Please input your flag:\n");
19 scanf("%s", Source);
20 v4 = strlen(Source);
21 for ( i = 0; i < v4; ++i )
22 {
23     if ( Source[i] < 65 || Source[i] > 90 )
24     {
25         printf("Wrong,try again!\n");
26         return -1;
27     }
28 }
29 if ( v4 >= 5 )
30 {
31     strncpy(&Destination, Source, 4u);
32     sub_401014(Str1, (int)&Destination, 4);
33     if ( !strcmp(Str1, "a3abe5e290d0bcc3c82ad572837b5d8d", 0x20u) )
34         printf("Correct!\n");
35     else
36         printf("Wrong,try again!\n");
37 }
38 else
39 {
40     printf("Wrong,try again!\n");
41 }
42 system("pause");
43 return 0;
44 }

```

MD5的轮常量表

```

2  v6 = ^a4;
3  sub_401028(&v9, v8, v7, v6, *a5, 0xD76AA478, 7);
4  sub_401028(&v6, v9, v8, v7, a5[1], 0xE8C7B756, 12);
5  sub_401028(&v7, v6, v9, v8, a5[2], 0x242070DB, 17);
6  sub_401028(&v8, v7, v6, v9, a5[3], 0xC1BDCEEE, 22);
7  sub_401028(&v9, v8, v7, v6, a5[4], -176418897, 7);
8  sub_401028(&v6, v9, v8, v7, a5[5], 1200080426, 12);
9  sub_401028(&v7, v6, v9, v8, a5[6], -1473231341, 17);
0  sub_401028(&v8, v7, v6, v9, a5[7], -45705983, 22);
1  sub_401028(&v9, v8, v7, v6, a5[8], 1770035416, 7);
2  sub_401028(&v6, v9, v8, v7, a5[9], -1958414417, 12);
3  sub_401028(&v7, v6, v9, v8, a5[10], -42063, 17);
4  sub_401028(&v8, v7, v6, v9, a5[11], -1990404162, 22);
5  sub_401028(&v9, v8, v7, v6, a5[12], 1804603682, 7);
6  sub_401028(&v6, v9, v8, v7, a5[13], -40341101, 12);

```

SEPT+任意大写字母  
长度大于等于5

test45.exe(RC4)

```

1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     signed int v4; // [esp+4Ch] [ebp-D0h]
4     signed int i; // [esp+50h] [ebp-CCh]
5     char v6[97]; // [esp+54h] [ebp-C8h] BYREF
6     __int16 v7; // [esp+B5h] [ebp-67h]
7     char v8; // [esp+B7h] [ebp-65h]
8     char Str[100]; // [esp+B8h] [ebp-64h] BYREF
9
10    memset(v6, 0, sizeof(v6));
11    v7 = 0;
12    v8 = 0;
13    puts("please input a correct string to encrypt:");
14    scanf("%s", Str);
15    v4 = strlen(Str);
16    sub_40100A(Str, (int)v6);
17    for ( i = 0; i < v4; ++i )
18    {
19        if ( v6[i] != byte_429B30[i] )
20        {
21            puts("Wrong!!!");
22            system("pause");
23            return -1;
24        }
25    }
26    puts("Great!!!");
27    puts("Can you tell me all the candidate flags???");
28    system("pause");
29    return 0;
30 }

```

加了取反

```

8 dword_42D23C = 0;
9 dword_42D240 = 0;
10 v5 = strlen(Str);
11 sub_401014();
12 for ( i = 0; i < v5; ++i )
13 {
14     v2 = Str[i];
15     *(_BYTE *)(i + a2) = ~(unsigned __int8)sub_401005() ^ v2;
16 }
17 result = i + a2;
18 *(_BYTE *)(i + a2) = 0;
19 return result;
20 }

```

```

def rc4_decrypt(ciphertext, key):
    s = list(range(256))
    j = 0
    for i in range(256):
        j = (j + s[i] + key[i % len(key)]) % 256
        s[i], s[j] = s[j], s[i]
    i, j = 0, 0
    plaintext = []

```

```

for byte in ciphertext:
    i = (i + 1) % 256
    j = (j + s[i]) % 256
    s[i], s[j] = s[j], s[i]
    plaintext.append(byte ^ ~(s[(s[i] + s[j]) % 256]) & 0xFF)
return bytes(plaintext)

ciphertext = [0x30, 0x2C, 0x02, 0xDC, 0xFB, 0x3B]
key = b'secrets'
plaintext = rc4_decrypt(ciphertext, key)
print(plaintext.decode())

```

SeeYou

## 20考题

### C++1.exe(位运算)

```

1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     char v4[20]; // [esp+50h] [ebp-2Ch]
4     char Str[20]; // [esp+64h] [ebp-18h] BYREF
5     int i; // [esp+78h] [ebp-4h]
6
7     printf("Please give me your input:\n");
8     sub_401005(Str, 18);
9     if ( strlen(Str) == 15 )
10    {
11        for ( i = 0; i < 15; ++i )
12            v4[i] = byte_427A30[14 - i] ^ Str[i];
13        for ( i = 0; i < 15 && v4[i] == byte_427A40[i]; ++i )
14            ;
15        if ( i == 15 )
16            printf("Congratulations! You are right!\n");
17        else
18            printf("2 Sorry, you are wrong!\n");
19        system("pause");
20        return 0;
21    }
22    else
23    {
24        printf("1 Sorry,you are wrong!\n");
25        system("pause");
26        return 0;
27    }
28 }

```

```

str1=[0x42, 0x61, 0x7F, 0x69, 0x5F, 0x62, 0x6C, 0x66, 0x41, 0x74,
0x6C, 0x61, 0x6D, 0x66, 0x72]
str2=[0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A,
0x0B, 0x0C, 0x0D, 0x0E, 0x0F]

str=[]
for i in range(15):
    str.append(str1[i]^str2[14-i])
str_output=''.join(chr(x) for x in str)
print("字符串:"+str_output)

```

MoreThenFriends

### C++2.exe(异常+位运算)

```

1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     char Str[28]; // [esp+54h] [ebp-28h] BYREF
4     int v5; // [esp+70h] [ebp-Ch]
5     int v6; // [esp+74h] [ebp-8h]
6     int v7; // [esp+78h] [ebp-4h]
7
8     v7 = 1;
9     v6 = 0;
10    printf("please input password : ");
11    scanf("%s", Str);
12    if ( strlen(Str) != 18 )
13    {
14        printf("1 Sorry,you are wrong!\n");
15        system("pause");
16    }
17    strncpy(Destination, Str, 0x10u);
18    v5 = v7 / v6;
19    printf("What a pity, you found a wrong way.\n");
20    system("pause");
21    return -1;
22 }

```

除0异常

打开OD，断在idiv时查看SEH链，发现执行函数401020

```

1 int sub_401020()
2 {
3     signed int v1; // [esp+4Ch] [ebp-8h]
4     signed int i; // [esp+50h] [ebp-4h]
5     signed int j; // [esp+50h] [ebp-4h]
6
7     v1 = strlen(Destination);
8     for ( i = 0; i < v1; ++i )
9         Destination[i] = (4 * (Destination[i] & 3)) | ((Destination[i] & 0xC) >> 2) | Destination[i] & 0xF0;
10    for ( j = 0; j < v1 && Destination[j] == byte_429A30[j]; ++j )
11        ;
12    if ( j == 16 )
13        printf("Congratulations! You are right!\n");
14    else
15        printf("2 Sorry, you are wrong!\n");
16    system("pause");
17    return 0;
18 }

```

```

target = [0x5C, 0x65, 0x6C, 0x75, 0x78, 0x66, 0x71, 0x76, 0x46, 0x7C,
0x50, 0x75, 0x7A, 0x7A, 0x63, 0x65]
result = []
for target_byte in target:
    for x in range(256):
        trans_x = (4 * (x & 3)) | ((x & 0xC) >> 2) | x & 0xF0
        if trans_x == target_byte:
            result.append(chr(x))
            break
print("".join(result))

```

SecurityIsPuzzle +任意2个字符

CPP3.exe(MD5)

```

11
12 memset(Str1, 0, 33);
13 memset(Source, 0, sizeof(Source));
14 v9 = 0;
15 v10 = 0;
16 Destination = 0;
17 v7 = 0;
18 printf("Please input your flag:\n");
19 scanf("%s", Source);
20 v4 = strlen(Source);
21 for ( i = 0; i < v4; ++i )
22 {
23     if ( Source[i] < 97 || Source[i] > 122 )
24     {
25         printf("Wrong,try again!\n");
26         return -1;
27     }
28 }
29 if ( v4 >= 5 )
30 {
31     strncpy(&Destination, Source, 4u);
32     sub_401014(Str1, (int)&Destination, 4);
33     if ( !strncmp(Str1, "b5c0b187fe309af0f4d35982fd961d7e", 0x20u) )
34         printf("Correct!\n");
35     else
36         printf("Wrong,try again!\n");
37 }
38 else
39 {
40     printf("Wrong,try again!\n");
41 }
42 system("pause");
43 return 0;

```

MD5爆破

TARGET\_HASH = "b5c0b187fe309af0f4d35982fd961d7e"

love +1个小写字母

CPP4.exe(仿射)



```

12
13 v12 = 9;
14 v11 = 7;
15 v10 = 3;
16 memset(Str1, 0, sizeof(Str1));
17 v8 = 0;
18 v9 = 0;
19 puts("please input a string:");
20 scanf("%s", Str1);
21 v6 = strlen(Str1);
22 for ( i = 0; i < v6; ++i )
23 {
24     if ( Str1[i] < 65 || Str1[i] > 90 )
25     {
26         printf("Sorry! Hang on!");
27         return -1;
28     }
29 }
30 for ( j = 0; j < v6; ++j )
31     Str1[j] = (v11 + v12 * (Str1[j] - 65)) % 26 + 65;
32 if ( !strcmp(Str1, Str2) )
33     puts("Ok, you know it. Just hang on.");
34 else
35     puts("Sorry! Hang on!");
36 return system("pause");
37 }

```

HANGON

CPP5.exe(DES)

```

5 char v7[12]; // [esp+6Ch] [ebp-Ch] BYREF
6
7
8 strcpy(v7, "TakeEasy");
9 puts("give me a string to encrypt:");
10 scanf("%s", Str);
11 if ( strlen(Str) == 8 )
12 {
13     sub_40100F(v7);
14     sub_401032(Str, v5);
15     for ( i = 0; i < 8; ++i )
16     {
17         if ( v5[i] != byte_42AA30[i] )
18         {
19             puts("Wrong!!");
20             system("pause");
21             return -1;
22         }
23     }
24     puts("G00d Job!!");
25     system("pause");
26     return 0;
27 }
28 else
29 {
30     system("pause");
31     return -1;
32 }
33 }

```

itiseasy

## 21考题

CPP1.exe(hook)

```

21 strcpy((char *)Buffer, "realpwd");
22 hModule = GetModuleHandleA("kernel32.dll");
23 WriteFile = (BOOL (__stdcall *))(HANDLE, LPCVOID, DWORD, LPDWORD, LPOVERLAPPED))GetProcAddress(hModule, "WriteFile");
24 lpAddress = WriteFile;
25 if ( VirtualProtect(WriteFile, 5u, 0x40u, &f10ldProtect) )
26 {
27     memcpy(&unk_42DC8C, lpAddress, 5u);
28     Src = (char *)sub_40100A - (char *)WriteFile - 5;
29     memcpy(&v10[1], &Src, 4u);
30     memcpy(WriteFile, v10, 5u);
31     VirtualProtect(WriteFile, 5u, f10ldProtect, &f10ldProtect);
32 }
33 hFile = CreateFileA("pwd.txt", 0x10000000u, 0, 0, 2u, 0x80u, 0);
34 v3 = strlen((const char *)Buffer);
35 ::WriteFile(hFile, Buffer, v3, (LPDWORD)&Buffer[2], 0);
36 CloseHandle(hFile);
37 Stream = fopen("pwd.txt", "r");
38 if ( Stream )
39     printf("File open success\n");
40 else
41     printf("File open fail\n");
42 if ( Stream )
43     fscanf(Stream, "%s", String2);
44 else
45     printf("scan fail\n");
46 if ( Stream )
47     fclose(Stream);
48 if ( !strcmpA((LPCSTR)Buffer, String2) )
49     printf("try again!\n");
50 else
51     printf("congratulations!\n");
52 system("Pause");
53 return 0;
54 }

1 int __thiscall sub_401100(void *this, void *a2, int a3, int a4, DWORD *a5, struct _OVERLAPPED *a6)
2 {
3     HMODULE ModuleHandleA; // eax
4     DWORD v7; // eax
5     int i; // [esp+4Ch] [ebp-20h]
6     char Str[8]; // [esp+50h] [ebp-1Ch] BYREF
7     char v11[16]; // [esp+58h] [ebp-14h] BYREF
8     BOOL (__stdcall *WriteFile)(HANDLE, LPCVOID, DWORD, LPDWORD, LPOVERLAPPED); // [esp+68h] [ebp-4h]
9
10    Str[0] = 26;
11    Str[1] = 10;
12    Str[2] = 14;
13    Str[3] = 7;
14    Str[4] = 17;
15    Str[5] = 7;
16    Str[6] = 13;
17    Str[7] = 0;
18    sub_401005(this);
19    printf("Please input your flag \n");
20    scanf("%s", v11);
21    for ( i = 0; i < 7; ++i )
22        Str[i] ^= v11[i];
23    ModuleHandleA = GetModuleHandleA("kernel32.dll");
24    WriteFile = (BOOL (__stdcall *))(HANDLE, LPCVOID, DWORD, LPDWORD, LPOVERLAPPED))GetProcAddress(
25                                                ModuleHandleA,
26                                                "WriteFile");
27    v7 = strlen(Str);
28    WriteFile(a2, Str, v7, a5, a6);
29    return 1;
30 }

```

```

40     else
41         printf("File open fail\n");
42     if ( Stream )
43         fscanf(Stream, "%s", String2);
44     else
45         printf("scan fail\n");
46     if ( Stream )
47         fclose(Stream);
48     if ( lstrcmpA((LPCSTR)Buffer, String2) )
49         printf("try again!\n");
50     else
51         printf("congratulations!\n");
52     system("Pause");
53     return 0;

```

```

019FEEC db 72h ; r
019FEED db 65h ; e
019FEEE db 61h ; a
019FEEF db 6Ch ; l
019FEF0 db 70h ; p
019FEF1 db 77h ; w
019FEF2 db 64h ; d
019FEF3 db 00h ; 

```

sb了，开头就有

```

*(_DWORD *)&v10[2] = 0;
strcpy((char *)Buffer, "realpwd");
hModule = GetModuleHandleA("kernel32.d
WriteFile = (BOOL (__stdcall *))(HANDLE
lpAddress = WriteFile;

```

hookapi

CPP3.exe(花指令+仿射)

```

loc_4010BE:                                     ; CODE
xor     eax, eax
jz      short near ptr loc_4010BE+1

loc_4010BE:                                     ; CODE
jmp     near ptr 0C8568Ah

```

```

--
12  v8 = 5;
13  v7 = 5;
14  memset(Str, 0, sizeof(Str));
15  v5 = 0;
16  v6 = 0;
17  puts("please input a string:");
18  scanf("%s", Str);
19  v3 = strlen(Str);
20  for ( i = 0; i < v3; ++i )
21  {
22      if ( Str[i] < 97 || Str[i] > 122 )
23          return -1;
24  }
25  for ( j = 0; j < v3; ++j )
26      Str[j] = (v7 + v8 * (Str[j] - 97)) % 26 + 97;
27  if ( !strcmp(Str, Str2) )
28      puts("ok, you really know");
29  else
30      puts("sorry");
31  return system("pause");
32 }

```

greatvalue

CPP4.exe(MD5)

```

12  memset(Str1, 0, 33);
13  memset(Source, 0, sizeof(Source));
14  v9 = 0;
15  v10 = 0;
16  Destination = 0;
17  v7 = 0;
18  printf("Please input your flag:\n");
19  scanf("%s", Source);
20  v5 = strlen(Source);
21  for ( i = 0; i < v5; ++i )
22  {
23      if ( Source[i] < 48 || Source[i] > 57 )
24          return -1;
25  }
26  if ( v5 >= 5 )
27  {
28      strncpy(&Destination, Source, 4u);
29      sub_401014(Str1, (int)&Destination, 4);
30      if ( !strcmp(Str1, "eb62f6b9306db575c2d596b1279627a4", 0x20u) )
31          printf("Correct!\n");
32      else
33          printf("Wrong,try again!\n");
34  }
35  else
36  {
37      printf("Wrong,try again!\n");
38  }
39  system("pause");
40  return 0;
41 }

```

0123 +任意数字

CPP5.exe

```

9
10 memset(v6, 0, sizeof(v6));
11 v7 = 0;
12 v8 = 0;
13 puts("please input a correct string to encrypt:");
14 scanf("%s", Str);
15 v4 = strlen(Str);
16 sub_40100A(Str, (int)v6);
17 for ( i = 0; i < v4; ++i )
18 {
19     if ( v6[i] != byte_429B30[i] )
20     {
21         puts("Wrong!!!");
22         system("pause");
23         return -1;
24     }
25 }
26 puts("Great!!!");
27 system("pause");
28 return 0;
29 }

9 dword_42D240 = 0;
10 v5 = strlen(Str);
11 sub_401014();
12 for ( i = 0; i < v5; ++i )
13 {
14     v2 = Str[i];
15     *(_BYTE *)(i + a2) = ~(unsigned __int8)sub_401005() ^ v2;
16 }
17 result = i + a2;
18 *(_BYTE *)(i + a2) = 0;
19 return result;
20 }

```

seeyounextyear

## 助考习题

Q2.exe(xor)

```

/
8  strcpy(v6, "@URU@KE");
9  v5[0] = 2;
10 v5[1] = 0;
11 v5[2] = 2;
12 v5[3] = 1;
13 v5[4] = 6;
14 v5[5] = 1;
15 v5[6] = 22;
16 v5[7] = 0;
17 printf("Plase give me your answer:\n");
18 scanf("%s", Str);
19 if ( strlen(Str) == 7 )
20 {
21     for ( i = 0; i < 7; ++i )
22         Str[i] ^= v5[i];
23     for ( i = 0; i < 7 && Str[i] == v6[i]; ++i )
24         ;
25     if ( i == 7 )
26         printf("Congratulations! You are right!\n");
27     else
28         printf("Sorry, you are wrong!\n");
29     system("pause");
30     return 0;
31 }
32 else
33 {
34     printf("Sorry,you are wrong!\n");
35     system("pause");
36     return 0;
37 }
38 }

```

BUPTFJS

Q3.exe(花指令)



```
loc_4010BA:                                ; C
xor     eax, eax
jz      short near ptr loc_4010BE+1
```

```
loc_4010BE:                                ; C
jmp     near ptr 0C8568Ah
```

```
10  int v8, // [esp+00h] [ebp-4h]
11
12  v8 = 5;
13  v7 = 5;
14  memset(Str, 0, sizeof(Str));
15  v5 = 0;
16  v6 = 0;
17  puts("please input a string:");
18  scanf("%s", Str);
19  v3 = strlen(Str);
20  for ( i = 0; i < v3; ++i )
21  {
22      if ( Str[i] < 97 || Str[i] > 122 )
23          return -1;
24  }
25  for ( j = 0; j < v3; ++j )
26      Str[j] = (v7 + v8 * (Str[j] - 97)) % 26 + 97;
27  if ( !strcmp(Str, Str2) )
28      puts("ok, you really know");
29  else
30      puts("sorry");
31  return system("pause");
32 }
```