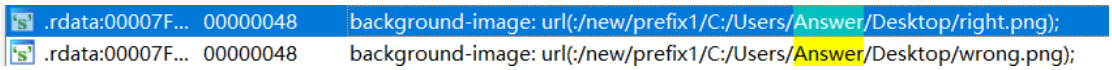
Answer's Windows

ida 中打开

搜索字符串 answer 找到 wright 和 wrong 的 png 图片地址



对输入进行加密的伪代码如下

```
do
{
  v18 = a1;
  if ( v14 >= 0x10 )
    v18 = *a1;
  v19 = &datattt;
  if ( v12 >= 0x10 )
    v19 = v13;
  *(v16 - 2) = *(v19 + (v18[k] >> 2));
  if ( v14 < 0x10 )
  {
    rcx1 = a1 + k;
    input1 = a1;
  }
  else
  {
    input1 = *a1;
    rcx1 = (k + *a1);
  }
  sstr1 = &datattt;
  if ( v12 >= 0x10 )
    sstr1 = v13;
  *(v16 - 1) = *(sstr1 + ((*(input1 + k + 1) >> 4) | (16i64 * (*rcx1 & 3))));
  if ( v14 < 0x10 )
  {
    v24 = a1 + 1;
    input2 = a1;
  }
  else
  {
    input2 = *a1;
    v24 = *a1 + 1;
  }
```

```
    rcx2 = (k + v24);
    sstr2 = &datattt;
    if ( v12 >= 0x10 )
      sstr2 = v13;
    *v16 = *(sstr2 + ((*(input2 + k + 2) >> 6) | (4i64 * (*rcx2 & 0xF))));
    input3 = a1;
    if ( v14 >= 0x10 )
      input3 = *a1;
    sstr3 = &datattt;
    if ( v12 >= 0x10 )
      sstr3 = v13;
    v16[1] = *(sstr3 + (*(input3 + k + 2) & 0x3F));
    k += 3i64;
    v16 += 4;
    --i;
  }
  while ( i );
```

加密后的字符串并不存在'\,.-@等符号

在字符串中寻找合适的字符串集合，编写如下代码

```
#include<stdio.h>
int main()
{
    char code[58] = ";'>B<76\\=82@-8.@=T\"@-7ZU:8*F=X2J<G>@=W^@-8.@9D2T:49U@1aa";
    char s[100] =
"!\"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\\]^_`abcdefghijklmnopqrst
uvwxyz{|}~";

    int i, a, b, c, out[4];
    i = 0;
    for (i = 0; i < 14; i++)
    {
        for (a = 0; a <= 127; a++)
        {
            for (b = 0; b <= 127; b++)
            {
                for (c = 0; c <= 127; c++)
                {
                    {
                    out[0] = s[a >> 2];
                    out[1] = s[(b >> 4) | (16 * (a & 3))];
                    out[2] = s[(c >> 6) | (4 * (b & 0xF))];
                    out[3] = s[c & 0x3F];
                    if (out[0] == code[4 * i] && out[1] == code[4 * i+1] && out[2] ==
code[4 * i+2] && out[3] == code[4 * i+3])
                    {
                        printf("%c%c%c", a, b, c);
                    }
```

```
                    }
                }
            }
        }
    }
    return 0;
}
```

```
hgame{qt_1s_s0_1nteresting_so_1s_b4se64
```

在最后补上},
得到 hgame{qt_1s_s0_1nteresting_so_1s_b4se64}

creakme3
打开 ida
查看汇编

```
loc_10000648:
lwz         r9, 0x20(r31)
slwi        r9, r9, 2
addi        r10, r31, 0x190
add         r9, r10, r9
addi        r9, r9, -0x168
lwz         r9, 0(r9)       # r9的第1个word
lis         r10, a@ha
addi        r10, r10, a@l
slwi        r9, r9, 3
add         r9, r10, r9
addi        r9, r9, 4
lwz         r10, 0(r9)      # r31的第？个dw
lwz         r9, 0x20(r31)
addi        r9, r9, -1      # 前四位
slwi        r9, r9, 2       # 前四位
addi        r8, r31, 0x190
add         r9, r8, r9
addi        r9, r9, -0x168
lwz         r9, 0(r9)       # r9的第0个word
lis         r8, a@ha
addi        r8, r8, a@l
slwi        r9, r9, 3
add         r9, r8, r9
addi        r9, r9, 4
lwz         r9, 0(r9)
cmpw        r10, r9
blt         loc_100006D0
```

```
loc_100006F4:
lwz         r9, 0x24(r31)
slwi        r9, r9, 2
addi        r10, r31, 0x190
add         r9, r10, r9
addi        r9, r9, -0x168
lwz         r9, 0(r9)
lis         r10, a@ha
addi        r10, r10, a@l
slwi        r9, r9, 3
add         r9, r10, r9
lwz         r9, 0(r9)
mr          r3, r9          # c
bl          putchar
lwz         r9, 0x24(r31)
addi        r9, r9, 1
stw         r9, 0x24(r31)
```

将 data 中的数据按从小到大的顺序进行比较，并输出该数据前面的 WORD

编写 c

```
#include<stdio.h>
#include<math.h>
int main()
{
    int a[800] =
{ 0,0,0,0x30,0,0,0x4E,0x7D,0,0,0,0x30,0,0,0x67,0xBD,0,0,0,0x30,0,0,0x7A,0x48,0,0,0,0x30
,0,0,0x82,0xA2,0,0,0,0x30,0,0,0x93,0x3E,0,0,0,0x31,0,0,0x9C,0x18,0,0,0,0x32,0,0,0x5A,0x
FF,0,0,0,0x32,0,0,0x6C,0xD7,0,0,0,0x32,0,0,0xA6,0xCA,0,0,0,0x32,0,0,0xBD,0x79,0,0,0,0x3
2,0,0,0xCE,0xBD,0,0,0,0x33,0,0,0x32,0x4A,0,0,0,0x33,0,0,0x32,0x92,0,0,0,0x33,0,0,0x39,5
,0,0,0,0x33,0,0,0x42,0x91,0,0,0,0x33,0,0,0x5A,0xDE,0,0,0,0x33,0,0,0x6E,0x9F,0,0,0,0x33,
0,0,0xA5,0x2A,0,0,0,0x33,0,0,0xBE,0x35,0,0,0,0x33,0,0,0xCB,0x63,0,0,0,0x35,0,0,0x7F,0x3
B,0,0,0,0x38,0,0,0x39,0x14,0,0,0,0x38,0,0,0xB2,0xAD,0,0,0,0x39,0,0,0x38,0xDA,0,0,0,0x39
,0,0,0x4E,0x50,0,0,0,0x39,0,0,0x6A,2,0,0,0,0x39,0,0,0xB1,0xF,0,0,0,0x42,0,0,0x78,0xE5,0
,0,0,0x5F,0,0,0x7E,0xF6,0,0,0,0x5F,0,0,0x89,0xA3,0,0,0,0x5F,0,0,0x8E,0xBD,0,0,0,0x5F,0,
0,0x95,0xE3,0,0,0,0x61,0,0,0x73,0xDA,0,0,0,0x64,0,0,0x53,0x8C,0,0,0,0x64,0,0,0x63,0x3B,
0,0,0,0x64,0,0,0x9E,0x9C,0,0,0,0x64,0,0,0xB7,0x8B,0,0,0,0x64,0,0,0xC8,0x66,0,0,0,0x65,0
,0,0x32,0xAE,0,0,0,0x65,0,0,0x76,0x79,0,0,0,0x66,0,0,0x2A,0xE7,0,0,0,0x66,0,0,0x4D,0x6A
,0,0,0,0x66,0,0,0x57,8,0,0,0,0x66,0,0,0x66,0x10,0,0,0,0x66,0,0,0xA2,0x58,0,0,0,0x66,0,0
,0xB8,0xC,0,0,0,0x66,0,0,0xC8,0x85,0,0,0,0x67,0,0,0x71,0xA,0,0,0,0x67,0,0,0x7C,0xF4,0,0
,0,0x68,0,0,0x3F,0x76,0,0,0,0x68,0,0,0x70,0x2B,0,0,0,0x68,0,0,0xA3,0xEE,0,0,0,0x68,0,0,
0xAD,0x50,0,0,0,0x68,0,0,0xBA,0xC7,0,0,0,0x69,0,0,0x40,0x24,0,0,0,0x69,0,0,0x8A,0x22,0,
0,0,0x69,0,0,0xC0,0x55,0,0,0,0x6A,0,0,0x2B,0x52,0,0,0,0x6A,0,0,0xC6,0x87,0,0,0,0x6B,0,0
,0x5F,0,0,0,0,0x6B,0,0,0xC4,0x17,0,0,0,0x6C,0,0,0x61,0x82,0,0,0,0x6D,0,0,0x75,0xDB,0,0,
0,0x6E,0,0,0x3C,0x61,0,0,0,0x6E,0,0,0x49,0x96,0,0,0,0x6E,0,0,0x5D,0xC1,0,0,0,0x6F,0,0,0
x2D,0x76,0,0,0,0x6F,0,0,0x7D,0x17,0,0,0,0x6F,0,0,0xA9,0x1B,0,0,0,0x70,0,0,0x9A,0xED,0,0
```

```
, 0, 0x72, 0, 0, 0x45, 0xD0, 0, 0, 0, 0x72, 0, 0, 0x84, 0x67, 0, 0, 0, 0x72, 0, 0, 0xAB, 0x5D, 0, 0, 0, 0x73, 0, 0,
0x50, 0x83, 0, 0, 0, 0x73, 0, 0, 0x62, 0x22, 0, 0, 0, 0x73, 0, 0, 0x8D, 0x93, 0, 0, 0, 0x73, 0, 0, 0x92, 0x3A, 0,
0, 0, 0x73, 0, 0, 0x97, 0x1E, 0, 0, 0, 0x73, 0, 0, 0xB4, 0xBA, 0, 0, 0, 0x73, 0, 0, 0xC7, 0x85, 0, 0, 0, 0x74, 0, 0
, 0x35, 0x58, 0, 0, 0, 0x74, 0, 0, 0x86, 0xBD, 0, 0, 0, 0x74, 0, 0, 0x97, 0x38, 0, 0, 0, 0x75, 0, 0, 0x37, 0x10, 0
, 0, 0, 0x75, 0, 0, 0x97, 0x79, 0, 0, 0, 0x77, 0, 0, 0x2F, 0x3F, 0, 0, 0, 0x77, 0, 0, 0x44, 0xDD, 0, 0, 0, 0x7B, 0,
0, 0x78, 0xE1, 0, 0, 0, 0x7D, 0, 0, 0x9F, 0x42 };
    int b[0x59], c[0x59];
    int i, j;
    int k;
    for (i = 0; i <= 0x58; i++)
    {
        c[i] = i;
        k = 8 * i + 6;
        b[c[i]] = (a[k] * 0x100) + a[k + 1];
    }
    for (i = 0; i <= 0x58; i++)
    {
        for (j = i; j <= 0x58; j++)
        {
            if (b[c[i]] >= b[c[j]])
            {
                k = c[i], c[i] = c[j], c[j] = k;
            }
        }
    }
    for (i = 0; i <= 0x58; i++)
    {
        printf("%c", a[8 * (c[i]) + 3]);
    }

    return 0;
}
fjow33etu938nhi3wrnf90sdf32nklsdf0923hgame{B0go_50rt_is_s0_stup1d}fh32orh98sdfh23ikjsdf32
```

得到 flag

hgame{B0go_50rt_is_s0_stup1d}