

[原创]看雪.纽盾 KCTF晋级赛2019 Q2 第六题 消失的岛屿 优

梦游枪手

中级



举报

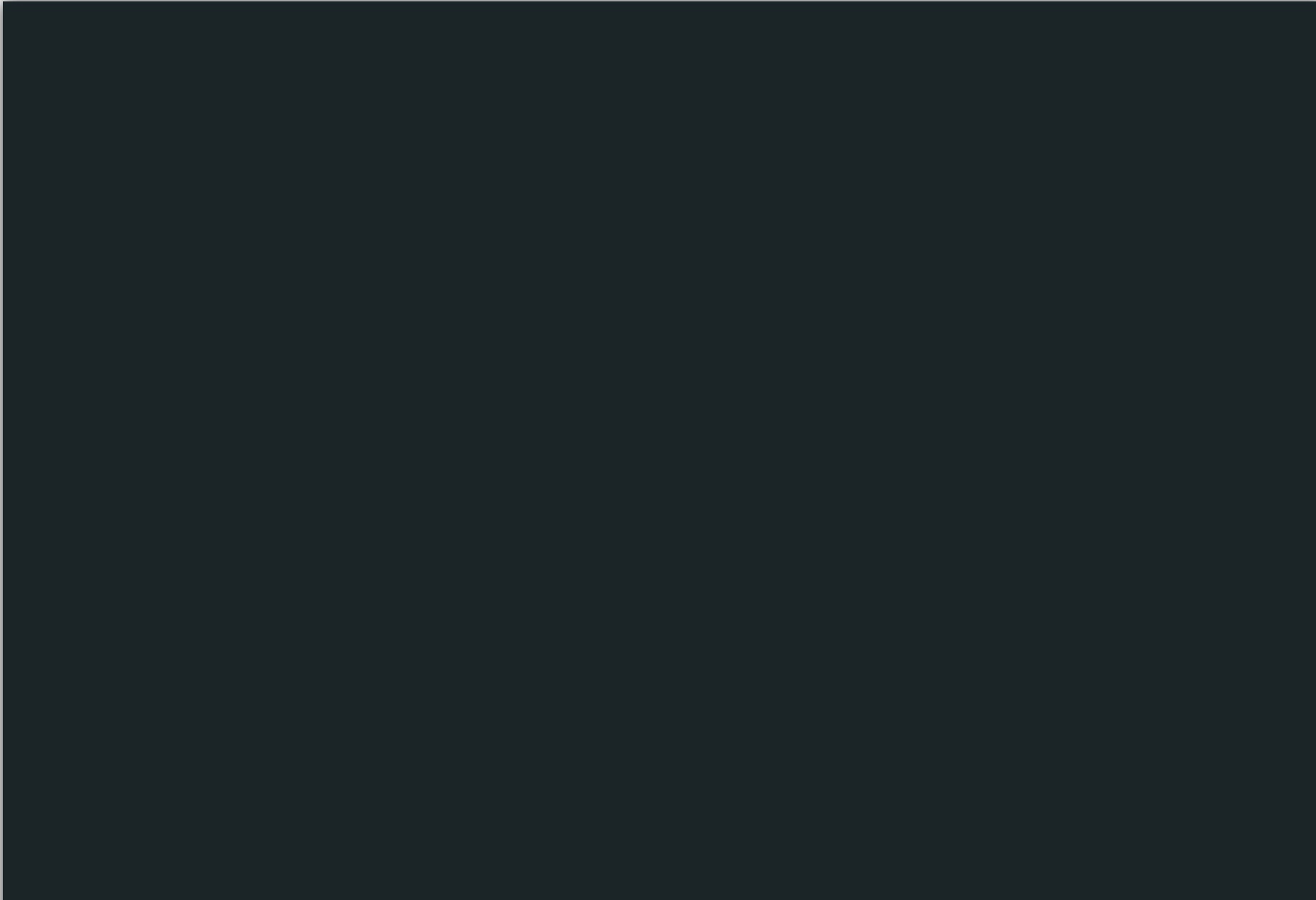
2019-6-18 14:37

328

用IDA载入文件，提示有DWARF debug information，选择yes，可以看到更多的信息。然后我们找到main函数。

```
1  int __cdecl main(int argc, const char **argv, const char **envp)
2  {
3      int v3; // eax
4      uint8_t bindata; // [esp+11h] [ebp-3Fh]
5      const char *v6; // [esp+48h] [ebp-8h]
6      char *v7; // [esp+4Ch] [ebp-4h]
7
8      __main();
9      printf("please enter Serial:");
10     scanf(" %s", &bindata);
11     if ( strlen((const char *)&bindata) > 0x31 )
12         puts("error");
13     v7 = (char *)calloc(1u, 0x400u);
14     v3 = strlen((const char *)&bindata);
15     base64_encode(&bindata, v7, v3);
16     v6 = "!NGV%,$h1f4S3%2P(hkQ94==";
17     if ( !strcmp("!NGV%,$h1f4S3%2P(hkQ94==", v7) )
18         puts("Success");
19     else
20         puts("Please Try Again");
21     free(v7);
22     system("pause");
23     return 0;
24 }
```

这几乎是源代码了，而且代码逻辑也很简单，输入的字符串base64编码以后跟"!NGV%,\$h1f4S3%2P(hkQ94=="比较，相同则成功。但是这个看起来不是标准base64编码，进入base64_encode函数看看。



```
1  int __cdecl base64_encode(const uint8_t *bindata, char *base64, int binlength)
2  {
3      int v3; // eax
4      char *v4; // ebx
5      int v5; // eax
6      int v6; // ST0C_4
7      char *v7; // ebx
8      int v8; // eax
9      int v9; // eax
10     char *v10; // ebx
11     int v11; // eax
12     int v12; // eax
13     char *v13; // ebx
14     uint8_t current; // [esp+Bh] [ebp-Dh]
15     uint8_t currenta; // [esp+Bh] [ebp-Dh]
16     int j; // [esp+Ch] [ebp-Ch]
17     int ja; // [esp+Ch] [ebp-Ch]
18     int jb; // [esp+Ch] [ebp-Ch]
19     int i; // [esp+10h] [ebp-8h]
20
21     i = 0;
22     j = 0;
23     while ( i < binlength )
24     {
25         v3 = j;
26         ja = j + 1;
27         v4 = &base64[v3];
28         *v4 = charEncrypt((bindata[i] >> 2) & 0x3F);
29         current = 16 * bindata[i] & 0x30;
30         if ( i + 1 >= binlength )
31         {
32             v5 = ja;
33             v6 = ja + 1;
34             v7 = &base64[v5];
35             *v7 = charEncrypt(current);
36             base64[v6] = 61;
37             v8 = v6 + 1;
38             j = v6 + 2;
39             base64[v8] = 61;
40             break;
41         }
42         v9 = ja;
43         jb = ja + 1;
44         v10 = &base64[v9];
45         *v10 = charEncrypt((bindata[i + 1] >> 4) | current);
46         currenta = 4 * bindata[i + 1] & 0x3C;
47         if ( i + 2 >= binlength )
48         {
49             base64[jb] = charEncrypt(currenta);
50             v11 = jb + 1;
51             j = jb + 2;
52             base64[v11] = '=';
53             break;
54         }
55         base64[jb] = charEncrypt((bindata[i + 2] >> 6) | currenta);
56         v12 = jb + 1;
57         j = jb + 2;
58         v13 = &base64[v12];
59         *v13 = charEncrypt(bindata[i + 2] & 0x3F);
60         i += 3;
61     }
62     base64[j] = 0;
63     return j;
64 }
```

charEncrypt函数

```
1  char __cdecl charEncrypt(int data)
2  {
3      int dataa; // [esp+18h] [ebp+8h]
4
5      dataa = aTuvwxTuImnopqr[data];
6      if ( dataa > '@' && dataa <= 'Z' )
7          return 0x9B - dataa;
8      if ( dataa > '`' && dataa <= 'z' )
9          return dataa - 0x40;
10     if ( dataa > '/' && dataa <= '9' )
11         return dataa + '2';
12     if ( dataa == '+' )
13         return 'w';
14     if ( dataa == '/' )
15         dataa = 'y';
16     return dataa;
17 }
```

我们可以通过 charEncrypt函数得到真实的码表，把代码复制一份，做点小修改

```
1  #include <stdio.h>
2  char b64_chr[] = "tuvwxTUlmnopqrs7YZabcdefghijklmnopqrstuvwxyz0123456VWXkABCDEFGHJKLMNOPS9+/";
3  char charEncrypt(int data)
4  {
5      int dataa; // [esp+18h] [ebp+8h]
6
7      dataa = data;
8      if ( dataa > '@' && dataa <= 'Z' )
9          return 0x9B - dataa;
10     if ( dataa > '`' && dataa <= 'z' )
11         return dataa - 0x40;
12     if ( dataa > '/' && dataa <= '9' )
13         return dataa + '2';
14     if ( dataa == '+' )
15         return 'w';
16     if ( dataa == '/' )
17         dataa = 'y';
18     return dataa;
19 }
20 int main(int argc, char const *argv[])
21 {
22     for (int i = 0; i < sizeof(b64_chr); ++i)
23     {
24         printf("%c",charEncrypt(b64_chr[i]));
25     }
26     return 0;
27 }
```

运行得到"45678GF,-./0123iBA!\ "\$%&'()*j9:bcdefghEDC+ZYXWVUTSRQPONMLKJIHkwy", 这个就是真正的码表。
用这个码表解密" !NGV%,\$h1f4S3%2P(hkQ94==", 得到key: KanXue2019ctf_st

[公告]看雪.纽盾 KCTF 2019晋级赛Q3攻击方规则，9月10日开赛，华为P30 Pro、iPad、kindle等你来拿！

0

☆ 收藏

0

👍 赞


¥

打赏

🔗

分享

最新回复 (0)



勇士小蓝

内容

高级回复

回帖

表情

返回