【BUUCTF】SimpleRev WriteUp

原创

古月浪子 于 2020-07-20 11:03:05 发布 459 收藏

分类专栏： BUUCTF - RE 文章标签： CTF

本文链接：https://blog.csdn.net/tqydyqt/article/details/107459090

buuoj上的一道稍微有点难度的逆向题

用IDA快速定位到main函数

```
int __cdecl __noreturn main(int argc, const char **argv, const char **envp)
{
  int v3; // eax
  char v4; // [rsp+Fh] [rbp-1h]

  while ( 1 )
  {
    while ( 1 )
    {
      printf("Welcome to CTF game!\nPlease input d/D to start or input q/Q to quit this program: ", argv, envp);
      v4 = getchar();
      if ( v4 != 'd' && v4 != 'D' )
        break;
      Decry();
    }
    if ( v4 == 113 || v4 == 81 )
      Exit();
    puts("Input fault format!");
    v3 = getchar();
    putchar(v3);
  }
}
```

可以看到关键逻辑应该在Decry函数里，跟进去看看

```
unsigned __int64 Decry()
{
  char v1; // [rsp+Fh] [rbp-51h]
  int v2; // [rsp+10h] [rbp-50h]
  int v3; // [rsp+14h] [rbp-4Ch]
  int i; // [rsp+18h] [rbp-48h]
  int v5; // [rsp+1Ch] [rbp-44h]
  char src[8]; // [rsp+20h] [rbp-40h]
  __int64 v7; // [rsp+28h] [rbp-38h]
  int v8; // [rsp+30h] [rbp-30h]
  __int64 v9; // [rsp+40h] [rbp-20h]
  __int64 v10; // [rsp+48h] [rbp-18h]
  int v11; // [rsp+50h] [rbp-10h]
  unsigned __int64 v12; // [rsp+58h] [rbp-8h]

  v12 = __readfsqword(0x28u);
  *src = 'SLCDN';                               // NDCLS
  v7 = 0LL;
  v8 = 0;
  v9 = 'wodah';                                 // hadow
  v10 = 0LL;
  v11 = 0;
  text = join(key3, &v9);                       // text = killshadow
  strcpy(key, key1);
  strcat(key, src);                             // key = ADSFKNDCLS
  v2 = 0;
  v3 = 0;
```

```
28  getchar();
29  v5 = strlen(key);
30  for ( i = 0; i < v5; ++i )
31  {
32    if ( key[v3 % v5] > 64 && key[v3 % v5] <= 90 )
33      key[i] = key[v3 % v5] + 32;
34    ++v3;
35  }                                             // key = adsfkndcls
36  printf("Please input your flag:", src);
37  while ( 1 )
38  {
39    v1 = getchar();
40    if ( v1 == 10 )
41      break;
42    if ( v1 == 32 )
43    {
44      ++v2;
45    }
46    else
47    {
48      if ( v1 <= 96 || v1 > 122 )
49      {
50        if ( v1 > 64 && v1 <= 90 )
51          str2[v2] = (v1 - 39 - key[v3++ % v5] + 97) % 26 + 97;
```

```
52      }
53      else
54      {
55        str2[v2] = (v1 - 39 - key[v3++ % v5] + 97) % 26 + 97;
56      }
57      if ( !(v3 % v5) )
58        putchar(32);
59      ++v2;
60    }
61  }
62  if ( !strcmp(text, str2) )
63    puts("Congratulation!\n");
64  else
65    puts("Try again!\n");
66  return __readfsqword(0x28u) ^ v12;
67}
```

这里要注意的是，IDA里面按R将整数转成字符串时，是反着的…
可以看到，程序是逐位进行运算然后和text比对，因此可以写爆破脚本

```cpp
#include <iostream>

using namespace std;

int main()
{
 char key[] = "adsfkndcls";
 char text[] = "killshadow";
 int v3 = 10;
 for (int i = 0; i < 10; i++)
 {
  for (int j = 0; j < 128; j++)
  {
   if (j < 'A' || j > 'z' || j > 'Z' && j < 'a')
   {
    continue;
   }
   if ((j - 39 - key[v3 % 10] + 97) % 26 + 97 == text[i])
   {
    cout << char(j);
    v3++;
    break;
   }
  }
 }
}
```

直接跑出结果~