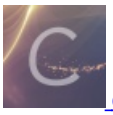
攻防世界Reverse进阶区-dmd-50-writeup

原创

y4ung 于 2020-09-19 12:17:48 发布 379 收藏 1

分类专栏： ctf 文章标签： ctf



1. 介绍

本题是xctf攻防世界中Reverse的进阶区的题dmd-50

题目来源： suctf-2016

2. 分析

```
$  file 4907915cc47e4b5bb02bbde6c445c924
4907915cc47e4b5bb02bbde6c445c924: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, inter
preter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 2.6.24, BuildID[sha1]=2643fecd383362fe9593ef8605a9ce882a85a38a
, not stripped

$ chmod +x 4907915cc47e4b5bb02bbde6c445c924

$  ./4907915cc47e4b5bb02bbde6c445c924
Enter the valid key!
123
Invalid Key! :(
```

扔进IDA里分析。`Enter the valid key!` 是在main函数中被引用。
第47行的代码内容，IDA里分析出的伪代码有个&edata，不知道是啥。。然后我又放到ghidra里去分析，得到这里的&edata是&std:cin，也就是读取用户输入，保存到v42,。接下来创建了string 对象v39，然后进行md5运算，赋值给v40，经由v40创建string对象v41。也就是v41是用户输入的md5值。

```
44
45  v43 = __readfsqword(0x28u);
46  std::operator<<<std::char_traits<char>>(&std::cout, "Enter the valid key!\n", envp);
47  std::operator>><char,std::char_traits<char>>(&edata, &v42);// 根据ghidra的分析结果，这里的&edata是&std:cin，读取用户的输入，保存到变量v42
48  std::allocator<char>::allocator(&v38);
49  std::string::string(&v39, &v42, &v38);
50  md5(&v40, &v39);
51  v41 = (_BYTE *)std::string::c_str((std::string *)&v40);
52  std::string::~string((std::string *)&v40);
53  std::string::~string((std::string *)&v39);
54  std::allocator<char>::~allocator(&v38);
```

继续往下看，在下面的代码中，可以看出，只有if条件不满足时，才会输出`The key is valid :)`。 if条件不满足，即用户输入的md5值必须等于`780438d5b6e29db0898bc4f0225935c0` 。经过MD5解码得到flag为： `b781cbb29054db12f88f08c6e161c199`

```
if ( *v41 != '7'
    || v41[1] != '8'
    || v41[2] != '0'
```

```
  || v41[3] != '4'
  || v41[4] != '3'
  || v41[5] != '8'
  || v41[6] != 'd'
  || v41[7] != '5'
  || v41[8] != 'b'
  || v41[9] != '6'
  || v41[10] != 'e'
  || v41[11] != '2'
  || v41[12] != '9'
  || v41[13] != 'd'
  || v41[14] != 'b'
  || v41[15] != '0'
  || v41[16] != '8'
  || v41[17] != '9'
  || v41[18] != '8'
  || v41[19] != 'b'
  || v41[20] != 'c'
  || v41[21] != '4'
  || v41[22] != 'f'
  || v41[23] != '0'
  || v41[24] != '2'
  || v41[25] != '2'
  || v41[26] != '5'
  || v41[27] != '9'
  || v41[28] != '3'
  || v41[29] != '5'
  || v41[30] != 'c'
  || v41[31] != '0' )
{
  v23 = std::operator<<<std::char_traits<char>>(&std::cout, 'I');
  v24 = std::operator<<<std::char_traits<char>>(v23, 'n');
  v25 = std::operator<<<std::char_traits<char>>(v24, 'v');
  v26 = std::operator<<<std::char_traits<char>>(v25, 'a');
  v27 = std::operator<<<std::char_traits<char>>(v26, 'l');
  v28 = std::operator<<<std::char_traits<char>>(v27, 'i');
  v29 = std::operator<<<std::char_traits<char>>(v28, 'd');
  v30 = std::operator<<<std::char_traits<char>>(v29, ' ');
  v31 = std::operator<<<std::char_traits<char>>(v30, 'K');
  v32 = std::operator<<<std::char_traits<char>>(v31, 'e');
  v33 = std::operator<<<std::char_traits<char>>(v32, 'y');
  v34 = std::operator<<<std::char_traits<char>>(v33, '!');
  v35 = std::operator<<<std::char_traits<char>>(v34, ' ');
  v36 = std::operator<<<std::char_traits<char>>(v35, ':');
  v37 = std::operator<<<std::char_traits<char>>(v36, '(');
  std::ostream::operator<<(v37, &std::endl<char,std::char_traits<char>>);
  result = 0;
}
else
{
  v3 = std::operator<<<std::char_traits<char>>(&std::cout, 'T');
  v4 = std::operator<<<std::char_traits<char>>(v3, 'h');
  v5 = std::operator<<<std::char_traits<char>>(v4, 'e');
  v6 = std::operator<<<std::char_traits<char>>(v5, ' ');
  v7 = std::operator<<<std::char_traits<char>>(v6, 'k');
  v8 = std::operator<<<std::char_traits<char>>(v7, 'e');
  v9 = std::operator<<<std::char_traits<char>>(v8, 'y');
  v10 = std::operator<<<std::char_traits<char>>(v9, ' ');
  v11 = std::operator<<<std::char_traits<char>>(v10, 'i');
  v12 = std::operator<<<std::char_traits<char>>(v11, 's');
```

```
    v12 = std::operator<<<std::char_traits<char>>(v11, 's');
    v13 = std::operator<<<std::char_traits<char>>(v12, ' ');
    v14 = std::operator<<<std::char_traits<char>>(v13, 'v');
    v15 = std::operator<<<std::char_traits<char>>(v14, 'a');
    v16 = std::operator<<<std::char_traits<char>>(v15, 'l');
    v17 = std::operator<<<std::char_traits<char>>(v16, 'i');
    v18 = std::operator<<<std::char_traits<char>>(v17, 'd');
    v19 = std::operator<<<std::char_traits<char>>(v18, ' ');
    v20 = std::operator<<<std::char_traits<char>>(v19, ':');
    v21 = std::operator<<<std::char_traits<char>>(v20, ')');
    std::ostream::operator<<(v21, &std::endl<char,std::char_traits<char>>);
    result = 0;
  }
  return result;
}
```