

# BUUCTF\_CrackRTF

原创

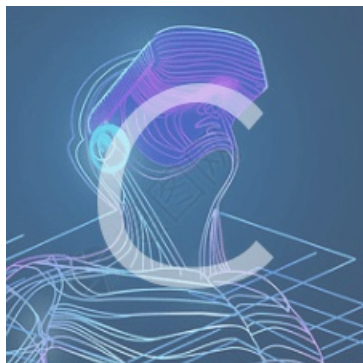
ZYen12138 于 2020-10-26 00:16:56 发布 238 收藏 2

分类专栏: [# BUUCTF CTF](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: [https://blog.csdn.net/weixin\\_46009088/article/details/109280340](https://blog.csdn.net/weixin_46009088/article/details/109280340)

版权



[BUUCTF 同时被 2 个专栏收录](#)

17 篇文章 2 订阅

订阅专栏



[CTF](#)

17 篇文章 0 订阅

订阅专栏

---

## BUUCTF\_CrackRTF

---

学到了很多, 尽量写的详细一点吧!!!

丢进IDA, 找到main\_0, F5反汇编(感觉写了这么多进去main函数就是main\_0下次直接找main\_0了)

```

int __cdecl main_0()
{
    DWORD v0; // eax
    DWORD v1; // eax
    CHAR String; // [esp+4Ch] [ebp-310h]
    int v4; // [esp+150h] [ebp-20Ch]
    CHAR String1; // [esp+154h] [ebp-208h]
    BYTE pbData; // [esp+258h] [ebp-104h]

    memset(&pbData, 0, 0x104u);
    memset(&String1, 0, 0x104u);
    v4 = 0;
    printf("pls input the first passwd(1): ");
    scanf("%s", &pbData);
    if ( strlen((const char *)&pbData) != 6 ) // fLag的长度为6
    {
        printf("Must be 6 characters!\n");
        ExitProcess(0);
    }
    v4 = atoi((const char *)&pbData); // 感觉是转换v4的, 具体没细看(感觉不太重要)
    if ( v4 < 100000 ) // v4 小于100000就退出
        ExitProcess(0);
    strcat((char *)&pbData, "@DBApp"); // 将@DBApp与pbData连接
    v0 = strlen((const char *)&pbData); // v0 == 12
    sub_40100A(&pbData, v0, &String1);
    if ( !_strcmpi(&String1, "6E32D0943418C2C33385BC35A1470250DD8923A9") )
    {
        printf("continue...\n\n");
        printf("pls input the first passwd(2): ");
        memset(&String, 0, 0x104u);
        scanf("%s", &String);
        if ( strlen(&String) != 6 ) // string == 6
        {
            printf("Must be 6 characters!\n");
            ExitProcess(0);
        }
        strcat(&String, (const char *)&pbData); // string == 18
        memset(&String1, 0, 0x104u);
        v1 = strlen(&String); // v1 == 18
        sub_401019((BYTE *)&String, v1, &String1);
        if ( !_strcmpi("27019e688a4e62a649fd99cadaafdb4e", &String1) )
        {
            if ( !sub_40100F(&String) )
            {
                printf("Error!!\n");
                ExitProcess(0);
            }
            printf("bye ~~\n");
        }
    }
    return 0;
}

```

关键的地方做一些注释，后面有个\_strcmpi判断字符串的，sub\_40100A应该是把pbData转换一下点进去看看(写一下备注)。

```

1 int __cdecl sub_401230(BYTE *pbData, DWORD dwDataLen, LPSTR lpString1)
2 {
3     int result; // eax
4     DWORD i; // [esp+4Ch] [ebp-28h]
5     CHAR String2; // [esp+50h] [ebp-24h]
6     char v6[20]; // [esp+54h] [ebp-20h]
7     DWORD pdwDataLen; // [esp+68h] [ebp-Ch]
8     HCRYPTHASH phHash; // [esp+6Ch] [ebp-8h]
9     HCRYPTPROV phProv; // [esp+70h] [ebp-4h]
10
11     if ( !CryptAcquireContextA(&phProv, 0, 0, 1u, 0xF0000000) )// 创建密钥
12         return 0;
13     if ( CryptCreateHash(phProv, 0x8004u, 0, 0, &phHash) )// 启动散列函数
14     {
15         if ( CryptHashData(phHash, pbData, dwDataLen, 0) )
16         {
17             CryptGetHashParam(phHash, 2u, (BYTE *)v6, &pdwDataLen, 0);// 检索控制哈希对象操作的数据
18             *lpString1 = 0;
19             for ( i = 0; i < pdwDataLen; ++i )
20             {
21                 wprintfA(&String2, "%02X", (unsigned __int8)v6[i]);
22                 lstrcatA(lpString1, &String2);
23             }
24             CryptDestroyHash(phHash);
25             CryptReleaseContext(phProv, 0);
26             result = 1;
27         }
28         else
29         {
30             CryptDestroyHash(phHash);
31             CryptReleaseContext(phProv, 0); // 释放密钥
32             result = 0;
33         }
34     }
35     else
36     {
37         CryptReleaseContext(phProv, 0); // 释放密钥
38         result = 0;
39     }
40     return result;
41 }


```

[https://blog.csdn.net/weixin\\_46009088](https://blog.csdn.net/weixin_46009088)

查一下MSDN发现CryptCreateHash函数最重要！

这个**CryptCreateHash**函数启动散列一条数据流。控件的句柄创建并返回给调用应用程序。密码服务提供者(CSP)散列对象。此句柄用于后续调用**CryptHashData**和**CryptHashSessionKey**散列会话密钥和其他数据流。

## 句法

C++	 复制
<pre> BOOL CryptCreateHash(     HCRYPTPROV hProv,     ALG_ID Algid,     HCRYPTKEY hKey,     DWORD dwFlags,     HCRYPTHASH *phHash ); </pre>	

[https://blog.csdn.net/weixin\\_46009088](https://blog.csdn.net/weixin_46009088)

ALG\_ID该值标识要使用的哈希算法我们去查一下ALG\_ID大全

CALG\_SHA1

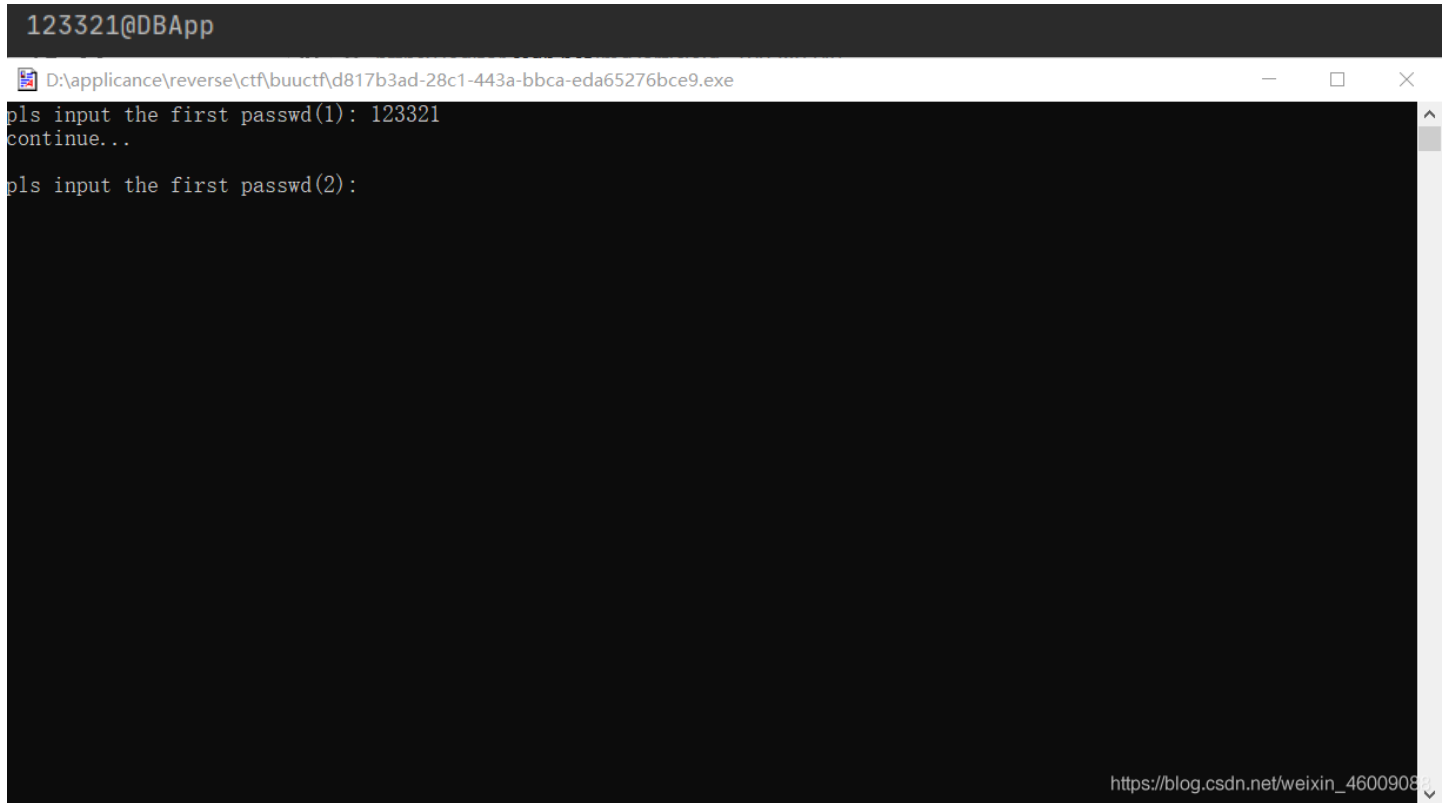
0x00008004

Same as CALG\_SHA. This algorithm is supported by the [Microsoft Base Cryptographic Provider](#).

发现这是SHA\_1加密我们写出，python脚本：

```
import hashlib
partflag = '@DBApp'
flag = ''
for i in range(100000,999999):
    flag = str(i) + partflag
    flaghex = hashlib.sha1(flag.encode('utf-8'))
    flaghex = flaghex.hexdigest()
    if "6e32d0943418c2c33385bc35a1470250dd8923a9" == flaghex:
        print(flag)
        break
```

运行得到答案：



```
123321@DBApp
D:\applicance\reverse\ctf\buuctf\d817b3ad-28c1-443a-bbca-eda65276bce9.exe
pls input the first passwd(1): 123321
continue..
pls input the first passwd(2):
```

[https://blog.csdn.net/weixin\\_46009089](https://blog.csdn.net/weixin_46009089)

答案对了！

接下来是sub\_401019点进去，如图：

```
1 int __cdecl sub_401040(BYTE *pbData, DWORD dwDataLen, LPSTR lpString1)
2 {
3     int result; // eax
4     DWORD i; // [esp+4Ch] [ebp-24h]
5     CHAR String2; // [esp+50h] [ebp-20h]
6     char u6[16]; // [esp+54h] [ebp-1Ch]
7     DWORD pdwDataLen; // [esp+64h] [ebp-Ch]
8     HCRYPTHASH phHash; // [esp+68h] [ebp-8h]
9     HCRYPTPROV phProv; // [esp+6Ch] [ebp-4h]
10
11     if ( !CryptAcquireContextA(&phProv, 0, 0, 1u, 0xF0000000) )
12         return 0;
13     if ( CryptCreateHash(phProv, 0x8003u, 0, 0, &phHash) )
14     {
15         if ( CryptHashData(phHash, pbData, dwDataLen, 0) )
16         {
17             CryptGetHashParam(phHash, 2u, (BYTE *)u6, &pdwDataLen, 0);
18             *lpString1 = 0;
19             For ( i = 0; i < pdwDataLen; ++i )
20             {
21                 wsprintfA(&String2, "%02X", (unsigned __int8)u6[i]);
22                 lstrcatA(lpString1, &String2);
23             }
24             CryptDestroyHash(phHash);
25             CryptReleaseContext(phProv, 0);
26             result = 1;
27         }
28         else
29         {
30             CryptDestroyHash(phHash);
31             CryptReleaseContext(phProv, 0);
32             result = 0;
33         }
34     }
35     else
36     {
37         CryptReleaseContext(phProv, 0);
38         result = 0;
39     }
40     return result;
41 }
```

[https://blog.csdn.net/weixin\\_46009088](https://blog.csdn.net/weixin_46009088)

和sub\_40100A差不多，关键的还是CryptCreateHash，一查，MD5！

CALG\_MD5

0x00008003

MD5 hashing algorithm. This algorithm is supported by the [Microsoft Base Cryptographic Provider](#).

方法一（有点走捷径）：

MD5作为一个会损坏原字符串的算法，想要用脚本解密是不太行的，只能拿这个字符串去解密网站<sup>1</sup>解密

## 输入让你无语的MD5

27019e688a4e62a649fd99cadaafdb4e

解密

md5

~!3a@0123321@DBApp



~!3a@0不就是答案嘛

方法二：

不管MD5,往下看sub\_40100F点进去看

```
if ( !sub_40100F(&String) )
```

```
char __cdecl sub_4014D0(LPCSTR lpString)
{
    LPCVOID lpBuffer; // [esp+50h] [ebp-1Ch]
    DWORD NumberOfBytesWritten; // [esp+58h] [ebp-14h]
    DWORD nNumberOfBytesToWrite; // [esp+5Ch] [ebp-10h]
    HGLOBAL hResData; // [esp+60h] [ebp-Ch]
    HRSRC hResInfo; // [esp+64h] [ebp-8h]
    HANDLE hFile; // [esp+68h] [ebp-4h]

    hFile = 0;
    hResData = 0;
    nNumberOfBytesToWrite = 0;
    NumberOfBytesWritten = 0;
    hResInfo = FindResourceA(0, (LPCSTR)0x65, "AAA");
    if ( !hResInfo )
        return 0;
    nNumberOfBytesToWrite = SizeofResource(0, hResInfo);
    hResData = LoadResource(0, hResInfo);
    if ( !hResData )
        return 0;
    lpBuffer = LockResource(hResData);
    sub_401005(lpString, (int)lpBuffer, nNumberOfBytesToWrite);
    hFile = CreateFileA("dbapp.rtf", 0x10000000u, 0, 0, 2u, 0x80u, 0);
    if ( hFile == (HANDLE)-1 )
        return 0;
    if ( !WriteFile(hFile, lpBuffer, nNumberOfBytesToWrite, &NumberOfBytesWritten, 0) )
        return 0;
    CloseHandle(hFile);
    return 1;
}
```

```
hResInfo = FindResourceA(0, (LPCSTR)0x65, "AAA");
```

FindResourceA为关键函数，如下：<sup>2</sup>

# FindResourceA函数(winbase.h)

12/05/2018 · 2分钟阅读

确定指定模块中具有指定类型和名称的资源的位置。

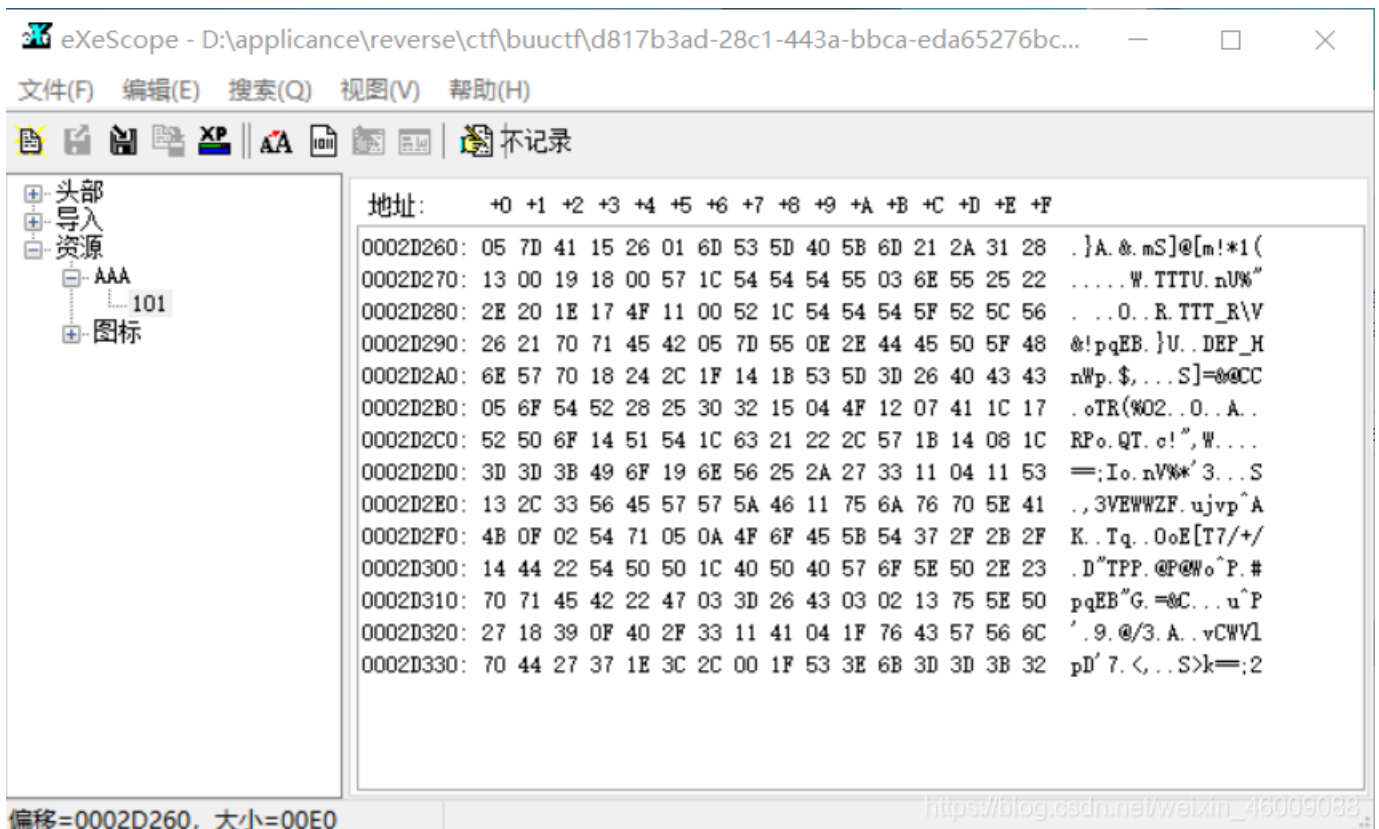
若要指定语言，请使用FindResourceEx功能。

## 句法

```
C++ 复制  
  
HRSRC FindResourceA(  
    HMODULE hModule,  
    LPCSTR lpName,  
    LPCSTR lpType  
);
```

[https://blog.csdn.net/weixin\\_46009088](https://blog.csdn.net/weixin_46009088)

用exescope打开查看AAA，Resource Hacker也阔以，现在看不懂不要紧往下看。



往下看，SizeofResource是计算长度的，LoadResource是加载资源的，具体看MASD！

```

if ( !hResInfo )
    return 0;
nNumberOfBytesToWrite = SizeofResource(0, hResInfo);
hResData = LoadResource(0, hResInfo);
if ( !hResData )
    return 0;
lpBuffer = LockResource(hResData);
sub_401005(lpString, (int)lpBuffer, nNumberOfBytesToWrite);
hFile = CreateFileA("dbapp.rtf", 0x10000000u, 0, 0, 2u, 0x80u, 0);

```

LockResource指向了AAA这个文件的资源MSDN文档<sup>3</sup>如下:

# LockResource函数(libloaderapi.h)

09/24/2020 • 2分钟阅读

检索指向内存中指定资源的指针。

## 句法

C++	Copy
<pre> LPVOID LockResource(     HGLOBAL hResData ); </pre>	

[https://blog.csdn.net/weixin\\_46009088](https://blog.csdn.net/weixin_46009088)

接下来的sub\_401005的函数对数据进行了操作(lpString是我们的密码, lpBuffer是资源, nNumberOfBytesToWrite是资源的大小):

```
sub_401005(lpString, (int)lpBuffer, nNumberOfBytesToWrite);
```

点进来, 如图:

```

1 unsigned int __cdecl sub_401420(LPCSTR lpString, int a2, int a3)
2 {
3     unsigned int result; // eax
4     unsigned int i; // [esp+4Ch] [ebp-Ch]
5     unsigned int v5; // [esp+54h] [ebp-4h]
6
7     v5 = lstrlenA(lpString);
8     for ( i = 0; ; ++i )
9     {
10        result = i;
11        if ( i >= a3 )
12            break;
13        *(_BYTE *)(i + a2) ^= lpString[i % v5];
14    }
15    return result;
16 }

```

[https://blog.csdn.net/weixin\\_46009088](https://blog.csdn.net/weixin_46009088)

异或便是整个函数的关键点, 现在理一下思路, 我们的密码和资源异或返回以后, 将得到一个dbapp.rtf的文件所以最后得到的是一个rtf文件:

```
hFile = CreateFileA("dbapp.rtf", 0x10000000u, 0, 0, 2u, 0x80u, 0);
```

我们用UltraEdit随意打开一个rtf文件看看, 如图:

```
{\rtf1\ansi\ansicpg936\deff0\deflang1033\deflangfe2052{\fonttbl{\f0\modern\fprq6\fcharset134 \cb\ce\cc'e5;}}
```

我们只是不知道 lpString 这个参数的前六位所以我只要取rtf文件的前6位进行运算就好了, 写出python脚本:

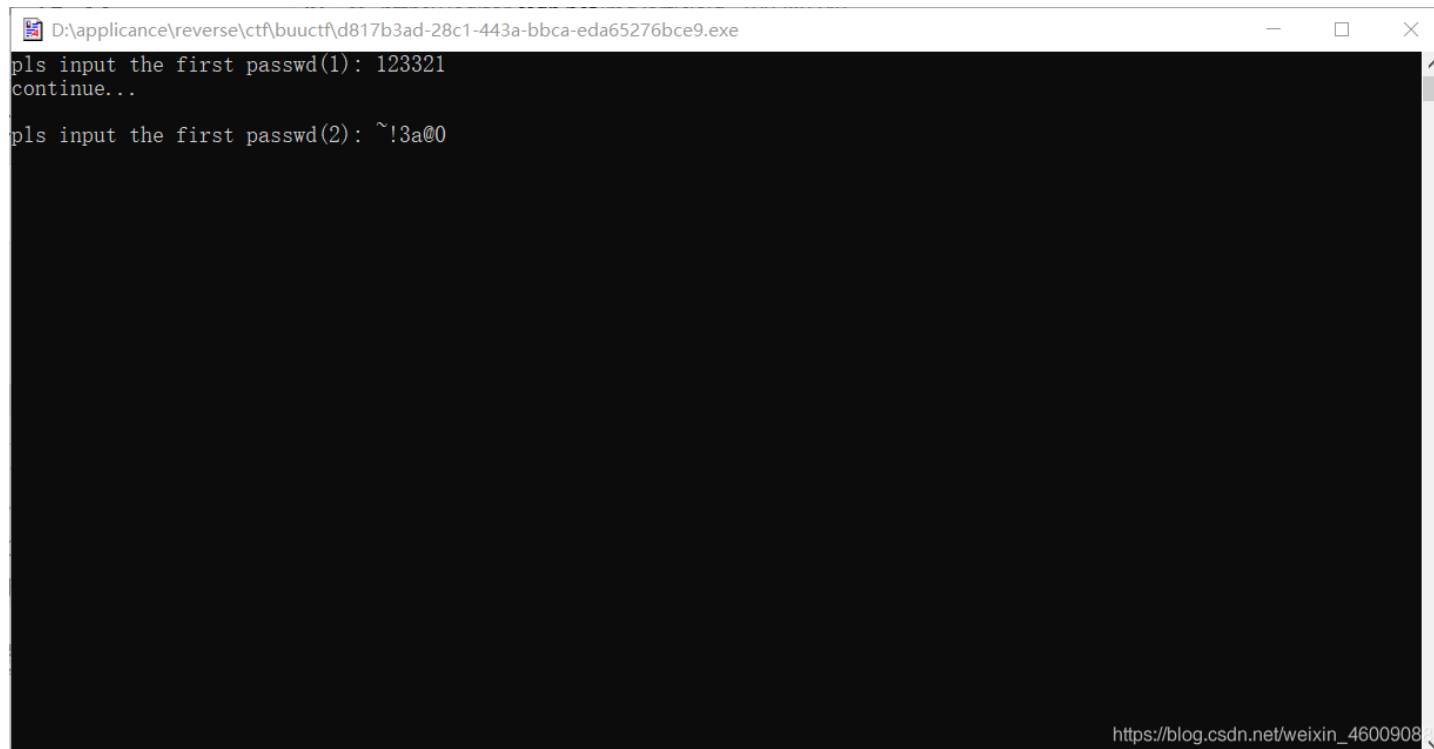


```
str = ['{', '\\', 'r', 't', 'f', '1', ]
resource = [0x05, 0x7D, 0x41, 0x15, 0x26, 0x01]
flag = ''
for i in range(len(str)):
    flag += chr(ord(str[i]) ^ resource[i])
print(flag)
```

运行得到结果:

```
~!3a@0
```

将结果输入到控制台中.



```
D:\appliance\reverse\ctf\buuctf\d817b3ad-28c1-443a-bbca-eda65276bce9.exe
pls input the first passwd(1): 123321
continue...
pls input the first passwd(2): ~!3a@0
~!3a@0
https://blog.csdn.net/weixin_4600908
```

控制台退出, 随后生成了一个rft文件, 打开便是flag!!!

Flag{N0\_M0re\_Free\_Bugs}

MD5解密网站

<https://www.somd5.com/>

FindResourceA MSDN文档: <https://docs.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-findresourcea>

LockResource MSDN文档: <https://docs.microsoft.com/en-us/windows/win32/api/libloaderapi/nf-libloaderapi-lockresource>