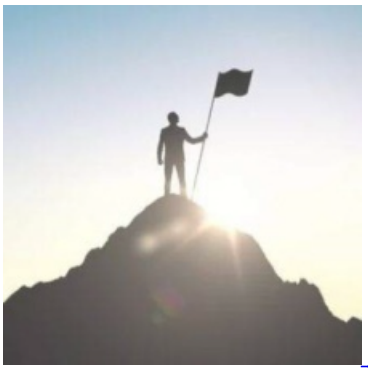
BUUCTF Reverse CrackRTF

原创

A_dmins 于 2019-07-20 22:53:43 发布 2405 收藏 4

分类专栏： CTF题 一天一道CTF BUUCTF

本文链接：https://blog.csdn.net/qq_42967398/article/details/96492843



CTF题 同时被 3 个专栏收录

115 篇文章 11 订阅

订阅专栏

一天一道CTF

52 篇文章 5 订阅

订阅专栏

BUUCTF

24 篇文章 2 订阅

订阅专栏

BUUCTF Reverse CrackRTF

一天一道**CTF**题目，能多不能少

题目描述：

下载文件，无表直接ida打开，查看主函数

下载文件，无完直接ida打开，查看主函数：

```
printf("pls input the first passwd(1): ");
scanf("%s", &pbData);
if ( strlen((const char *)&pbData) != 6 )
{
  printf("Must be 6 characters!\n");
  ExitProcess(0);
}
v4 = atoi((const char *)&pbData);
if ( v4 < 100000 )
  ExitProcess(0);
strcat((char *)&pbData, "@DBApp");
v0 = strlen((const char *)&pbData);
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 )
  {
    printf("Must be 6 characters!\n");
    ExitProcess(0);
  }
  strcat(&String, (const char *)&pbData);
  memset(&String1, 0, 0x104u);
  v1 = strlen(&String);
  sub_401019((BYTE *)&String, v1, &String1);
  if ( !_strcmpi("27019e688a4e62a649fd99cadaafdb4e", &String1) )
  {
    if ( !(unsigned __int8)sub_40100F(&String) )
    {
      printf("Error!!\n");
      ExitProcess(0);
    }
    printf("bye ~~\n");
```


可以看到，需要我们输入两次密码，而且每次输入密码都必须是6位数
否则退出，并且有两次判断~~
先看第一次，第一次要求我们输入6位数，然后连接上@DBApp，
通过一个sub_40100A函数进行加密，然后与`6E32D0943418C2C33385BC35A1470250DD8923A9`进行匹配
如果一样则继续往下走~~，否则退出~

```
printf("pls input the first passwd(1): ");
scanf("%s", &pbData);
if ( strlen((const char *)&pbData) != 6 )
{
  printf("Must be 6 characters!\n");
  ExitProcess(0);
}
v4 = atoi((const char *)&pbData);
if ( v4 < 100000 )
  ExitProcess(0);
strcat((char *)&pbData, "@DBApp");
v0 = strlen((const char *)&pbData);
sub_40100A(&pbData, v0, &String1);
if ( !_strcmpi(&String1, "6E32D0943418C2C33385BC35A1470250DD8923A9") )
{
```


先进入第一个加密函数中去看看，内容如下：

```
int __cdecl sub_401230(BYTE *pbData, DWORD dwDataLen, LPSTR lpString1)
{
  int result; // eax
  DWORD i; // [esp+4Ch] [ebp-28h]
  CHAR String2; // [esp+50h] [ebp-24h]
  BYTE v6[20]; // [esp+54h] [ebp-20h]
  DWORD pdwDataLen; // [esp+68h] [ebp-Ch]
  HCRYPTHASH phHash; // [esp+6Ch] [ebp-8h]
  HCRYPTPROV phProv; // [esp+70h] [ebp-4h]

  if ( !CryptAcquireContextA(&phProv, 0, 0, 1u, 0xF0000000) )
    return 0;
  if ( CryptCreateHash(phProv, 0x8004u, 0, 0, &phHash) )
  {
    if ( CryptHashData(phHash, pbData, dwDataLen, 0) )
    {
      CryptGetHashParam(phHash, 2u, v6, &pdwDataLen, 0);
      *lpString1 = 0;
      for ( i = 0; i < pdwDataLen; ++i )
      {
        wsprintfA(&String2, "%02X", v6[i]);
        lstrcatA(lpString1, &String2);
      }
      CryptDestroyHash(phHash);
      CryptReleaseContext(phProv, 0);
      result = 1;
    }
    else
    {
      CryptDestroyHash(phHash);
      CryptReleaseContext(phProv, 0);
      result = 0;
    }
  }
  else
  {
```


这利用到了一个windows加密的加密库函数~
一般第二个参数是加密的方式，但是这里看上去不是很明显，，，，
不过还是可以猜一下的，，，，，
经过发现`6E32D0943418C2C33385BC35A1470250DD8923A9`是40位的加密后的字符串
很有可能是sha1加密，先来爆破试一试~
编写脚本：

```
import hashlib

flag = "@DBApp"

for i in range(100000,999999):
 s = str(i)+flag
 x = hashlib.sha1(s.encode())
 cnt = x.hexdigest()
 if "6e32d0943418c2c" in cnt:
  print(cnt)
  print(str(i)+flag)
```

爆出第一次密码~~

```
6e32d0943418c2c33385bc35a1470250dd8923a9
123321@DBApp
```

运行exe，输入检查正确~

接下来走第二次输入~

第二次输入同理，不过他把123321@DBApp加在了第二次密码的后面，并且进行加密~

```
  ExitProcess(0);
strcat((char *)&pbData, "@DBApp");
v0 = strlen((const char *)&pbData);
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 )
  {
    printf("Must be 6 characters!\n");
    ExitProcess(0);
  }
  strcat(&String, (const char *)&pbData);
  memset(&String1, 0, 0x104u);
  v1 = strlen(&String);
  sub_401019((BYTE *)&String, v1, &String1);
  if ( !_strcmpi("27019e688a4e62a649fd99cadaafdb4e", &String1) )
  {
    if ( !(unsigned __int8)sub_40100F(&String) )
    {
```

直接进入sub_401019函数进行查看~：

```
int __cdecl sub_401040(BYTE *pbData, DWORD dwDataLen, LPSTR lpString1)
{
  int result; // eax
  DWORD i; // [esp+4Ch] [ebp-24h]
  CHAR String2; // [esp+50h] [ebp-20h]
  BYTE v6[16]; // [esp+54h] [ebp-1Ch]
  DWORD pdwDataLen; // [esp+64h] [ebp-Ch]
  HCRYPTHASH phHash; // [esp+68h] [ebp-8h]
  HCRYPTPROV phProv; // [esp+6Ch] [ebp-4h]

  if ( !CryptAcquireContextA(&phProv, 0, 0, 1u, 0xF0000000) )
    return 0;
  if ( CryptCreateHash(phProv, 0x8003u, 0, 0, &phHash) )
  {
    if ( CryptHashData(phHash, pbData, dwDataLen, 0) )
    {
      CryptGetHashParam(phHash, 2u, v6, &pdwDataLen, 0);
      *lpString1 = 0;
      for ( i = 0; i < pdwDataLen; ++i )
      {
        wsprintfA(&String2, "%02X", v6[i]);
        lstrcatA(lpString1, &String2);
      }
      CryptDestroyHash(phHash);
      CryptReleaseContext(phProv, 0);
      result = 1;
    }
    else
    {
      CryptDestroyHash(phHash);
      CryptReleaseContext(phProv, 0);
      result = 0;
    }
```

发现大体好像差不多，不过第一个参数不同了，应该换了一种加密方式~~

发现只得好像是不多，不过第二个参数不同了，应该换了一种加密方式
再次查看比较字符串的长度~
发现是32位的，一般来说就是MD5了
不过什么提示都没有，，，我怎么进行爆破啊
6位全字符，这得爆破到一年，，，，陷入迷惘！！！
先继续往下走走看~
发现if里面还有一个函数~
进入到这个比较中的sub_40100F函数看看！！

```
39    v1 = strlen(&String);
40    sub_401019((BYTE *)&String, v1, &String1);
41    if ( !_strcmpi("27019e688a4e62a649fd99cadaafdb4e", &String1) )
42    {
43      if ( !(unsigned __int8)sub_40100F(&String) )
44      {
45        printf("Error!!\n");
46        ExitProcess(0);
47      }
48      printf("bye ~~\n");
49    }
```


看到内容如下，应该是从一个AAA的东西里面取出数据，
然后将从AAA中取出的数据和我们传入的参数——也就是输入的第二次密码连接后的字符串
传入一个sub_401005函数
最后生成一个dbapp.rtf文件，看到这想起了题目的RTF~
这个函数应该有东西~

```
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;
}
```


进入sub_401005函数查看一下，
就是将我们从AAA取出的值和第二次密码连接后的字符串进行异或嘛：

```
unsigned int __cdecl sub_401420(LPCSTR lpString, int a2, int a3)
{
  unsigned int result; // eax
  unsigned int i; // [esp+4Ch] [ebp-Ch]
  unsigned int v5; // [esp+54h] [ebp-4h]

  v5 = lstrlenA(lpString);
  for ( i = 0; ; ++i )
  {
    result = i;
    if ( i >= a3 )
      break;
    *(_BYTE *)(i + a2) ^= lpString[i % v5];
  }
  return result;
}
```


不过从AAA中的值不知道，，，，

后面才知道，有一款工具可以直接查看文件中的资源——ResourceHacker

名称	修改日期	类型	大小
help	2019/7/18 23:54	文件夹	
samples	2019/7/18 23:54	文件夹	
changes.txt	2019/7/18 23:54	文本文档	4 KB
ReadMe.txt	2019/7/18 23:54	文本文档	2 KB
ResourceHacker.exe	2019/7/18 23:54	应用程序	5,351 KB
ResourceHacker.ini	2019/7/18 23:56	配置设置	1 KB

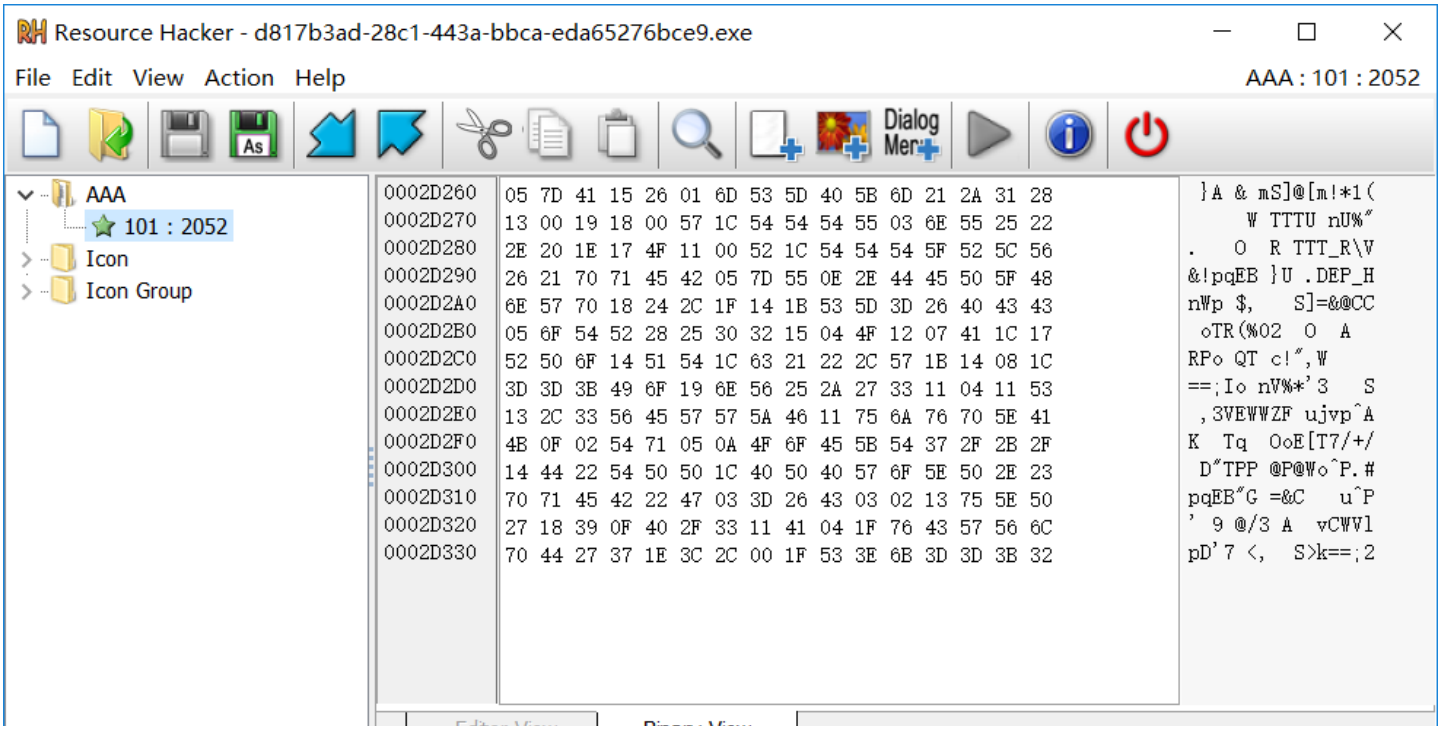
附上下载链接吧~

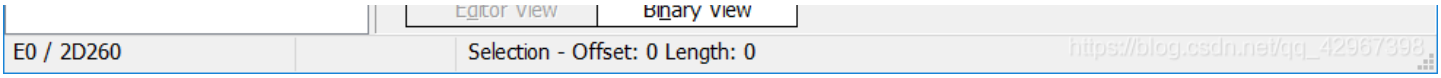
链接：https://pan.baidu.com/s/10_uSM2JG5BElDywaTsAldg

提取码：a9ng

复制这段内容后打开百度网盘手机App，操作更方便哦

打开得到数据~~





这就好办了，因为要生成一个.rtf的文件，那肯定缺少不了文件头
直接搜索.rtf的文件头~
得到：`{\rtf1\ansi\ansicpg936\deff0\deflang1033等等等`
这里我们去前6位即可，因为我们密码是6位数的
所以前六位（{\rtf1）直接与AAA中数据异或就能得出第二次加密的密码啦~
直接写脚本：

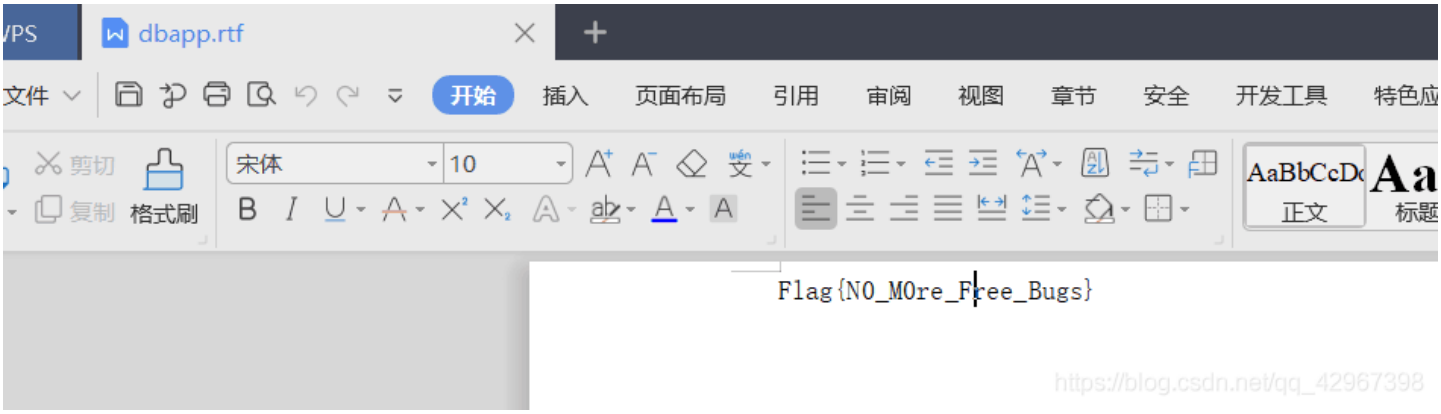
```
s = "{\\rtf1"

a = [0x05,0x7D,0x41,0x15,0x26,0x01]

flag = ""
for i in range(0,len(s)):
 x = ord(s[i]) ^ a[i]
 flag += chr(x)
print(flag)
```

得到第二次密码：`~!3a@0`
验证正确~

最后在本地查找dbapp.rtf文件，最后得到flag~



flag：`flag{N0_M0re_Free_Bugs}`