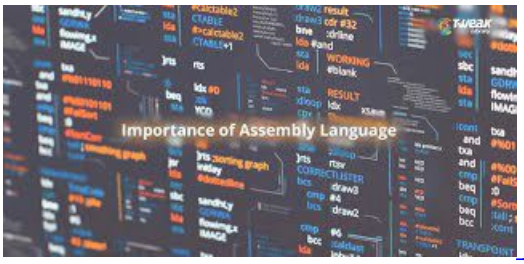
BUUCTF reverse wp 11 - 20

原创

fa1c4 于 2022-01-08 18:40:43 发布 372 收藏

分类专栏： 逆向工程 文章标签： windows逆向 Android逆向

本文链接：https://blog.csdn.net/qq_33976344/article/details/121262268



逆向工程 专栏收录该内容

58 篇文章 1 订阅

订阅专栏

Java逆向解密

java逆向, 用java-decompile逆

```
import java.util.ArrayList;
import java.util.Scanner;

public class Reverse {
  public static void main(String[] args) {
    Scanner s = new Scanner(System.in);
    System.out.println("Please input the flag );
    String str = s.next();
    System.out.println("Your input is );
    System.out.println(str);
    char[] stringArr = str.toCharArray();
    Encrypt(stringArr);
  }

  public static void Encrypt(char[] arr) {
    ArrayList<Integer> Resultlist = new ArrayList<>();
    for (int i = 0; i < arr.length; i++) {
      int result = arr[i] + 64 ^ 0x20;
      Resultlist.add(Integer.valueOf(result));
    }
    int[] KEY = {
        180, 136, 137, 147, 191, 137, 147, 191, 148, 136,
        133, 191, 134, 140, 129, 135, 191, 65 };
    ArrayList<Integer> KEYList = new ArrayList<>();
    for (int j = 0; j < KEY.length; j++)
      KEYList.add(Integer.valueOf(KEY[j]));
    System.out.println("Result:");
    if (Resultlist.equals(KEYList)) {
      System.out.println("Congratulations);
    } else {
      System.err.println("Error);
    }
  }
}
```

当`Resultlist.equals(KEYList)`时通过检测
`Resultlist`的逻辑

```
    for (int i = 0; i < arr.length; i++) {
      int result = arr[i] + 64 ^ 0x20;
      Resultlist.add(Integer.valueOf(result));
    }
```

逆回去就是先用Keylist ^ 0x20, 然后减去64(注意java中`^`的优先级是比`+`更低的,所以是先`+`再`^`

```
Keylist = [180, 136, 137, 147, 191, 137, 147, 191, 148, 136, \
           133, 191, 134, 140, 129, 135, 191, 65]

flag = ''
for key in Keylist:
    flag += chr((key ^ 0x20) - 64)

print(flag)
```

结果记得加flag{}

[GXYCTF2019]luck_guy

file看是x64文件, 没壳拖进IDA64

```
unsigned __int64 get_flag()
{
  unsigned int v0; // eax
  int i; // [rsp+4h] [rbp-3Ch]
  int j; // [rsp+8h] [rbp-38h]
  __int64 s; // [rsp+10h] [rbp-30h] BYREF
  char v5; // [rsp+18h] [rbp-28h]
  unsigned __int64 v6; // [rsp+38h] [rbp-8h]

  v6 = __readfsqword(0x28u);
  v0 = time(0LL);
  srand(v0);
  for ( i = 0; i <= 4; ++i )
  {
    switch ( rand() % 200 )
    {
      case 1:
        puts("OK, it's flag:");
        memset(&s, 0, 0x28uLL);
        strcat((char *)&s, f1);
        strcat((char *)&s, &f2);
        printf("%s", (const char *)&s);
        break;
      case 2:
        printf("Solar not like you");
        break;
      case 3:
        printf("Solar want a girlfriend");
        break;
      case 4:
        s = 0x7F666F6067756369LL;
        v5 = 0;
        strcat(&f2, (const char *)&s);
        break;
      case 5:
        for ( j = 0; j <= 7; ++j )
        {
          if ( j % 2 == 1 )
            *(&f2 + j) -= 2;
          else
            --*(&f2 + j);
        }
        break;
      default:
        puts("emmm,you can't find flag 23333");
        break;
    }
  }
  return __readfsqword(0x28u) ^ v6;
}
```

```
.data:0000000000601078 ; char f1[]
.data:0000000000601078 f1              db 'GXY{do_not_',0      ; DATA XREF: get_flag+9E↑o
.data:0000000000601078 _data           ends
.data:0000000000601078
```

处理一下s字符串, 然后就是flag后半段

```
flag = "GXY{do_not_"
s = [0x7F, 0x66, 0x6F, 0x60, 0x67, 0x75, 0x63, 0x69] # 0x7F666F6067756369LL
s = s[::-1]
for i in range(8):
    if i % 2 == 1: s[i] -= 2
    else: s[i] -= 1
    flag += chr(s[i])

print(flag)
```

GXY{do_not_hate_me}
提交时改成flag{}

刮开有奖

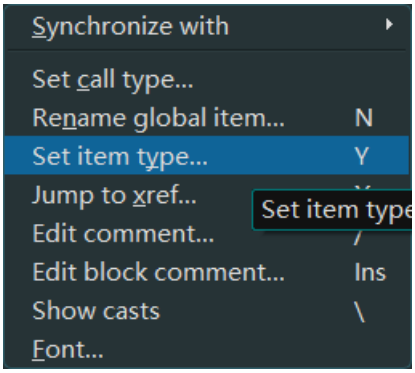
```
INT_PTR __stdcall DialogFunc(HWND hDlg, UINT a2, WPARAM a3, LPARAM a4)
{
  const char *v4; // esi
  const char *v5; // edi
  int v7[2]; // [esp+8h] [ebp-20030h] BYREF
  int v8; // [esp+10h] [ebp-20028h]
  int v9; // [esp+14h] [ebp-20024h]
  int v10; // [esp+18h] [ebp-20020h]
  int v11; // [esp+1Ch] [ebp-2001Ch]
  int v12; // [esp+20h] [ebp-20018h]
  int v13; // [esp+24h] [ebp-20014h]
  int v14; // [esp+28h] [ebp-20010h]
  int v15; // [esp+2Ch] [ebp-2000Ch]
  int v16; // [esp+30h] [ebp-20008h]
  CHAR String[65536]; // [esp+34h] [ebp-20004h] BYREF
  char v18[65536]; // [esp+10034h] [ebp-10004h] BYREF

  if ( a2 == 272 )
    return 1;
  if ( a2 != 273 )
    return 0;
  if ( a3 == 1001 )
  {
    memset(String, 0, 0xFFFFu);
    GetDlgItemTextA(hDlg, 1000, String, 0xFFFF);
    if ( strlen(String) == 8 )
    {
      v7[0] = 90;
      v7[1] = 74;
      v8 = 83;
      v9 = 69;
      v10 = 67;
      v11 = 97;
      v12 = 78;
      v13 = 72;
      v14 = 51;
      v15 = 110;
      v16 = 103;
      sub_4010F0(v7, 0, 10);
      memset(v18, 0, 0xFFFFu);
      v18[0] = String[5];
      v18[2] = String[7];
      v18[1] = String[6];
      v4 = sub_401000(v18, strlen(v18));
```

```
      memset(v18, 0, 0xFFFFu);
      v18[1] = String[3];
      v18[0] = String[2];
      v18[2] = String[4];
      v5 = sub_401000(v18, strlen(v18));
      if ( String[0] == v7[0] + 34
        && String[1] == v10
        && 4 * String[2] - 141 == 3 * v8
        && String[3] / 4 == 2 * (v13 / 9)
        && !strcmp(v4, "ak1w")
        && !strcmp(v5, "V1Ax") )
      {
        MessageBoxA(hDlg, "U g3t 1T!", "@_@", 0);
      }
    }
    return 0;
  }
  if ( a3 != 1 && a3 != 2 )
    return 0;
  EndDialog(hDlg, a3);
  return 1;
}
```

输入8字符的string, 通过if校验后为flag

```
      if ( String[0] == v7[0] + 34
        && String[1] == v10
        && 4 * String[2] - 141 == 3 * v8
        && String[3] / 4 == 2 * (v13 / 9)
        && !strcmp(v4, "ak1w")
        && !strcmp(v5, "V1Ax") )
```

v7 到 v16会经过`sub_4010F0`函数, 这里懒得分析了, 直接修改为c语言run, 可以在IDA中修改函数的item type, a1修改为char[]类型



双击v7局部变量, 进入栈窗口, 右键修改为数组类型

```
19      a1[0] = 'Z';
20      a1[1] = 'J';
21      a1[2] = 'S';
22      a1[3] = 'E';
23      a1[4] = 'C';
24      a1[5] = 'a';
25      a1[6] = 'N';
26      a1[7] = 'H';
27      a1[8] = '3';
28      a1[9] = 'n';
29      a1[10] = 'g';
30      sub_4010F0(a1, 0, 10);
31      memset(v9, 0, 0xFFFFu);
```


伪代码中的`4 *`进行数组索引是不合理的, 去掉之后得到符合语法的C代码如下

```
#include<iostream>
using namespace std;

int sub_4010F0(char a1[], int a2, int a3)
{
  int result; // eax
  int i; // esi
  int v5; // ecx
  int v6; // edx

  result = a3;
  for ( i = a2; i <= a3; a2 = i )
  {
    v5 = i;
    v6 = *&a1[i];
    if ( a2 < result && i < result )
    {
      do
      {
        if ( v6 > *&a1[result] )
        {
          if ( i >= result )
            break;
          ++i;
          *&a1[v5] = *&a1[result];
          if ( i >= result )
            break;
          while ( *&a1[i] <= v6 )
          {
            if ( ++i >= result )
              goto LABEL_13;
          }
          if ( i >= result )
            break;
          v5 = i;
          *&a1[result] = *&a1[i];
        }
        --result;
      }
      while ( i < result );
    }
LABEL_13:
```

```
    *&a1[result] = v6;
    sub_4010F0(a1, a2, i - 1);
    result = a3;
    ++i;
  }
  return result;
}

/*
      v7[0] = 90;
      v7[1] = 74;
      v8 = 83;
      v9 = 69;
      v10 = 67;
      v11 = 97;
      v12 = 78;
      v13 = 72;
      v14 = 51;
      v15 = 110;
      v16 = 103;
*/

int main() {
    char arr[] = {90, 74, 83, 69, 67, 97, 78, 72, 51, 110, 103, 0};
    cout << arr << endl;
    sub_4010F0(arr, 0, 10);
    cout << arr << endl;
    return 0;
}
```

> ZJSECaNH3ng
> 3CEHJNSZagn

可以确定string的前四位字符

```
_BYTE *__cdecl sub_401000(int a1, int a2)
{
  int v2; // eax
  int v3; // esi
  size_t v4; // ebx
  _BYTE *v5; // eax
  _BYTE *v6; // edi
  int v7; // eax
  _BYTE *v8; // ebx
  int v9; // edi
  int v10; // edx
  int v11; // edi
  int v12; // eax
  int i; // esi
  _BYTE *result; // eax
  _BYTE *v15; // [esp+Ch] [ebp-10h]
  _BYTE *v16; // [esp+10h] [ebp-Ch]
  int v17; // [esp+14h] [ebp-8h]
  int v18; // [esp+18h] [ebp-4h]

  v2 = a2 / 3;
  v3 = 0;
  if ( a2 % 3 > 0 )
```

```
  ++v2;
v4 = 4 * v2 + 1;
v5 = malloc(v4);
v6 = v5;
v15 = v5;
if ( !v5 )
  exit(0);
memset(v5, 0, v4);
v7 = a2;
v8 = v6;
v16 = v6;
if ( a2 > 0 )
{
  while ( 1 )
  {
    v9 = 0;
    v10 = 0;
    v18 = 0;
    do
    {
      if ( v3 >= v7 )
        break;
      ++v10;
      v9 = *(v3 + a1) | (v9 << 8);
      ++v3;
    }
    while ( v10 < 3 );
    v11 = v9 << (8 * (3 - v10));
    v12 = 0;
    v17 = v3;
    for ( i = 18; i > -6; i -= 6 )
    {
      if ( v10 >= v12 )
      {
        *(&v18 + v12) = (v11 >> i) & 0x3F;
        v8 = v16;
      }
      else
      {
        *(&v18 + v12) = 64;
      }
      *v8++ = byte_407830[*(&v18 + v12++)];
      v16 = v8;
    }
    v3 = v17;
    if ( v17 >= a2 )
      break;
    v7 = a2;
  }
  v6 = v15;
}
result = v6;
*v8 = 0;
return result;
}
```

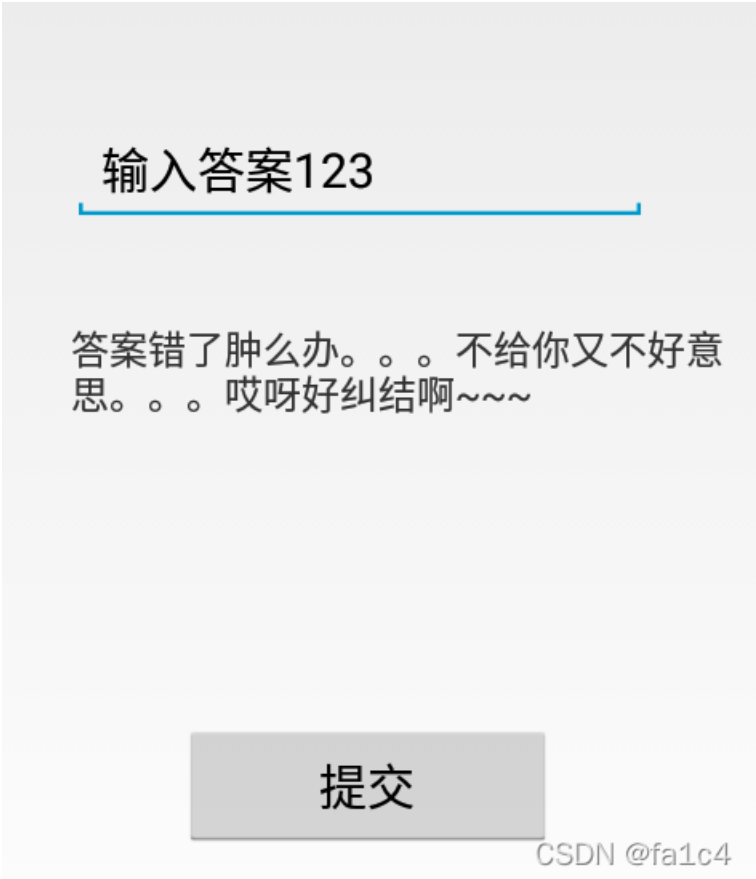
分析`sub_401000`函数, 看到`byte_407830`数组

```
.rdata:00407830 byte_407830     db 41h                  ; DATA XREF: sub_401000+C0↑r
.rdata:00407831 aBcdefghijklmno db 'BCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=',0
.rdata:00407872                 align 4
.rdata:00407874 aAk1w           db 'ak1w',0             ; DATA XREF: DialogFunc+24D↑o
```

所以直接猜测为Base64加密, (因为传入3字节, 输出4字节

```
import base64

"""
      if ( String[0] == a1[0] + 34
        && String[1] == a1[4]
        && 4 * String[2] - 141 == 3 * a1[2]
        && String[3] / 4 == 2 * (a1[7] / 9)
        && !strcmp(v4, "ak1w")
        && !strcmp(v5, "V1Ax") )
"""
s = "3CEHJNSZagn"
arr = [ord(_) for _ in s]
print(arr)
strings = [arr[0] + 34, arr[4], (3 * arr[2] + 141) // 4, int(2 * (arr[7] / 9) * 4)]
print(strings)

flag = "".join([chr(_) for _ in strings])
# print(flag)

b64str1 = "ak1w"
b64str2 = "V1Ax"
print(base64.b64decode(b64str1))
print(base64.b64decode(b64str2))
flag += "1jMp"
print("flag{" + flag + "}")
```

findit


输入答案123
答案错了肿么办。。。不给你又不好意思。。。哎呀好纠结啊~~~
提交
CSDN @fa1c4

```java
package com.example.findit;

import android.os.Bundle;
import android.support.v7.app.ActionBarActivity;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends ActionBarActivity {
    /* access modifiers changed from: protected */
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView((int) R.layout.activity_main);
        final EditText edit = (EditText) findViewById(R.id.widget2);
        final TextView text = (TextView) findViewById(R.id.widget1);
        final char[] a = {'T', 'h', 'i', 's', 'I', 's', 'T', 'h', 'e', 'F', 'l', 'a', 'g', 'H', 'o', 'm', 'e'};
        final char[] b = {'p', 'v', 'k', 'q', '{', 'm', '1', '6', '4', '6', '7', '5', '2', '6', '2', '0', '3', '
3', 'l', '4', 'm', '4', '9', 'l', 'n', 'p', '7', 'p', '9', 'm', 'n', 'k', '2', '8', 'k', '7', '5', '}'};
        ((Button) findViewById(R.id.widget3)).setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                char[] x = new char[17];
                char[] y = new char[38];
                for (int i = 0; i < 17; i++) {
                    if ((a[i] < 'I' && a[i] >= 'A') || (a[i] < 'i' && a[i] >= 'a')) {
                        x[i] = (char) (a[i] + 18);
                    } else if ((a[i] < 'A' || a[i] > 'Z') && (a[i] < 'a' || a[i] > 'z')) {
                        x[i] = a[i];
                    } else {
                        x[i] = (char) (a[i] - 8);
                    }
                }
                if (String.valueOf(x).equals(edit.getText().toString())) {
                    for (int i2 = 0; i2 < 38; i2++) {
                        if ((b[i2] < 'A' || b[i2] > 'Z') && (b[i2] < 'a' || b[i2] > 'z')) {
                            y[i2] = b[i2];
                        } else {
                            y[i2] = (char) (b[i2] + 16);
                            if ((y[i2] > 'Z' && y[i2] < 'a') || y[i2] >= 'z') {
                                y[i2] = (char) (y[i2] - 26);
                            }
                        }
                    }
                    text.setText(String.valueOf(y));
                    return;
                }
                text.setText("答案错了肿么办。。。不给你又不好意思。。。哎呀好纠结啊~~~");
            }
        });
    }

    public boolean onOptionsItemSelected(MenuItem item) {
        if (item.getItemId() == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
}
```

输入与x相同则输出flag, 图省事, java转cpp比较方便, 所以直接用cpp写exp

exp.cpp

```cpp
#include<iostream>
#include<string>
using namespace std;

int main() {
    char a[] = {'T', 'h', 'i', 's', 'I', 's', 'T', 'h', 'e', 'F', 'l', 'a', 'g', 'H', 'o', 'm', 'e'};
    char x[17];

    for (int i = 0; i < 17; i++) {
    if ((a[i] < 'I' && a[i] >= 'A') || (a[i] < 'i' && a[i] >= 'a')) {
        x[i] = (char) (a[i] + 18);
    } else if ((a[i] < 'A' || a[i] > 'Z') && (a[i] < 'a' || a[i] > 'z')) {
        x[i] = a[i];
    } else {
        x[i] = (char) (a[i] - 8);
        }
    }

    cout << x << endl;
}
```

[BJDCTF2020]JustRE

shift + F12, 找到flag字符串

```
INT_PTR __stdcall DialogFunc(HWND hWnd, UINT a2, WPARAM a3, LPARAM a4)
{
  CHAR String[100]; // [esp+0h] [ebp-64h] BYREF

  if ( a2 != 272 )
  {
    if ( a2 != 273 )
      return 0;
    if ( (_WORD)a3 != 1 && (_WORD)a3 != 2 )
    {
      sprintf(String, Format, ++dword_4099F0);
      if ( dword_4099F0 == 19999 )
      {
        sprintf(String, " BJD{%d%d2069a45792d233ac}", 19999, 0);
        SetWindowTextA(hWnd, String);
        return 0;
      }
      SetWindowTextA(hWnd, String);
      return 0;
    }
    EndDialog(hWnd, (unsigned __int16)a3);
  }
  return 1;
}
```

记得将BJD{} 改为flag{}

简单注册器

```
package com.example.flag;

import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v7.app.ActionBarActivity;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends ActionBarActivity {
    /* access modifiers changed from: protected */
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView((int) R.layout.activity_main);
        if (savedInstanceState == null) {
            getSupportFragmentManager().beginTransaction().add((int) R.id.container, (Fragment) new PlaceholderF
ragment()).commit();
        }
        final TextView textview = (TextView) findViewById(R.id.textView1);
        final EditText editview = (EditText) findViewById(R.id.editText1);
        ((Button) findViewById(R.id.button1)).setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                int flag = 1;
                String xx = editview.getText().toString();
                if (!(xx.length() == 32 && xx.charAt(31) == 'a' && xx.charAt(1) == 'b' && (xx.charAt(0) + xx.cha
```

```
rAt(2)) - 48 == 56)) {
                    flag = 0;
                }
                if (flag == 1) {
                    char[] x = "dd2940c04462b4dd7c450528835cca15".toCharArray();
                    x[2] = (char) ((x[2] + x[3]) - 50);
                    x[4] = (char) ((x[2] + x[5]) - 48);
                    x[30] = (char) ((x[31] + x[9]) - 48);
                    x[14] = (char) ((x[27] + x[28]) - 97);
                    for (int i = 0; i < 16; i++) {
                        char a = x[31 - i];
                        x[31 - i] = x[i];
                        x[i] = a;
                    }
                    textview.setText("flag{" + String.valueOf(x) + "}");
                    return;
                }
                textview.setText("输入注册码错误");
            }
        });
    }

    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    public boolean onOptionsItemSelected(MenuItem item) {
        if (item.getItemId() == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }

    public static class PlaceholderFragment extends Fragment {
        public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
            return inflater.inflate(R.layout.fragment_main, container, false);
        }
    }
}
```

```
#include<iostream>
#include<string>
using namespace std;

int main() {
    string x = "dd2940c04462b4dd7c450528835cca15";
    x[2] = (x[2] + x[3]) - 50;
    x[4] = (x[2] + x[5]) - 48;
    x[30] = (x[31] + x[9]) - 48;
    x[14] = (x[27] + x[28]) - 97;
    for (int i = 0; i < 16; i++) {
        char a = x[31 - i];
        x[31 - i] = x[i];
        x[i] = a;
    }

    cout << "flag{" << x << "}" << endl;
}
```

[GWCTF 2019]pyre

python逆向

`python -m pip install uncompyle`

`uncompyle6 ./attachment.pyc > rev.py`

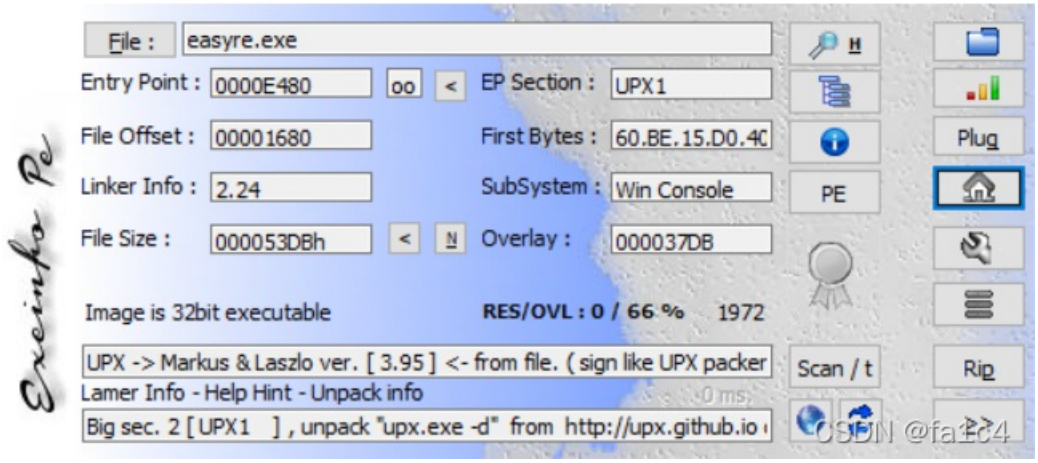
```
# uncompyle6 version 3.8.0
# Python bytecode 2.7 (62211)
# Decompiled from: Python 3.8.10 (tags/v3.8.10:3d8993a, May  3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)]
# Embedded file name: encode.py
# Compiled at: 2019-08-19 21:01:57
print 'Welcome to Re World!'
print 'Your input1 is your flag~'
l = len(input1)
for i in range(l):
    num = ((input1[i] + i) % 128 + 128) % 128
    code += num

for i in range(l - 1):
    code[i] = code[i] ^ code[(i + 1)]

print code
code = ['\x1f', '\x12', '\x1d', '(', '0', '4', '\x01', '\x06', '\x14', '4', ',', '\x1b', 'U', '?', 'o', '6', '*'
, ':', '\x01', 'D', ';', '%', '\x13']
# okay decompiling .\attachment.pyc
```

从后往前逆回去就行, `code += num`看起来不合理, 先不理它

exp

```
code = ['\x1f', '\x12', '\x1d', '(', '0', '4', '\x01', '\x06', '\x14', '4', ',', '\x1b', 'U', '?', 'o', '6', '*'
, ':', '\x01', 'D', ';', '%', '\x13']

l = len(code)

for i in range(l - 2, -1, -1):
    code[i] = chr(ord(code[i]) ^ ord(code[i + 1]))

for i in range(l):
    code[i] = chr((ord(code[i]) - i) % 128)

flag = ""
for i in range(l):
    flag += code[i]
print(flag)
```

[ACTF新生赛2020]easyre



脱壳, 拖进IDA

`upx -d easyre.exe`

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  _BYTE v4[12]; // [esp+12h] [ebp-2Eh] BYREF
  _DWORD v5[3]; // [esp+1Eh] [ebp-22h]
  _BYTE v6[5]; // [esp+2Ah] [ebp-16h] BYREF
  int v7; // [esp+2Fh] [ebp-11h]
  int v8; // [esp+33h] [ebp-Dh]
  int v9; // [esp+37h] [ebp-9h]
  char v10; // [esp+3Bh] [ebp-5h]
  int i; // [esp+3Ch] [ebp-4h]

  __main();
  qmemcpy(v4, "*F'\"N,\"(I?+@", sizeof(v4));
  printf("Please input:");
  scanf("%s", v6);
  if ( v6[0] != 65 || v6[1] != 67 || v6[2] != 84 || v6[3] != 70 || v6[4] != 123 || v10 != 125 )
    return 0;
  v5[0] = v7;
  v5[1] = v8;
  v5[2] = v9;
  for ( i = 0; i <= 11; ++i )
  {
    if ( v4[i] != _data_start__[*((char *)v5 + i) - 1] )
      return 0;
  }
  printf("You are correct!");
  return 0;
}
```

```
.data:00402000 ; char _data_start__[]
.data:00402000 __data_start__  db '~'                  ; DATA XREF: _main+EC↑r
.data:00402001 aZyxwvutsrqponm db '}|{zyxwvutsrqponmlkjihgfedcba`_^]\[ZYXWVUTSRQPONMLKJIHGFEDCBA@?>='
.data:00402001                 db '<;:9876543210/.-,+*)(',27h,'&%$# !"',0
.data:00402060                 align 40h
```

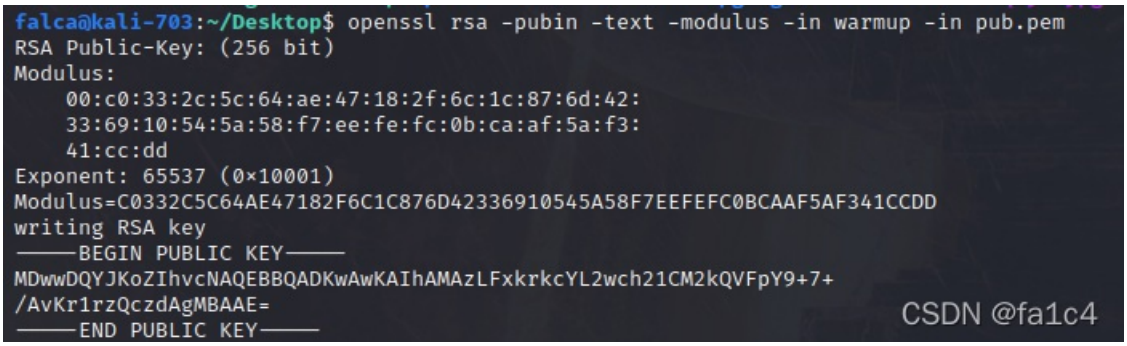
强逆, 可从hex窗口拷贝string


```
teststr = "*F'\"N,\"(I?+@"
print(len(teststr))

data = "~}|{zyxwvutsrqponmlkjihgfedcba`_^]\[ZYXWVUTSRQPONMLKJIHGFEDCBA@?>="
data += "<;:9876543210/.-,+*)("
data += chr(0x27)
data += "&%$# !\""

indexs = []
for ch in teststr:
    for i in range(len(data)):
        if ch == data[i]:
            indexs.append(chr(i + 1))
            print(i + 1)

flag = "".join(indexs)
print("ACTF{" + flag + "}")
print("flag{" + flag + "}")
```

rsa

密码学题目, 怎么丢进reverse来了…

pub.key改成pem文件, 拖进kali用openssl



n = 869344822960481191906660620034948005889056560172030256172166540583783221035 17
http://www.factordb.com/index.php?query=869344822960481191906660620034948005889056560172030256172166540583783221035 17
分解得到p, q
p = 285960468890451637935629440372639283459
q = 304008741604601924494328155975272418463

解密exp

```
import gmpy2
import rsa

e = 65537
n = 869344822960481191906660620034948005889056560172030256172166540583783221035 17
p = 285960468890451637935629440372639283459
q = 304008741604601924494328155975272418463
d = gmpy2.invert(e, (q-1)*(p-1))

key = rsa.PrivateKey(n, e, int(d), p, q)

with open("./flag.enc", "rb+") as f:
    f = f.read()
    print(rsa.decrypt(f, key))
```

CrackRTF

```
int __cdecl main_0(int argc, const char **argv, const char **envp)
{
  DWORD v3; // eax
  DWORD v4; // eax
  char Str[260]; // [esp+4Ch] [ebp-310h] BYREF
  int v7; // [esp+150h] [ebp-20Ch]
  char String1[260]; // [esp+154h] [ebp-208h] BYREF
  char Destination[260]; // [esp+258h] [ebp-104h] BYREF

  memset(Destination, 0, sizeof(Destination));
  memset(String1, 0, sizeof(String1));
  v7 = 0;
  printf("pls input the first passwd(1): ");
  scanf("%s", Destination);
  if ( strlen(Destination) != 6 )
  {
    printf("Must be 6 characters!\n");
    ExitProcess(0);
  }
  v7 = atoi(Destination);
  if ( v7 < 100000 )
    ExitProcess(0);
  strcat(Destination, "@DBApp");
  v3 = strlen(Destination);
  sub_40100A((BYTE *)Destination, v3, String1);
  if ( !_strcmpi(String1, "6E32D0943418C2C33385BC35A1470250DD8923A9") )
  {
    printf("continue...\n\n");
    printf("pls input the first passwd(2): ");
    memset(Str, 0, sizeof(Str));
    scanf("%s", Str);
    if ( strlen(Str) != 6 )
    {
      printf("Must be 6 characters!\n");
      ExitProcess(0);
    }
    strcat(Str, Destination);
    memset(String1, 0, sizeof(String1));
    v4 = strlen(Str);
    sub_401019((BYTE *)Str, v4, String1);
    if ( !_strcmpi("27019e688a4e62a649fd99cadaafdb4e", String1) )
    {
      if ( !(unsigned __int8)sub_40100F(Str) )
      {
        printf("Error!!\n");
        ExitProcess(0);
      }
      printf("bye ~~\n");
    }
  }
  return 0;
}
```

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

  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
  {
    CryptReleaseContext(phProv, 0);
    result = 0;
  }
  return result;
}
```

`sub_401230`用到了一个windows加密的加密库函数

观察发现`6E32D0943418C2C33385BC35A1470250DD8923A9`是40位的字符串

猜测可能是sha1加密

用sha1爆破一下6位密码

```python
import hashlib

hash1 = "6E32D0943418C2C33385BC35A1470250DD8923A9".lower()
hash2 = "27019e688a4e62a649fd99cadaafdb4e".lower()
flag = "@DBApp"

for i in range(100000, 999999):
    s = str(i) + flag
    x = hashlib.sha1(s.encode())
    if x.hexdigest() == hash1:
        flag = s
        print(flag)
        break
```

> 123321@DBApp

验证通过

```
pls input the first passwd(1): 123321
continue...

pls input the first passwd(2):
```

第二次hash是32位的字符串, 猜测是MD5, 但是直接全字符爆破不实际

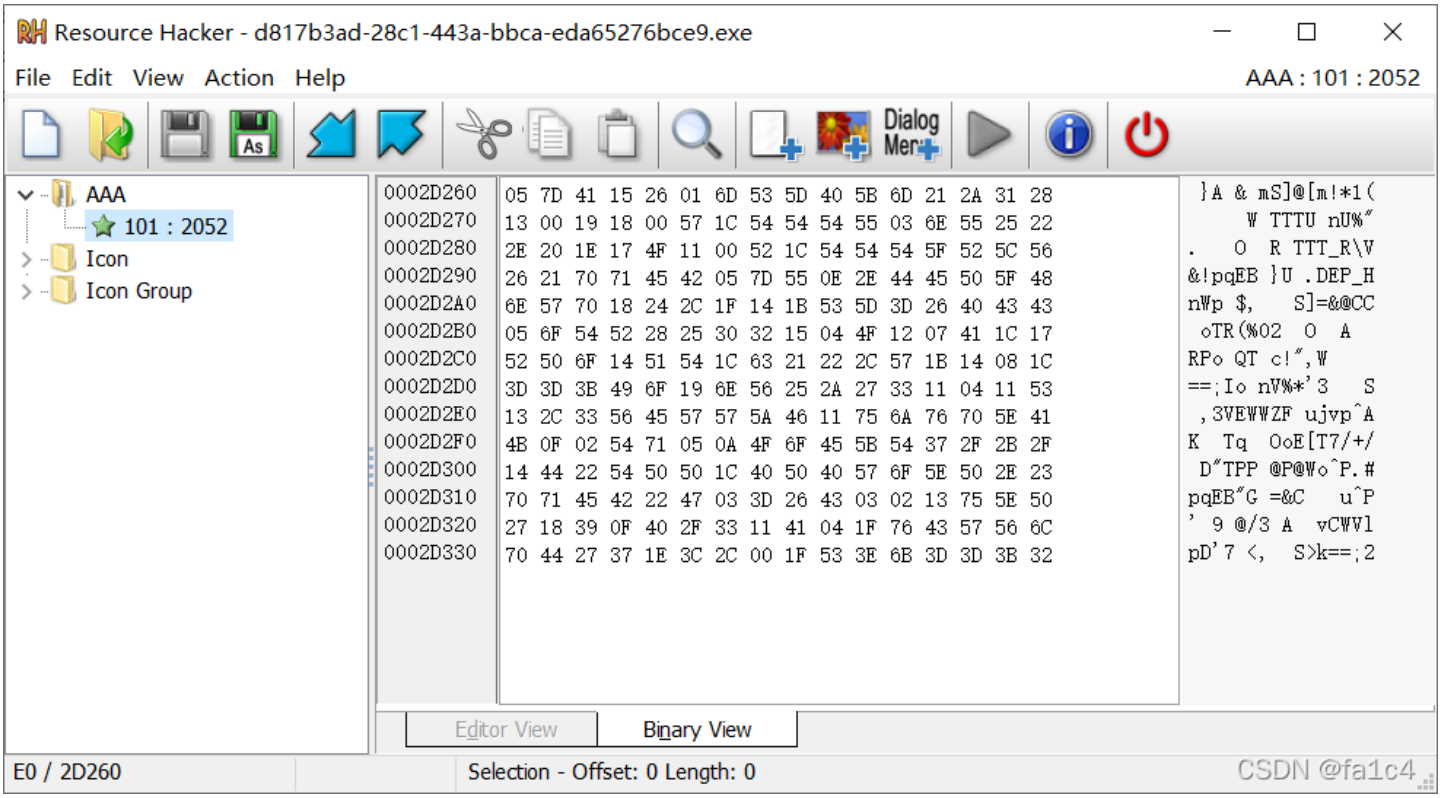
分析其他线索

```
char __cdecl sub_4014D0(LPCSTR lpString)
{
  LPCVOID lpBuffer; // [esp+50h] [ebp-1Ch]
  DWORD NumberOfBytesWritten; // [esp+58h] [ebp-14h] BYREF
  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;
}
```

```
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;
}
```

从"AAA"resource中读取数据, 然后和输入的6个字符异或得到结果, 写入`.rtf`文件, 因为rtf文件头是确定的, 所以只要找到AAA中的数据就能得到第二个密钥

用resource-hacker



.rtf文件头

写exp解密第二个key, 这里rtf头有5字节, 往后多取一个字节, 为0x31

```
rtf_header = [0x7B, 0x5C, 0x72, 0x74, 0x66, 0x31]
arr = [0x05, 0x7D, 0x41, 0x15, 0x26, 0x01]

key2 = ''
for i in range(6):
    key2 += chr(rtf_header[i] ^ arr[i])
print(key2)
```

> ~!3a@0

输入exe的第二个key, 生成rtf文件, 用wps打开得到flag