BUUCTF reverse21-25 WP

原创

是Mumuzi 于 2021-04-04 11:10:53 发布 201 收藏 1

分类专栏： ctf reverse buuctf 文章标签： 反编译 信息安全

本文链接：https://blog.csdn.net/qq_42880719/article/details/115341087

ctf 同时被 3 个专栏收录

75 篇文章 28 订阅

订阅专栏

reverse

8 篇文章 4 订阅

订阅专栏

buuctf

15 篇文章 2 订阅

订阅专栏

写脚本技术很菜，所以和其他wp相比看起来比较麻烦

笔记是按照当时的题目排序写的，顺序可能有出入

做题是从0开始做起，所以前面部分也会尽可能写的详细一点

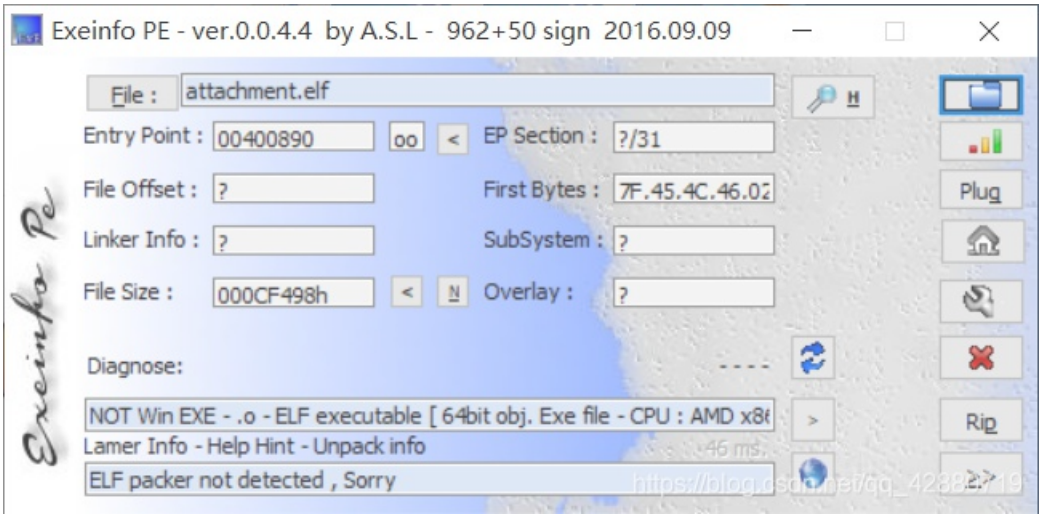
只要能记录并且了解到怎么做即可，所以就没有去在意排版

遇到不会的函数尽可能去百度了解

题目：[2019红帽杯]easyRE、Youngter-drive、[ACTF新生赛2020]rome、[FlareOn4]login、[GUET-CTF2019]re

21.[2019纨幃杵]easyRE

憟谨 purecall 帆傋揖価颞直~徝制皊 flag 诽匋丐 flag{} 揖弈ザ



无壳，IDA打开，搜索字符串找到you found me，然后跟进跳转，F5反编译查看函数（sub_4009C6）

```
v18 = 110;
v19 = 98;
v20 = 40;
v21 = 111;
v22 = 99;
v23 = 121;
v24 = 127;
v25 = 121;
v26 = 46;
v27 = 105;
v28 = 127;
v29 = 100;
v30 = 96;
v31 = 51;
v32 = 119;
v33 = 125;
v34 = 119;
v35 = 101;
v36 = 107;
v37 = 57;
v38 = 123;
v39 = 105;
v40 = 121;
v41 = 61;
v42 = 126;
v43 = 121;
v44 = 76;
v45 = 64;
v46 = 69;
v47 = 67;
memset(v48, 0, sizeof(v48));
v49 = 0;
v50 = 0;
sub_4406E0(0LL, v48, 37LL);
v50 = 0;
if ( sub_424BA0(v48) == 36 )
{
  for ( i = 0; i < (unsigned __int64)sub_424BA0(v48); ++i )
  {
    if ( (unsigned __int8)(v48[i] ^ i) != *(&v12 + i) )
    {
      result = 4294967294LL;
      goto LABEL_13;
    }
  }
  sub_410CC0("continue!");
  memset(&v51, 0, 0x40uLL);
  v53 = 0;
  sub_4406E0(0LL, &v51, 64LL);
  v52 = 0;
  if ( sub_424BA0(&v51) == 39 )
  {
    v1 = sub_400E44(&v51);
    v2 = sub_400E44(v1);
    v3 = sub_400E44(v2);
    v4 = sub_400E44(v3);
    v5 = sub_400E44(v4);
```

```
      v6 = sub_400E44(v5);
      v7 = sub_400E44(v6);
      v8 = sub_400E44(v7);
      v9 = sub_400E44(v8);
      v10 = sub_400E44(v9);
      if ( !(unsigned int)sub_400360(v10, off_6CC090) )
      {
        sub_410CC0("You found me!!!");
        sub_410CC0("bye bye~");
      }
      result = 0LL;
    }
    else
    {
      result = 4294967293LL;
    }
  }
  else
  {
    result = 0xFFFFFFFFLL;
  }
LABEL_13:
  if ( __readfsqword(0x28u) != v54 )
    sub_444020();
  return result;
}
```

首先简单分成两部分，第一部分是continue前面

```
LODWORD(v2) = sub_424BA0((const __m128i *)v53);
if ( v2 == 36 )
{
  for ( i = 0; ; ++i )
  {
    v1 = v53;
    LODWORD(v4) = sub_424BA0((const __m128i *)v53);
    if ( i >= v4 )
      break;
    if ( (unsigned __int8)(v53[i] ^ i) != *(&v17 + i) )
    {
      result = 4294967294LL;
      goto LABEL_13;
    }
  }
  sub_410CC0((const __m128i *)"continue!");
```


可以看到，与i异或之后得到从v17-v52，所以将v17-v52依次与i异或，就可以还原加密前的函数

```
enc = [73,111,100,108,62,81,110,98,40,111,99,121,127,121,46,105,127,100,96,51,119,125,119,101,107,57,123,105,121
,61,126,121,76,64,69,67]
flag = ''
for i in range(len(enc)):
    flag += chr(enc[i]^i)
print(flag)
```

```
C:\Users\mumuzi\PycharmProjects\python
Info:The first four chars are `flag`
```

解密后得到提示，前四个字符是flag，下面继续分析后半部分

```
v6 = (const __m128i *)sub_400E44((const __m128i *)&v56);
v7 = (const __m128i *)sub_400E44(v6);
v8 = (const __m128i *)sub_400E44(v7);
v9 = (const __m128i *)sub_400E44(v8);
v10 = (const __m128i *)sub_400E44(v9);
v11 = (const __m128i *)sub_400E44(v10);
v12 = (const __m128i *)sub_400E44(v11);
v13 = (const __m128i *)sub_400E44(v12);
v14 = (const __m128i *)sub_400E44(v13);
```

跳转之后查看，发现这串是对指定字符串进行base64解码，手动解码后得到https://bbs.pediy.com/thread-254172.htm（？）

Base64 编码或解码的结果:

https://bbs.pediy.com/thread-254172.htm

好像没什么用，继续查看后面

```
v15 = sub_400E44(v14);
v0 = off_6CC090;
v1 = (char *)v15;
if ( !(unsigned int)sub_400360(v15, off_6CC090) )
{
  sub_410CC0((const __m128i *)"You found me!!!");
  v1 = "bye bye~";
  sub_410CC0((const __m128i *)"bye bye~");
```

这里一直找都不知道找什么，去看了其他师傅的wp，说是最后解码的base64串后面有一段常量，跟进常量之后发现被sub_400D35引用（？为什么要这样找啊看了几篇WP都没解释），查看sub_400D35

```
v9 = __readfsqword(0x28u);
  v2 = 0LL;
  v5 = sub_43FD20(0LL) - qword_6CEE38;
  for ( i = 0; i <= 1233; ++i )
  {
    v2 = v5;
    sub_40F790(v5);
    sub_40FE60();
    sub_40FE60();
    v5 = (unsigned __int64)sub_40FE60() ^ 0x98765432;
  }
  v8 = v5;
  if ( ((unsigned __int8)v5 ^ byte_6CC0A0[0]) == 102 && (HIBYTE(v8) ^ (unsigned __int8)byte_6CC0A3) == 103 )
  {
    for ( j = 0; j <= 24; ++j )
    {
      v2 = (unsigned __int8)(byte_6CC0A0[j] ^ *((_BYTE *)&v8 + j % 4));
      sub_410E90(v2);
    }
  }
  v4 = __readfsqword(0x28u);
  result = v4 ^ v9;
  if ( v4 != v9 )
    sub_444020(v2, a2);
  return result;
}
```

查看关键部分

```
  v8 = v5;
  if ( ((unsigned __int8)v5 ^ byte_6CC0A0[0]) == 'f' && (HIBYTE(v8) ^ (unsigned __int8)byte_6CC0A3) == 'g' )
  {
    for ( j = 0; j <= 24; ++j )
    {
      v2 = (unsigned __int8)(byte_6CC0A0[j] ^ *((_BYTE *)&v8 + j % 4));
      sub_410E90(v2);
    }
  }
  v4 = __readfsqword(0x28u);
  result = v4 ^ v9;
  if ( v4 != v9 )
    sub_444020(v2, a2);
  return result;
```


v5与byte_6CC0A0[0]异或为f，v8与byte_6CC0A3异或之后为g，根据前面得到的信息，v5-v8异或之后应该为“flag”，所以可以反过来推敲出一个“key”，然后这个“key”再去和之前看到的常量进行异或，就可以得到flag了，写脚本：

```
enc2 = [0x40,0x35,0x20,0x56,0x5d,0x18,0x22,0x45,0x17,0x2f,0x24,0x6e,0x62,0x3c,0x27,0x54,0x48,0x6c,0x24,0x6e,0x72
,0x3c,0x32,0x45,0x5b]
key = ''
realflag = ''
enc3 = 'flag'
for j in range(4):
    key += chr(ord(enc3[j])^enc2[j])
for j in range(len(enc2)):
    realflag += chr(ord(key[j%4])^enc2[j])
print(realflag)
```

```
C:\Users\mumuzi\PycharmProjects\pythonPr
Info:The first four chars are `flag`
flag{Act1ve_Defen5e_Test}
```

> flag{Act1ve_Defen5e_Test}

22. Youngter-drive

幺轮帢舌杀丐践二 泮愕 x 值制帢 flag 诽匈丐 flag{} 揖奕

下载之后告诉是高危文件)
先查壳，发现有UPX壳，脱壳即可，然后32位IDA打开
查找字符串，找到input your flag，跟进跳转

```
int sub_411BD0()
{
  printf(
    "1111111111111111111111111111111111111111111111111111111111111111111111111111111111\n"
    "********************************************************************************\n"
    "***************             ************************************************\n"
    "***************   ********   ***********************                ************\n"
    "***************   *********  ***********************   ****************************\n"
    "***************   *********  ***********************   ****************************\n"
    "***************   *********  ***********************   ****************************\n"
    "***************   *******   ************************   ****************************\n"
    "***************   ****   **************************   ****************************\n"
    "***************   *     ****************************                ************\n"
    "***************   ***    ***************************   ****************************\n"
    "***************   ******   *************************   ****************************\n"
    "***************   ********   ***********************   ****************************\n"
    "***************   **********   *********************   ****************************\n"
    "***************   ***********    ********************                ************\n"
    "********************************************************************************\n"
    "1111111111111111111111111111111111111111111111111111111111111111111111111111111111\n");
  printf("input flag:\n");
  return scanf("%36s", &Source);
}
```


emm，好像没啥用，还是去看main函数

```
int __thiscall main_0(void *this)
{
  HANDLE v2; // [esp+D0h] [ebp-14h]
  HANDLE hObject; // [esp+DCh] [ebp-8h]

  sub_4110FF(this);
  ::hObject = CreateMutexW(0, 0, 0);
  j_strcpy(Dest, &Source);
  hObject = CreateThread(0, 0, StartAddress, 0, 0, 0);
  v2 = CreateThread(0, 0, sub_41119F, 0, 0, 0);
  CloseHandle(hObject);
  CloseHandle(v2);
  while ( dword_418008 != -1 )
    ;
  sub_411190();
  CloseHandle(::hObject);
  return 0;
}
```

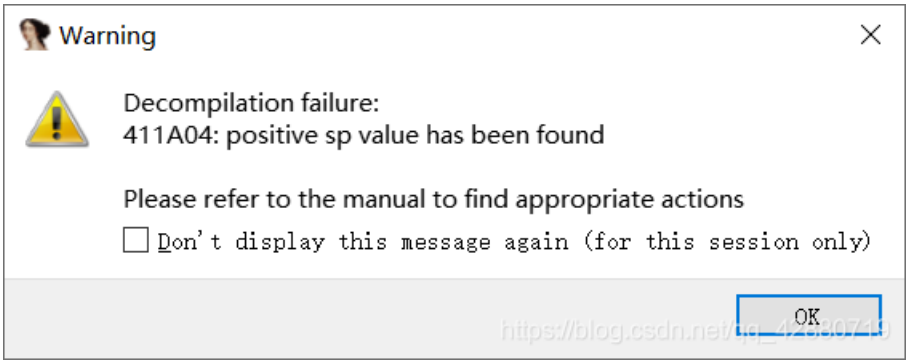
首先进入sub_4110FF函数（就是刚刚的那个）输入字符串，然后把输入的字符串粘贴到Dest，之后进入StartAddress进行加密，进入sub_41119F加密，最后进入sub_411190
查看411190

```
  for ( i = 0; i < 29; ++i )
  {
    if ( Source[i] != off_418004[i] )
      exit(0);
  }
  return printf("\nflag{%s}\n\n", Dest);
}
```

发现就是将变换后的函数相比较，其中off_418004中内容为TOiZiZtOrYaToUwPnToBsOaOapsyS，正确即输出flag，flag内容即Dest内容。
查看StartAddress函数

```
1  void __stdcall StartAddress_0(int a1)
2  {
3    while ( 1 )
4    {
5      WaitForSingleObject(hObject, 0xFFFFFFFF);
6      if ( dword_418008 > -1 )
7      {
8        sub_41112C(Source, dword_418008);
9        --dword_418008;
10       Sleep(0x64u);
11     }
12     ReleaseMutex(hObject);
13   }
14 }
```


跟进sub_41112C函数，但是打开sub_411940函数的时候，出现错误提示，萌新第一次遇见这种东西，看了看其他师傅的wp，意思是说堆栈不平衡导致的无法反编译



> 所谓堆栈平衡，就是说在函数开始和结束时，栈顶指针 SP 必须指向同一个地方，否则称为堆栈不平衡或堆栈平衡破坏。我们知道函数的局部变量是在栈上分配，所谓分配其实就是抬高栈顶，减少 SP 的值，划出一块内存空间给局部变量用。函数结束时，要回收分配的空间，也就是降低栈顶，增加 SP 的值。当时分配了多少空间，就应当回收多少，因此 SP 指向的地方应当是不变的。IDA 的做法是在函数开始时，假设 SP 为 0，函数中间可能会增减 SP，最后结束时 SP 应当回到 0，而这里 IDA 识别出现了错误，SP 大于 0，因此报错。这种情况下一般有两种办法：一是直接看汇编，不看伪代码了，IDA 的反汇编还是有保证的；二是手动修复 SP。
> (https://hx1997.github.io/2018/07/22/anheng-july-re-youngter-drive/）

```
004119F1
004119F1      loc_4119F1:
004119F1 0D8 pop      edi
004119F2 0D4 pop      esi
004119F3 0D0 pop      ebx
004119F4 0CC add      esp, 0CCh
004119FA 000 cmp      ebp, esp
004119FC 000 call     j___RTC_CheckEsp
00411A01 000 mov      esp, ebp
00411A03 000 pop      ebp
00411A04 -04 retn
00411A04      sub_411940 endp ; sp-analysis failed
00411A04
00411A04      ; ---------------------------------------------------------------------
```


看不到堆栈的话，option->general,把Stack pointer复选的勾打上即可
找到这个，按 Alt+K，把-04改成00，再按F5，成功查看伪代码

```
000 pop
000 retn
```

```
{
  char *result; // eax
  char v3; // [esp+D3h] [ebp-5h]

  v3 = *(_BYTE *)(a2 + a1);
  if ( (v3 < 97 || v3 > 122) && (v3 < 65 || v3 > 90) )
    exit(0);
  if ( v3 < 97 || v3 > 122 )
  {
    result = off_418000[0];
    *(_BYTE *)(a2 + a1) = off_418000[0][*(char *)(a2 + a1) - 38];
  }
  else
  {
    result = off_418000[0];
    *(_BYTE *)(a2 + a1) = off_418000[0][*(char *)(a2 + a1) - 96];
  }
  return result;
}
```


其中a1是source，a2是dword_418008（1Dh=29）
v3即为source的ascii码+a2，根据前面的函数可以知道，每进入一次411940的时候，a2的值就会减少1
这里还是看了其它师傅的wp，说两个线程是交替执行的

> CreateThread API 会创建新线程，这道题涉及到多线程。CreateMutex 创建一个互斥体，用于防止多线程中出现资源争用，即多个线程同时读写同一个资源的情况，所创建的互斥体的句柄会存到全局变量 hObject 中（注意前面的两个冒号表示是全局变量，而不是这个函数里同名的局部变量）。这里创建了两个线程，入口点分别位于函数 StartAddress 和 sub_41119F，且这两个函数都没有传入参数.
> 查 MSDN 知，可以用 WaitForSingleObject 等待互斥体的使用权（ownership）空闲出来，并获取使用权，然后再访问和其他线程共享的资源，访问完后，用 ReleaseMutex 释放使用权，给其他线程使用的机会。通过比较两线程的函数，很容易知道所共享的资源就是全局变量 dword_418008，它的初值是 29。而这两个线程一前一后创建，理论上是 StartAddress 先获得使用权，后来的 sub_41119F 进入等待状态，前者执行一次循环后释放使用权，与此同时后者等待结束、获得使用权，进入循环，循环完后释放使用权，前者又获得使用权，如此循环往复。也就是说，两个线程的操作是交替进行的
> (https://hx1997.github.io/2018/07/22/anheng-july-re-youngter-drive/)

既然已经知道是交互进行的，并且查看sub_41119F，并不会调用sub_41112C，而是直接将dword_418008的值减1，所以可以写解密脚本

```
alp = 'QWERTYUIOPASDFGHJKLZXCVBNMqwertyuiopasd'
enc = "TOiZiZtOrYaToUwPnToBsOaOapsyS"
flag = ''
for i in range(len(enc)):
    if(i%2==0):
        flag += enc[i]
    else:
        if enc[i] in alp[1:26]:
            flag += chr(alp.find(enc[i]) + 96)
        if enc[i] in alp[27:52]:
            flag += chr(alp.find(enc[i]) + 38)
print(flag)
```

```
C:\Users\mumuzi\PycharmProjects\p
ThisisthreadofwindowshahaIsES
```

ThisisthreadofwindowshahaIsES
可是，交不上，又只能跑去看其它师傅们的博客，说是需要30位，但是只有29位，所以只能自己手动试？网上说是E，那就把E添上去提交吧

> flag{ThisisthreadofwindowshahaIsESE}

23.[ACTF闊甥赡2020]rome

值制皆 flag 诽匈丐 flag{} 揖弈ザ

下载，拖进winhex，没UPX字样，扔进IDA，找到func函数

```
int func()
{
  int result; // eax
  int v1; // [esp+14h] [ebp-44h]
  int v2; // [esp+18h] [ebp-40h]
  int v3; // [esp+1Ch] [ebp-3Ch]
  int v4; // [esp+20h] [ebp-38h]
  unsigned __int8 v5; // [esp+24h] [ebp-34h]
  unsigned __int8 v6; // [esp+25h] [ebp-33h]
  unsigned __int8 v7; // [esp+26h] [ebp-32h]
  unsigned __int8 v8; // [esp+27h] [ebp-31h]
  unsigned __int8 v9; // [esp+28h] [ebp-30h]
  int v10; // [esp+29h] [ebp-2Fh]
  int v11; // [esp+2Dh] [ebp-2Bh]
  int v12; // [esp+31h] [ebp-27h]
  int v13; // [esp+35h] [ebp-23h]
  unsigned __int8 v14; // [esp+39h] [ebp-1Fh]
  char v15; // [esp+3Bh] [ebp-1Dh]
  char v16; // [esp+3Ch] [ebp-1Ch]
  char v17; // [esp+3Dh] [ebp-1Bh]
  char v18; // [esp+3Eh] [ebp-1Ah]
  char v19; // [esp+3Fh] [ebp-19h]
  char v20; // [esp+40h] [ebp-18h]
  char v21; // [esp+41h] [ebp-17h]
  char v22; // [esp+42h] [ebp-16h]
```

```
char v23; // [esp+43h] [ebp-15h]
char v24; // [esp+44h] [ebp-14h]
char v25; // [esp+45h] [ebp-13h]
char v26; // [esp+46h] [ebp-12h]
char v27; // [esp+47h] [ebp-11h]
char v28; // [esp+48h] [ebp-10h]
char v29; // [esp+49h] [ebp-Fh]
char v30; // [esp+4Ah] [ebp-Eh]
char v31; // [esp+4Bh] [ebp-Dh]
int i; // [esp+4Ch] [ebp-Ch]

v15 = 81;
v16 = 115;
v17 = 119;
v18 = 51;
v19 = 115;
v20 = 106;
v21 = 95;
v22 = 108;
v23 = 122;
v24 = 52;
v25 = 95;
v26 = 85;
v27 = 106;
v28 = 119;
v29 = 64;
v30 = 108;
v31 = 0;
printf("Please input:");
scanf("%s", &v5);
result = v5;
if ( v5 == 65 )
{
  result = v6;
  if ( v6 == 67 )
  {
    result = v7;
    if ( v7 == 84 )
    {
      result = v8;
      if ( v8 == 70 )
      {
        result = v9;
        if ( v9 == 123 )
        {
          result = v14;
          if ( v14 == 125 )
          {
            v1 = v10;
            v2 = v11;
            v3 = v12;
            v4 = v13;
            for ( i = 0; i <= 15; ++i )
            {
              if ( *((_BYTE *)&v1 + i) > 64 && *((_BYTE *)&v1 + i) <= 90 )
                *((_BYTE *)&v1 + i) = (*((char *)&v1 + i) - 51) % 26 + 65;
              if ( *((_BYTE *)&v1 + i) > 96 && *((_BYTE *)&v1 + i) <= 122 )
                *((_BYTE *)&v1 + i) = (*((char *)&v1 + i) - 79) % 26 + 97;
            }
```

```
              for ( i = 0; i <= 15; ++i )
              {
                result = (unsigned __int8)*(&v15 + i);
                if ( *((_BYTE *)&v1 + i) != (_BYTE)result )
                  return result;
              }
              result = printf("You are correct!");
            }
          }
        }
      }
    }
  }
  return result;
}
```

分析代码

ASCII在65-90之间即大写字母，97-122即小写字母，就是将大写字母和小写字母分开进行了变换，所以可以直接进行爆破加密前的字符。

```
enc = [81,115,119,51,115,106,95,108,122,52,95,85,106,119,64,108]
flag = ''
for i in range(len(enc)):
    for j in range(0,128):
        k=j
        if(j > 64) and (j<=90):
            j = (j-51)%26+65
        if(j > 96) and (j<=122):
            j = (j-79)%26+97
        if(j==enc[i]):
            flag += chr(k)
print('flag{'+flag+'}')
```

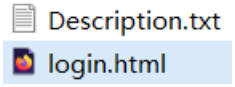
```
C:\Users\mumuzi\PycharmPr
flag{Cae3ar_th4_Gre@t}
```

> flag{Cae3ar_th4_Gre@t}

24. [FlareOn4]login

值制岭 flag 诽匈丐 flag{} 揖弈ザ

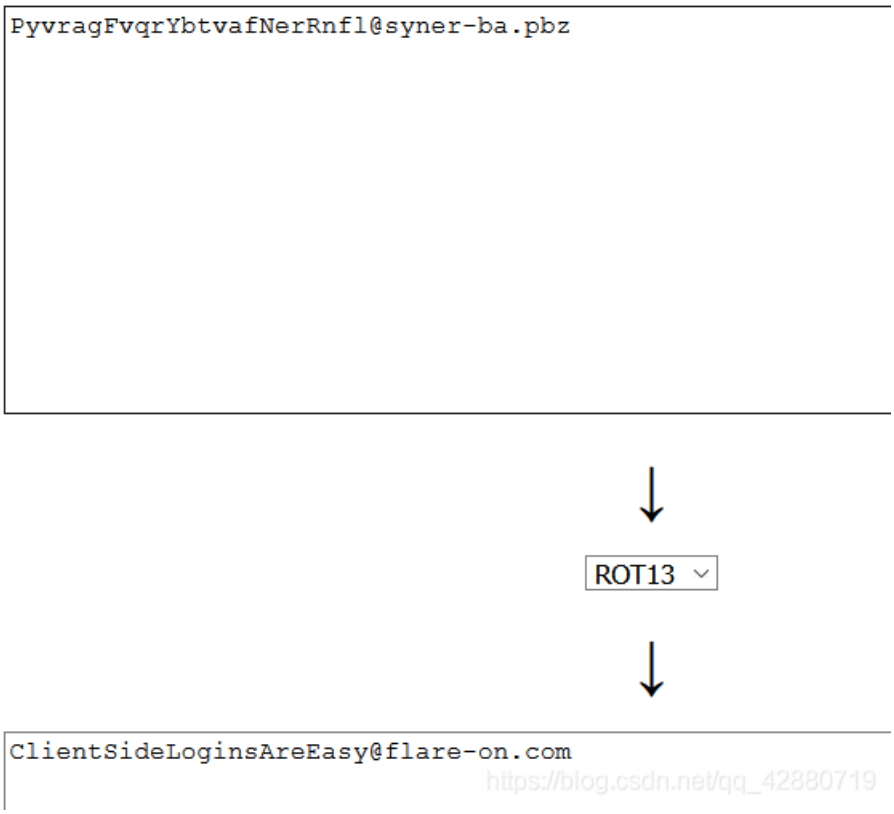
下载得到，用notepad++打开



直接查看加密部分

```
document.getElementById("prompt").onclick = function () {
    var flag = document.getElementById("flag").value;
    var rotFlag = flag.replace(/[a-zA-Z]/g, function(c){return String.fromCharCode((c <= "Z" ? 90 : 122) >= (c = c.charCodeAt(0) + 13) ? c : c - 26);});
    if ("PyvragFvqrYbtvafNerRnfl@syner-ba.pbz" == rotFlag) {
        alert("Correct flag!");
```

可以发现rotflag要和那个字符串进行比较，加密部分是判断C是否为大写，判断之后再判断+13是否大于C，大于即超出90，要减去26，是明显的rot13编码，直接找一个在线网站即可



> flag{ClientSideLoginsAreEasy@flare-on.com}

25. [GUET-CTF2019]re

值制岭 flag 诽匈丐 flag{} 揖弈ザ

加了壳，upx -d脱壳即可

搜字符串找到input your flag，找到函数，F5大法

要求输入字符串，进入sub_4009AE然后判断

```
_BOOL8 __fastcall sub_4009AE(char *a1)
{
  if ( 1629056 * *a1 != 166163712 )
    return 0LL;
  if ( 6771600 * a1[1] != 731332800 )
```

```
  return 0LL;
if ( 3682944 * a1[2] != 357245568 )
  return 0LL;
if ( 10431000 * a1[3] != 1074393000 )
  return 0LL;
if ( 3977328 * a1[4] != 489211344 )
  return 0LL;
if ( 5138336 * a1[5] != 518971936 )
  return 0LL;
if ( 7532250 * a1[7] != 406741500 )
  return 0LL;
if ( 5551632 * a1[8] != 294236496 )
  return 0LL;
if ( 3409728 * a1[9] != 177305856 )
  return 0LL;
if ( 13013670 * a1[10] != 650683500 )
  return 0LL;
if ( 6088797 * a1[11] != 298351053 )
  return 0LL;
if ( 7884663 * a1[12] != 386348487 )
  return 0LL;
if ( 8944053 * a1[13] != 438258597 )
  return 0LL;
if ( 5198490 * a1[14] != 249527520 )
  return 0LL;
if ( 4544518 * a1[15] != 445362764 )
  return 0LL;
if ( 3645600 * a1[17] != 174988800 )
  return 0LL;
if ( 10115280 * a1[16] != 981182160 )
  return 0LL;
if ( 9667504 * a1[18] != 493042704 )
  return 0LL;
if ( 5364450 * a1[19] != 257493600 )
  return 0LL;
if ( 13464540 * a1[20] != 767478780 )
  return 0LL;
if ( 5488432 * a1[21] != 312840624 )
  return 0LL;
if ( 14479500 * a1[22] != 1404511500 )
  return 0LL;
if ( 6451830 * a1[23] != 316139670 )
  return 0LL;
if ( 6252576 * a1[24] != 619005024 )
  return 0LL;
if ( 7763364 * a1[25] != 372641472 )
  return 0LL;
if ( 7327320 * a1[26] != 373693320 )
  return 0LL;
if ( 8741520 * a1[27] != 498266640 )
  return 0LL;
if ( 8871876 * a1[28] != 452465676 )
  return 0LL;
if ( 4086720 * a1[29] != 208422720 )
  return 0LL;
if ( 9374400 * a1[30] == 515592000 )
  return 5759124 * a1[31] == 719890500;
return 0LL;
}
```

算出来就好了，而且没有a1[6]，需要手动爆破

```
a1 = chr(166163712 // 1629056)
a2 = chr(731332800 // 6771600)
a3 = chr(357245568 // 3682944)
a4 = chr(1074393000 // 10431000)
a5 = chr(489211344 // 3977328)
a6 = chr(518971936 // 5138336)

a8 = chr(406741500 // 7532250)
a9 = chr(294236496 // 5551632)
a10 = chr(177305856 // 3409728)
a11 = chr(650683500 // 13013670)
a12 = chr(298351053 // 6088797)
a13 = chr(386348487 // 7884663)
a14 = chr(438258597 // 8944053)
a15 = chr(249527520 // 5198490)
a16 = chr(445362764 // 4544518)
a17 = chr(981182160 // 10115280)
a18 = chr(174988800 // 3645600)
a19 = chr(493042704 // 9667504)
a20 = chr(257493600 // 5364450)
a21 = chr(767478780 // 13464540)
a22 = chr(312840624 // 5488432)
a23 = chr(1404511500 // 14479500)
a24 = chr(316139670 // 6451830)
a25 = chr(619005024 // 6252576)
a26 = chr(372641472 // 7763364)
a27 = chr(373693320 // 7327320)
a28 = chr(498266640 // 8741520)
a29 = chr(452465676 // 8871876)
a30 = chr(208422720 // 4086720)
a31 = chr(515592000 // 9374400)
a32 = chr(719890500 // 5759124)
print (a1+a2+a3+a4+a5+a6,a8+a9+a10+a11+a12+a13+a14+a15+a16+a17+a18+a19+a20+a21+a22+a23+a24+a25+a26+a27+a28+a29+a
30+a31+a32)
```

得到flag{e 65421110ba03099a1c039337}
中间那位拿去试，得到1

> flag{e165421110ba03099a1c039337}