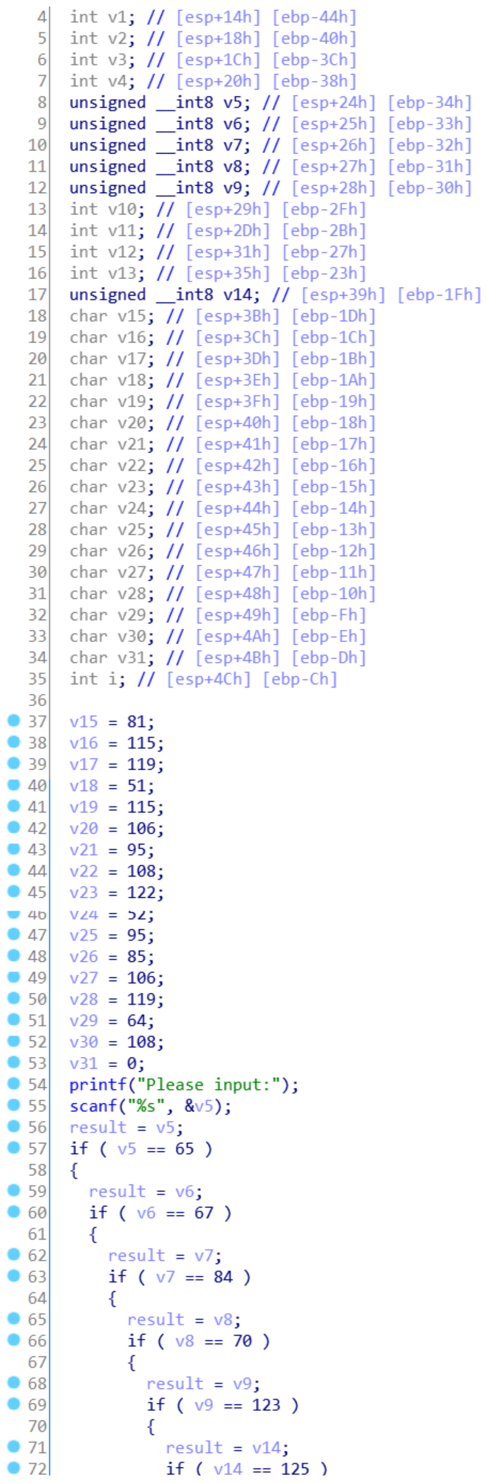
# buuctf——[ACTF新生赛2020]rome && buuctf——[GUET-CTF2019]re && buuctf——[FlareOn4]login

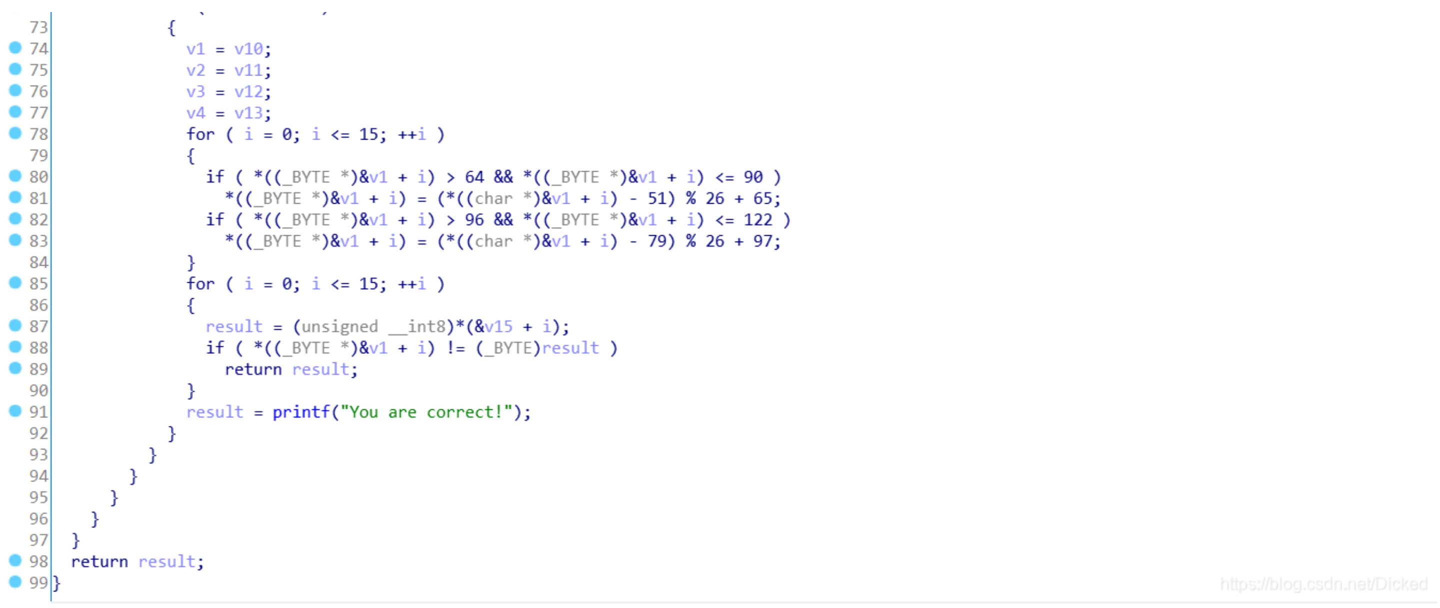
原创

Dicked 于 2020-12-03 22:18:18 发布 99 收藏

## [ACTF新生赛2020]rome

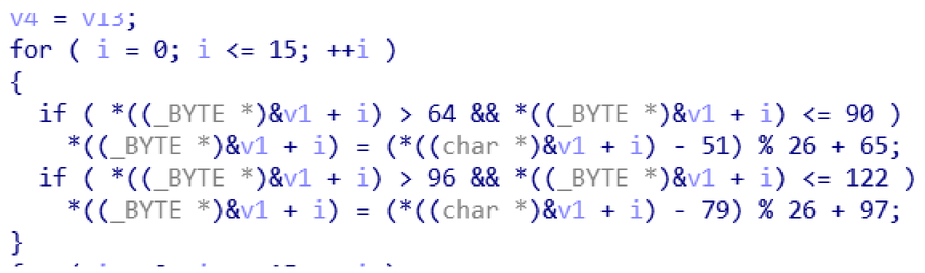


```
int v1; // [esp+14h] [ebp-44h]
int v2; // [esp+18h] [ebp-40h]
int v3; // [esp+1Ch] [ebp-3Ch]
int v4; // [esp+20h] [ebp-38h]
unsigned __int8 v5; // [esp+24h] [ebp-34h]
unsigned __int8 v6; // [esp+25h] [ebp-33h]
unsigned __int8 v7; // [esp+26h] [ebp-32h]
unsigned __int8 v8; // [esp+27h] [ebp-31h]
unsigned __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;
}
```

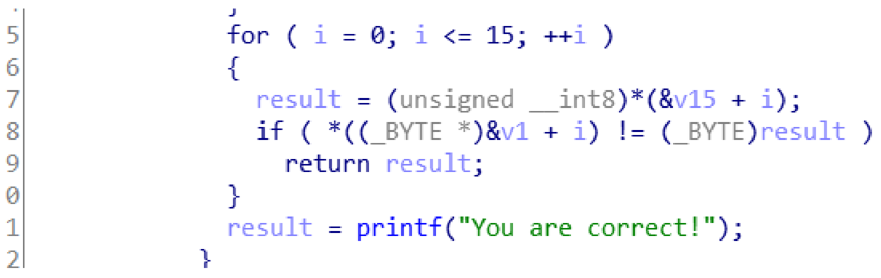
加密运算



```
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;
}
```

很像凯撒密码

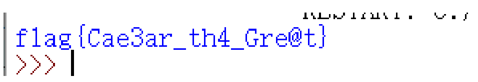
第二个for循环进行了校验，把输入的值和v15-v30的初始值进行对比。



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

程序很简单，就是让我们输入一个字符串，然后判断大小写，进行相应的运算，最后得到了程序开头的数组
v15= [ ‘Q’,‘s’,‘w’,‘3’,‘s’,‘j’, ‘’,‘l’,‘z’,‘4’,’’,‘U’,‘j’,‘w’,’@’,‘l’ ]
由于加密运算里的那个%运算的逆运算很神奇，所以我就采取了最简单值观的暴力破解。

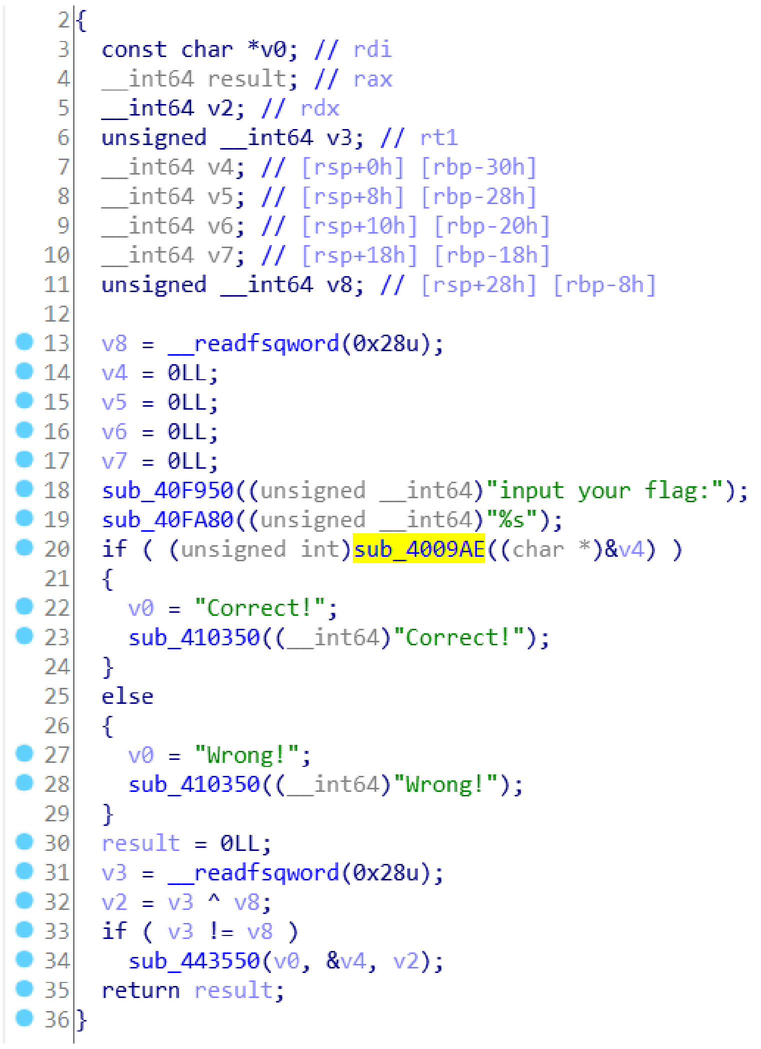
```python
v15= [ 'Q','s','w','3','s','j', '_','l','z','4','_','U','j','w','@','l' ]
flag=""

for i in range(16):
    for j in range(128):#ascii表上有127个字符，一个一个试吧
        x=j
        if chr(x).isupper():
            x=(x-51)%26+65
        if chr(x).islower():
            x=(x-79)%26+97
        if chr(x)==v15[i]:
            flag+=chr(j)

print ('flag{'+flag+'}')
```



```
flag{Cae3ar_th4_Gre@t}
>>> 
```

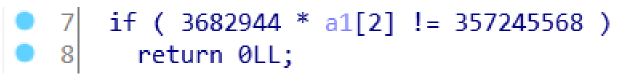
## [GUET-CTF2019]re

upx，脱

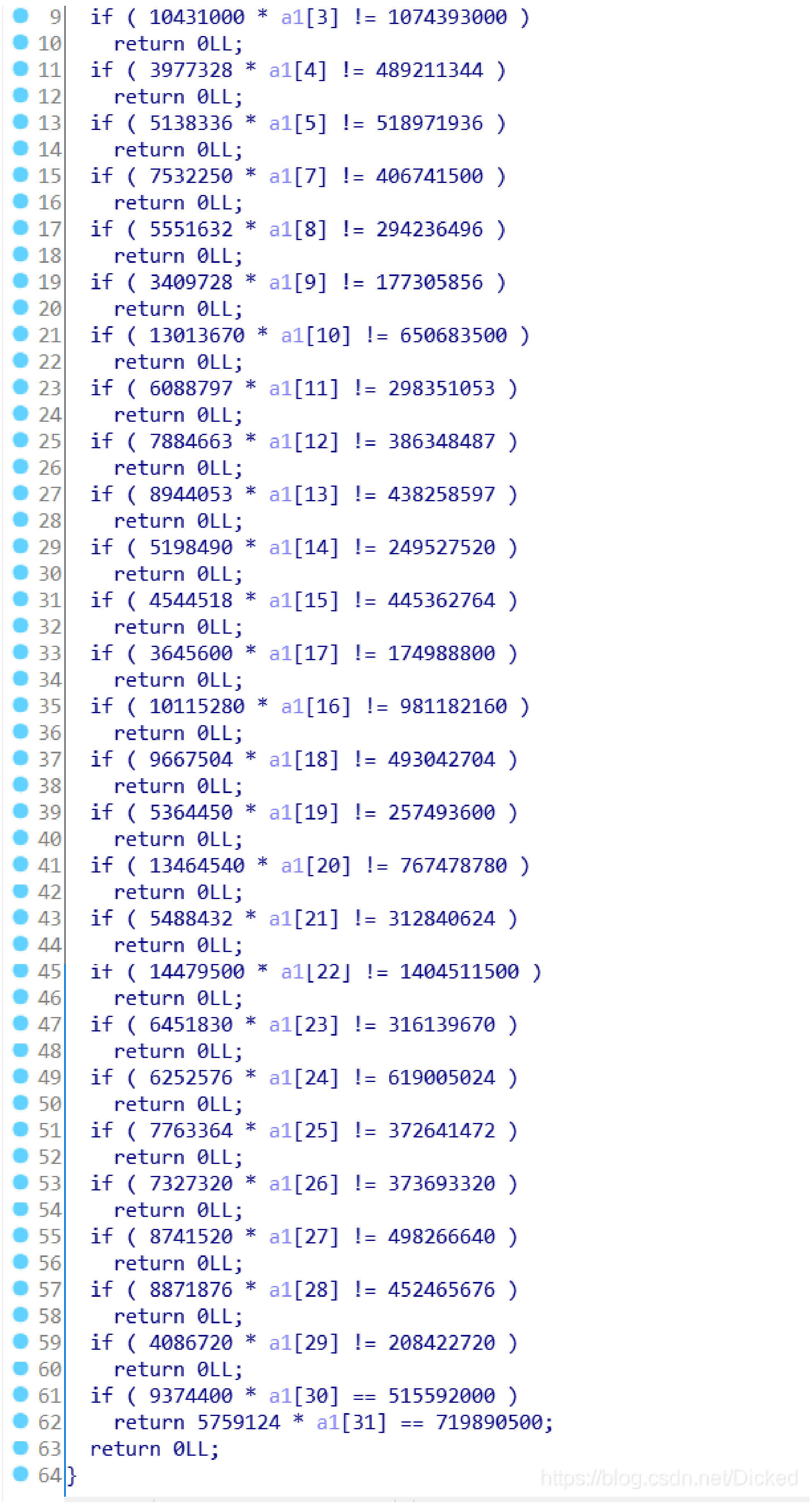


```c
{
  const char *v0; // rdi
  __int64 result; // rax
  __int64 v2; // rdx
  unsigned __int64 v3; // rt1
  __int64 v4; // [rsp+0h] [rbp-30h]
  __int64 v5; // [rsp+8h] [rbp-28h]
  __int64 v6; // [rsp+10h] [rbp-20h]
  __int64 v7; // [rsp+18h] [rbp-18h]
  unsigned __int64 v8; // [rsp+28h] [rbp-8h]

  v8 = __readfsqword(0x28u);
  v4 = 0LL;
  v5 = 0LL;
  v6 = 0LL;
  v7 = 0LL;
  sub_40F950((unsigned __int64)"input your flag:");
  sub_40FA80((unsigned __int64)"%s");
  if ( (unsigned int)sub_4009AE((char *)&v4) )
  {
    v0 = "Correct!";
    sub_410350((__int64)"Correct!");
  }
  else
  {
    v0 = "Wrong!";
    sub_410350((__int64)"Wrong!");
  }
  result = 0LL;
  v3 = __readfsqword(0x28u);
  v2 = v3 ^ v8;
  if ( v3 != v8 )
    sub_443550(v0, &v4, v2);
  return result;
}
```

进入sub_4009AE

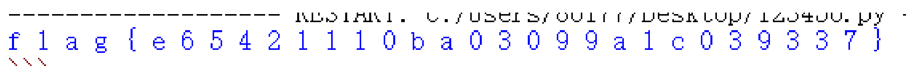


```c
  if ( 3682944 * a1[2] != 357245568 )
    return 0LL;
```



```
  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;
}
```

```python
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,a30,a31,a32)
```



~~~
==================== RESTART: C:/Users/86177/Desktop/123456.py =
f l a g { e 6 5 4 2 1 1 1 0 b a 0 3 0 9 9 a 1 c 0 3 9 3 3 7 }
```
~~~

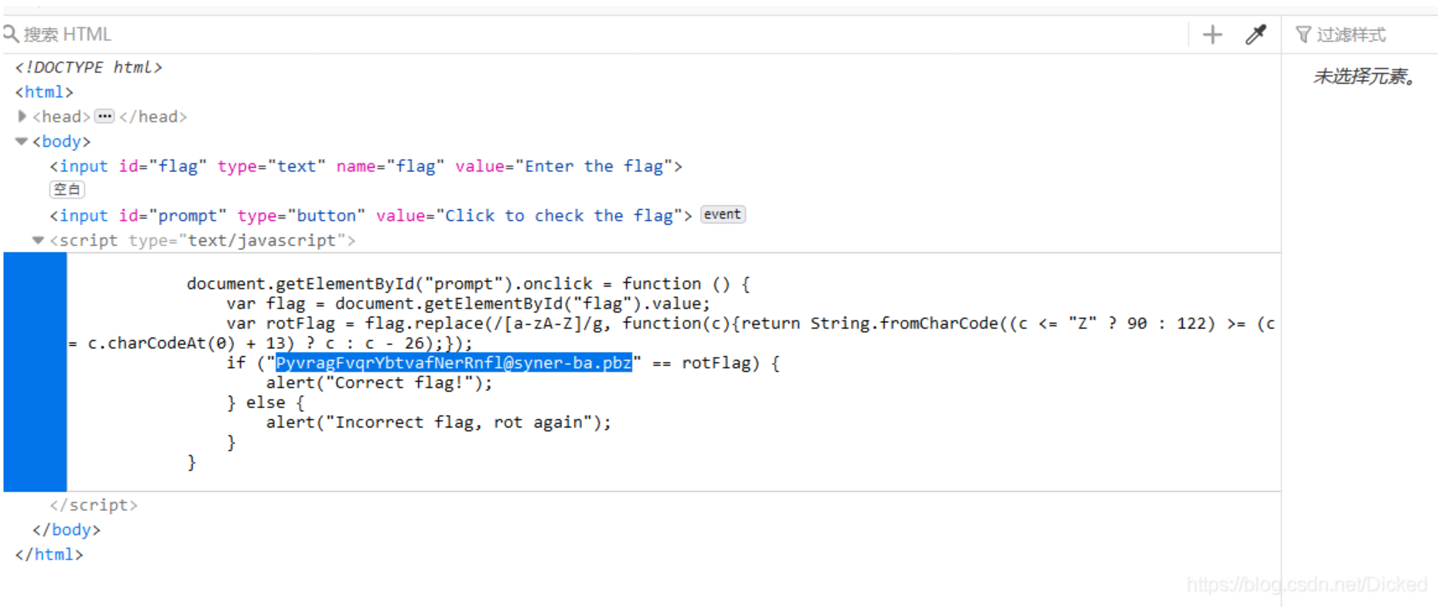
但是少了a7，即

flag{e？65421110ba03099a1c039337}这里少了一位，爆破得到1

即flag{e165421110ba03099a1c039337}

## [FlareOn4]login

html打开f12



```
<!DOCTYPE html>
<html>
<head>…</head>
<body>
  <input id="flag" type="text" name="flag" value="Enter the flag">
  空白
  <input id="prompt" type="button" value="Click to check the flag"> event
  <script type="text/javascript">

            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!");
                } else {
                    alert("Incorrect flag, rot again");
                }
            }

  </script>
</body>
</html>
```

rot13
解密
flag{ClientSideLoginsAreEasy@flare-on.com}