【攻防世界】二十三 --- lottery

原创

通地塔 于 2021-01-03 22:38:14 发布 141 收藏

分类专栏： 攻防世界 文章标签： ctf 网络安全

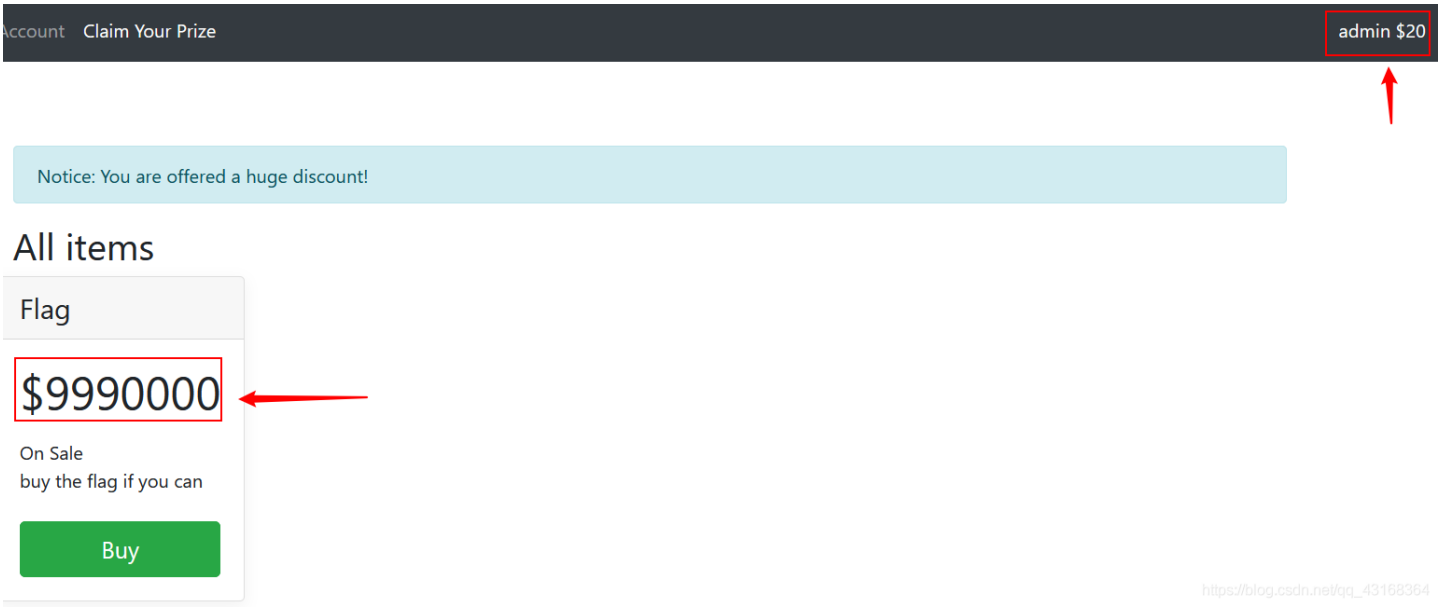
本文链接：https://blog.csdn.net/qq_43168364/article/details/112099972

题目 ---- lottery

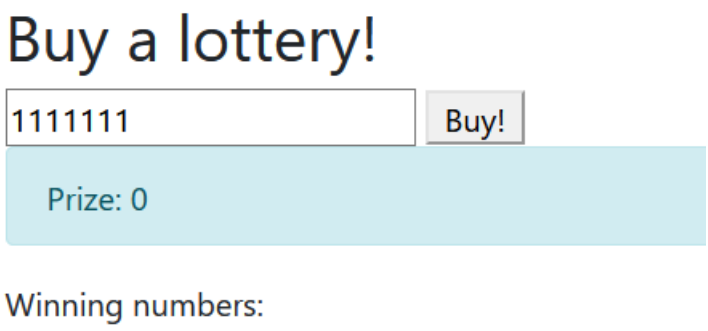
真正的页面是这样的

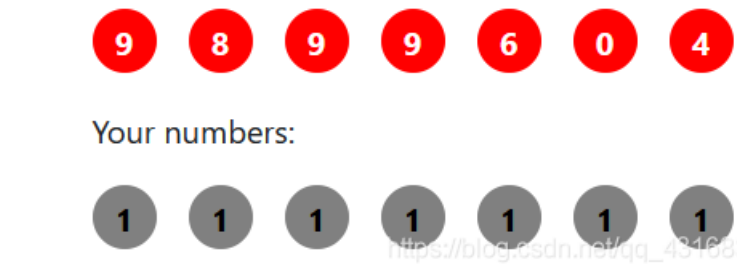
一、writeup

通过熟悉网站的业务流程可知，我们要通过玩游戏获得足够的钱才能去购买flag，我们的初始金币有20，而flag需要9990000金币才可以购买

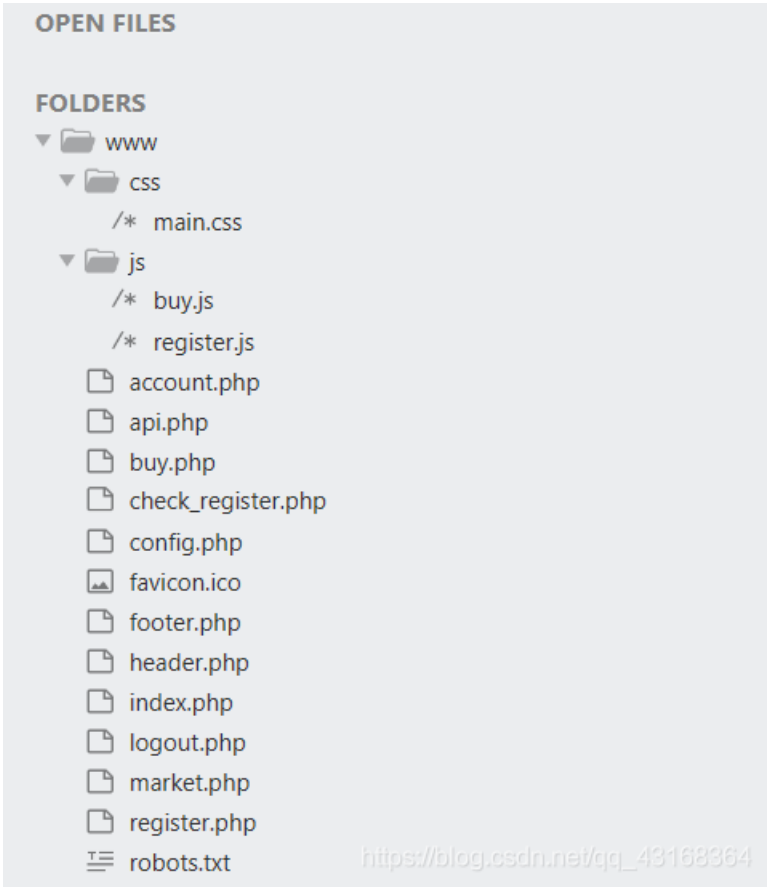


在buy页面玩几次之后会发现想要通过玩游戏赚取到买flag的金币根本不可能



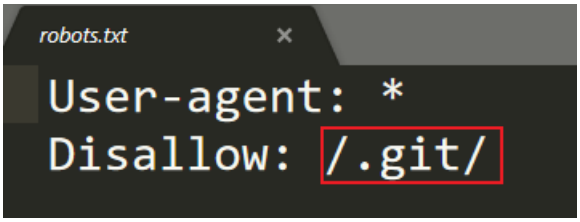


接下来我们审计题目中给出的附件，即网站的源代码。目录结构如下所示

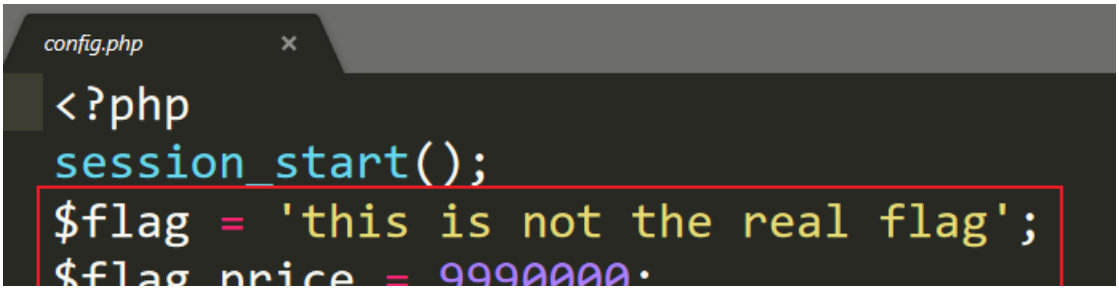


查看几个关键文件中的内容：`robots.txt`、`config.php`、`index.php`

robots.txt文件给了一个`/.git/`目录（这个题原本应该是没有直接给网站源代码的，需要我们自己能想到`git源码泄露`）



`config.php`中启动了一个会话，给了两个变量



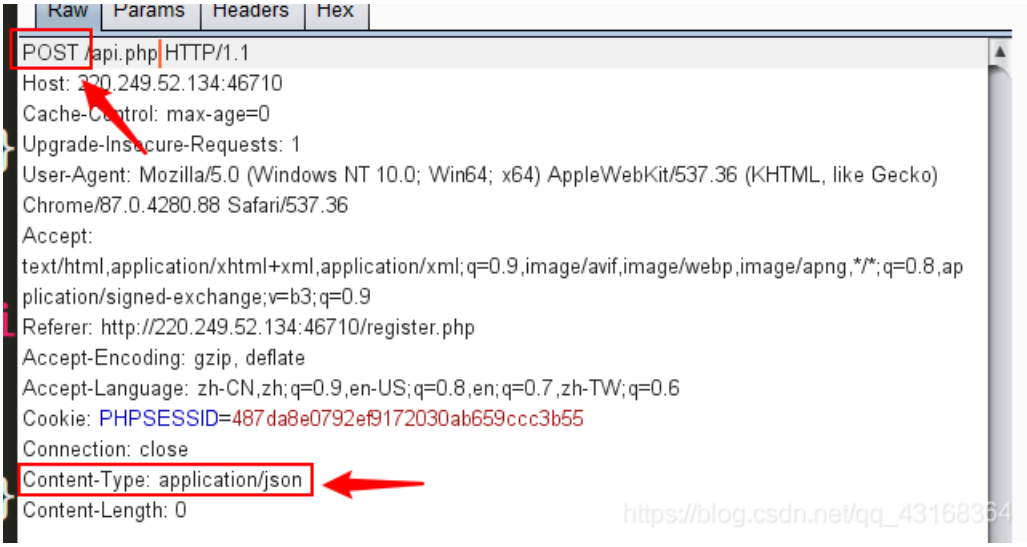
```
$flag_price = 9990000;
```


剩下的常规文件我都看了没有什么漏洞点，需要重点关照的就一个`api.php`文件，该文件是一个接口文件，我们一步一步的对其进行审计（这里先跳过它的一些函数）。首先它会包含`config.php`文件，然后设置`content-type`为`application/json`

```php
<?php
require_once('config.php');
header('Content-Type: application/json');
```

构造bp中的http头，使其满足下面的if中的条件

```php
if($_SERVER["REQUEST_METHOD"] != 'POST' || !isset($_SERVER["CONTENT_TYPE"]) || $_SERVER["CONTENT_TYPE"] != 'application/json'){
    response_error('please post json data');
}
```



获取用户输入的json数据放入`$data`变量，判断json的格式是否正确，然后使用`require_keys函数（该函数在api.php中有定义）`检查json数据中是否含有`action键`

```php
// 用json解编码拿到的内容
$data = json_decode(file_get_contents('php://input'), true);

// json_last_error --- 如果有错误,返回 JSON 编码解码时最后发生的错误
if(json_last_error() != JSON_ERROR_NONE){
    response_error('invalid json');
}

require_keys($data, ['action']);
```


接下来又是一堆函数先跳过，直接到`switch代码`的部分，会检测用户输入的json数据中的`action`对应的键值然后执行不同的代

码

```php
switch ($data['action']) {
    case 'buy':
        require_keys($data, ['numbers']);
        buy($data);
        break;

    case 'flag':
        flag($data);
        break;

    case 'register':
        require_keys($data, ['name']);
        register($data);
        break;

    default:
        response_error('invalid request');
        break;
}
```


我们先让`action`为`buy`看看buy中的代码会干什么事情，要访问到`buy函数`还需要给json中的数据再加上一个`numbers键`

```php
function buy($req){
    require_registered();
    require_min_money(2);

    $money = $_SESSION['money'];
    $numbers = $req['numbers'];
    $win_numbers = random_win_nums();
    $same_count = 0;
    for($i=0; $i<7; $i++){
        if($numbers[$i] == $win_numbers[$i]){
            $same_count++;
        }
    }
    switch ($same_count) {
        case 2:
            $prize = 5;
            break;
        case 3:
            $prize = 20;
            break;
        case 4:
            $prize = 300;
            break;
        case 5:
            $prize = 1800;
            break;
        case 6:
            $prize = 200000;
```

检测是否注册

检测用户的资金是否足够

取得用户输入的数字

生成随机的数字

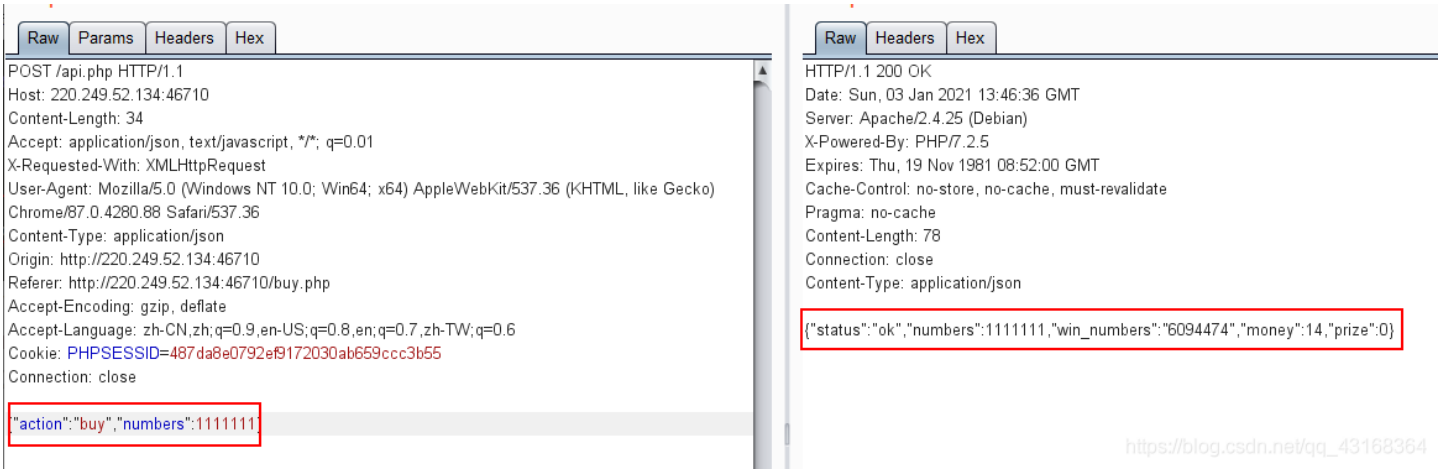
统计位置和值相同的值得个数

```
            break;
        case 7:
            $prize = 5000000;
            break;
        default:
            $prize = 0;
            break;
    }
    $money += $prize - 2;
    $_SESSION['money'] = $money;
    response(['status'=>'ok','numbers'=>$numbers, 'win_numbers'=>$win_numbers, 'money'=>$money, 'prize'=>$prize]);
}
```

返回的json格式

扣两元钱，更新用户的钱

在bp中运行如下



还看了其他的函数没有漏洞，现在要想办法通过传入的`numbers`来让用户的金币实现暴增。可以看到程序在获取numbers中的值时名没有检查参数的类型。并且在比较数值时使用的是 `==` 而非 `===` 。

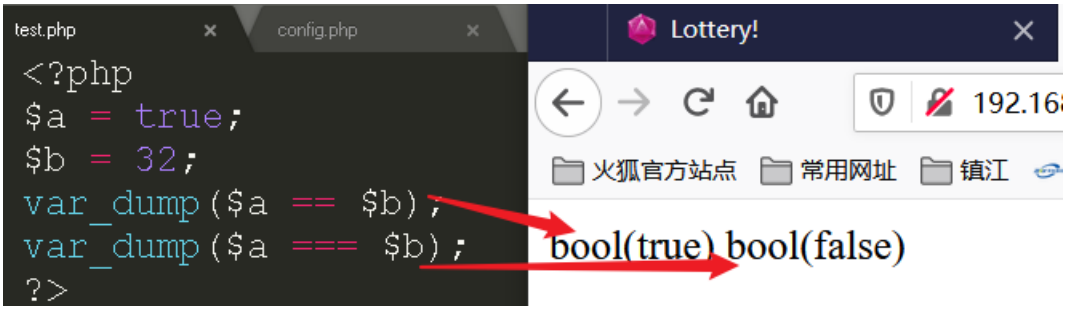
```
function buy($req){
    require_registered();
    require_min_money(2);

    $money = $_SESSION['money'];
    $numbers = $req['numbers'];
    $win_numbers = random_win_nums();
    $same_count = 0;
    for($i=0; $i<7; $i++){
        if($numbers[$i] == $win_numbers[$i]){
            $same_count++;
        }
    }
    switch ($same_count) {
        case 2:
```

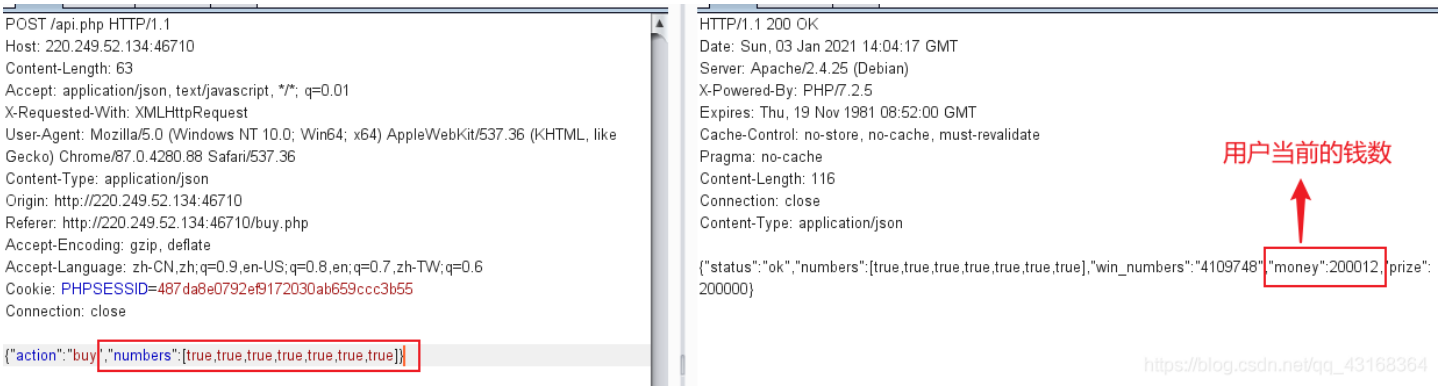
直接获取，没有判断类型

使用的是==符号

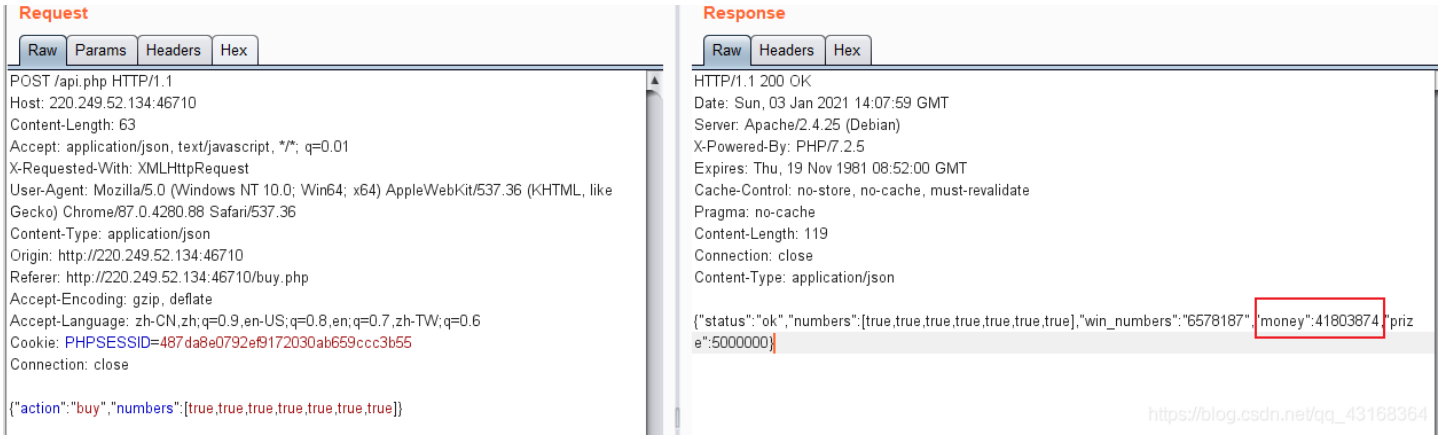
因此我们可以让`numbers=[true,true,true,true,true,true,true]`，在php中使用`==`判断时`true`和`任何数字`是相等的



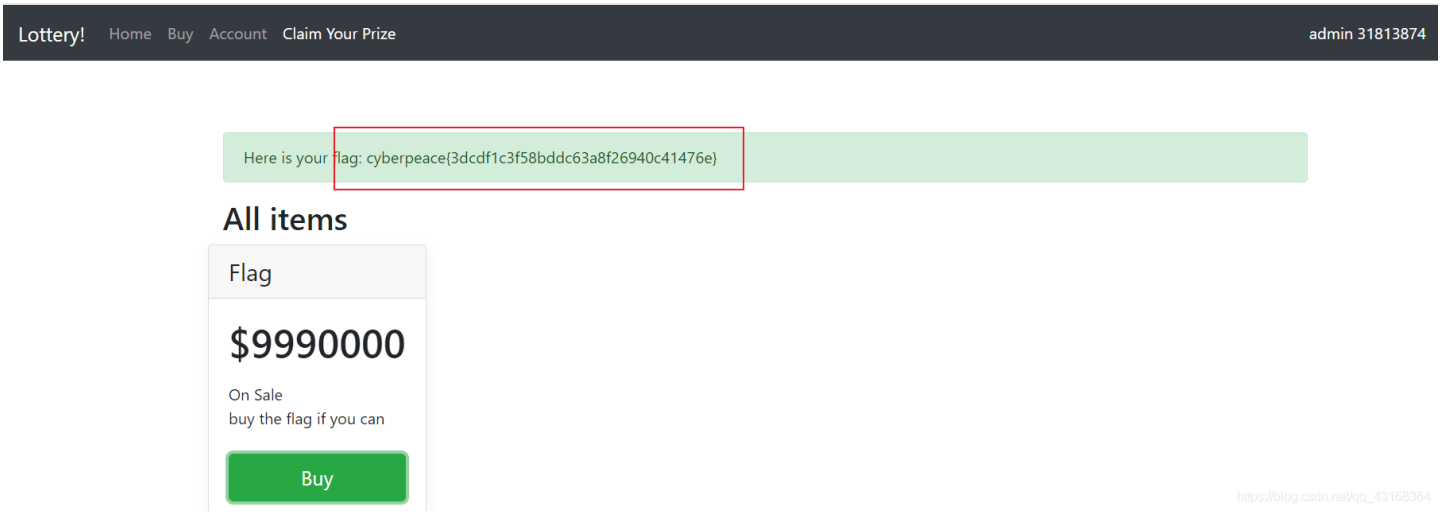
执行后结果如下



现在的钱数还是不够买flag，多执行几次即可获得足够的钱



成功买到了flag



二、知识点

- php代码审计
- php弱类型比较