simple_hash_series-php:03

原地址：原地址：GZCTF-challenges/simple_hash_series-php/03

访问页面看到如下内容

```php
<?php
error_reporting(0);
include("flag.php");
if(isset($_GET['r'])){
    $r = $_GET['r'];
    mt_srand(hexdec(substr(md5($flag), 0,8)));
    $rand = intval($r)-intval(mt_rand());
    if((!$rand)){
        if($_COOKIE['token']==(mt_rand()+mt_rand())){
            echo $flag;
        }
    }else{
        echo $rand;
    }
}else{
    highlight_file(__FILE__);
    echo system('cat /proc/version');
} Linux version 6.16.8+kali-amd64 (devel@kali.org) (x86_64-linux-gnu-gcc-14 (Debian 14.3.0-8) 14.3.0, GNU ld (GNU Binutils for Debian) 2.45) #1 SMP PREEMPT_DYNAMIC Kali 6.16.8-1kali1 (2025-09-24) Linux version 6.16.8+kali-amd64 (devel@kali.org) (x86_64-linux-gnu-gcc-14 (Debian 14.3.0-8) 14.3.0, GNU ld (GNU Binutils for Debian) 2.45) #1 SMP PREEMPT_DYNAMIC Kali 6.16.8-1kali1 (2025-09-24)
```

访问 `IP:PORT/?r=0` 得到第一个随机数

```
-541523574
```

取正数 `541523574`

利用这个随机数倒推种子值，使用 php_mt_seed 工具

```
┌──(kali㉿kali)-[~/Desktop/tool/php_mt_seed-4.0][09:23:47]
```

```
└─$ ./php_mt_seed 541523574
Pattern: EXACT
Version: 3.0.7 to 5.2.0
Found 0, trying 0x60000000 - 0x7fffffff, speed 80530.6 Mseeds/s
seed = 0x64d49eae = 1691655854 (PHP 3.0.7 to 5.2.0)
seed = 0x64d49eaf = 1691655855 (PHP 3.0.7 to 5.2.0)
Found 2, trying 0xe0000000 - 0xffffffff, speed 62634.9 Mseeds/s
Version: 5.2.1+
Found 2, trying 0x50000000 - 0x5fffffff, speed 571.1 Mseeds/s
seed = 0x5805fcb8 = 1476787384 (PHP 5.2.1 to 7.0.x; HHVM)
seed = 0x5805fcb8 = 1476787384 (PHP 7.1.0+)
seed = 0x51d617f9 = 1372985337 (PHP 5.2.1 to 7.0.x; HHVM)
seed = 0x51d617f9 = 1372985337 (PHP 7.1.0+)
Found 6, trying 0x80000000 - 0x8fffffff, speed 569.6 Mseeds/s
seed = 0x8ec34361 = 2395161441 (PHP 5.2.1 to 7.0.x; HHVM)
Found 7, trying 0xa0000000 - 0xafffffff, speed 568.7 Mseeds/s
seed = 0xa66f13b0 = 2792297392 (PHP 7.1.0+)
Found 8, trying 0xf0000000 - 0xffffffff, speed 569.5 Mseeds/s
Found 8
```

我们查看网页返回的响应头可知 `PHP/8.2.29` ，所以选择 `PHP 7.1.0+` 的种子值——`1476787384`、`1372985337`、`2792297392`

根据爆破出的种子值去计算三次伪随机数的值以及 `token` 的值（`token` 的值是第二、三次的值的和）

```php
<?php
// 已知第一个mt_rand()值（r参数）
$expected_first = 541523574;

// 候选种子列表
$candidate_seeds = [1476787384,1372985337,2792297392];

echo "=== PHP 种子验证与token计算 ===\n";
echo "预期第一个mt_rand()值: {$expected_first}\n\n";

foreach ($candidate_seeds as $seed) {
    echo "测试种子: {$seed}\n";
    mt_srand($seed);
    $first = mt_rand();

    if ($first != $expected_first) {
        echo "❌ 种子不匹配（第一个随机数：{$first}）\n\n";
        continue;
    }

    // 计算token（后续两个mt_rand()的和）
    $a = mt_rand();
    $b = mt_rand();
    $token = $a + $b;

    echo "✅ 种子匹配！\n";
    echo "第一个随机数：{$first}\n";
    echo "后续两个随机数：{$a} + {$b} = {$token}\n";
    echo "token值：{$token}\n\n";
}
?>
```

得到结果

```
=== PHP 种子验证与token计算 ===
预期第一个mt_rand()值：541523574

测试种子：1476787384
✅ 种子匹配！
第一个随机数：541523574
后续两个随机数：599738816 + 287389330 = 887128146
token值：887128146

测试种子：1372985337
✅ 种子匹配！
第一个随机数：541523574
后续两个随机数：1144411632 + 484291881 = 1628703513
token值：1628703513

测试种子：2792297392
✅ 种子匹配！
第一个随机数：541523574
后续两个随机数：304456083 + 1899581667 = 2204037750
token值：2204037750
```

接下来发包尝试获取 flag

```python
import requests

# 目标URL
TARGET_URL = "http://192.168.128.131:32779/"
# 固定r参数（第一个mt_rand()值）
R_VALUE = 541523574
# 已计算的有效token（种子 => token）
VALID_TOKENS = {
    1476787384: 887128146,
    1372985337: 1628703513,
    2792297392: 2204037750
}

def send_request(token):
    """发送包含r参数和token Cookie的请求"""
    params = {"r": R_VALUE}
    cookies = {"token": str(token)}
    try:
        # 忽略HTTPS证书验证
        response = requests.get(
            TARGET_URL,
            params=params,
            cookies=cookies,
            timeout=10,
            verify=False
        )
        return response.text
    except Exception as e:
        return f"请求失败：{str(e)}"

def main():
    print("=== 批量请求工具 ===")
    print(f"目标URL：{TARGET_URL}")
    print(f"r参数固定值：{R_VALUE}\n")

    # 忽略requests的HTTPS证书警告
    requests.packages.urllib3.disable_warnings()

    for seed, token in VALID_TOKENS.items():
        print(f"测试种子：{seed}，使用token：{token}")
        response_text = send_request(token)

        # 输出响应结果，优先检测flag
        print("响应内容：")
        if "ctfshow" in response_text.lower():
            print(f"🎉 找到flag：{response_text.strip()}\n")
        else:
            # 显示前500字符，避免输出过长
            print(f"{response_text[:500].strip()}{'...' if len(response_text) > 500 else ''}\n")
        print("-" * 60)

if __name__ == "__main__":
    main()
```

得到结果

```
=== 批量请求工具 ===
目标URL：http://192.168.128.131:32779/
r参数固定值：541523574

测试种子：1476787384，使用token：887128146
响应内容：


------------------------------------------------------------
测试种子：1372985337，使用token：1628703513
响应内容：


------------------------------------------------------------
测试种子：2792297392，使用token：2204037750
响应内容：
flag{GZCTF_dynamic_flag_test}

------------------------------------------------------------
```