

CTF实验吧让我进去writeup

转载

weixin_30496751 于 2019-03-26 23:28:00 发布 208 收藏

原文链接: <http://www.cnblogs.com/yunen/p/10604681.html>

版权

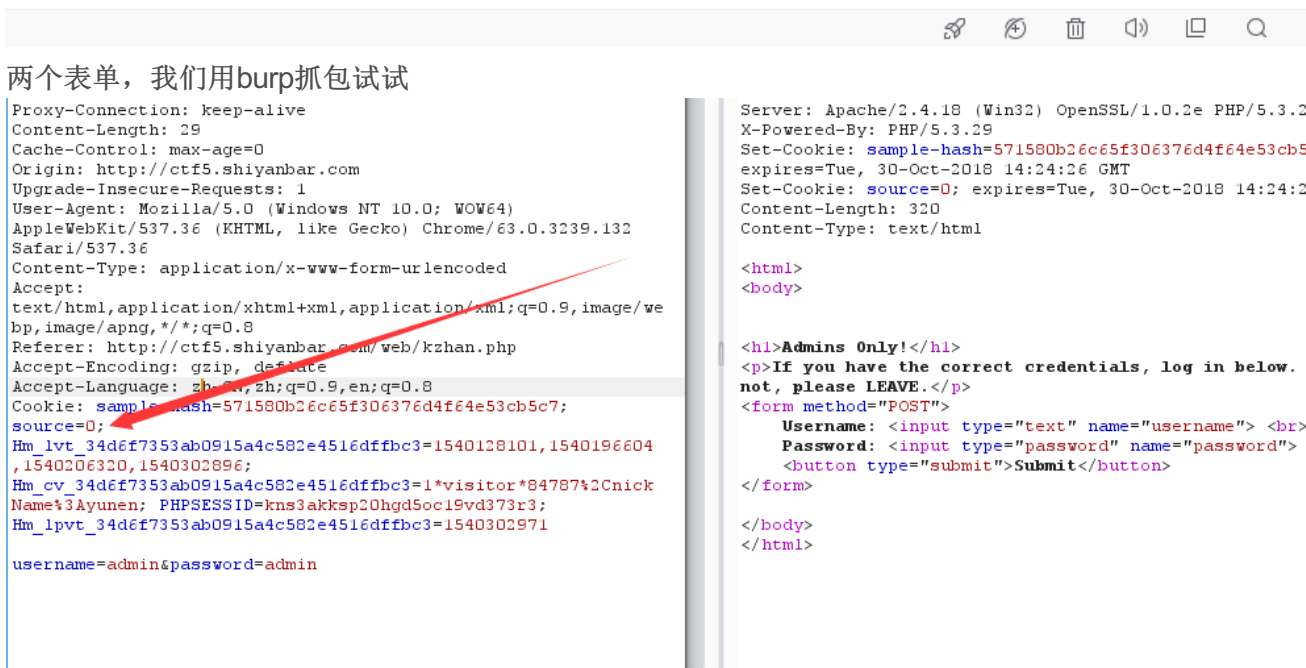
初探题目



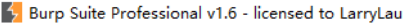
Admins Only!

If you have the correct credentials, log in below. If not, please LEAVE.

Username:
Password:





这时候我们发现Cookie值里有个很奇怪的值是source，这个单词有起源的意思，我们就可以猜测这个是判断权限的依据，让我们来修改其值为1，发送得到如下显示：


_ □ ×

Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts

1 × 2 × 3 × 4 × 5 × ...

Go Cancel < >
Target: http://ctf5.shiyanbar.com  

Request

Raw Params Headers Hex

```

POST /web/kzhan.php HTTP/1.1
Host: ctf5.shiyanbar.com
Proxy-Connection: keep-alive
Content-Length: 29
Cache-Control: max-age=0
Origin: http://ctf5.shiyanbar.com
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132
Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Referer: http://ctf5.shiyanbar.com/web/kzhan.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Cookie: sample-hash=571580b26c65f306376d4f64e53cb5c7;
source=1;
Hm_lvt_34d6f7353ab0915a4c582e4516dffbc3=1540128101,1540196604,1540206320,1540302896;
Hm_cv_34d6f7353ab0915a4c582e4516dffbc3=1*visitor*84787%2CnickName%3Ayunen; PHPSESSID=kns3akks20hgdsoc19vd373r3;
Hm_lpvt_34d6f7353ab0915a4c582e4516dffbc3=1540302971

username=admin&password=admin
          
```

? < + > 0 matches

Done

Response

Raw Headers Hex HTML Render

```

X-Powered-By: PHP/5.3.29
Set-Cookie: sample-hash=571580b26c65f306376d4f64e53cb5c7;
expires=Tue, 30-Oct-2018 14:32:22 GMT
Content-Length: 1303
Content-Type: text/html

<html>
<body>

<pre>
$flag = "XXXXXXXXXXXXXXXXXXXXXXXXX";
$secret = "XXXXXXXXXXXXXXXXX"; // This secret is 15 characters
long for security!

$username = $_POST["username"];
$password = $_POST["password"];

if (!empty($_COOKIE["getmein"])) {
    if (urldecode($username) === "admin" &&
        urldecode($password) !== "admin") {
        if ($_COOKIE["getmein"] === md5($secret .
            urldecode($username . $password))) {
            echo "Congratulations! You are a registered
            user.\n";
            die ("The flag is ". $flag);
        }
        else {
            die ("Your cookies don't match up! STOP HACKING
            THIS SITE.");
        }
    }
    else {
          
```

? < + > 0 matches

1,586 bytes | 51 millis

代码审计

发现爆出了源代码，让我们来审计一下

```

$flag = "XXXXXXXXXXXXXXXXXXXXXXXXXX";
$secret = "XXXXXXXXXXXXXXXXXX"; // This secret is 15 characters long for security!

$username = $_POST["username"];
$password = $_POST["password"];

if (!empty($_COOKIE["getmein"])) {
    if (urldecode($username) === "admin" && urldecode($password) != "admin") {
        if ($_COOKIE["getmein"] === md5($secret . urldecode($username . $password))) {
            echo "Congratulations! You are a registered user.\n";
            die ("The flag is ". $flag);
        }
        else {
            die ("Your cookies don't match up! STOP HACKING THIS SITE.");
        }
    }
    else {
        die ("You are not an admin! LEAVE.");
    }
}

setcookie("sample-hash", md5($secret . urldecode("admin" . "admin")), time() + (60 * 60 * 24 * 7));

if (empty($_COOKIE["source"])) {
    setcookie("source", 0, time() + (60 * 60 * 24 * 7));
}
else {
    if ($_COOKIE["source"] != 0) {
        echo ""; // This source code is outputted here
    }
}
}

```

我们如果需要获得flag，需要满足以下条件：

- 1.Cookie中getmein的值不能为空
- 2.username必须为admin和密码不能为admin
- 3.Cookie中的getmein必须等于md5(\$secret.urldecode(\$username.\$password))

满足这三个条件才可获得flag，可是我们无法得知\$secret的值为多少

```
setcookie("sample-hash", md5($secret . urldecode("admin" . "admin")), time() + (60 * 60 * 24 * 7));
```

发现下面有行代码是这样写的，将输出的md5(\$secret . urldecode("admin" . "admin"))作为cookie输出，结合前面的数据包我们可以知道输出的值为571580b26c65f306376d4f64e53cb5c7 可是这串md5是由\$secret+'adminadmin'转md5而得到的，如果我们在password输入admin将不满足前面所需的三个条件

死局转生

我们知道常见的md5是16位的，而这里的md5正是16位，我们的\$secret是十五位的,加上'adminadmin'就变成25位了，很明显这里的md5肯定会出现重复，所以我们可以哈希长度拓展攻击绕过这个死局

这里附两个讲述具体原理的链接：

<http://www.freebuf.com/articles/web/69264.html> <https://www.cnblogs.com/p00mj/p/6288337.html>

