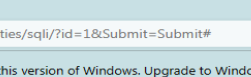
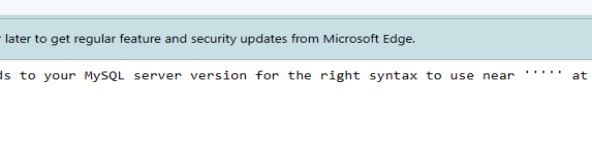


首先先随意输入一个id



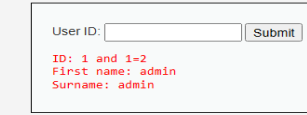
确定为GET型

输入‘ 判断是否有错误反馈



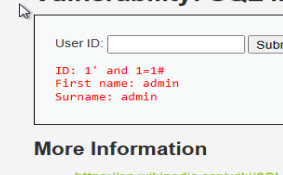
确定为MYSQL

输入 1 and 1=1, 1and 1=2 均有结果



确定不是数字型

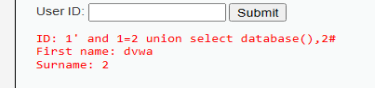
输入 1’ and 1=1# 有结果 1’ 1=2# 无结果 #的作用注释掉后面的引号



说明为字符串型注入

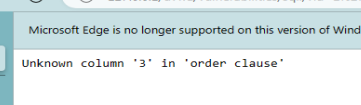
输入 1’ and 1=2 union select database(),2#

前半部分使正常查询无效，后半部分利用database（）函数查询到当前的数据库



说明当前数据库的名字为dvwa

下一步判断当前表中有几个字段



当order by 1 和 order by 2 时有数据显示

但 order by 3 时会报错 说明表中只有两个字段

所以我们接下来收集的信息都将基于这两个输出接口

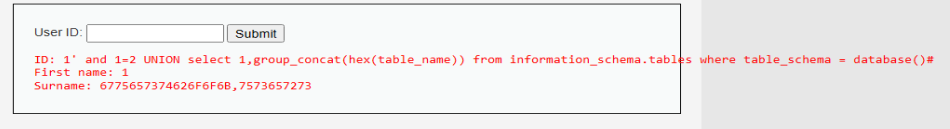
接下来判断数据库中一共有几个表

最初的方法收集

Concat(table\_name) 出现了一些小问题，编码形式上的问题

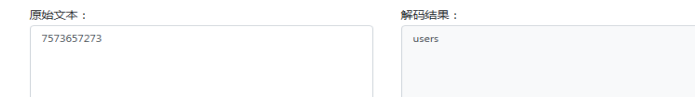
解决方案：

采取收集十六进制解决



将收集到的16进制 转化



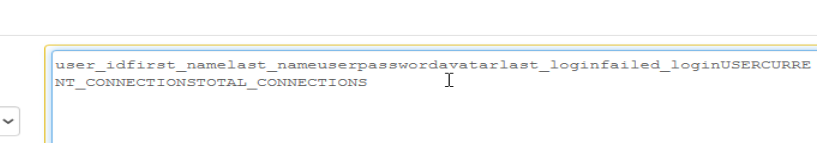


得到两个表单 猜测users 表单中存放着用户的用户名和密码

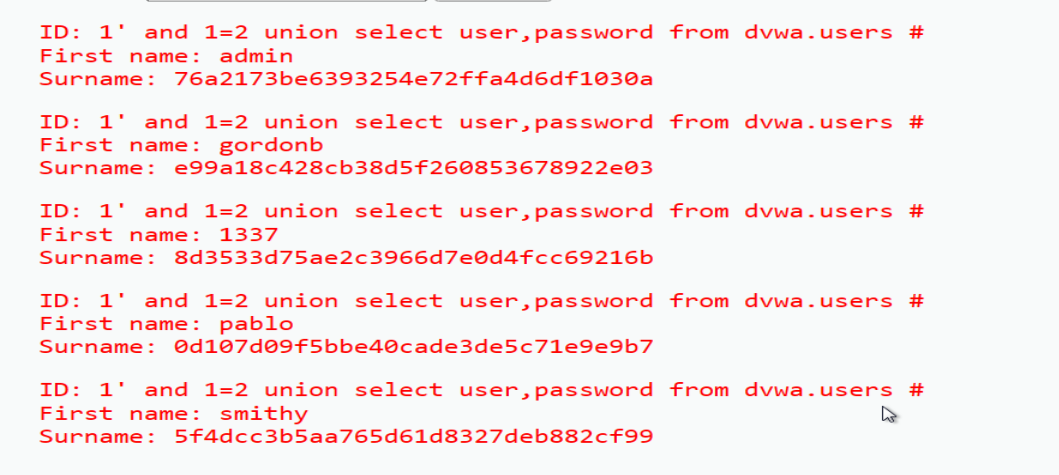
查看users中的所有字段



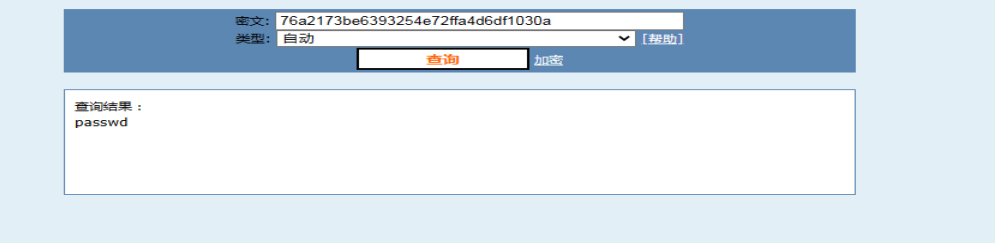
进行逐一解码工作：

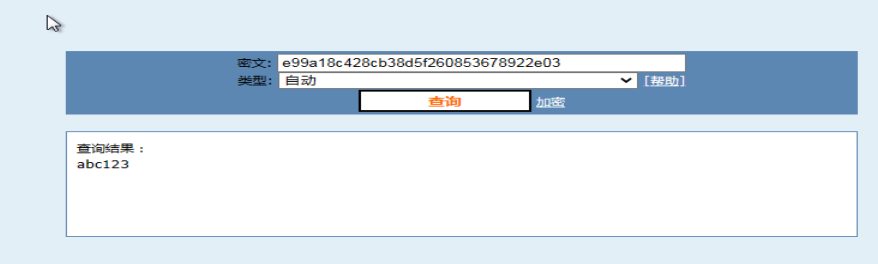


发现其中有user 和 password 字段，猜测其存储内容为用户名和密码



将其解码可以得到密码的明文





这样就拿到了数据库中保存的账号和密码

Medium：

Medium 模式 将GET请求方式改为POST，由于页面中没有提供输入信息的接口，所以需要用到抓包工具 burpsuite

在ID值后添加 ‘ 浏览器报错，说明网页同样存在sql Injection 漏洞

输入 id = 1 and 1=1 id =1 and 1=2 后者没有数据返回

之后与low级别的思路相同

Id = 1 and 1=2 union select 1,group\_concat(column\_name) from information\_schema.columns where table\_name = ‘users’ 查询字段名，发现页面报错

Medium 在id值上添加了mysql\_real\_escape\_string函数 ，该函数可以对单引号进行转义

方法： 对表名进行16进制 HEX编码，可以不使用‘’ ，成功查出字段名user password

总结：

1、medium模式使用了post提交，比get好一点，但是并没有什么用，可以用抓包工具

1. 使用mysql\_real\_escape\_string函数 将‘’过滤掉 可以使用hex进行编码，可以不使用‘’

High: 级别

分离了输入页面和显示结果的页面，防止了SQLMAP的自动注入

输入 1’ and 1=1# 1’ and 1=2# 前者返回数据 后者没有返回数据