详解强大的 SQL 注入工具——SQLMAP

Akast [N.S.T]

1. 前言

Windows 下的注入工具好的又贵，免费的啊 D、明小子等又不好用，我们根本没必要花时间去找什么破解的 havij、pangolin 什么的，特别是破解的工具很可能被绑了木马。其实 Linux 下的注入工具也是非常强大的，不过分的说，可以完全取代 Windows 下面的所有注入工具。

就如 backtrack 系统里面就有非常丰富的注入工具，对 MSSQL、MYSQL、oracle 等各种数据库的应有尽有了，而且这些工具都是免费的，并且是开放源代码的，我们还可以用来修改为合适自己使用的注入工具。

本文给大家介绍的 SqlMap 是一个开放源码的渗透测试工具，它可以自动探测和利用 SQL 注入漏洞来接管数据库服务器。它配备了一个强大的探测引擎，为最终渗透测试人员提供很多猥琐的功能，可以拖库，可以访问底层的文件系统，还可以通过带外连接执行操作系统上的命令。

2. SQLMAP 命令详解

为了方便使用我把 sqlmap 的选项都翻译出来了，当然可能会存在一些不恰当的地方，请大家指出，可以给我发邮件：akast@ngsst.com。如果我有时间会把这个工具出个中文版。

```
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


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

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，您还可以运行您自己的 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
    --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）
    -D DB                要进行枚举的数据库名
    -T TBL               要进行枚举的数据库表
    -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

  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

  General（一般）：

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
```

3. 什么是 OOB？

传输层协议使用带外数据(out-of-band,OOB)来发送一些重要的数据,如果通信一方有重要的数据需要通知对方时,协议能够将这些数据快速地发送到对方.为了发送这些数据,协议一般不使用与普通数据相同的通道,而是使用另外的通道.linux 系统的套接字机制支持低层协议发送和接受带外数据.但是 TCP 协议没有真正意义上的带外数据.为了发送重要协议,TCP 提供了一种称为紧急模式(urgentmode)的机制.TCP 协议在数据段中设置 URG 位,表示进入紧急模式.接收方可以对紧急模式采取特殊的处理.很容易看出来,这种方式数据不容易被阻塞,可以通过在我们的服务器端程序里面捕捉 SIGURG 信号来及时接受数据或者使用带 OOB 标志的 recv 函数来接受.

4. 注入实例

没时间写，大家自己发挥吧。