SQLMap v1.0

Options:

-h/**-hh** help/advanced help --version show version number
-v VERBOSE verbosity level: 0-6 (default 1)

Target: (At least one of these options has to be provided)

-u URL target URL -d DIRECT direct connection to the db
-m FILE targets in a file -l LOGFILE parse from Burp/WebScarab
-r FILE load HTTP request file -g GDORK google dork as target
-c CONFIGFILE load options from a configuration INI file

Request: (specify how to connect to the target URL)

--data=DATA data string to be sent through POST
--param-del=PDEL character used for splitting parameter values
--cookie=COOKIE HTTP Cookie header
--cookie-del=CDEL character used for splitting cookie values
--load-cookies=L.. file containing cookies in Netscape/wget format
--drop-set-cookie ignore Set-Cookie header from response
--user-agent=AGENT **--random-agent**
--host=HOST --referer=REFERER --headers=HEADERS
--auth-type=AUTH.. Basic, Digest, NTLM or PKI
--auth-cred=AUTH.. name:password
--auth-private=A.. PEM private key file
--proxy=PROXY --proxy-cred=PRO.. name:password
--proxy-file=PRO.. list from a file --ignore-proxy ignore system settings
--tor --tor-port=TPORT --tor-type=TYPE HTTP (dflt), SOCKS4, SOCKS5
--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 (default 30)
--retries=RETRIES retries when the connection timeouts (default 3)
--randomize=RPARAM randomly change value for given parameter(s)
--safe-url=SAFURL URL address to visit frequently during testing
--safe-freq=SAFREQ test requests between two visits to a given safe URL
--skip-urlencode skip URL encoding of payload data
--force-ssl force usage of SSL/HTTPS
--hpp use HTTP parameter pollution
--eval=EVALCODE evaluate provided Python code before the request (e.g. "import hashlib;id2=hashlib.md5(id).hexdigest()")

SecurityByDefault.com

Optimization:

-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:

-p TESTPARAMETER testable parameter(s)
--skip=SKIP skip testing for given parameter(s)
--dbms=DBMS force back-end DBMS to this value
--dbms-cred=DBMS.. DBMS authentication credentials (user:password)
--os=OS force backend DBMS OS to this value
--invalid-bignum use big numbers for invalidating values
--invalid-logical/--invalid-string use logical/random for invalidating values
--no-cast/--no-escape turn off payload casting/escaping
--prefix=PREFIX/--suffix=SUFFIX injection payload prefix/suffix string
--tamper=TAMPER use given script(s) for tampering injection data

Detection: **(**used to customize/improve the detection phase**)**

--level=LEVEL level of tests to perform (1-**5**, default 1)
--risk=RISK risk of tests to perform (0-3, default 1)
--string=STRING/--not-string=NOT.. match when query is evaluated to True/False
--regexp=REGEXP regexp to match when query is evaluated to True
--code=CODE HTTP code to match when query is evaluated to True
--text-only/--titles compare pag based only on the textual content/ titles

Techniques: **(**used to tweak testing of specific SQL injection)

--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
--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-order=S.. resulting page URL searched for second-order response

Enumeration: (enumerate the back-end database, structure and data contained)

-a, --all retrieve everything	-b retrieve banner
--is-dba check if user is DBA	
--current-user/**--current-db**/--hostname	retrieve DBMS current user/database/hostname
--users/**--passwords**	enumerate DBMS users / users password hashes
--privileges/--roles	enumerate DBMS users privileges/roles
--dbs/**--tables**/**--columns**/**--schema**	enumerate DBMS dbs/tables/columns/schema
--count retrieve num of entries for table(s)	--search search column(s), table/db name
--dump-all dump all DBMS dbs tables entries	**--dump** dump DBMS db table entries
-U USER DBMS user to enumerate	--exclude-sysdbs exclude system dbs
--comments retrieve DBMS comments	-X EXCLUDECOL table column(s) to not enum
-D DB / **-T TBL** / **-C COL**	DBMS database to enumerate / tables / columns
--where=DUMPWHERE	use WHERE condition while table dumping
--start=LIMITSTART/--stop=LIMITSTOP	first/last query output entry to retrieve
--first=FIRSTCHAR/--last=LASTCHAR	first/last query output word character to retrieve
--sql-file=SQLFILE	execute SQL statements from given file(s)
--sql-shell	prompt for an interactive SQL shell
--sql-file=FILE	execute SQL statements from given file(s)

General:

-s SESSIONFILE	load session from .sqlite file	-t TRAFFICFILE	log all HTTP traffic
--batch	never ask for input	--eta	display for each eta
--save	save options to a configuration INI file	**--update**	update sqlmap

--charset=CHARSET	force character encoding used for data retrieval
--crawl=CRAWLDEPTH	crawl the website starting from the target URL
--csv-del=CSVDEL	delimiting character used in CSV output (default ",")
--dump-format=DU..	format of dumped data (CSV (default), HTML or SQLITE)
--flush-session	flush session files for current target
--forms	parse and test forms on target URL
--fresh-queries	ignore query results stored in session file
--hex	use DBMS hex function(s) for data retrieval
--output-dir=ODIR	custom output directory path
--parse-errors	parse and display DBMS error messages from responses
--pivot-column=P..	pivot column name
--scope=SCOPE	regexp to filter targets from provided proxy log
--test-filter=TE..	select tests by payloads and/or titles (e.g. ROW)

SQLMap v1.0

Fingerprint: **-f, --fingerprint** perform an extensive DBMS version fingerprint

Brute Force: --common-tables/--common-columns check common tables/columns

User-defined function injection:

--udf-inject inject custom functions --shared-lib=SHLIB local path of the shared lib

File system access:

--file-read=RFILE/--file-write=WFILE	read/write local file on the DBMS file system
--file-dest=DFILE	back-end DBMS absolute filepath to write to

Operating system access:

--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/--tmp-path=TMPPATH	local Metasploit/Remote tmp path

Windows registry access:

--reg-read/--reg-add/--reg-del		read/write/delete a win registry key value
--reg-key=REGKEY	win registry key	--reg-value=REGVAL win reg key value
--reg-data=REGDATA	win reg key data	--reg-type=REGTYPE win reg key value type

Miscellaneous:

-z MNEMONICS	use short mnemonics (e.g. "flu,bat,ban,tec=EU")
--alert=ALERT	run host OS command(s) when SQL injection is found
--answers=ANSWERS	set question answers (e.g. "quit=N,follow=N")
--check-waf/--identify-waf	WAF/IPS/IDS protection
--cleanup	clean up the DBMS from sqlmap specific UDF and tables
--dependencies	check for missing (non-core) sqlmap dependencies
--gpage=GOOGLEPAGE	Use Google dork results from specified page number
--mobile	imitate smartphone through HTTP User-Agent header
--page-rank	display page rank (PR) for Google dork results
--purge-output	safely remove all content from output directory
--smart	conduct through tests only if positive heuristic(s)
--disable-coloring	--beep if sql injection is found.
--wizard	wizard interface for beginner users