

Sqlmap

Miscellaneous

- Use short mnemonics (e.g. "flu,bat,ban,tec=EU") `--z MNEMONICS`
- Run host OS command(s) when SQL injection is found `--alert=ALERT`
- Beep on question and/or when vulnerability is found `--beep`
- Check for missing (optional) sqlmap dependencies `--dependencies`
- Disable console output coloring `--disable-coloring`
- Display list of available tamper scripts `--list-tampers`
- Disable logging to a file `--no-logging`
- Work in offline mode (only use session data) `--offline`
- Safely remove all content from sqlmap data directory `--purge`
- Location of CSV results file in multiple targets mode `--results-file=R..`
- Prompt for an interactive sqlmap shell `--shell`
- Local directory for storing temporary files `--tmp-dir=TMPDIR`
- Adjust options for unstable connections `--unstable`
- Update sqlmap `--update`
- Simple wizard interface for beginner users `--wizard`

General

- Load session from a stored (.sqlite) file `--s SESSIONFILE`
- Log all HTTP traffic into a textual file `--t TRAFFICFILE`
- Set predefined answers (e.g. "quit=N,follow=N") `--answers=ANSWERS`
- Parameter(s) containing Base64 encoded data `--base64=BASE64P..`
- Use URL and filename safe Base64 alphabet (RFC 4648) `--base64-safe`
- Never ask for user input, use the default behavior `--batch`
- Result fields having binary values (e.g. "digest") `--binary-fields=.`
- Check Internet connection before assessing the target `--check-internet`
- Clean up the DBMS from sqlmap specific UDF and tables `--cleanup`
- Crawl the website starting from the target URL `--crawl=CRAWLDEPTH`
- Regex to exclude pages from crawling (e.g. "logout") `--crawl-exclude=.`
- Delimiting character used in CSV output (default ";") `--csv-del=CSVDEL`
- Blind SQL injection charset (e.g. "0123456789abcdef") `--charset=CHARSET`
- Store dumped data to a custom file `--dump-file=DUMP..`
- Format of dumped data (CSV (default), HTML or SQLITE) `--dump-format=DU..`
- Character encoding used for data retrieval (e.g. GBK) `--encoding=ENCOD..`
- Display for each output the estimated time of arrival `--eta`
- Flush session files for current target `--flush-session`
- Parse and test forms on target URL `--forms`
- Ignore query results stored in session file `--fresh-queries`
- Use Google dork results from specified page number `--gpage=GOOGLEPAGE`
- Log all HTTP traffic into a HAR file `--har=HARFILE`
- Use hex conversion during data retrieval `--hex`
- Custom output directory path `--output-dir=OUT..`
- Parse and display DBMS error messages from responses `--parse-errors`
- Use given script(s) for preprocessing (request) `--preprocess=PRE..`
- Use given script(s) for postprocessing (response) `--postprocess=PO..`
- Redump entries having unknown character marker (?) `--repair`
- Save options to a configuration INI file `--save=SAVECONFIG`
- Regex for filtering targets `--scope=SCOPE`
- Skip heuristic detection of vulnerabilities `--skip-heuristics`
- Skip heuristic detection of WAF/IPS protection `--skip-waf`
- Prefix used for temporary tables (default: "sqlmap") `--table-prefix=T..`
- Select tests by payloads and/or titles (e.g. ROW) `--test-filter=TE..`
- Skip tests by payloads and/or titles (e.g. BENCHMARK) `--test-skip=TEST..`
- Web server document root directory (e.g. "/var/www") `--web-root=WEBROOT`

Windows registry access

- Read a Windows registry key value `--reg-read`
- Write a Windows registry key value data `--reg-add`
- Delete a Windows registry key value `--reg-del`
- Windows registry key `--reg-key=REGKEY`
- Windows registry key value `--reg-value=REGVAL`
- Windows registry key value data `--reg-data=REGDATA`
- Windows registry key value type `--reg-type=REGTYPE`

Operating system access

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

File system access

- Read a file from the back-end DBMS file system `--file-read=FILE..`
- Write a local file on the back-end DBMS file system `--file-write=FIL..`
- Back-end DBMS absolute filepath to write to `--file-dest=FILE..`

User-defined function injection

- Inject custom user-defined functions `--udf-inject`
- Local path of the shared library `--shared-lib=SHLIB`

Brute force

- Check existence of common tables `--common-tables`
- Check existence of common columns `--common-columns`
- Check existence of common files `--common-files`

Enumeration

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

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)

Target

- `-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

Request

- `-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, Bearer, ...)
- `--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)
- `--retry-on=RETRYON` Retry request on regex matching content (e.g. "drop")
- `--randomize=RPARA` 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-data=CSRF..` POST data to send 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

- `-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)
- `--skip-static` Skip testing parameters that not appear to be dynamic
- `--param-exclude=.` Regex 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

- `--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` Regex 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

- `--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

Fingerprint

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



@hackinarticles



https://github.com/ignitetechnologies



https://in.linkedin.com/company/hackingarticles