

Forum:[Hack Forums](#)

Thread: [SQLiBar Pentesting & SQLi Firefox Addon](#)

•^v^-[bLacKn3x]-^v^• - August 10, 2026, 11:17 am

SQLiBar – Pentesting & SQLi Firefox Addon

A Developer Tools extension for security testing, real-time SQL injection pattern detection, request manipulation, parameter discovery, advanced payload encoding, and multi-language UI support.

Description

SQLiBar is an advanced web penetration testing tool built into Firefox Developer Tools. Designed for security researchers, penetration testers, and web developers, it streamlines parameter analysis, HTTP request building, SQL injection vulnerability testing, and response inspection directly from your browser.

SQL Injection Detection & Pattern Analysis

Automated response analysis against 100+ database-specific error patterns (MySQL, MariaDB, PostgreSQL, MSSQL, Oracle, SQLite, DB2, Sybase, Access, H2, Firebird, and common ORMs such as PDO, SQLAlchemy, Django, Laravel, Hibernate, Entity Framework, Prisma, and more)

Baseline comparison: set response length and timing baselines to catch subtle anomalies (Boolean-based, Union-based, Error-based, and Time-based indicators)

Reflection detection: highlights payload fragments that appear in the response

Diff view: line-based comparison against the stored baseline body

Visual severity indicators (high / medium / low) with code snippet context for identified error signatures

Header SQLi Mass-Test

One-click batch testing of custom headers for SQL injection

Sends one request per header, injecting the payload into that header only while leaving the rest unchanged

Options: risky only (limit to known high-risk headers such as X-Forwarded-For, Cookie, Authorization, etc.) or test all headers from the textarea

Current payload mode uses your Payload field; otherwise a built-in detection set is used (error-based and time-based probes)

Live progress, abort support, and a results list with status, timing, body size, and badges (``clean` / ``HIGH` / ``TIME` / ``error`)

Load any result back into the tester for manual follow-up

JSON Key SQLi Mass-Test

Automatic SQLi testing of every value in a JSON request body

Walks nested objects and arrays and tests one leaf value at a time

Append mode: the original value is kept and the payload is concatenated — nothing is replaced, and all other fields stay intact

Options: use the current payload or the built-in detection set; optionally restrict to leaf values only

Progress indicator, abort, and a detailed results panel (path, original value, payload, status, timing, badges)

-> Body loads the single injected JSON for that test into the Body field for further manual testing

`Content-Type: application/json` is set automatically when missing

Smart Page Parameter Scanner

Scans the current page DOM to discover form fields, hidden inputs, URL query parameters, link parameters, data-* attributes, and cookies

Instantly push discovered parameters to the URL query, request body, or payload field with a single click

Live Network Capture & Parameter Aggregator

Real-time monitoring and capturing of HTTP / XHR / Fetch requests

Automatic extraction and aggregation of parameters from headers, query strings, URL-encoded bodies, and nested JSON payloads

Inspect raw request/response headers, bodies, and response previews directly

Filter by All / XHR / Documents / POST and search by URL or method

One-click send to Tester, Replay, cURL export, and parameter copy

Payload Presets & Request Manipulation

Built-in payload preset manager with dynamic column generator for UNION-based SQLi tests

Interactive JSON body tree examiner for navigating and selecting deeply nested keys (copy path, push to Payload / URL)

Custom HTTP header injector (X-Forwarded-For, Authorization, custom cookies, Host rewrite, and more)

Method selection (GET, POST, PUT, PATCH, DELETE, HEAD, OPTIONS), body editor, and optional page navigation

One-click cURL export of the current request

Response Viewer

Syntax highlighting for JSON and HTML response bodies

Search within the response (case-sensitive option, next/previous match)

Copy full response or body only

Boolean Blind Mini-Test

A quick True vs False comparison to help spot boolean-based blind SQL injection without leaving the panel.

Sends two requests in sequence: one with a True expression and one with a False expression

Compares status, response length, timing, and a simple body hash

Shows a clear verdict when responses differ (possible boolean-blind) or look the same

Feeds the True response into the existing SQLi detector and response view

Multi-Format Encoder / Decoder

Live preview – encode/decode updates as you type

2-step chain – optional second encoding stage (e.g. URL -> Base64 in one step)

Swap Input <-> Output for rapid multi-pass encoding

Push results directly into Payload, URL, or Request Body

Formats: URL, Double URL, Triple URL, Selective URL (SQLi-focused), Base64, Base64 URL-Safe, Hex (spaced / 0xAABB... / \x.. / 0xAA,0xBB), Binary, ASCII/Decimal, SQL CHA*R(), SQL CHA*R(0x...), SQL CON*CAT(CHAR...), Unicode \u, Unicode %u, Fullwidth Unicode, HTML Entities, JSON, JWT Decode (header / payload / signature), ROT13

Themes & Localization

Multiple built-in color schemes (Neon Green, Cyber Cyan, Violet, Amber, Hot Pink, Electric Blue, Matrix Lime, Mono) plus custom accent color picker

Multi-language interface: English, Deutsch, Español, Français, Português, ???????, ???, ??, Türkçe, ??????? (saved preference)

Permissions Justification

devtools: Integrates natively into Firefox DevTools for a seamless workflow

webRequest / cookies / activeTab: Required to capture live network traffic, inspect response headers, inject payload parameters, and manage cookies for security testing

Disclaimer

SQLiBar is strictly intended for authorized security testing, educational purposes, and vulnerability research on systems you own or have explicit permission to test.

<https://github.com/blackn3x/SQLiBar>

<https://addons.mozilla.org/addon/sqlibar/>

Dotexe - August 10, 2026, 11:21 am

Seems like a good thing, perhaps a good alternative to those who can't afford BurpSuite professional

•^v^-[bLacKn3x]-^v^• - August 10, 2026, 11:46 am

(Aug 10, 2026 11:21 AM)Dotexe Wrote: [\[CITE\]](#) Seems like a good thing, perhaps a good alternative to those who can't afford BurpSuite professional

Thanks, I hope it can be a helpful tool for some people. I'm already working on new features and optimizations.

R5N3T - August 10, 2026, 4:22 pm

Wow that nice one, thank you for sharing

snadder - August 12, 2026, 9:22 am

This is interesting. I have been using hackbar which is not perfect. This is a whole new level for a browser addon.

Somali - August 13, 2026, 7:23 am

(Aug 10, 2026 11:21 AM)Dotexe Wrote: [\[CITE\]](#) Seems like a good thing, perhaps a good alternative to those who can't afford BurpSuite professional

What makes BurpSuite pro better for manual browsing website then this SQLiBar ?

If u about to say "AI", that is not true since all exports from extension u can feed to AI later and it will even give better response then checking each single request/response via AI as they come.

Dotexe - August 13, 2026, 10:16 am

(Aug 13, 2026 07:23 AM)Somali Wrote: [\[CITE\]](#) What makes BurpSuite pro better for manual

browsing website then this SQLiBar ?

If u about to say "AI", that is not true since all exports from extension u can feed to AI later and it will even give better response then checking each single request/response via AI as they come.

I couldn't care less about the AI part, to be honest. If you can't differentiate between the two, I'd recommend taking the first CS course you can find. That said, I don't plan on spamming OP's thread with your stuff, nor am I going to spend any more time putting together a comparison table for you. OP's tool is good, and that's what matters.

Somali - August 13, 2026, 1:16 pm

(Aug 13, 2026 10:16 AM)Dotexe Wrote: [\[CITE\]](#) I couldn't care less about the AI part, to be honest. If you can't differentiate between the two, I'd recommend taking the first CS course you can find. That said, I don't plan on spamming OP's thread with your stuff, nor am I going to spend any more time putting together a comparison table for you. OP's tool is good, and that's what matters.

Haha brother why u answer question with a recommendation. I mean i just asked simple personal question what u personally use.

Why for u Burp Suite is better, that is personal question not overall question of features/features.

That would have even helped people to understand how this tool can help people since everyone uses burp suite already its like most known app on this forum, but u decided to stay ignorant.

I mentioned AI just as example, but your post was similar to spam it haven't answered anything and haven't helped anyone :(

•^v^-[bLacKn3x]-^v^• - August 14, 2026, 3:37 am

Update Notes – v1.0.1 & v1.0.2

Encoder / Decoder

New encodings

Selective URL Encode – only encodes characters that typically break SQLi payloads (' " < > and similar), leaving the rest readable

Triple URL Encode – extra encoding layer for stricter WAFs

SQL CHA*R (hex) – CHA*R(0x41,0x42,...)

SQL CONCA*T(CHAR...) – MySQL-style reconstruction: CONC*AT(CHA*R(97),CHA*R(98))

Hex (0xAA,0xBB) – comma-separated hex bytes

JWT Decode – proper header / payload / signature view as formatted JSON

UX

Live preview – encode/decode updates as you type (toggleable)

Swap (?) – move Output <-> Input for easy chaining

2-step chain – optional second encoding stage (e.g. URL -> Base64 in one go)

-> Payload / -> URL / -> Body – push the result straight into the tester fields

Clear button and status line (length / errors)

Encode-only types disable Decode (and vice versa for JWT)

Grouped type list (URL, Base64, Hex, SQL, Unicode, Other)

Reliability

Correct UTF-8 handling for Base64 (no more broken non-ASCII)

More tolerant hex/binary parsers (spaces, commas, 0x prefixes)

SQL CHA*R() / CONCAT values can be decoded back to text

Clearer error messages instead of silent failures

Live Network

Parameter <-> request association

Parameters are now linked to the requests they came from (request ID + URL)

Each request in the list shows a parameter badge (e.g. 3p)

Click a request to expand it and see its parameters inline, with URL / Body / PL actions

Click the same request again to collapse the panel

In the global parameter list, the source line is clickable and jumps to the matching request

New actions

-> Tester – load URL, method, headers, and body into the URL/Payload tab

cURL – copy the request as a cURL command

Copy Params – copy this request's parameters as name=value lines

? Replay – send to Tester and fire the request immediately

Parameter list improvements

Type filter: query / body / json / cookie / header

Selected only – show only parameters of the currently selected request

Clearer source labels, including how many requests share a parameter

Request details include a compact parameter summary for the selected request

Languages

Added Français (fr), Português (pt), and ??? (ja)

Full UI translations for encoder and network features

Language dropdown order: English · Deutsch · Español · Français · Português · ???????? · ??? · ??

CyberXero - August 16, 2026, 2:07 pm

Seems like it's actively updated, v1.0.6 now

I gotta check this out!

•^v^-[bLacKn3x]-^v^• - August 17, 2026, 4:15 am

(Aug 16, 2026 02:07 PM)CyberXero Wrote: [\[CITE\]](#) Seems like it's actively updated, v1.0.6 now

I gotta check this out!

I've actually been working on this project for quite a while, but it was about time to finally release it. I hope it can be useful for people doing manual SQL injection testing.

If anyone has any ideas for future features or updates, feel free to let me know!

Regards

CL1B - August 17, 2026, 4:31 am

Sick! Has anyone used this yet? I don't have time to look at the repo now but I'm curious about it!

GL!

CyberXero - August 17, 2026, 1:15 pm

(Aug 17, 2026 04:15 AM)•^v^-[bLacKn3x]-^v^• Wrote: [\[CITE\]](#) I've actually been working on this project for quite a while, but it was about time to finally release it. I hope it can be useful for people doing manual SQL injection testing.

If anyone has any ideas for future features or updates, feel free to let me know!

RegardsNo being rude, but I am looking into manual sql injection too, I usually automate my process and thought if I want more success on individual targets manual poking is the way, I believe this is gonna help a lot,

(Aug 17, 2026 04:31 AM)MarlboroMan Wrote: [\[CITE\]](#) Sick! Has anyone used this yet? I don't have time to look at the repo now but I'm curious about it!

GL!!I'm looking into it today

ItsDatchii - September 2, 2026, 11:48 pm

added this to my tools will be useful also nice project

CyberXero - September 3, 2026, 1:22 pm

Hey allgood if I slide into your DMs I have your SQLiBar but I'm confused w it