SQL Injection (SQLi) - what it is, why it’s dangerous, and how to prevent it

Gridinsoft Help Center

What it is

SQL Injection is a way attackers trick a website into running their database commands. Instead of treating a form field (like "username") as plain text, a vulnerable site stuffs that text straight into a database query. If an attacker types special characters and SQL words, the site may run them - letting the attacker peek at tables, change data, or even delete it. Think of it as slipping a second instruction into a message meant for the database. Well-built apps block this; rushed ones sometimes don’t.

Why it matters

A single SQLi hole can expose logins, emails, payment info - or let someone take over accounts and wipe records. Breaches from SQLi often lead to identity theft and big service outages.

How it works

- Find input: attacker tests fields/URLs for errors or odd responses.
- Inject: adds SQL pieces (quotes, OR/AND, UNION, etc.) to change the query.
- Extract/modify: dumps tables, changes passwords, or deletes rows.
- Pivot: uses the database foothold to move deeper into the app or network.

Red flags

- Weird database error messages shown to users (with "SQL" or table names).
- Sudden spikes in requests with quotes, UNION, or odd parameters.
- Unexplained data changes or new admin accounts.
- WAF/EDR alerts about injection patterns in web traffic.

Prevent it

- Parameterized queries (prepared statements): never build SQL by string-concatenating user input.
- ORM/Query builders: use safe APIs that bind values for you.
- Input handling: validate types/lengths; reject unexpected characters where possible.
- Least privilege: app DB user should only have the permissions it needs.
- WAF & monitoring: block common SQLi patterns and alert on anomalies.
- Tests: run automated scanners and add unit/integration tests for SQLi cases.