What is NoSQL injection?

NoSQL injection occurs when an application builds a NoSQL query, for example a MongoDB query, from user input without keeping that input strictly as data. Because many NoSQL drivers accept query objects, an attacker who can inject operators such as $ne, $gt, or $regex can alter the query's logic to bypass authentication or extract data, and where server-side JavaScript like $where is enabled, reach code execution. It is closely related to SQL injection but exploits the document and operator model rather than SQL syntax.

HOW IT WORKS

01 The abuse and payload

Shown for defensive context:

- Authentication bypass: send a JSON body `{"username": {"$ne": null}, "password": {"$ne": null}}` so the query matches any user, or target a known user with `{"username": "admin", "password": {"$ne": ""}}`.
- Operator injection for extraction: `$regex` lets an attacker infer values character by character, for example `{"password": {"$regex": "^a"}}` to learn the first character from the response.
- Server-side JavaScript: where `$where` or `mapReduce` runs JavaScript, an injected expression can execute logic on the database, in some setups reaching code execution.
- Operator injection in query strings: frameworks that parse `user[$ne]=1` into an object reintroduce the flaw even without a JSON body.

These logic-level bypasses are a standard check against authentication and search endpoints during web application penetration testing.

HOW TO DEFEND

- Cast and validate types: ensure fields that should be strings are strings before they reach the query, rejecting objects and arrays where a scalar is expected.
- Use the driver's safe query construction and parameterization rather than merging raw request objects into a query.
- Disable server-side JavaScript such as `$where` and `mapReduce` on untrusted input, and turn it off at the database level if unused.
- Validate against a schema so unexpected operators and nested structures are rejected at the boundary.

SOURCES

[1] OWASP: Testing for NoSQL Injection

[2] PortSwigger: NoSQL injection

[3] MITRE ATT&CK: Exploit Public-Facing Application (T1190)

Find the injection before an attacker does.

securelayer7.net/learn/web-exploitation/what-is-nosql-injection

Open online