Code scanning alerts / #5

Missing rate limiting

In pull request in `refs/pull/170/merge` 5 days ago

backend/routes/**auth.js:14**

```
11    }
12  });
13
14  router.post("/login", async function (req, res) {
15    if (req.body.username && req.body.password) {
16      authorize(req.body.username, req.body.password, function
17        if (user) {
18          res.send({ token: user["token"] });
19        } else {
20          res.status(401).send({
21            error: err.message,
22          });
23        }
24      });
25    } else {
26      res.status(400).send({ error: "Specify username and passw
27    }
28  });
```

This route handler performs ▶ authorization, but is not rate-limit

CodeQL

```
29
30  export default router;
```

Tool	Rule ID	Query
CodeQL	`js/missing-rate-limiting`	View source

HTTP request handlers should not perform expensive operations such as accessing the file system, executing an operating system command or interacting with a database without limiting the rate at which requests are accepted. Otherwise, the application becomes vulnerable to denial-of-service attacks where an attacker can cause the application to crash or become unresponsive by issuing a large number of requests at the same time.

Recommendation

Severity

High

Affected branches

None

Tags

security

Weaknesses

CWE-307
CWE-400
CWE-770

A rate-limiting middleware should be used to prevent such attacks.

Example

The following example shows an Express application that serves static files without rate limiting:

```
var express = require('express');
var app = express();

app.get('/:path', function(req, res) {
  let path = req.params.path;
  if (isValidPath(path))
    res.sendFile(path);
});
```

To prevent denial-of-service attacks, the `express-rate-limit` package can be used:

```
var express = require('express');
var app = express();

// set up rate limiter: maximum of five requests per minute
var RateLimit = require('express-rate-limit');
var limiter = RateLimit({
  windowMs: 15 * 60 * 1000, // 15 minutes
  max: 100, // max 100 requests per windowMs
});

// apply rate limiter to all requests
app.use(limiter);

app.get('/:path', function(req, res) {
  let path = req.params.path;
  if (isValidPath(path))
    res.sendFile(path);
});
```

References

- OWASP: Denial of Service Cheat Sheet.
- Wikipedia: Denial-of-service attack.
- NPM: express-rate-limit.
- Common Weakness Enumeration: CWE-770.
- Common Weakness Enumeration: CWE-307.
- Common Weakness Enumeration: CWE-400.

Show less

First detected in commit last week