keystonejs / **keystone** Public

Code Issues 88 Pull requests 29 Discussions Actions ···

NODE_ENV defaults to development with esbuild

Critical **dcousens** published **GHSA-25mx-2mxm-6343** 23 days ago

Package

@keystone-6/core (npm)

Affected versions	Patched versions
3.0.0,3.0.1	3.0.2

Description

Impact

`@keystone-6/core@3.0.0 || 3.0.1` users that use `NODE_ENV` in their own code (**not dependencies**) to trigger security-sensitive functionality in a production build are vulnerable to `NODE_ENV` being inlined to `"development"` for user code.

If your dependencies use `NODE_ENV` to trigger particular behaviours (optimisations, security or otherwise), they should still respect your environment's configured `NODE_ENV` variable and thereby be unaffected.

If you do not use `NODE_ENV` in your own code to trigger security-sensitive functionality, **you are not impacted** by this vulnerability.
An example of code that would be affected, might be the following:

```
if (process.env.NODE_ENV !== 'production') {
  // this code would unintentionally run in your production builds
}
```

Technical Description

The problem comes from esbuild defaulting `NODE_ENV` to `"development"` when a platform configuration is undefined.
You can read about why `esbuild` has that behaviour in their documentation, but the result for Keystone users is that user Typescript was compiled, and had inlined `NODE_ENV` to the constant `"development"` .

Your application's dependencies, as found in `node_modules` (including `@keystone-6/core`), are typically not compiled as part of this process, and thus should be unaffected. Therefore any libraries that used `NODE_ENV` to trigger particular behaviours (optimisations, security or otherwise) should still respect your environment's `NODE_ENV` .
We have tested this assumption by verifying that `NODE_ENV=production yarn keystone start` still uses secure cookies when using `statelessSessions` .

Thereby, the severity of this vulnerability is dependent on what functionality users conditionally triggered, in their own code, depending on the expectation that `NODE_ENV` would be correctly configured in their application. In accordance with Common Vulnerability Scoring System `2.3.3. Assume Vulnerable Configurations` , this security advisory assumes vulnerable configurations and is thus marked as *critical*, but you should evaluate the true security impact for your application to determine a relevant score.

Patches

This vulnerability has been fixed in `@keystone-6/core@3.0.2` , thanks to **@mmachatschek** in #8031.
We have added regression tests for this vulnerability in #8063.

Workarounds

If you cannot upgrade your `@keystone-6/core` version for any reason, your best alternative is to remove any code that uses `NODE_ENV` in a way that may reasonably impact your application security.

References

- https://esbuild.github.io/api/#platform
- #8031
- #8063

For more information

Thanks to Austin Burdine for reporting this problem as a potential security vulnerability.

If you have any questions around this security advisory, please don't hesitate to contact us at security@keystonejs.com, or open an issue on GitHub.

If you have a security flaw to report for any software in this repository, please see our SECURITY policy.

Severity

Critical 9.8 / 10

CVSS base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	None
User interaction	None
Scope	Unchanged
Confidentiality	High
Integrity	High
Availability	High

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

CVE ID

CVE-2022-39382

Weaknesses

No CWEs

Credits

acburdine