Improper handling of email input

High balazsorban44 published GHSA-pgjx-7f9g-9463 on Jul 6

Package

next-auth (npm)

Affected versions	Patched versions
<3.29.8, <4.9.0	3.29.8, 4.9.0

Description

Impact

An attacker can pass a compromised input to the e-mail signin endpoint that contains some malicious HTML, tricking the e-mail server to send it to the user, so they can perform a phishing attack. Eg.: `balazs@email.com, <a href="http://attacker.com">Before signing in, claim your money!</a>` . This was previously sent to `balazs@email.com` , and the content of the email containing a link to the attacker's site was rendered in the HTML. This has been remedied in the following releases, by simply not rendering that e-mail in the HTML, since it should be obvious to the receiver what e-mail they used:

next-auth v3 users before version 3.29.8 are impacted. (We recommend upgrading to v4, as v3 is considered unmaintained. See our migration guide)

next-auth v4 users before version 4.8.0 are impacted.

Patches

We've released patches for this vulnerability in:

- v3 - `3.29.8`
- v4 - `4.9.0`

You can do:

```
npm i next-auth@latest
# or
yarn add next-auth@latest
```

```
#
pnpm add next-auth@latest
```

(This will update to the latest v4 version, but you can change `latest` to `3` if you want to stay on v3. This is not recommended.)

Workarounds

If for some reason you cannot upgrade, the workaround requires you to sanitize the `email` parameter that is passed to `sendVerificationRequest` and rendered in the HTML. If you haven't created a custom `sendVerificationRequest`, you only need to upgrade. Otherwise, make sure to either exclude `email` from the HTML body or efficiently sanitize it. Check out https://next-auth.js.org/providers/email#customizing-emails

References

Related documentation:

- https://next-auth.js.org/providers/email#customizing-emails
- https://next-auth.js.org/getting-started/upgrade-v4

A test case has been added so this kind of issue will be checked before publishing. See: https://github.com/nextauthjs/next-auth/blob/cd6ccfde898037290ae949d500ace8a378376cd8/packages/next-auth/tests/email.test.ts

For more information

If you have any concerns, we request responsible disclosure, outlined here: https://next-auth.js.org/security#reporting-a-vulnerability

Timeline

The issue was reported 2022 June 29th, a response was sent out to the reporter in less than 1 hour, and after identifying the issue a patch was published within 4 working days.

Severity

High

CVE ID

CVE-2022-31127

Weaknesses

CWE-20

Credits

Sandiipmaity