Middleware-based route protection bypass

Critical **nikosdouvlis** published **GHSA-vqx2-fgx2-5wq9** last week

Package

@clerk/astro (npm)

Affected versions	Patched versions
>= 0.0.1, <= 2.17.9, >= 3.0.0, <= 3.0.14	1.5.7, 2.17.10, 3.0.15

@clerk/nextjs (npm)

>= 5.0.0, <= 6.39.1, >= 7.0.0, <= 7.2.0	5.7.6, 6.39.2, 7.2.1

@clerk/nuxt (npm)

>= 1.1.0, <= 1.13.27, >= 2.0.0, <= 2.2.1	1.13.28, 2.2.2

@clerk/shared (npm)

>= 2.20.17, <= 3.47.3, >= 4.0.0, <= 4.8.0	2.22.1, 3.47.4, 4.8.1

Description

Summary

`createRouteMatcher` in `@clerk/nextjs`, `@clerk/nuxt`, and `@clerk/astro` can be bypassed by certain crafted requests, allowing them to skip middleware gating and reach downstream handlers.

Sessions are not compromised and no existing user can be impersonated - the bypass only affects the middleware-level gating decision.

Who is affected

All apps using `createRouteMatcher` should upgrade to the patched versions. Patches are drop-in with no API changes. The information below describes the scope of the bypass and helps you understand whether you are potentially affected, but is not a reason to delay the upgrade.

Apps relying only on middleware gating via `createRouteMatcher` are affected, because a crafted request can skip middleware checks and reach downstream handlers (API routes, server components, etc.). This middleware pattern permits the bypass:

```
// Next.js example, equivalent patterns exist in Nuxt and Astro
const isProtectedRoute = createRouteMatcher(['/admin(.*)']);

export default clerkMiddleware(async (auth, req) => {
  if (isProtectedRoute(req)) {
    await auth.protect();
  }
});
```

That said, the bypass is limited to the middleware-level route-matching gate. `clerkMiddleware` still authenticates the request and `auth()` reflects the real authentication state of the caller. Auth checks performed inside your route handlers, server components, or server actions continue to work correctly and are not affected. Whether your app is affected in practice depends on whether you have those downstream checks.

External APIs that authenticate each request with a token are also unaffected on those endpoints, since token verification runs independently.

Additionally, this common middleware pattern correctly blocks the bypass at the middleware layer:

```
// Next.js example, equivalent patterns exist in Nuxt and Astro
const isPublicRoute = createRouteMatcher(['/docs(.*)']);

export default clerkMiddleware(async (auth, req) => {
  if (!isPublicRoute(req)) {
    await auth.protect();
  }
});
```

`@clerk/shared` is usually not imported directly in application code, but if you import `createPathMatcher` from an affected `@clerk/shared` version, you are also affected. Run `npm why @clerk/shared` (or your package manager's equivalent) to check your installed version.

Recommended actions

Install the patched version for your framework (pick the one matching your current major):

`@clerk/nextjs`

- v7.x: fixed in `7.2.1`
- v6.x: fixed in `6.39.2`
- v5.x: fixed in `5.7.6`

`@clerk/nuxt`

- v2.x: fixed in `2.2.2`
- v1.x: fixed in `1.13.28`

`@clerk/astro`

- v3.x: fixed in `3.0.15`
- v2.x: fixed in `2.17.10`
- v1.x: fixed in `1.5.7`

`@clerk/shared`

- v4.x: fixed in `4.8.1`
- v3.x: fixed in `3.47.4`
- v2.x: fixed in `2.22.1`

Workaround

If you cannot upgrade immediately, adding server-side auth checks (`auth()`) inside your route handlers, server components, or server actions provides defense-in-depth against this bypass.

Timeline

This issue was reported on 13 APR 2026, patched on 15 APR 2026, and publicly disclosed on 15 APR 2026.

Thanks to Christiaan Swiers for the responsible disclosure of this vulnerability.

Severity

Critical 9.1 / 10

CVSS v3 base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	None

User interaction	None
Scope	Unchanged
Confidentiality	High
Integrity	High
Availability	None

Learn more about base metrics

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

CVE ID

CVE-2026-41248

Weaknesses

- CWE-436
- CWE-863

Credits

YouGina Reporter